UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

EVALUATION OF EXTREME LEARNING MACHINE AS CHANNEL EQUALIZER FOR COLOR-SHIFT KEYING-BASED VISIBLE LIGHT COMMUNICATION SYSTEMS EMPLOYED IN UNDERGROUND MINING SCENARIOS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

RAIMUNDO DANIEL BECERRA PARRA

PROFESOR GUÍA:
CESAR AZURDIA MEZA

PROFESOR CO-GUÍA:
ISMAEL SOTO GÓMEZ

COMISIÓN:
JORGE SANDOVAL ARENAS

Este trabajo ha sido parcialmente financiado por:
FONDECYT N°1211132 2021-2022

SANTIAGO DE CHILE
2022

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: RAIMUNDO DANIEL BECERRA PARRA
FECHA: 2022
PROF. GUÍA: CESAR AZURDIA MEZA

EVALUACIÓN DE EXTREME LEARNING MACHINE COMO ECUALIZADOR DE CANAL PARA SISTEMAS DE COMUNICACIÓN POR LUZ VISIBLE BASADOS EN COLOR-SHIFT KEYING EMPLEADOS EN ESCENARIOS DE MINERÍA SUBTERRÁNEA

En el contexto de la minería subterránea, los sistemas de comunicación por luz visible se han propuesto como sistemas auxiliares a la comunicación inalámbrica tradicional, pues tienen el potencial de otorgarle mayor fiabilidad y flexibilidad a esta última. El canal de comunicación por luz visible en una mina subterránea es mucho más complejo que el de entornos in-door comunes (por ejemplo, oficinas, centros comerciales, etcétera): polvo, maquinaria pesada, paredes irregulares son solo algunos de los factores que degradan la señal de luz en una mina subterránea. Dada la potencial hostilidad del canal de comunicación en el entorno minero, es interesante evaluar un método de ecualización complejo y compararlo con el método de ecualización estándar para sistemas de comunicación por luz visible.

Extreme learning machine es un algoritmo basado en redes neuronales que ha demostrado ser efectivo como ecualizador de canales altamente hostiles. Color-shift keying es una de las técnicas de modulación utilizadas en sistemas de comunicación por luz visible que alcanza altas tasas de datos. El objetivo de este trabajo es evaluar la factibilidad de utilizar un ecualizador basado en extreme learning machine en un sistema de comunicación por luz visible, desplegado en una mina subterránea y haciendo uso de la modulación color-shift keying.

Para cumplir el objetivo, se diseñó un escenario de simulación y un montaje experimental computacional, cuyo resultado es la probabilidad de error de bit del sistema simulado. El escenario de simulación consistió en una mina subterránea en la que un único transmisor enviaba datos a un único receptor, usando un enlace de comunicación por luz visible basado color-shift keying y ateniéndose a las especificaciones del estándar IEEE 802.15.7. Usando la probabilidad de error de bit obtenida como métrica, los métodos de ecualización basados en extreme learning machine se compararon con el método de ecualización tradicional, propuesto por el estándar IEEE 802.15.7 para color-shift keying. Los fenómenos inherentes a las minas subterráneas que degradan la señal de luz se consideraron en el escenario de simulación incorporando un modelo de canal de mina subterránea del estado del arte.

Los resultados dan a entender que, para las tasas de datos definidas en el estándar IEEE 802.15.7 y para el entorno de simulación evaluado, el método estándar de ecualización de canal es perfectamente capaz de ecualizar la señal; de hecho, el método estándar tiene mejor rendimiento que la ecualización basada en extreme learning machine, y además requiere menor complejidad computacional. Sin embargo, para descartar completamente extreme learning machine como método de ecualización de canales por luz visible, es necesario futuro trabajo, ya sea evaluándolo en simulaciones más complejas o incluso implementando el sistema en la realidad.

EVALUATION OF EXTREME LEARNING MACHINE AS CHANNEL EQUALIZER FOR COLOR-SHIFT KEYING-BASED VISIBLE LIGHT COMMUNICATION SYSTEMS EMPLOYED IN UNDERGROUND MINING SCENARIOS

In the context of underground mining, visible light communication systems have been proposed as auxiliary systems to traditional wireless communications, as they have the potential to provide greater reliability and flexibility to the latter. The visible light communication channel in an underground mine is much more complex than that of common in-door environments (e.g. offices, shopping malls, etc.): dust, heavy machinery, irregular walls are just some of the factors that degrade the light signal in an underground mine. Given the potential harshness of the communication channel in underground mining environments, it is interesting to evaluate a complex equalization method and compare it to the standard equalization method for visible light communication systems.

Extreme learning machine is a neural network-based algorithm that has proven to be effective as channel equalizer of extremely harsh channels. Color-shift keying is one of the modulation techniques used in visible light communication systems which reaches the high data rates. The objective of this work is to evaluate the feasibility of using an equalizer based on extreme learning machines in a visible light communication system, deployed in an underground mine and making use of color-shift keying modulation.

To meet the objective, a simulation scenario and a computational experimental setup were designed, the result of which is the bit error probability of the simulated system. The simulation scenario consisted of an underground mine in which a single transmitter sent data to a single receiver using a visible light communication link based on color-shift keying and confined to the specifications of the IEEE 802.15.7 standard. Using the bit error probability obtained as metric, the extreme learning machine-based equalization methods were compared with the standard equalization method proposed by the IEEE 802.15.7 standard for color-shift keying. The phenomena inherent to underground mines that degrade the light signal were considered in the simulation scenario by incorporating a state-of-the-art underground mine channel model.

The results imply that, for the data rates defined in the IEEE 802.15.7 standard and for the simulation environment evaluated, the standard channel equalization is perfectly capable of equalizing the signal; in fact, the standard equalization method performs better than equalization based on extreme learning machines, and also requires less computational complexity. However, to completely rule out extreme learning machine as a visible light channel equalization method, future work is needed, either by evaluating it in more complex simulations or even by implementing the system in reality.

Esta tesis está dedicada a mis padres

Acknowledgments

Quiero dar muchas gracias al Dr. César Azurdia por su inmenso apoyo durante estos últimos años, al Dr. Ismael Soto, al Dr. David Zabala-Blanco y a Pablo Palacios por su indispensable ayuda a lo largo de este trabajo. Muchas gracias a mi polola, a mi familia, a mis amigos, a mi mamá. Su amor, consejos y apoyo incondicional son impagables y no podría haber llegado hasta acá, el final de la carrera, sin todos ustedes.

Table of Content

List of Tables

List of Figures

Acronyms

AWGN additive white Gaussian noise

BER bit error ratio

CIE International Commission on Illumination

CIR channel impulse response

CSK color-shift keying

DC direct current

DCS digital communication system

ELM extreme learning machine

flops floating point operations

FOV field of vision

IEEE Institute of Electrical and Electronics Engineers

IM/DD intensity modulation with direct detection

ISI intersymbol interference

LED light-emitting diode

LOS line-of-sight

LS light source, least squares

MAC medium access control layer

NLOS non-line-of-sight

OOK on-off keying

PD photodiode, photodetector

PHY phyisical layer

RF radio frequency communication

RGB red, green and blue

SELM supervised extreme learning machine

SISO single-input single-output

SNR signal-to-noise ratio

SPD spectral power distribution

SSELM semi-supervised extreme learning machine

USELM unsupervised extreme learning machine

VLC visible light communication

VLS virtual light source

VPPM variable pulse position modulation

YCM yellow, cyan and magenta

Chapter 1

Introduction

1.1. Motivation

Mining can be defined as the activity, occupation and industry concerned with the extraction of minerals. [1] This activity can be further divided into two groups according to the type of excavation: strip mining and underground mining. The differences between these two groups is that in the former, people and machinery operate at ground level, while the latter operates underground [1]. Underground mining is considered one of the most dangerous occupations, mainly for two reasons:

- Work is performed in hazardous environments, with hazards such as fires, explosions, roof collapses, toxic gases, floods and vehicle accidents that might result in serious injuries or disabilities, and in the worst cases, fatalities [1].
- When an accident occurs during underground mining operations, the emergency response is met with a lot more obstacles than in other work environments. This is due to the fact that the underground mine's own structure does not allow for a swift emergency response [1].

Despite the extreme work conditions, productivity and efficiency in mining operations must always be maintained as to meet market demand. In other words, mining operations are expected to extract minerals as efficiently, and safely, as possible. Communications are at the center of both concerns: daily mining operations require a communication system for their coordination and monitoring, while in the case of an accident, the information flow is essential in order to locate workers in urgent need [1]. Currently, the most common underground mine communication systems consist of cable or radio communication systems. Both kinds of systems have respective disadvantages. On the one hand, wired systems are prone to damage, while radio signals are extremely attenuated by the mining environment, leading to low data rates. In recent years, a new communication system has been proposed to complement existing systems over short distances: visible light communications (VLC) [2].

These VLC systems have multiple applications as alternatives to traditional wireless systems: hospitals and aircraft cabin (both are electromagnetic sensitive areas), Vehicle-to-X communications, museums and street lighting (as to give access to information/entertainment in public areas), industrial applications (specially where resistance to electromagnetic interference is important), mobile underwater communications, sensor node communications, etc. [3]

Some advantages of this kind of system are that both the illumination and communication functions are integrated, and that the devices needed, light-emitting diodes (LEDs) and photodiodes (PDs), are relatively cheap [2]. Another advantage is that the frequency spectrum of visible light is entirely different from radio communications, and thus there is no electromagnetic interference and the spectrum is completely unlicensed, allowing for faster data rates [4]. One modulation scheme exclusive to VLC systems known as color-shift keying (CSK) is of particular interest due to its high data rates.

The disadvantage is that underground mining environments are usually more hostile towards light transmission than other environments where visible light communications are traditionally deployed, such as offices or shopping complexes. Factors such as LED and photodiode tilting, irregular walls, light being obstructed by large machinery, light scattering due to dust, all contribute to the deterioration of the visible light communication signal in underground mining environments [4]. In order to reduce the detrimental impact of these factors, not seen in conventional visible light communication channels, unconventional channel estimation and equalization methods are needed.

In recent years, demand for faster and more reliable wireless communications has spiked research interest in machine learning-based solutions for channel estimation and equalization. These machine learning algorithms use data sequences known both at the transmitter and at the receiver to fine-tune its own internal parameters iteratively, in a process known as training. This method entails two problems: on the one hand, the need for data sequences used in training requires significant amounts of overhead. On the other hand, the need for iterative training requires significant computational power in order to perform the training operations as fast as real-time communications demand it. The family of machine learning-based algorithms known as extreme learning machine (ELM) was proposed as a solution to both these problems. The main advantage of extreme learning machine is that, at its most basic implementation, it performs the training process in only one iteration, making it a simple and fast alternative.

Extreme learning machines-based algorithms have proven effective in radio frequency (RF) channel equalization [5, 6, 7], and as post-distortion algorithms used to mitigate the inherent LED non-linearity in visible light communications [8]. Yet to be proven is their effectiveness in channel equalization for visible light communications, let alone in underground mining scenarios, but given their performance in radio frequency channels and simplicity, they are an attractive candidate for such task. In this work, ELM-based algorithms are proposed as VLC channel estimation and equalization algorithms under the CSK modulation scheme, in order to compare their performance against the estimation and equalization method proposed by the Institute of Electrical and Electronics Engineers (IEEE) in their IEEE 802.15.7 standard [9], commonly known as least squares (LS) equalizer.

This work is concerned with two main scientific inquiries: firstly, the assumption that the least squares equalizer is a feasible equalization method for CSK-based VLC systems, even in harsher channel conditions such as underground mining scenarios, to the best of our knowledge, has not been put to the test in the literature. And secondly, even though extreme learning machine has been studied as an equalizer for harsh channel conditions in wireless systems, these are in the RF bands. Extreme learning machines-based equalizers, to the best of our knowledge, have not been employed in CSK-based visible light communication systems under harsh environments such as underground mining scenarios.

1.2. Hypothesis

"It is feasible, in terms of bit error probability and computational complexity, to employ the least squares estimator as the equalizer in a single-input single-output visible light communication system based on the color-shift keying modulation scheme and deployed in an underground mine scenario, all the while adhering to the system specifications of the IEEE 802.15.7 standard."

1.3. Objectives

1.3.1. General Objective

The general objective of this work can be formulated as

"To evaluate and analyze the feasibility of employing extreme learning machine as channel equalizer for a color-shift keying-based visible light communication system in an underground mine"

1.3.2. Specific Objectives

The specific objectives are

- To analyze the extreme learning machines-based equalizer and compare it to the equalizer proposed in the IEEE 802.15.7 standard, in terms of probability of error
- To analyze the extreme learning machines-based equalizer and compare it to the equalizer proposed in the IEEE 802.15.7 standard, in terms of computational complexity
- To evaluate and analyze the communication system considering the technical specifications presented in the IEEE 802.15.7 standard
- To evaluate and analyze the performance of the equalizers considering different communication system parameters

1.4. Structure

This thesis is structured as follows. Chapter 2 presents an overview of the theoretical aspects of this work. Chapter 3 covers the methods used in this work to attain the objectives, which can be summarized as the color-shift keying modulation, the extreme learning machine algorithm and the VLC system simulation. Chapter 4 shows the results obtained from the experiments laid out in Chapter 3 and then discusses them. Finally, Chapter 5 presents the conclusions drawn from the entire experience.

Chapter 2

Literature Review

2.1. Digital Communication Systems

The principal characteristic of a digital communication system (DCS) is that during a finite time interval, it transmits a waveform from a finite set of possible waveforms. In opposition, analog communication systems transmit a waveform from an infinite set of waveform shapes. The goal of a DCS at the receiver is not to reproduce a transmitted waveform accurately but rather to determine, by looking at the received signal, which waveform from the finite set of waveforms was sent by the transmitter [10].

The source output, be it analog or digital, is converted into a sequence of binary digits by the *source encoder*, in a process called source encoding or data compression. This process seeks to represent the source output using as few binary digits as possible to represent the source output. The binary digits from the source encoder are then passed to the *channel encoder*, which, contrary to the source encoder, seeks to add redundancy in the binary sequence, making the communication system more robust against noise and interference. The channel encoder outputs a binary sequence which is then passed to the *digital modulator* [11].

The digital modulator functions as the interface to the channel, mapping the binary sequence into signal waveforms that are compatible with the requirements imposed by the channel. In the modulation process, the binary sequence is divided into groups of m bits, with each group being treated as a single unit called a *symbol*. Each symbol s_i, for $i = 1, \ldots, M$, is found in the finite symbol set $\{s_1, \ldots, s_M\}$, also known as the alphabet. The size of the alphabet is given by $M = 2^m$, where m is the number of bits found in each symbol. For a given data rate R_b (bits per second, bps), the symbol rate is given by R_b/m (symbols per second, baud) [10].

Each symbol takes the form of a compatible waveform. Depending on the channel requirements, either a *baseband waveform* or a *bandpass waveform* is used. These two kinds of waveforms are differentiated by their frequency spectrum: in a baseband waveform, usually known as a pulse, the frequency spectrum of the signal extends from DC up to a finite value, while in a bandpass waveform, the baseband waveform is translated in the frequency space by a carrier wave to a frequency that is much larger than the bandwidth of the baseband waveform [10].

In the transmission process, the symbol waveform is affected by the channel. This interaction can be described by the channel impulse response (CIR). Additive random noise also impacts the symbol waveform. At the receiver, the *demodulator* restores the received symbol to an optimally shaped baseband pulse. It also applies several filters in order to prepare

the signal for detection: filters are applied to remove unwanted high frequency components and for pulse shaping. A special case of filter is called *equalizer*. Equalization is the filtering process in or after the demodulator in which the degrading channel effects on the signal are reversed. Equalization is essential when the channel impulse response is of very poor quality, since its function is to compensate any signal distortion caused by a non-ideal CIR. After being processed in the demodulator, the received symbol is sampled and assigned one of the finite symbols in $\{s_1, \ldots, s_M\}$.

2.1.1. Channel effects

The channel effects comprise two main sources of corruption: channel noise and intersymbol interference (ISI). Channel noise is caused by the inherent thermal noise of the physical medium and interference from other users or other external sources. Intersymbol interference refers to the spreading of a signal pulse past the symbol duration, causing successive signal pulses to overlap. For a signal pulse not to be spread by the channel, the channel would have to be a perfect filter, or, in other words, the channel would have to be band-limited. The closer the channel bandwidth is to the signal bandwidth, the more pulse spread there is and thus, the worse the ISI becomes [10].

2.1.2. Bit error ratio

Given a time period T and a data rate of R_b, then the total number of bits transmitted in the period T is given by R_bT. The bit error ratio (BER) is defined as

$$\mathrm{BER}[T] = \frac{\#\text{ of incorrectly received bits during } T}{R_bT} \tag{2.1}$$

The bit error probability p_e is the expected value of the BER,

$$p_e = \lim_{T\to\infty} \mathrm{BER}[T] \tag{2.2}$$

due to the law of large numbers. In practical terms, the BER is an approximation of the bit error probability.

2.1.3. Mathematical models for communication channels

Most communication channels can be classified in the following three categories:

- **Additive noise channel:** In this model, the transmitted signal $s(t)$ is corrupted by an additive random noise process $n(t)$. Additionally, the signal can be attenuated by the channel by a factor a, called the attenuation factor. Thus, the received signal $r(t)$ can be expressed as [11]
$$r(t) = as(t) + n(t) \tag{2.3}$$
- **Linear filter channel:** In some channels, filters are used to ensure that the transmitted signals fall in a specified frequency band, as to not interfere with one another. Such channels can be modeled as linear filter channels with additive white Gaussian noise (AWGN). For a linear filter with impulse response $h(t)$, the output of the linear filter

channel can be expressed as, [11]

$$\begin{aligned} r(t) &= s(t) * h(t) + n(t) \\ &= \int_{-\infty}^{\infty} h(\tau) s(t-\tau) d\tau + n(t) \end{aligned} \tag{2.4}$$

where $*$ denotes convolution.

- **Linear time-variant filter channel:** There are channels that transmit the signal over multiple paths which vary over time. These channels may be modeled as time-variant linear filters. These linear filters are defined by the time-variant channel impulse response $h(\tau,\, t)$, where $h(\tau,\, t)$ designates the response of the channel at time t, due to an impulse applied at time $t-\tau$. The output of the time-variant linear filter channel can be expressed as, [11]

$$\begin{aligned} r(t) &= s(t) \circledast h(\tau,\, t) + n(t) \\ &= \int_{-\infty}^{\infty} h(\tau,\, t) s(t-\tau) d\tau + n(t). \end{aligned} \tag{2.5}$$

2.2. Machine Learning

Machine learning is a field of study concerned with understanding and developing algorithms, methods or models that can improve their performance at a certain task, i.e. learn, from data. The classical machine learning approach to the learning process is to consider a *training set* $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and feed it to a machine learning model, which in turn uses the given data to tune its internal parameters [12].

In the *supervised learning* approach, the training set is complemented with a *label set* $\{\mathbf{t}_1, \ldots, \mathbf{t}_N\}$, where $\mathbf{t}_i$ is the desired model output for the model input $\mathbf{x}_i$, with $i = 1 \ldots, N$. The *supervised training dataset*, $\mathcal{S}$, can be expressed as

$$\mathcal{S} = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^{N}\,. \tag{2.6}$$

In the *training phase*, also known as the *learning phase*, the machine learning model tunes its internal parameters such that it generates a desired output given a certain input. What a "desired output" is, the model learns from the dataset. The result of the training phase is a machine learning model Θ, which takes a new input $\mathbf{x}$ i.e. $\mathbf{x} \notin \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, and outputs a vector $\mathbf{y}$, encoded in the same way as the vectors in the label set, namely

$$\Theta(\mathbf{x}) = \mathbf{y}\,. \tag{2.7}$$

In other words, after training, the model can identify new inputs $\mathbf{x}$ by associating a label $\mathbf{y}$ to them. The set of new inputs that are given to the model after training are known as the *test set*. The ability to correctly identify inputs different from the ones in the training set is known as *generalization*. When the output labels can take the form of just one of a finite number of discrete categories, then the task at hand is called *classification*. When the output labels can take the form of one or continuous variables, then the task at hand is called *regression* [12].

In the *unsupervised learning* approach, the training dataset consists of a set of input vectors

without a corresponding label set. The *unsupervised training dataset*, $\mathcal{U}$, can be expressed as

$$\mathcal{U} = \{\mathbf{x}_i\}_{i=1}^{N} . \tag{2.8}$$

Usually, the goal of unsupervised learning is to form *clusters*, or groups, of input vectors with similar characteristics, process known as *clustering*. It is also useful for determining the distribution of data in the input space, process known as *density estimation* [12].

2.2.1. Hyperparameters

Machine learning models usually employ parameters in the training phase that do not change during training and also do not depend on the training data, but rather must be pre-set. These parameters are called *hyperparameters* and can have an important impact on the efficacy, speed, and quality of the training phase. Consequently, choosing appropriate hyperparameters is an important step in the machine learning model design. Unfortunately, there is no clear-cut way to choose the hyperparameters, meaning that heuristic methods must be employed.

The method known as *grid search* is the most well known heuristic employed in hyperparameter selection. It requires the selection of a set of candidate values for each hyperparameter, found by trial and error or by an educated guess, and then training the model for all combination of these candidate values. The trained models are then evaluated according to an objective function which must be maximized or minimized. Examples of such objective functions are the accuracy (to be maximized), or the mean squared error (to be minimized), of the model predictions in the test set. The combination of hyperparameter values that generates the model that achieves the minimum (or maximum) objective function value out of all the other models, is selected as the optimal hyperparameter value combination.

2.3. Visible Light Communication Systems

Visible light communication (VLC) is founded on the idea of providing both illumination and data transmission simultaneously. This is achieved via the modulation of the light source, be it white light or other selected colors in the visible spectrum. The most used light source are light-emitting diodes (LEDs), which can be turned on and off millions of times per second without decreasing their lifespan. VLC systems are as much an alternative as a complementary system to RF communications. Some key advantages of VLC systems are: [3]

- *Energy efficiency:* No extra power is needed for data transmission. The same power used in illumination is re-used for communication purposes.
- *Large unregulated bandwidth:* The wavelengths usable with LEDs in the visible range go from 380 to 780 nm, which roughly translates into 750 THz of signal bandwidth. This large bandwidth allows for Gbps data rates, at least in theory. Furthermore, international radio frequency spectrum regulations usually go up to 3 THz, meaning that license-free operation is possible.
- *Interference immunity:* Light does not penetrate through walls, unlike radio waves in the microwave frequency range. This means that the entire light bandwidth can be re-used in adjacent rooms, without causing interference.

- *Low-cost hardware:* VLC receiver and transmitter hardware is much simpler than RF front-ends, at least below 1 Gbps, meaning that low-cost consumer products are possible.

2.3.1. Fundamentals of Light

Light can be thought of as a quantum particle, called photon, or as an electromagnetic wave. A classical wave has its energy spread out in space, while a classical particle is located at a single point in space with all of its energy in said location. The property of light to be interpreted in these two contradictory ways is called the wave-particle duality [3]. Light visible by human eyes, or simply visible light, is located in the electromagnetic spectrum from 390 nm (770 THz) to roughly 720 nm (420 THz) [3]. For electromagnetic waves, given a particular *wavelength* λ (in m), the equivalent *frequency* f (in Hz) is given by

$$f = \frac{c}{\lambda}\,, \tag{2.9}$$

where c is the speed of light, roughly 3×10^8 m/s in free space.

2.3.2. Radiometry

Radiometry is the field concerned with the measurement of electromagnetic radiation, independent of the detector or observer of said radiation. Radiometry measures the energy or the power of optical radiation and works with four basic measures: radiant power, radiant intensity, irradiance and radiance. Radiant power or radiant flux Φ_e is the total electromagnetic energy dQ_e emitted by a light source per unit time dt. It is measured in watts (W) and it is given by [13]

$$\Phi_e = \frac{dQ_e}{dt}\,. \tag{2.10}$$

Radiant intensity I_e is the radiant flux $d\Phi_e$ emitted, reflected, transmitted or received per unit solid angle $d\Omega$. It is measured in watts per steradian (W/sr) and is given by [13]

$$I_e = \frac{d\Phi_e}{d\Omega}\,. \tag{2.11}$$

Irradiance E_e is the radiant flux $d\Phi_e$ received by a surface per unit area dA. It is expressed in watts per square meter (W/m^2) and it is given by [13]

$$E_e = \frac{d\Phi_e}{dA}\,. \tag{2.12}$$

When considering a point source, which obeys the inverse square law, a detector with area dA at a distance r from the source defines the solid angle $d\Omega = dA/r^2$. Considering this, it is possible to relate the radiant intensity and the irradiance by replacing $d\Phi_e$ from Equation 2.11 into Equation 2.12, resulting in [13]

$$E_e = \frac{I_e d\Omega}{dA} = \frac{I_e}{r^2}\,. \tag{2.13}$$

Radiance L_e is the radiant flux $d\Phi_e$ emitted, reflected, transmitted or received by a surface, per unit area dA_e per unit solid angle $d\Omega$. It is expressed in watts per steradian per square

meter ($\mathrm{W} \cdot \mathrm{sr}^{-1} \cdot \mathrm{m}^{-2}$) and it is given by [13]

$$L_e = \frac{d^2\Phi_e}{dA_e d\Omega}\,. \tag{2.14}$$

2.3.3. Photometry

Photometry performs analogous measurements to radiometry, but the former takes into account the visual perception of the human eye. It does this by taking the radiometric measurements and weight-averaging them over the visible spectrum. The weights in this averaging process correspond to the light sensitivity of the human eye in function of the wavelength. Radiant flux, radiant intensity, irradiance and radiance are analogous to the photometric measures known as luminous flux Φ_v, luminous intensity I_v, illuminance E_v and luminance L_v, respectively. These are measured in lumen (lm), lumen per steradian (lm/sr, or candela — cd), lumen per square meter ($\mathrm{lm/m^2}$, or lux — lx), and candela per square meter ($\mathrm{cd/m^2}$), respectively. In particular, the luminous flux is given by [13]

$$\Phi_v = K_m \int_{380\text{ nm}}^{780\text{ nm}} \Phi_e(\lambda) V(\lambda) d\lambda \tag{2.15}$$

where $\Phi_e(\lambda)$ is the spectral power distribution (SPD), that is, the radiated flux per unit wavelength, and $K_m = 683$ lm/W is the constant that establishes the relationship between the watt and the lumen. $V(\lambda)$ is the spectral sensitivity curve of the human eye. The other photometric measures are calculated by taking the weighted integral of their respective radiometric measures.

2.3.4. Lambertian sources

A light source or reflecting surface is said to be *Lambertian* when the emitted luminous intensity is proportional to the cosine of the azimuth angle θ, for all θ, namely [13]

$$I_v(\theta) = I_v(0) \cos\theta\,. \tag{2.16}$$

This is called the *Lambert's cosine law*. Similarly, a light source or reflecting surface which obeys

$$I_v(\theta) = I_v(0) \frac{m+1}{2} \cos^m\theta\,, \tag{2.17}$$

is said to be *generalized Lambertian*. The mode number m, also called the order of the Lambertian, is given by [13]

$$m = -\frac{1}{\log_2(\cos\theta_{1/2})} \tag{2.18}$$

where $\theta_{1/2}$ is the angle between the axis of maximum luminous intensity and the axis where the intensity is half of that maximum value, usually known as the *half-power angle*. Note that for $m = 1$ (and thus $\theta_{1/2} = 60°$), the source or surface is Lambertian.

2.3.5. Color mixing

Not all colors that humans perceive are generated by individual beams of light. Colors can mix to form other colors. There are two ways in which colors are mixed, additive mixing and subtractive mixing. In *additive mixing*, two or more light beams superimpose. The *primary colors* in additive mixing are red, green and blue (RGB), as when all three primary colors are superimposed in equal amounts, the result is white. In the absence of all three primary colors, the result is black. Secondary colors are obtained when two primary colors are superimposed, red and green results in yellow, green and blue results in cyan, and blue and red results in magenta. Yellow, cyan and magenta (YCM) are the three secondary colors in additive mixing. When two light beams superimpose, the original wavelengths are preserved. White light, being the mixture of RGB, cannot be expressed by a single wavelength [3].

In *subtractive mixing*, two or more substances (like paint or ink) are mixed. In this case, the three primary colors are yellow, cyan and magenta. The subtractive mixing of YCM in equal amounts results in black. RGB are the three secondary colors in subtractive mixing. Mixing yellow and cyan results in green, cyan and magenta results in blue, and magenta and yellow results in red [3].

2.3.6. CIE 1931 XYZ color space

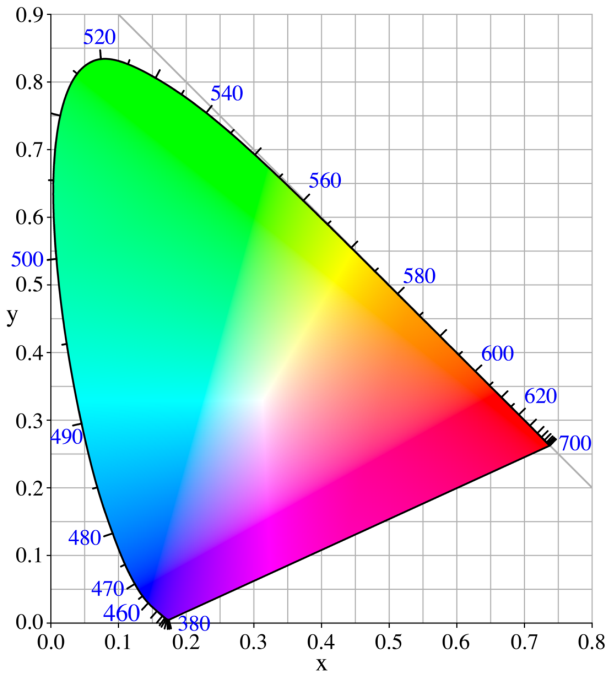


Figure 2.1: CIE 1931 xy color space diagram. Obtained from the public domain [14].

A *color model* is a mathematical representation of color, modeling all or most colors perceivable by humans as numbers. A *color space* is the multidimensional set of all colors which can be generated by a particular color model. The *CIE 1931 XYZ color space* or simply CIE 1931 color space, was proposed in 1931 by the International Commission on Illumination (CIE, for the French name Commission Internationale de l'Éclairage). It is

the first color space based on experimental results of human color perception, and it can represent all colors that the standard human eye can perceive. [3] The experimental basis of the standard comes from the three standard colorimetric observer functions $\bar{x}(\lambda), \bar{y}(\lambda)$ and $\bar{z}(\lambda)$, also called color-matching functions. These functions represent the average spectral responsivity of the human retina, determined by experiments with multiple test subjects, all of them with different colorimetric sensitivity [3].

The *tristimulus values* X, Y, and Z, for a given light source of radiance $L_e(\lambda)$ are given by [3]

$$\mathrm{X} = \int_{380\ \mathrm{nm}}^{780\ \mathrm{nm}} L_e(\lambda) \cdot \bar{x}(\lambda)\, d\lambda\,, \tag{2.19}$$

$$\mathrm{Y} = \int_{380\ \mathrm{nm}}^{780\ \mathrm{nm}} L_e(\lambda) \cdot \bar{y}(\lambda)\, d\lambda\,, \tag{2.20}$$

and

$$\mathrm{Z} = \int_{380\ \mathrm{nm}}^{780\ \mathrm{nm}} L_e(\lambda) \cdot \bar{z}(\lambda)\, d\lambda\,, \tag{2.21}$$

respectively. X and Z represent the *chrominance* and Y represents the *luminance* or brightness. This stems from the fact that $\bar{y}(\lambda)$ is identical to the spectral sensitivity function $V(\lambda)$. The chromaticity coordinates xyz are obtained by normalizing the tristimulus values XYZ: [3]

$$x = \frac{X}{X+Y+Z}\,, \quad y = \frac{Y}{X+Y+Z}\,, \quad z = \frac{Z}{X+Y+Z}\,. \tag{2.22}$$

Since $x+y+z = 1$, it is possible to express z in terms of x and y, as $z = 1-x-y$. This means that the chromaticity coordinates can be represented entirely in a two-dimensional plane, as shown in the chromaticity diagram presented in Figure 2.1, where y is plotted against x. This diagram is known as the *CIE 1931 xy chromaticity diagram*, and it is not a complete color space, but rather a two-dimensional projection of the CIE 1931 XYZ color space. If additionally to xy, one considers the luminance Y, then the entire CIE 1931 XYZ color space can be recovered: given x, y and Y, X can be recovered with $X = x \cdot Y/y$ and Z can be recovered with $Z = (1-x-y) \cdot Y/y$ [3]. The color space given by x, y and Y is called *CIE xyY color space*.

2.4. VLC Channel Modeling

Most VLC systems use intensity modulation with direct detection (IM/DD), due to its low cost and reduced complexity. In this type of system, the transmitters are LEDs whose instantaneous optical power $\Phi_e(t)$ is modulated in proportion to the driving electrical current $i_t(t)$, which in turn is modulated according the data to be transmitted. After passing through the channel, the optical power signal reaches the receiver surface, usually a photodiode, also known as a photodetector (PD). The received optical power induces a proportional photocurrent $i_r(t)$ in the photodiode.

2.4.1. Receiver: Photodiode

Photodiodes convert optical power signals into electrical signals, meaning that they are photoelectric transducers. For an incident optical power at a given wavelength $\Phi(\lambda)$, the photodiode will produce a proportional output current i_r. The proportionality constant at

said wavelength is called responsivity, $R(\lambda)$ and is given by [15]

$$R(\lambda) = \frac{i_r}{\Phi(\lambda)}\,, \tag{2.23}$$

and it is expressed in amperes per watt (A/W). The responsivity curve of a PD depends on the materials it is made of and its structure.

2.4.2. Transmitter: LED

Light emitting diodes are solid state semiconductor light sources that emit light when current flows through them. They do this thanks to the electroluminescence effect, where the forward current induces electrons in the semiconductor to recombine with electron holes, releasing energy in the form of photons. The wavelength of the emitted light is given by the energy required for electrons to cross the band gap of the semiconductor. There are two types of LEDs widely used to produce white light: phosphor LEDs and RGB LEDs. The former uses a blue LED and coats it with a yellow phosphor to produce white light. The latter uses the additive mixture of red, green and blue light sources [15].

2.4.3. Channel model

IM/DD VLC systems are usually modeled in their baseband equivalent, given by [16]

$$i_r(t) = \bar{R} \cdot h(t) * \Phi_e(t) + n(t)\,, \tag{2.24}$$

where $*$ is the convolution operation, $i_r(t)$ is the received current signal, $\bar{R}$ is the luminosity flux to current ratio for a given SPD, $\Phi_e(t)$ is the modulated optical power signal, $h(t)$ is the baseband channel impulse response (CIR) and $n(t)$ is the additive white Gaussian noise (AWGN). $\Phi_e(t)$ can be further expressed by [15]

$$\Phi_e(t) = h_{\mathrm{LED}}(t) * i_t(t)\,, \tag{2.25}$$

where $i_t(t)$ is the LED drive current and $h_{\mathrm{LED}}(t)$ is the current to luminous flux impulse response of the LED. $\bar{R}$ can be calculated by [15]

$$\bar{R} = \int_\lambda R(\lambda)\Phi(\lambda)\,d\lambda\,, \tag{2.26}$$

where $\Phi(\lambda)$ is the SPD of the LED. Since $\Phi_e(t)$ represents optical power, it is non-negative, i.e.

$$\Phi_e(t) \geq 0\,. \tag{2.27}$$

Additionally, the optical signal must be constrained for human-appropriate illumination. This translates to maintaining the average optical power of the source at a desired $P_{\mathrm{T,avg}}$, i.e.

$$P_{\mathrm{T,avg}} = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} \Phi_e(t)\,dt\,. \tag{2.28}$$

There is also a peak optical flux P_{peak}, which corresponds to the optical flux induced in the LED by the maximum drive current, [15]

$$\Phi_e(t) \leq P_{\text{peak}} \,. \tag{2.29}$$

2.4.4. Channel response

In general, VLC channels are comprised of a line-of-sight (LOS) component and a non-line-of-sight (NLOS) or diffuse component. The first component corresponds to the light beams that reach the PD directly, while the latter corresponds to the light beams that are reflected in one or more objects before reaching the PD. Consequently, the CIR $h(t)$ can be expressed as [3]

$$h(t) = h_{\text{LOS}}(t) + h_{\text{NLOS}}(t) \,. \tag{2.30}$$

2.4.4.1. LOS link

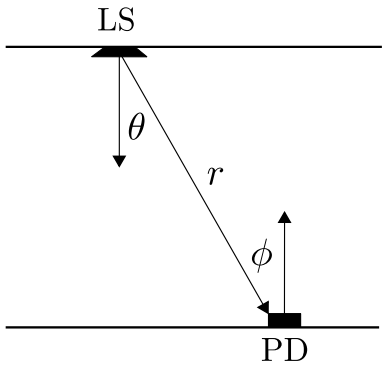


Figure 2.2: LOS link diagram. Reproduced from [3]

In order to obtain $h_{\text{LOS}}(t)$, it is sufficient to consider the most basic VLC link: a single light source (LS), which can be monochromatic or multichromatic, and a single PD, in an indoor free-space environment. These two components are shown in Figure 2.2. The optical received power $P_{\text{R,opt}}$ can be expressed, when considering LS a point source from the point of view of the PD, as [3]

$$P_{\text{R,opt}} = P_{\text{T,opt}} G_{\text{conc}} G_{\text{filter}} f(\theta, \theta_{1/2}) \frac{A_{\text{R,eff}}}{\pi r^2} \quad \text{for } d \gg \lambda \text{ and } A_R \gg \lambda^2 \,, \tag{2.31}$$

where $P_{\text{T,opt}}$ is the optical transmit power, $G_{\text{conc}} \geq 1$ is optical concentrator gain, $G_{\text{filter}} \leq 1$ is optical filtering loss, r is the distance between the LS and the PD, $\theta_{1/2}$ is the half-power angle of the light beam, A_{R} is the aperture area of the PD, and $A_{\text{R,eff}}$ is the effective aperture area of the PD such that [3]

$$A_{\text{R,eff}} = A_{\text{R}} \cos \phi \,. \tag{2.32}$$

The condition $d \gg \lambda$ stems from the point source assumption, while the condition $A_{\text{R}} \gg \lambda^2$ implies that the optical power detection process at the PD is deterministic. Note that when ϕ exceeds the field of vision (FOV) of the PD, ϕ_{POV}, then $P_{\text{R,opt}} = 0$ [3].

For generalized Lambertian sources, the directivity function $f(\cdot)$ can be expressed as [3]

$$f(\theta, \theta_{1/2}) = \frac{1}{2}(m+1)\cos^m(\theta)\,, \tag{2.33}$$

where $m = -\frac{1}{\log_2(\cos\theta_{1/2})}$. Optical concentrators are collimators, devices that focus light rays. Color filters are used for spectral shaping, polarization control, ambient light mitigation and interference suppression. G_{conc} is primarily angle-dependent, while G_{filter} is predominantly wavelength-dependent, but also angle-dependent [3].

From Equation 2.31, and considering, for conciseness, that $G_{\text{conc}}G_{\text{filter}} = 1$, it follows that $h_{\text{LOS}}(t)$ can be expressed as [3]

$$h_{\text{LOS}}(t) = \frac{P_{\text{R,opt}}}{P_{\text{T,opt}}}\delta\left(t - \frac{r}{c}\right) = f(\theta, \theta_{1/2})\frac{A_{\text{R,eff}}}{\pi r^2}\delta\left(t - \frac{r}{c}\right)\,, \tag{2.34}$$

where $\delta(\cdot)$ is the Dirac function.

2.4.4.2. NLOS link

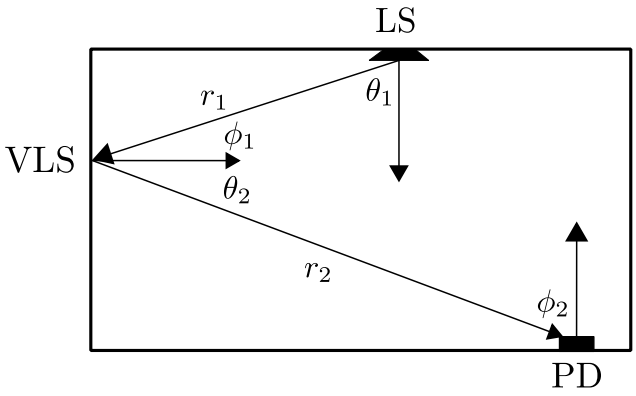


Figure 2.3: NLOS link diagram, one VLS, single-hop. Reproduced from [3].

In RF communications, multipath causes stochastic and time-varying signal distortion, causing microwave channels to be modeled as random. In VLC, however, the multipath channel is deterministic, since $A_{\text{R}} \gg \lambda^2$, or in other words, the PD captures the optical signal over an area millions greater than a square wavelength. As long as the objects in the room are fixed, the indoor VLC channel is time-invariant. Nonetheless, multipath propagation can still cause problems in VLC systems: at high data rates, it causes intersymbol interference [3].

In order to obtain $h_{\text{NLOS}}(t)$, it is convenient to start with a single reflector, as shown in Figure 2.3. This reflector acts as a *virtual light source* (VLS). Most reflections are diffuse, meaning that the angle of irradiance θ_2 is not necessarily the same as the angle of incidence ϕ_1. Additionally, reflections can usually be modeled as Lambertian. Denoting the distance between LS and VLS by r_1, and the distance between VLS and PD by r_2, and considering $G_{\text{conc}}G_{\text{filter}} = 1$, Equation 2.31 can be extended as [3]

$$P_{\text{R,opt}} = P_{\text{T,opt}} f(\theta_1, \theta_{1/2})\frac{A_{\text{refl,eff}}}{\pi r_1^2} \cdot \rho(\lambda) f(\theta_2, 60°)\frac{A_{\text{R,eff}}}{\pi r_2^2}\,, \tag{2.35}$$

for $d_1, d_2 \gg \lambda$ and $A_{\text{refl}}, A_{\text{R}} \gg \lambda^2$. If LS is considered generalized Lambertian, then

$$f(\theta_1, \theta_{1/2}) = \frac{1}{2}(m+1)\cos^m(\theta_1)\,. \tag{2.36}$$

Likewise, if VLS is considered Lambertian, then

$$f(\theta_2, 60°) = \cos(\theta_2)\,. \tag{2.37}$$

$A_{\text{refl,eff}}$ and $A_{\text{R,eff}}$ are the effective areas of VLS and PD, respectively, with [3]

$$A_{\text{refl,eff}} = A_{\text{refl}}\cos\phi_1 \tag{2.38}$$
$$A_{\text{R,eff}} = A_{\text{R}}\cos\phi_2\,. \tag{2.39}$$

$\rho(\lambda) \in [0, 1)$ is the reflection coefficient or reflectance, accounting for the light absorption of the reflective surface material. Note that when ϕ_2 exceeds the field of vision (FOV) of the PD, ϕ_{FOV}, then $P_{\text{R,opt}} = 0$. For multichromatic transmission, the SPD of the LS, $\Phi(\lambda)$ must be considered. Note that

$$P_{\text{T,opt}} = \int_\lambda \Phi(\lambda)\,d\lambda\,. \tag{2.40}$$

The wavelength-dependency of $\rho(\lambda)$ can be taken into account by defining

$$\Gamma = \int_\lambda \Phi(\lambda)\rho(\lambda)\,d\lambda\,. \tag{2.41}$$

With this, Equation 2.35 can be rewritten for the multichromatic case as [3]

$$P_{\text{R,opt}} = \Gamma f(\theta_1, \theta_{1/2})\frac{A_{\text{refl,eff}}}{\pi r_1^2} f(\theta_2, 60°)\frac{A_{\text{R,eff}}}{\pi r_2^2}\,. \tag{2.42}$$

From this, the impulse response for the single-hop single-reflector, can be expressed as

$$h^{(1)}_{\text{NLOS, mono}}(t) = f(\theta_1, \theta_{1/2})\frac{A_{\text{refl}}\cos\phi_1}{\pi r_1^2}\cdot\rho(\lambda) f(\theta_2, 60°)\frac{A_{\text{R}}\cos\phi_2}{\pi r_2^2}\cdot\delta\left(t - \frac{r_1 + r_2}{c}\right), \tag{2.43}$$

for the monochromatic case and [3]

$$h^{(1)}_{\text{NLOS, multi}}(t) = \frac{\Gamma}{P_{\text{T,opt}}} f(\theta_1, \theta_{1/2})\frac{A_{\text{refl}}\cos\phi_1}{\pi r_1^2} f(\theta_2, 60°)\frac{A_{\text{R}}\cos\phi_2}{\pi r_2^2}\cdot\delta\left(t - \frac{r_1 + r_2}{c}\right), \tag{2.44}$$

for the multichromatic case. In order to consider more reflectors, one can generalize Equations 2.43 and 2.44 for the n-th reflector, with $1, \ldots N$, where N is the total number of reflectors. The result is [3]

$$h^{(n)}_{\text{NLOS, mono}}(t) = f(\theta_{1,n}, \theta_{1/2})\frac{A_{\text{refl,n}}\cos\phi_{1,n}}{\pi r_{1,n}^2}\cdot\rho_n(\lambda) f(\theta_{2,n}, 60°)\frac{A_{\text{R}}\cos\phi_{2,n}}{\pi r_{2,n}^2}\cdot\delta\left(t - \frac{r_{1,n} + r_{2,n}}{c}\right), \tag{2.45}$$

for monochromatic LS and [3]

$$h^{(n)}_{\text{NLOS, multi}}(t) = \frac{\Gamma_n}{P_{\text{T,opt}}} f(\theta_{1,n}, \theta_{1/2}) \frac{A_{\text{refl,n}} \cos\phi_{1,n}}{\pi r^2_{1,n}} f(\theta_{2,n}, 60°) \frac{A_{\text{R}} \cos\phi_{2,n}}{\pi r^2_{2,n}} \cdot \delta\left(t - \frac{r_{1,n} + r_{2,n}}{c}\right) \tag{2.46}$$

for multichromatic LS. The total CIR of the NLOS scenario with single LS, multiple reflectors and single-hop reflections is given by

$$h_{\text{NLOS}}(t) = \sum_{n=1}^{N} h^{(n)}_{\text{NLOS}}(t)\,. \tag{2.47}$$

Note that in this sum, paths in which the reflected light beams reach the receiver FOV are considered.

Let us consider the case of a multi-hop NLOS link. Considering a general Lambertian LS and Lambertian reflectors, the CIR of a light beam that reaches the FOV of the receiver after k bounces is given by [17]

$$h^{(k)}_{\text{NLOS}}(t) = \frac{1}{P_{\text{T,opt}}} \int_{s\in\mathbb{S}} \left[L_1 L_2 \cdots L_{k+1} \Gamma^{(k)} \cdot \delta\left(t - \frac{d_1 + d_2 + \cdots + d_{k+1}}{c}\right)\right] dA_s\,, \tag{2.48}$$

where [17]

$$L_1 = \frac{A_s(m+1)\cos\phi_1 \cos^m\theta_1}{2\pi d_1^2} \tag{2.49}$$

$$L_{k+1} = \frac{A_{\text{R}} \cos\phi_{k+1} \cos\theta_{k+1}}{\pi d^2_{k+1}} \tag{2.50}$$

and in general, for $i = 2, \ldots, k$, [17]

$$L_i = \frac{A_s \cos\phi_i \cos\theta_i}{\pi d_i^2}\,. \tag{2.51}$$

The integration in Equation 2.48 goes through all the reflectors $s \in \mathbb{S}$, where A_s is the area of the surface of reflector s. $\Gamma^{(k)}$ is the optical power of the reflected light beam after the k-th bounce, and it is given by [17]

$$\Gamma^{(k)} = \int_\lambda \Phi(\lambda)\rho_1(\lambda)\rho_2(\lambda)\cdots\rho_k(\lambda)\,d\lambda \tag{2.52}$$

where $\rho_i(\lambda)$ is the reflectance of the surface at the i-th bounce, for $i = 1, \ldots, k$.

2.4.5. Channel noise

The total noise at the receiver is modeled as an AWGN process with variance σ^2. This variance can be expressed as [15]

$$\sigma^2 = \sigma^2_{\text{shot}} + \sigma^2_{\text{thermal}}\,, \tag{2.53}$$

where σ^2_{shot} accounts for the shot noise, which stems from the stochastic process of photon absorption and electron-hole recombination in semiconductors. Shot noise depends on the

received number of photons, and consequently, it is affected by the ambient lighting and the mean luminous intensity of the LED itself. $\sigma^2_{\text{thermal}}$ accounts for the thermal noise, which stems from the random, temperature dependent, vibration of charges in the receiver circuit. The optical and electrical SNR are defined by [15]

$$\text{SNR}_o = \frac{P_{\text{R,avg}}}{\sigma} \tag{2.54}$$

$$\text{SNR}_e = \frac{\bar{R}^2 P^2_{\text{R,avg}}}{\sigma^2} \tag{2.55}$$

where $P_{\text{R,avg}}$ represents the average received optical power, given by [15]

$$P_{\text{R,avg}} = H(0) P_{\text{T,avg}}\,, \tag{2.56}$$

where $H(0)$ is the DC channel gain, given by [15]

$$H(0) = \int_{-\infty}^{\infty} h(t)\, dt\,. \tag{2.57}$$

2.5. VLC Systems in Underground Mining Scenarios

Authors who have attempted to model a VLC channel in an underground mine environment typically assume that it can be modeled in the same way as an indoor VLC channel [2, 18, 19, 20], namely, a generalized Lambertian channel model. In general, effects such as *light scattering* due to dust, reflections off irregular walls, light obstruction, also known as *shadowing* and LED and PD tilting and rotation with respect to each other, are not taken into account or are not included in the CIR, and when they are, not all at once. In [4] it is argued that ignoring these effects does not allow a modeling of the mining channel that considers the complexities inherent in this environment, resulting in very optimistic expectations regarding the performance of the VLC system. This is why they propose a channel model that includes these effects explicitly in the channel transfer function, in addition to including the LED and photodetector rotation and tilt angles as parameters. Thus, in order to perform a simulation of a mining environment as close to reality as possible, the mining VLC channel model proposed by [4] will be used in this work. Explaining in mathematical detail the components of this channel model is out of the scope of this work, but a general overview should be sufficient to understand the important differences with respect to the traditional channel model. For a given position, rotation and tilt of a single LED and a single PD, the CIR of the VLC underground mining channel between these two components is expressed as [4]

$$h_{\text{mine}}(t) = h_{\text{LOS}_{(\text{sh})}}(t) + h^{(1)}_{\text{NLOS}_{(\text{sh})}}(t) + h_{\text{sca}}(t)\,. \tag{2.58}$$

What each of these components encompass is explained next.

2.5.1. LOS link in underground mining scenario

The $h_{\mathrm{LOS_{sh}}}(t)$ component represents the LOS component affected by shadowing and can be further expressed as [4]

$$h_{\mathrm{LOS}_{(\mathrm{sh})}}(t) = h_{\mathrm{LOS}_{(\mathrm{tilt})}}(t) \cdot \mathbb{P}_{\mathrm{LOS}}\,, \tag{2.59}$$

where $h_{\mathrm{LOS}_{(\mathrm{tilt})}}(t)$ is the commonly used VLC LOS channel model, but adjusted geometrically such that it accounts for the tilt and rotation of the LED and PD with respect to each other. $\mathbb{P}_{\mathrm{LOS}} \in [0, 1]$ is a weighting value proportional to the probability of the LOS link to not be obstructed by random obstacles. $\mathbb{P}_{\mathrm{LOS}}$ is inversely proportional to the average amount of shadowing experienced by the LOS link, and thus can be viewed as a deterministic expression to an otherwise stochastic process.

2.5.2. NLOS link in underground mining scenario

The $h^{(1)}_{\mathrm{NLOS}_{(\mathrm{sh})}}(t)$ component refers to the single-hop NLOS component, affected by shadowing. This model does not consider multi-hop since the contributions of further bounces after the first one are deemed negligible in an underground mine scenario. The single-hop NLOS component can be further expressed as [4]

$$h^{(1)}_{\mathrm{NLOS}_{(\mathrm{sh})}}(t) = \sum_{n=1}^{N} h^{(1,n)}_{\mathrm{NLOS}_{(\mathrm{tilt})}}(t) \cdot \mathbb{P}^{(n)}_{\mathrm{NLOS},1} \cdot \mathbb{P}^{(n)}_{\mathrm{NLOS},2}\,, \tag{2.60}$$

where N is the number of single-hop reflectors, $h^{(1,n)}_{\mathrm{NLOS}_{(\mathrm{tilt})}}(t)$ is the commonly used VLC single-hop NLOS channel model for the n-th reflector, but adjusted geometrically such that it accounts for the tilt and rotation of the LED and PD with respect to each other, and also for the non-flat irregular walls of the mine. The model accounts for the non-flat irregular walls by modeling the reflector surfaces as having random tilt and rotation angles, drawn from a uniform distribution in the range $[0°, 180°)$. $\mathbb{P}^{(n)}_{\mathrm{NLOS},1}, \mathbb{P}^{(n)}_{\mathrm{NLOS},2} \in [0, 1]$ are a weighting values proportional to the probability of the NLOS paths to not be obstructed by random obstacles. $\mathbb{P}^{(n)}_{\mathrm{NLOS},1}$ is inversely proportional to the average amount of shadowing experienced by the path between the LS and the n-th reflector, while $\mathbb{P}^{(n)}_{\mathrm{NLOS},2}$ is inversely proportional to the average amount of shadowing experienced by the path between the n-th reflector and the PD [4].

2.5.3. Scattering

Scattering is the reflection or absorption of photons by particles suspended in the atmosphere. This effect is not considered in traditional VLC models, since indoor scenarios do not have to deal with noticeable amounts of dust. In underground mines, however, the presence of dust is inevitable and ever-present, and it can potentially affect the VLC channel. This channel model accounts for the scattering effect by considering N dust particles, which are inside a two-dimensional disk around the PD, as a single-hop reflector. The aforementioned disk is contained in the 2D plane which contains both the LS and the PD and is perpendicular to the ground plane. Theoretically, the number of dust particles in this disk is

so large that the CIR may be expressed as [4]

$$h_{\text{sca}}(t) = \lim_{N\to\infty} \sum_{n=1}^{N} h^{(n)}_{\text{NLOS}_{(\text{sca})}}(t)\,, \tag{2.61}$$

namely, as the infinite sum of multiple single-hop reflections. Different atmospheric characteristics of the dust particles can be considered by selecting appropriate reflectances $\rho_n(\lambda)$.

2.5.4. Time-invariability of the channel model

Note that this channel model is modeled as time-invariant. This is due to the fact that the channel model accounts for highly dynamic, stochastic processes as deterministic variables in the form of averages. Nevertheless, the time-invariability assumption is only valid for a small period of time, since even the average of the stochastic variables change in such dynamic environments, although slower than the stochastic variable itself. This consideration will be important when simulating and estimating the channel-induced distortion in the following chapters.

2.6. The IEEE 802.15.7 Standard

The IEEE 802.15.7 standard defines the physical (PHY) and medium access control (MAC) layer for optimal short-range communications using visible light. The standard considers three modes of physical layer operation: [9]

- **PHY I:** This mode of operation is intended for out-door use with applications that require low data rates. There are two types of modulation supported in this mode: On-off keying (OOK) and variable pulse position modulation (VPPM).
- **PHY II:** This mode is intended for in-door use with moderate data rate requirement applications. This mode uses OOK and VPPM.
- **PHY III:** This mode is intended for applications using color-shift keying (CSK) with multiple light sources and detectors.

2.6.1. On-off keying

In on-off keying (OOK) modulation, the signal is transmitted by turning the LED on and off, making it the simplest method of transmitting a VLC signal. It is a type of amplitude modulation, where the carrier signal (the visible light) has two amplitude levels, 0 (W) and P_{peak}, representing the bits 0 and 1 respectively. In this modulation scheme, the flickering of the lights has to be control in order to not disrupt the illumination capabilities, since rapid change of brightness can cause detrimental physiological effects in humans [21]. The IEEE standard allows for data rates from 11.67 kbps up to 100 kbps in PHY I, and 6 Mbps up to 96 Mbps in PHY III, with this type of modulation [9].

2.6.2. Variable pulse position modulation

Variable pulse position modulation (VPPM) is a modulation method in which symbols are represented according to the position of the light pulse within the symbol period. Binary 0 occurs when the pulse is aligned to the left of the symbol period and binary 1 occurs when the pulse is aligned to the right of the symbol period. By modulating each pulse duration, dimming control is achieved, meaning that $P_{\mathrm{T,avg}}$ can be decreased or increased according to the illumination requirements. The average optical power of the binary 1 and binary 0 are equal. The advantage of this modulation over OOK is that it offers protection against light flicker, since each symbol, for a given level of attenuation, has the same average brightness [21], while in OOK the average optical power for the binary 1 is P_{peak}, and for the binary 0 is 0 (W). The IEEE standard allows for data rates from 35.65 kbps up to 266.6 kbps in PHY I, and 1.25 Mbps up to 5 Mbps in PHY II, with this type of modulation [9].

2.6.3. Color-shift keying

The color-shift keying (CSK) modulation method is implemented using three RGB LEDs, each transmitting a primary color, red, green or blue, and three photodetectors with optical filters that let through one of the primary colors each. The symbols are encoded in the CIE 1931 xy color space so that the luminance of the light source remains constant while the instantaneous chromaticity varies. The data is sent in the instantaneous color of the RGB triplet, but the perceived average chromaticity is kept at a constant white light. The constant luminance guarantees the absence of light flicker at all data rates [15]. The IEEE standard allows for data rates from 35.65 kbps up to 266.6 kbps in PHY I, and 1.25 Mbps up to 5 Mbps in PHY II, with this type of modulation [9]. The IEEE standard allows for data rates from 12 Mbps up to 24 Mbps with 4-CSK, and 18 Mbps up to 72 Mbps with 8-CSK and 48 Mbps up to 96 Mbps with 8-CSK [9].

Chapter 3

Methods

3.1. Color-Shift Keying

Color-shift keying will be further explained, since it is the modulation scheme used in this work. A CSK-based VLC system consists of three LEDs and three PDs, with each LED-PD pair being configured as to transmit and receive light signals from a color band centered around a primary color. The center of the color bands can be expressed in CIE 1931 xy space coordinates as $\mathbf{s}_{\mathrm{R}}, \mathbf{s}_{\mathrm{G}}, \mathbf{s}_{\mathrm{B}} \in \mathbb{R}^2$, known as the *center of band symbols*. All the colors that can be reproduced by the LEDs via additive mixing form a triangle in the CIE xy space, with $\mathbf{s}_{\mathrm{R}}, \mathbf{s}_{\mathrm{G}}$ and $\mathbf{s}_{\mathrm{B}}$ being the vertices. This set of all reproducible colors by the three LEDs is called the *gamut* of the system and can be expressed mathematically as the convex combination of the center of band symbols in the CIE xy space, as in

$$\mathcal{G} = \{\alpha_1 \mathbf{s}_{\mathrm{R}} + \alpha_2 \mathbf{s}_{\mathrm{G}} + \alpha_3 \mathbf{s}_{\mathrm{B}} \in \mathbb{R}^2 \mid \alpha_1 + \alpha_2 + \alpha_3 = 1, \quad \alpha_1, \alpha_2, \alpha_3 \geq 0\}\,, \tag{3.1}$$

where $\mathcal{G}$ is the gamut of the system. In order to modulate binary information, the *modulation alphabet* must be defined such that binary information can be encoded into distinct symbols. Given the *modulation order* M, such that $M = 2^m$ and $m \in \{2, 3, 4, \dots\}$, the modulation alphabet defines the matching up of all M possible binary sequences of length m to M specific symbols taken from $\mathcal{G}$, in a one-to-one correspondence. In mathematical terms, the modulation alphabet is defined by the bijective function $\mathcal{A}\colon \{0,1\}^{1\times m} \to \mathcal{K} \subset \mathcal{G}$, where $\mathcal{K} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_M\}$ is known as the *alphabet symbol set* or *constellation set*. Note that $\mathbf{s}_i \in \mathcal{G}$ is a distinct symbol from all the others in $\mathcal{K}$, for $i = 1, \dots, M$. The existence of alphabet function $\mathcal{A}$ implies that

$$\forall \mathbf{b} \in \{0,1\}^{1\times m},\, \exists k \in \{1 \dots, M\}, \quad \mathcal{A}(\mathbf{b}) = \mathbf{s}_k\,, \tag{3.2}$$

and

$$\forall k \in \{1 \dots, M\},\, \exists \mathbf{b} \in \{0,1\}^{1\times m},\, \mathcal{A}^{-1}(\mathbf{s}_k) = \mathbf{b}\,, \tag{3.3}$$

where $\mathcal{A}^{-1}$ is the inverse function of $\mathcal{A}$ and $\mathbf{s}_k \in \mathcal{K}$. Once the constellation set $\mathcal{K}$, the alphabet function $\mathcal{A}$ and the center of band symbols $\{\mathbf{s}_{\mathrm{R}}, \mathbf{s}_{\mathrm{G}}, \mathbf{s}_{\mathrm{B}}\}$ are defined, then the M-CSK modulation scheme has been completely defined.

3.1.1. Standard CSK constellations

The IEEE 802.15.7 standard defines some guidelines for designing M-CSK constellations, i.e. selecting the set $\mathcal{K}$ consisting of M constellation symbols. The constellation for 4-CSK is presented in Figure 3.1. The vertices RGB correspond to the center of the three primary color bands on the CIE 1931 xy space. $\{\mathbf{s}_1, \ldots, \mathbf{s}_4\}$ are the 4-CSK constellation symbols expressed in xy coordinates. $\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3$ are the vertices of the RGB triangle, while $\mathbf{s}_4$ is its centroid [9].

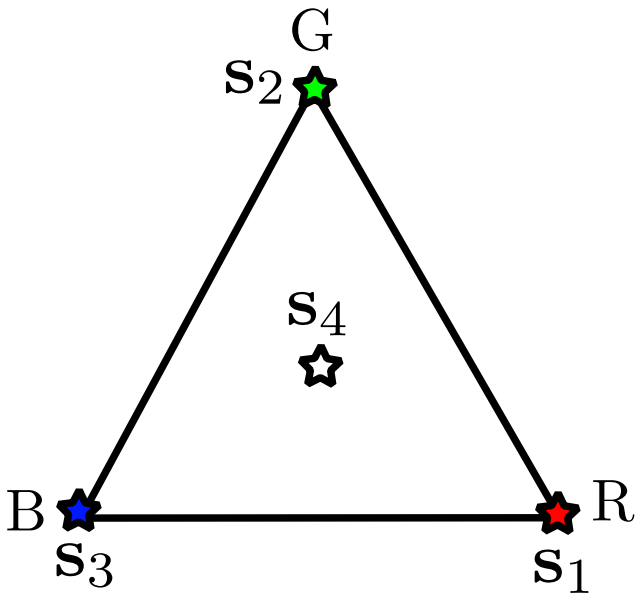


Figure 3.1: 4-CSK constellation.

The constellation for 8-CSK is presented in Figure 3.2. $\mathbf{s}_1, \ldots, \mathbf{s}_8$ are the 8-CSK constellation symbols expressed in xy coordinates. $\mathbf{s}_1, \mathbf{s}_4, \mathbf{s}_8$ are the vertices of the RGB triangle. $\mathbf{s}_2$ and $\mathbf{s}_3$ are points that divide side BG and side RG in the ratio 1:2. Point L and M are the midpoints of lines BG and RG. $\mathbf{s}_6$ is the midpoint of the line BR. Point O is the centroid of triangle R-M-$\mathbf{s}_6$. Point N is the centroid of triangle B-L-$\mathbf{s}_6$. $\mathbf{s}_7$ is the point that divides line OM in the ratio 1:2. $\mathbf{s}_5$ is the point that divides line NL in the ratio 1:2 [9].

The constellation for 16-CSK is presented in Figure 3.3. $\mathbf{s}_1, \ldots, \mathbf{s}_{16}$ are the 16-CSK symbols expressed in xy coordinates. $\mathbf{s}_1, \mathbf{s}_4, \mathbf{s}_7$ are the vertices of the RGB triangle. $\mathbf{s}_2$ and $\mathbf{s}_3$ are points that divide line RG in three equal parts. $\mathbf{s}_5$ and $\mathbf{s}_6$ are points that divide line BG in three equal parts. $\mathbf{s}_8$ and $\mathbf{s}_9$ are points that divide line BR in three equal parts. $\mathbf{s}_{16}$ is the centroid of triangle RGB. $\mathbf{s}_{10}, \mathbf{s}_{11}, \mathbf{s}_{12}, \mathbf{s}_{13}, \mathbf{s}_{14}$ and $\mathbf{s}_{15}$ are the centroids of each of the smaller triangles [9].

3.1.2. Modulation

Given a M-CSK system with alphabet function $\mathcal{A}$, constellation set $\mathcal{K} = \{\mathbf{s}_1, \ldots, \mathbf{s}_M\}$ and center of band symbols $\{\mathbf{s}_\mathrm{R}, \mathbf{s}_\mathrm{G}, \mathbf{s}_\mathrm{B}\}$, then any binary sequence of length m, $\mathbf{b} \in \{0,1\}^{1\times m}$, can be encoded as a CIE 1931 xy space symbol $\mathbf{s}_k \in \mathcal{K}$, with $k \in \{1, \ldots, M\}$, given by $\mathbf{s}_k = \mathcal{A}(\mathbf{b})$. Let us denote the explicit CIE xy coordinates of symbol $\mathbf{s}_k$ as $\mathbf{s}_k = [x_k, y_k]^\mathrm{T}$.

If constant luminance is considered, then each CSK symbol can be expressed as both a two-dimensional CIE xy coordinate and as a three-dimensional radiant flux vector. The symbol $\mathbf{s}_k$ can be expressed as radiant flux vector $\mathbf{\Phi} = [\Phi_\mathrm{R}, \Phi_\mathrm{G}, \Phi_\mathrm{B}]^\mathrm{T} \in \mathbb{R}^3_{\geq 0}$, where $\Phi_\mathrm{R}, \Phi_\mathrm{G}, \Phi_\mathrm{B}$

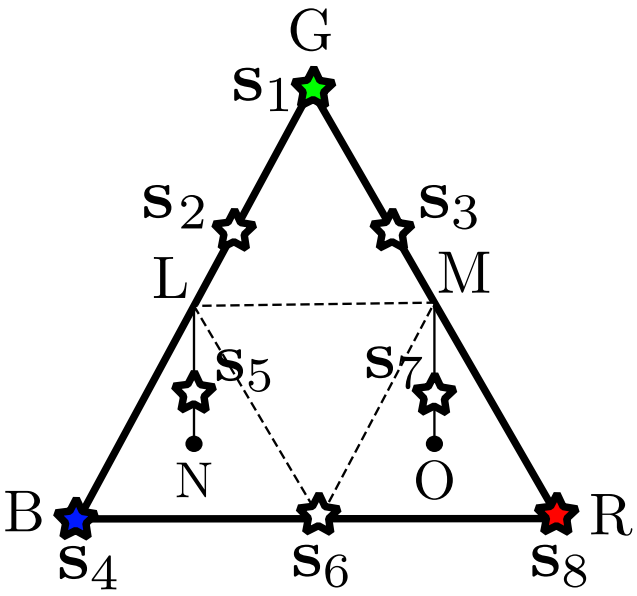


Figure 3.2: 8-CSK constellation.

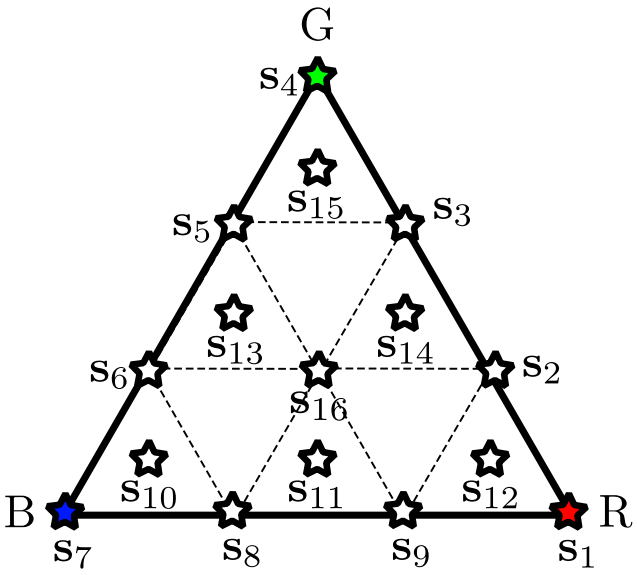


Figure 3.3: 16-CSK constellation.

represent the radiant flux to be emitted by the red, green and blue LED respectively. The radiant flux vector $\mathbf{\Phi}$ can be obtained by solving the system of equations [9]

$$\begin{aligned} x_k &= \Phi_\mathrm{R} x_\mathrm{R} + \Phi_\mathrm{G} x_\mathrm{G} + \Phi_\mathrm{B} x_\mathrm{B} \\ y_k &= \Phi_\mathrm{R} y_\mathrm{R} + \Phi_\mathrm{G} y_\mathrm{G} + \Phi_\mathrm{B} y_\mathrm{B} \\ 1 &= \Phi_\mathrm{R} + \Phi_\mathrm{G} + \Phi_\mathrm{B}\,. \end{aligned} \tag{3.4}$$

Note that this transformation works the other way around too, meaning that for a given radiant flux vector $\mathbf{\Phi}$, using the Equations 3.4, it is possible to find $\mathbf{s}_k = [x_k, y_k]^\mathrm{T}$.

3.1.3. Color calibration: Least squares equalizer

Due to the potential overlapping of SPDs of each LED and the responsivities of the PDs in conjunction with their optical filter, the overall system is subject to chromacity distortions. In the IEEE standard, a color calibration scheme, i.e. an equalizer, is recommended to reduce the color distortion. If one considers only the LOS component of the channel, then the baseband equivalent of the system can be written as [15]

$$\hat{\mathbf{\Phi}}^{(i)} = \mathbf{H}\mathbf{\Phi}^{(i)} + \mathbf{n}^{(i)} \tag{3.5}$$

where $\mathbf{\Phi^{(i)}} \in \mathbb{R}^3_{\geq 0}$ is the i-th transmitted symbol (expressed as a radiant flux vector), $\mathbf{n}^{(i)}$ is the AWGN noise, $\hat{\mathbf{\Phi}}^{(i)}$ is the i-th received symbol (also expressed as a radiant flux vector) and $\mathbf{H} \in \mathbb{R}^{3\times 3}$ is the channel matrix. One can estimate this matrix via channel estimation sequences and then the resulting estimation can be used to equalize the received symbols, as

$$\mathbf{z}^{(i)} = \hat{\mathbf{H}}^{-1}\hat{\mathbf{\Phi}}^{(i)} \tag{3.6}$$

where $\mathbf{z}^{(i)}$ is the equalized i-th symbol and $\hat{\mathbf{H}}$ is the estimated channel matrix. The IEEE standard defines a channel estimation sequence of 24 symbol periods: each color band sends a Walsh code of code length 4. Each bit of the Walsh code is transmitted for two symbol periods. For the red band, the Walsh code is $\boldsymbol{w}_{\mathrm{R}} = \{1, 0, 1, 0\}$, for the green band, the Walsh code is $\boldsymbol{w}_{\mathrm{G}} = \{1, 1, 0, 0\}$ and for the blue band, the Walsh code is $\boldsymbol{w}_{\mathrm{B}} = \{1, 0, 0, 1\}$. For each band, the corresponding Walsh code is transmitted twice in the estimation sequence. Considering one symbol period per Walsh bit, the channel estimation sequence can be expressed as [9]

$$\mathbf{w} = P_{\mathrm{T,avg}} \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3\times 12}_{\geq 0}, \tag{3.7}$$

In order to have two symbol periods per Walsh bit, it suffices to repeat each symbol of $\mathbf{w}$. The channel estimation sequence, as defined in the IEEE standard, is then given by $\mathbf{W} \in \mathbb{R}^{3\times 24}_{\geq 0}$, with its i-th symbol being

$$\mathbf{W}^{(i)} = \mathbf{w}^{(\lceil i/2 \rceil)}, \tag{3.8}$$

for $i = 1, \ldots, 24$. With the estimation sequence now established, consider $\hat{\mathbf{W}}$ to be the received channel estimation sequence, with

$$\hat{\mathbf{W}} = \mathbf{H}\mathbf{W} + \mathbf{N} \tag{3.9}$$

where $\mathbf{N}$ is the AWGN noise. Note that for a perfect estimation, one would like to find $\mathbf{H}$ such that

$$\hat{\mathbf{W}} = \mathbf{H}\mathbf{W}. \tag{3.10}$$

Unfortunately, this is an overdetermined system, meaning that there are more equations than unknowns. This means that, at best, one can hope to find $\mathbf{H}$ such that $||\hat{\mathbf{W}} - \mathbf{H}\mathbf{W}||^2$ is minimized. The classical least squares method provides an approximated solution for over-determined systems, given by

$$\hat{\mathbf{H}} = \hat{\mathbf{W}}\mathbf{W}^{\dagger}, \tag{3.11}$$

where $\mathbf{W}^{\dagger}$ denotes the Moore-Penrose pseudoinverse of the matrix $\mathbf{W}$ and $\hat{\mathbf{H}}$ denotes the estimated channel matrix via least squares. The equalization method consists of applying the inverse of $\hat{\mathbf{H}}$ to the received symbols, as shown in Equation 3.6.

3.1.4. Demodulation

Given a M-CSK system with alphabet function $\mathcal{A}$, constellation set $\mathcal{K} = \{\mathbf{s}_1, \ldots, \mathbf{s}_M\}$ and center of band symbols $\{\mathbf{s}_{\mathrm{R}}, \mathbf{s}_{\mathrm{G}}, \mathbf{s}_{\mathrm{B}}\}$, then any three-dimensional radiant flux vector $\mathbf{\Phi} = [\Phi_{\mathrm{R}}, \Phi_{\mathrm{G}}, \Phi_{\mathrm{B}}]^{\mathrm{T}} \in \mathbb{R}^3_{\geq 0}$ can be decoded into a binary sequence of length m. First, $\mathbf{\Phi}$ is converted into its CIE xy space equivalent via Equations 3.4, resulting in symbol $\mathbf{c} = [x, y]^{\mathrm{T}}$. Note that, in general, $\mathbf{c} \notin \mathcal{K}$, and thus function $\mathcal{A}^{-1}$ cannot be directly applied on it to find the corresponding binary sequence. Instead, the constellation symbol closest to $\mathbf{c}$ in the CIE xy space will be selected as the symbol from which to extract the binary sequence by the system:

$$\mathbf{s}_k = \underset{\mathbf{s} \in \mathcal{K}}{\operatorname{argmin}} \|\mathbf{s} - \mathbf{c}\|\,, \tag{3.12}$$

where $\mathbf{s}_k \in \mathcal{K}$, with $k \in \{1, \ldots, M\}$, is the constellation symbol the system chooses to decode the binary sequence from. In effect, the binary sequence encoded by $\mathbf{\Phi}$ is estimated by the system to be $\mathcal{A}^{-1}(\mathbf{s}_k)$.

3.2. Extreme Learning Machine

Extreme learning machine (ELM) is a machine learning algorithm, based on the architecture of artificial neural networks. Neural networks are machine learning models that usually require multiple iterations in the training phase: it feeds the training data forward into the network from input to output, and then it updates its internal parameters backwards, in a process known as *backpropagation*. Due to its iterative nature, this training method is a computationally expensive process. ELM promises to perform reasonably well but at a much lower cost, since its training process is not iterative, but rather involves solving a linear system (underconstrained or overdetermined, depending on the hyperparameter selection).

Given a training set $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_{\mathcal{N}}]^{\mathrm{T}} \in \mathbb{R}^{\mathcal{N} \times \mathcal{L}}$ with a corresponding label set $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_{\mathcal{N}}]^{\mathrm{T}} \in \mathbb{R}^{\mathcal{N} \times \mathcal{P}}$, then $\mathcal{L}$ denotes the number of input nodes of the network, $\mathcal{P}$ denotes the number of output nodes of the network and $\mathcal{N}$ denotes the number of training examples. The output of the ELM algorithm is denoted by $\mathbf{Y} \in \mathbb{R}^{\mathcal{N} \times \mathcal{P}}$ and can be found with [6]

$$\mathbf{Y} = \mathbf{\Lambda}\boldsymbol{\beta}\,, \tag{3.13}$$

where $\mathbf{\Lambda} \in \mathbb{R}^{\mathcal{N} \times \mathcal{K}}$ is the hidden layer output matrix and $\boldsymbol{\beta} \in \mathbb{R}^{\mathcal{K} \times \mathcal{P}}$ is the output weights matrix. $\mathcal{K}$ is the number of hidden neurons. $\mathbf{\Lambda}$ contains all the outputs from the hidden layer, which can be expressed as [6]

$$\mathbf{\Lambda} = \begin{bmatrix} g(\boldsymbol{\omega}_1^T \mathbf{x}_1 + \mu_1) & \cdots & g(\boldsymbol{\omega}_{\mathcal{K}}^T \mathbf{x}_1 + \mu_{\mathcal{K}}) \\ \vdots & \ddots & \vdots \\ g(\boldsymbol{\omega}_1^T \mathbf{x}_{\mathcal{N}} + \mu_1) & \cdots & g(\boldsymbol{\omega}_{\mathcal{K}}^T \mathbf{x}_{\mathcal{N}} + \mu_{\mathcal{K}}) \end{bmatrix} \tag{3.14}$$

where $\boldsymbol{\omega} = [\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_{\mathcal{K}}]^{\mathrm{T}} \in \mathbb{R}^{\mathcal{K} \times \mathcal{L}}$ is the input weights matrix, $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_{\mathcal{K}}]^{\mathrm{T}} \in \mathbb{R}^{\mathcal{K}}$ is the bias vector.

3.2.1. Training phase

In the training phase, $\boldsymbol{\omega}$ and $\boldsymbol{\mu}$ are randomly initialized. The objective of the training phase is to find $\boldsymbol{\beta}$ such that $\mathbf{T} = \boldsymbol{\Lambda}\boldsymbol{\beta}$. If $\mathcal{N} \geq \mathcal{K}$, i.e. there are more training examples than hidden neurons, then the linear system $\mathbf{T} = \boldsymbol{\Lambda}\boldsymbol{\beta}$ is *overdetermined* and there is no solution. In this case, the best one can hope for is to find $\boldsymbol{\beta}$ such that $||\mathbf{T} - \boldsymbol{\Lambda}\boldsymbol{\beta}||^2$ is minimized [22]. If $\mathcal{N} < \mathcal{K}$, i.e. there are more hidden neurons than training examples, then the linear system $\mathbf{T} = \boldsymbol{\Lambda}\boldsymbol{\beta}$ is *underconstrained* and there are multiple solutions. In this case, the best one can hope for is to find $\boldsymbol{\beta}$ such that $||\boldsymbol{\beta}||^2$ is minimized [22]. In any case, the optimal solution, $\hat{\boldsymbol{\beta}}$ can always be expressed as [22]

$$\hat{\boldsymbol{\beta}} = \boldsymbol{\Lambda}^{\dagger}\mathbf{T}\,, \tag{3.15}$$

where $(\cdot)^{\dagger}$ denotes the generalized Moore-Penrose pseudoinverse.

3.2.2. Testing phase

For the testing set $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_{\mathcal{Q}}]^{\mathrm{T}} \in \mathbb{R}^{\mathcal{Q}\times\mathcal{L}}$, the predicted output $\hat{\mathbf{Y}} \in \mathbb{R}^{\mathcal{Q}\times\mathcal{P}}$ is given by

$$\hat{\mathbf{Y}} = \begin{bmatrix} g(\boldsymbol{\omega}_1^T\mathbf{u}_1 + \mu_1) & \cdots & g(\boldsymbol{\omega}_{\mathcal{K}}^T\mathbf{u}_1 + \mu_{\mathcal{K}}) \\ \vdots & \ddots & \vdots \\ g(\boldsymbol{\omega}_1^T\mathbf{u}_{\mathcal{Q}} + \mu_1) & \cdots & g(\boldsymbol{\omega}_{\mathcal{K}}^T\mathbf{u}_{\mathcal{Q}} + \mu_{\mathcal{K}}) \end{bmatrix} \hat{\boldsymbol{\beta}}\,. \tag{3.16}$$

3.2.3. Regression and classification in ELM

ELM can be used to solve both classification or regression problems. In case of a regression problem, it suffices to follow Equation 3.15 for training and then Equation 3.16 for obtaining regression outputs from inputs the model has not been trained on. For a classification problem, some pre-processing must be done in the training data. In a classification problem, it is assumed that the label set has the form $\mathbf{T} = [t_1, \ldots, t_{\mathcal{N}}]^{\mathrm{T}} \in \{1, 2, \ldots, \mathcal{Z}\}^{\mathcal{N}}$, where $\mathcal{Z}$ is the number of classes. These labels are transformed into a more convenient form via *one-hot encoding*. The one-hot encoded labels are arranged in matrix $\mathbf{B} = [B_{ij}]_{\mathcal{N}\times\mathcal{Z}} \in \{0, 1\}^{\mathcal{N}\times\mathcal{Z}}$, which has the form

$$B_{ij} = \begin{cases} 1, & \text{if } t_i = j \\ 0 & \text{otherwise}\,. \end{cases} \tag{3.17}$$

Now $\mathbf{B}$ can be used as the new label set instead of $\mathbf{T}$. Suppose that once trained, the ELM is given as input $\mathbf{F} = [\mathbf{f}_1, \ldots, \mathbf{f}_{\mathcal{M}}]^{\mathrm{T}} \in \mathbb{R}^{\mathcal{M}\times\mathcal{O}}$ such that the model output is a matrix in the form $\mathbf{M} = [M_{uv}]_{\mathcal{M}\times\mathcal{Z}} \in \mathbb{R}^{\mathcal{M}\times\mathcal{Z}}$, encoded in one-hot encoding. In order to get the labels back in their numerical form, the following equation is used:

$$g_u = \underset{v}{\operatorname{argmax}}\, M_{uv} \quad \text{for } u = 1, \ldots, \mathcal{M}\,, \tag{3.18}$$

where g_u is the predicted numerical label for input $\mathbf{f}_u$. With this, the predicted labels, in numerical form, for input $\mathbf{F}$, are given by $\mathbf{G} = [g_1, \ldots, g_{\mathcal{M}}]^{\mathrm{T}} \in \{1, 2, \ldots, \mathcal{Z}\}^{\mathcal{M}}$.

3.2.4. Semi-supervised and unsupervised ELM

While the original ELM algorithm is intended to use a training set with labels, modifications have been proposed in the literature where only a part of the training set is labeled or none at all [23]. The version of the ELM algorithm that uses a labeled training set will be referred from now on as *supervised ELM* (SELM), the one where only a part of the training set is labeled, is known as *semi-supervised ELM* (SSELM) and the one that exclusively uses unlabeled data is known as *unsupervised ELM* (USELM). Explaining the SSELM and USELM algorithms is out of the scope of this work, but the details of their functioning can be found on [23], the original paper where they were proposed.

3.3. Simulation of VLC System in Underground Mining Scenario

In order to compare the performance of different equalization schemes, one would like to calculate the bit error ratio (BER) of the system at different noise levels, which results in the BER curves of the system. These curves are the main metric in which different digital communication systems are compared. Obtaining the BER curves for a VLC system operating in an underground mine can be complicated. For instance, one could build an experimental setup either in an actual underground mine or some kind of chamber which simulates an underground mine environment. Alternatively, one could perform a hardware-in-the-loop simulation, where channel parameters are programmed into an embedded system. Both of these approaches are expensive and time-consuming, mainly because they require the deployment of costly and complex hardware. Consequently, they are out of the scope of this work. Instead, in this work, the entire VLC system will be simulated in software, namely, in MATLAB. Channel models for VLC systems in underground mine scenarios exist in the literature, and are relatively straight-forward to implement in software. Of note is the channel model proposed in [4], which accounts for elements such as scattering, shadowing, LED and PD tilting and irregular walls, which are not found in other channel models, at least not all at once.

In the following sections, a detailed description of the simulated VLC systems created to obtain the BER curves is presented.

3.3.1. Simulation overview

The communication system simulated in this work implement a simplified version of the IEEE 802.15.7 PHY protocol — namely, PHY frames are made up of nothing more than the raw binary data to be transmitted and certain binary sequences that allow the receiver to perform the channel equalization. There is no synchronization header because it is assumed that transmitter and receiver are perfectly synchronized. There are no checksums, channel coding or any other error correction methods, since the objective is to evaluate how well the data is received after purely performing equalization.

Consider a M-CSK system with alphabet function $\mathcal{A}$, constellation set $\mathcal{K} = \{\mathbf{s}_1, \ldots, \mathbf{s}_M\}$ and center of band symbols $\{\mathbf{s}_\mathrm{R}, \mathbf{s}_\mathrm{G}, \mathbf{s}_\mathrm{B}\}$, where $M = 2^m$. In each simulation execution of this system, a random binary sequence $\mathbf{a} = (a^{(1)}, a^{(2)}, \ldots, a^{(n_a)}) \in \{0,1\}^{1\times n_a}$ of length n_a is transmitted. From this transmission, binary sequence $\hat{\mathbf{a}} = (\hat{a}^{(1)}, \ldots, \hat{a}^{(n_a)}) \in \{0,1\}^{1\times n_a}$ is

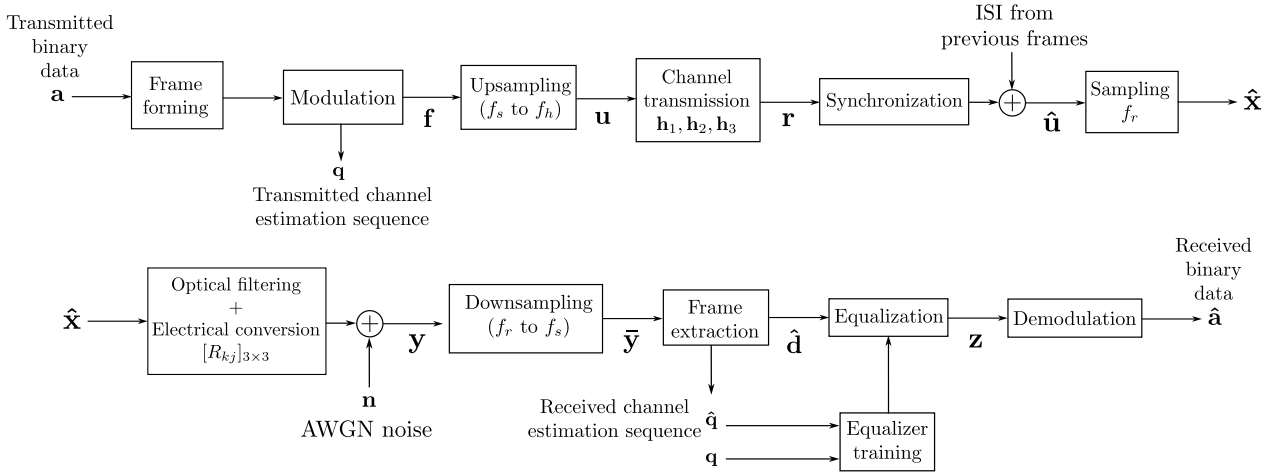


Figure 3.4: Simulation diagram.

recovered at the receiver. By counting the number different bits between $\mathbf{a}$ and $\hat{\mathbf{a}}$, the bit error ratio (BER) is calculated. Figure 3.4 presents a diagram that attempts to summarize the following sections, meant to be used as a visual guide. The code used in this work for simulating the VLC system and obtaining the BER values is open source; for more information, the interested reader can consult the Annex.

3.3.2. Frame forming

Before transmission, $\mathbf{a}$ is formatted into a simplified form of the IEEE 802.15.7 PHY frame, consisting of a channel estimation sequence and the data. Denoting by ℓ_a the number of data bits allowed in each frame, then the number of frames to be transmitted is given by

$$N_f = n_a/\ell_a\,, \tag{3.19}$$

where n_a is chosen so that N_f is an integer. Now consider that the channel estimation sequence is composed of n_s symbols. Considering a M-CSK modulation, where $m = \log_2 M$ denote the bits per symbol, then the bit length of the channel estimation sequence ℓ_s, is given by

$$\ell_s = m \times n_s\,. \tag{3.20}$$

By denoting as $\mathbf{q} = (\mathbf{q}^{(1)}, \mathbf{q}^{(2)}, \ldots, \mathbf{q}^{(n_s)}) \in \mathbb{R}_{\geq 0}^{3\times n_s}$ the modulated channel estimation sequence, then the binary channel estimation sequence is denoted by $\mathbf{c} = (c^{(1)}, c^{(2)}, \ldots, c^{(\ell_s)}) \in \{0,1\}^{1\times \ell_s}$, where

$$(c^{(i)}, c^{(i+1)}, \ldots, c^{(i+m-1)}) = \mathcal{A}^{-1}(\mathbf{q}^{(i)})\,, \tag{3.21}$$

for $i = 1, \ldots, n_s$. This means that one can design the channel estimation sequence as a modulated signal i.e. as a sequence of optical power vectors, and then convert this sequence into a binary sequence. The k-th frame, $\mathbf{o}_{(k)} \in \{0,1\}^{1\times \ell_f}$, where $\ell_f = \ell_s + \ell_a$, can be expressed as the binary sequence

$$\mathbf{o}_{(k)} = (o_{(k)}^{(1)}, \ldots, o_{(k)}^{(\ell_f)}) = (c^{(1)}, \ldots, c^{(\ell_s)}, a^{(k)}, a^{(k+1)}, \ldots, a^{(k+\ell_a-1)})\,. \tag{3.22}$$

In this step of the process, the binary sequence $\mathbf{a}$ is transformed into a set of N_f binary sequences of length ℓ_f, $\{\mathbf{o}_{(1)}, \mathbf{o}_{(2)}, \ldots, \mathbf{o}_{(N_f)}\}$ called frames. Note that depending on the equalization method used, the channel estimation sequence $\mathbf{q}$ is going to be different. Namely, in this work there are three different channel estimation sequences that were used, denoted by $\mathbf{q}_{\text{LS}}$, $\mathbf{q}_{\text{ELM}}$ and $\mathbf{q}_{\text{USELM}}$, depending on the equalization method. From now on, all M-CSK symbols are considered to be a three-dimensional non-negative optical power vector, unless explicitly stated otherwise.

3.3.2.1. Channel estimation sequence for LS

Following the IEEE standard, the channel estimation sequence for LS consists in three successive Walsh code sequences of length 4, with each symbol being transmitted twice. This translates into $n_s = 24$. This channel estimation sequence was already defined in Equation 3.8. One can simply define

$$\mathbf{q}_{\text{LS}} = \mathbf{W}\,. \tag{3.23}$$

3.3.2.2. Channel estimation sequence for S-ELM and SS-ELM

This sequence is a modification of the channel estimation sequence defined in the IEEE standard, but the assigned number of symbols $n_s = 24$ is still maintained, so that the comparison to LS is fair. The reason for the modification is that the LS channel estimation sequence does not contain all the possible M symbols, and in consequence the ELM algorithm could not learn to identify them. Defining

$$r_s = n_s \bmod M\,, \tag{3.24}$$

and

$$k_s = (n_s - r_s)/M\,, \tag{3.25}$$

then the proposed sequence consists of the sequence of constellation symbols $(\mathbf{s}_1, \ldots, \mathbf{s}_M)$, repeated successively k_s times, adding the sequence $(\mathbf{s}_1, \ldots, \mathbf{s}_{r_s})$ as a remainder at the end. The sequence can be defined as

$$\mathbf{q}_{\text{ELM}} = \begin{cases} (\mathbf{s}_1, \ldots, \mathbf{s}_M, \overset{k_s}{\cdots}, \mathbf{s}_1, \ldots, \mathbf{s}_M), & \text{if } r_s = 0 \\ (\mathbf{s}_1, \ldots, \mathbf{s}_M, \overset{k_s}{\cdots}, \mathbf{s}_1, \ldots, \mathbf{s}_M, \mathbf{s}_1, \ldots, \mathbf{s}_{r_s}), & \text{otherwise,} \end{cases} \tag{3.26}$$

or more compactly

$$\mathbf{q}_{\text{ELM}} = (\mathbf{q}^{(i)})_{i=1}^{n_s}\,, \tag{3.27}$$

where

$$\mathbf{q}^{(i)} = \mathbf{s}_j, \quad \text{with } j = i - M\left\lfloor \frac{i-1}{M} \right\rfloor\,, \tag{3.28}$$

for $i = 1, \ldots, n_s$. This ensures that all symbols in the constellation set $\mathcal{K}$ are in the estimation sequence at least once, as long as $n_s \geq M$.

3.3.2.3. Channel estimation sequence for US-ELM

Even though no channel estimation sequence is needed for performing the US-ELM equalization, it is useful to send a sequence that can then help match up each cluster found by

the US-ELM algorithm and its corresponding symbol. This sequence simply consists of the M constellation symbols in sequence

$$\mathbf{q}_{\text{USELM}} = (\mathbf{s}_1, \ldots, \mathbf{s}_M)\,, \tag{3.29}$$

and thus $n_s = M$.

3.3.3. Modulation

Once the N_f frames are correctly formatted, they are modulated into a sequence of M-CSK symbols. Each frame is divided into sequences of length m, and each of these sequences are then codified into one symbol. If there are leftover bits at the end of the frame, then zeros are used as padding in order to complete the sequence of m bits. In mathematical terms, one can define

$$r_f = \ell_f \bmod m \tag{3.30}$$

as the number of leftover bits after modulation. If $r_f = 0$, then there are no leftover bits and the binary sequence can be modulated directly. If $r_f > 0$, then the leftover bits cannot be modulated into a symbol with m bits of information, since $m > r_f$. In order to modulate the leftover bits, the binary sequence is padded with zeros until completion of the last modulation symbol. The number of padding bits required, p_f, is then given by

$$p_f = \begin{cases} 0, & \text{if } r_f = 0 \\ m - r_f, & \text{otherwise}\,. \end{cases} \tag{3.31}$$

From these definitions, the number of symbols for each frame, n_c, can be expressed as

$$n_c = \left\lceil \frac{\ell_f}{m} \right\rceil\,, \tag{3.32}$$

where $\lceil \cdot \rceil$ denotes the ceiling function. The number of symbols which contain data, as opposed to channel estimation symbols, is denoted by n_d, with

$$n_d = n_c - n_s\,. \tag{3.33}$$

In between frames, there is a period of time when all three LEDs are turned off. This period is given in number of optical clocks periods, and is denoted by N_{opt}, while the optical clock frequency is denoted by f_{opt}. The number of symbols periods to be used in the interframe period, n_o, is given by

$$n_o = \left\lceil \frac{1}{m} \left(R_b \frac{N_{\text{opt}}}{f_{\text{opt}}} - p_f \right) \right\rceil\,, \tag{3.34}$$

where R_b is the data rate of the system. The padding bits p_f are subtracted since their bit periods are lent from the interframe period to the preceding frame.

3.3.4. Transmission

The result of the previous modulation process are sequences of n_c CSK symbols grouped in individual frames. In this section, the simulation of the transmission of the N_f frames is

explained. First, we calculate how many symbols periods are transmitted. Consider N_f frames of n_c symbols each. Plus, in keeping with the IEEE standard, consider $N_f - 1$ interframe periods of n_o symbol periods each. In total then, the modulated signal has N symbols periods, calculated as

$$N = N_f n_c + (N_f - 1) n_o \,. \tag{3.35}$$

The modulated signal, consisting of the N_f frames separated by interframe periods, can then be expressed as the vector $\mathbf{x} \in \mathbb{R}_{\geq 0}^{3\times N}$, which can can be represented as

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}^{(1)} & \cdots & \mathbf{x}^{(N)} \end{bmatrix} , \tag{3.36}$$

where $\mathbf{x}^{(i)}$ represents the i-th symbol in the signal, for $i = 1, \ldots, N$. The i-th symbol can then be represented as the radiant flux vector

$$\mathbf{x}^{(i)} = \begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \\ x_3^{(i)} \end{bmatrix} , \tag{3.37}$$

where $x_1^{(i)}$, $x_2^{(i)}$, $x_3^{(i)}$ is the radiant flux or optical power transmitted, in the i-th symbol, by the red, green and blue LED respectively. Alternatively, one can define

$$\mathbf{x}_j = \begin{bmatrix} x_j^{(1)} & \cdots & x_j^{(N)} \end{bmatrix} , \tag{3.38}$$

as the optical signal transmitted by the j-th LED, where $j = 1, 2, 3$ represents the red, green and blue LED respectively. Note that in the signal $\mathbf{x}$, there is one sample per symbol. Consequently, the sampling frequency is given by

$$f_s = \frac{R_b}{m} \tag{3.39}$$

3.3.4.1. Channel transmission

As discussed in Section 2.5.4, time-invariability of the underground mine VLC channel can be considered for only small periods of time. In this work, it will be assumed that for the time duration of a frame and its subsequent interframe period, the channel is time-invariant. This means that each frame will be subjected to a different, randomly-generated underground mine CIR. The k-th frame, for $k = 1, \ldots, N_f$, is subjected to the CIRs $h_1^{(k)}(t)$, $h_2^{(k)}(t)$ and $h_3^{(k)}(t)$ for the red, green and blue wavelengths respectively. Note that these CIRs consider both the CIR of the LED and the CIR of the underground mine channel, i.e. for the j-th LED, with $j = 1, 2, 3$, $h_j^{(k)}(t)$ is found with

$$h_j^{(k)}(t) = h_{\text{mine},\,j}^{(k)}(t) * h_{\text{LED},\,j}(t) \,. \tag{3.40}$$

In order to simulate the analog, physical interaction between the channel and the optical signal, the CIRs will be sampled at a high sampling frequency f_h. This sampling results in the vectors $\mathbf{h}_{j,(k)} \in \mathbb{R}^{1\times N_h}$, for $j = 1, 2, 3$, where N_h is the total number of samples in each CIR vector. Let us define the isolated frame signal for the k-th frame $\mathbf{f}_{(k)} \in \mathbb{R}^{3\times N}$, by defining

its i-th symbol as

$$\mathbf{f}_{(k)}^{(i)} = \begin{cases} \mathbf{x}^{(i)}, & \text{if } i \geq 1 + (k-1)(n_c + n_o) \text{ and } i \leq k(n_c + n_o) \\ \mathbf{0}, & \text{otherwise} \end{cases}, \tag{3.41}$$

for $i = 1, \ldots, N$ and where $\mathbf{0} \in \mathbb{R}^3$ is the three-dimensional zero vector. From this, $\mathbf{f}_{(k)}$ can be expressed as

$$\mathbf{f}_{(k)} = \begin{bmatrix} \mathbf{f}_{(k)}^{(1)} & \cdots & \mathbf{f}_{(k)}^{(N)} \end{bmatrix}, \tag{3.42}$$

and it represents the modulated signal but with the LEDs turned off outside the time window corresponding to the k-th frame. This signal will be referred to as the k-th frame optical signal, and it is useful because we want to simulate each frame being independently affected by the channel. Note that $\mathbf{f}_{1,(k)}, \mathbf{f}_{2,(k)}$ and $\mathbf{f}_{3,(k)}$ represent the red, green and blue component of this optical signal, respectively.

The optical signal obtained after transmitting the $\mathbf{f}_{j,(k)}$ signal through the channel defined by $\mathbf{h}_{j,(k)}$ is denoted by $\mathbf{r}_{j,(k)}$ and can be simulated as the convolution between $\mathbf{h}_{j,(k)}$ and $\mathbf{f}_{j,(k)}$, for $j = 1, 2, 3$. For the resulting convolution to have physical meaning, both vectors must have the same sampling frequency, but the former has a sampling frequency of f_h and the latter of f_s. In general, $f_h > f_s$. This means that $\mathbf{f}_{(k)}$ has to be upsampled by a factor of $\alpha = \lfloor f_h / f_s \rfloor$. In this work, $\mathbf{f}_{(k)}$ is upsampled through linear interpolation, and the result is denoted by $\mathbf{u}_{j,(k)} \in \mathbb{R}^{1 \times \alpha N}$. With this, $\mathbf{r}_{j,(k)}$ is defined as

$$\mathbf{r}_{j,(k)} = \mathbf{h}_{j,(k)} * \mathbf{u}_{j,(k)}, \tag{3.43}$$

with

$$\mathbf{r}_{j,(k)} = \begin{bmatrix} r_{j,(k)}^{(1)} & \cdots & r_{j,(k)}^{(N_h + \alpha N + 1)} \end{bmatrix}. \tag{3.44}$$

Note that due to the convolution, $\mathbf{r}_{j,(k)} \in \mathbb{R}^{1 \times (N_h + \alpha N_u + 1)}$ is ahead of $\mathbf{u}_{j,(k)}$ by $D_{j,(k)}$ samples. $D_{j,(k)}$ depends exclusively on the CIR vector, $\mathbf{h}_{j,(k)}$. At the receiver, $D_{j,(k)}$ is known since, in this work, perfect synchronization is assumed. The synchronized, channel-distorted k-th frame optical signal from the j-th LED, is denoted by

$$\hat{\mathbf{u}}_{j,(k)} = \begin{bmatrix} \hat{u}_{j,(k)}^{(1)} & \cdots & \hat{u}_{j,(k)}^{(\alpha N)} \end{bmatrix} = \begin{bmatrix} r_{j,(k)}^{(1 + D_{j,(k)})} & \cdots & r_{j,(k)}^{(\alpha N + D_{j,(k)})} \end{bmatrix}. \tag{3.45}$$

The synchronized k-th frame optical signal $\hat{\mathbf{u}}_{j,(k)}$ is then combined with the optical signals from the other frames, by

$$\hat{\mathbf{u}}_j = \begin{bmatrix} \hat{u}_j^{(1)} & \cdots & \hat{u}_j^{(\alpha N)} \end{bmatrix} = \sum_{k=1}^{N_f} \hat{\mathbf{u}}_{j,(k)}. \tag{3.46}$$

Note that this sum is done exclusively to take into account the unlikely case where the channel induces ISI from one frame into another. If there is no ISI between adjacent frames, then they can be considered independent of each other, making this sum unnecessary. The optical signal $\hat{\mathbf{u}}_j$ represents the N symbols received at the PD, after being channel-distorted and synchronized and sampled at a sampling frequency of f_h. Nonetheless, it is assumed that the receiver is only capable of sampling at a much lower sampling frequency, denoted by f_r. The number of samples per symbol, after sampling at the receiver, is then given by $N_s = \lfloor f_s / f_r \rfloor$. The sampled, synchronized, channel-distorted optical signal from the j-th LED is denoted

by $\hat{\mathbf{x}}_j \in \mathbb{R}^{1\times N_u}$, with

$$\hat{x}_j^{(i)} = \hat{u}_j^{(1+\lceil \frac{\alpha}{N_s}(i-1)\rceil)}, \tag{3.47}$$

for $i = 1, 2, \ldots, N_u$, with $N_u = N \cdot N_s$.

3.3.4.2. Electrical conversion

The electrical signal obtained from the photodetection of the sampled, synchronized, channel-distorted optical signal $\hat{\mathbf{x}}_j$ in the j-th PD, for $j = 1, 2, 3$, is denoted by $\mathbf{y}_j \in \mathbb{R}_{\geq 0}^{3\times N_u}$, with

$$\mathbf{y}_j = \begin{bmatrix} y_j^{(1)} & \cdots & y_j^{(N_u)} \end{bmatrix} = \sum_{k=1}^{3} R_{kj}\hat{\mathbf{x}}_k + \sum_{k=1}^{3} \mathbf{n}_{kj}\,, \tag{3.48}$$

where R_{kj} is the j-th PD net responsivity (i.e. including optical filters) for light coming from the k-th LED, and $\mathbf{n}_{kj}$ is the noise induced on the j-th photodiode due to light coming from the k-th LED. The latter can be expressed as

$$\mathbf{n}_{kj} = \frac{H_{0,j} P_{\mathrm{T,avg}} R_{kj}}{\sqrt{\mathrm{SNR}}}\mathbf{n}\,, \tag{3.49}$$

where $P_{\mathrm{T,avg}}$ is the average transmission power of the three LEDs combined, SNR is the electrical signal-to-noise ratio under which the system is evaluated, $\mathbf{n}$ is a vector of length N_u for which every element is drawn from the standard normal distribution $\mathcal{N}(0, 1)$ and $H_{0,j}$ is the DC component of the channel coming from the j-th LED, averaged over all the frames, i.e.

$$H_{0,j} = \frac{1}{N_f}\sum_{k=1}^{N_f}\sum_{i=1}^{N_h} h_{j,(k)}^{(i)}\,. \tag{3.50}$$

Note that the three electrical signals $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3$ can be also thought of as one demultiplexed electrical signal $\mathbf{y} \in \mathbb{R}^{3\times N_u}$, where each i-th sample is given by three electrical values, namely,

$$\mathbf{y}^{(i)} = \begin{bmatrix} y_1^{(i)} \\ y_2^{(i)} \\ y_3^{(i)} \end{bmatrix}\,. \tag{3.51}$$

This signal is downsampled back to one sample per symbol, by averaging the N_s samples received in each symbol. The downsampled signal is denoted by $\bar{\mathbf{y}} \in \mathbb{R}_{\geq 0}^{3\times N}$, and its i-th symbol is given by

$$\bar{\mathbf{y}}^{(i)} = \frac{1}{N_s}\sum_{k=1}^{N_s} \mathbf{y}^{(k+N_s(i-1))}\,. \tag{3.52}$$

3.3.4.3. Frame extraction

From the received electrical signal $\bar{\mathbf{y}}$, one can extract individual frames. For instance, the first received frame, denoted by $\hat{\mathbf{f}}_{(1)} \in \mathbb{R}_{\geq 0}^{3\times n_c}$ is simply

$$\hat{\mathbf{f}}_{(1)} = \begin{bmatrix} \bar{\mathbf{y}}^{(1)} & \cdots & \bar{\mathbf{y}}^{(n_c)} \end{bmatrix} . \tag{3.53}$$

The second frame $\hat{\mathbf{f}}_{(2)}$ would start after the entire first frame and first interframe period are transmitted, and so the starting index would be shifted by $n_c + n_o$ symbols. The index of the last symbol in the frame is simply the starting index shifted by $n_c - 1$ symbols. In consequence, $\hat{\mathbf{f}}_{(2)} \in \mathbb{R}_{\geq 0}^{3\times n_c}$ can be denoted as

$$\hat{\mathbf{f}}_{(2)} = \begin{bmatrix} \bar{\mathbf{y}}^{(1+n_c+n_o)} & \cdots & \bar{\mathbf{y}}^{(2n_c+n_o)} \end{bmatrix} . \tag{3.54}$$

In general, the t-th frame, for $t = 1, \ldots, N_f$, is denoted by $\hat{\mathbf{f}}_{(t)} \in \mathbb{R}_{\geq 0}^{3\times n_c}$, with

$$\hat{\mathbf{f}}_{(t)} = \begin{bmatrix} \hat{\mathbf{f}}_{(t)}^{(1)} & \cdots & \hat{\mathbf{f}}_{(t)}^{(n_c)} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{y}}^{(1+(t-1)(n_c+n_o))} & \cdots & \bar{\mathbf{y}}^{(tn_c+(t-1)n_o)} \end{bmatrix} . \tag{3.55}$$

Once the frames are extracted, it is easier to define the received channel estimation sequences and the received data. The received modulated channel estimation sequence in the t-th frame, $\hat{\mathbf{q}}_{(t)} \in \mathbb{R}_{\geq 0}^{3\times n_s}$ is given by

$$\hat{\mathbf{q}}_{(t)} = \begin{bmatrix} \hat{\mathbf{f}}_{(t)}^{(1)} & \cdots & \hat{\mathbf{f}}_{(t)}^{(n_s)} \end{bmatrix} , \tag{3.56}$$

and the received modulated data in the t-th frame, $\hat{\mathbf{d}}_{(t)} \in \mathbb{R}_{\geq 0}^{3\times n_d}$ is given by

$$\hat{\mathbf{d}}_{(t)} = \begin{bmatrix} \hat{\mathbf{f}}_{(t)}^{(1+n_s)} & \cdots & \hat{\mathbf{f}}_{(t)}^{(n_c)} \end{bmatrix} . \tag{3.57}$$

3.3.5. Equalization

The equalization process is performed sequentially, frame by frame. Consider a nonspecific received modulated frame, with received modulated data denoted by $\hat{\mathbf{d}} \in \mathbb{R}_{\geq 0}^{3\times n_d}$ and received modulated channel estimation sequence denoted by $\hat{\mathbf{q}} = (\hat{\mathbf{q}}^{(1)}, \ldots, \hat{\mathbf{q}}^{(n_s)})$. Also consider that the receiver has access to the transmitted modulated channel sequence $\mathbf{q} = (\mathbf{q}^{(1)}, \ldots, \mathbf{q}^{(n_s)})$, where

$$\mathbf{q}^{(i)} = \begin{bmatrix} q_1^{(i)} \\ q_2^{(i)} \\ q_3^{(i)} \end{bmatrix} \tag{3.58}$$

is the i-th symbol in the channel estimation sequence, with $i = 1, \ldots n_s$ and where $q_1^{(i)}$, $q_2^{(i)}$, $q_3^{(i)} \in \mathbb{R}_{\geq 0}$ are the optical powers associated to the aforementioned symbol, set to be transmitted from the red, green and blue LED respectively. The definition of $\hat{\mathbf{q}}^{(i)}$ is equivalent,

$$\hat{\mathbf{q}}^{(i)} = \begin{bmatrix} \hat{q}_1^{(i)} \\ \hat{q}_2^{(i)} \\ \hat{q}_3^{(i)} \end{bmatrix}, \tag{3.59}$$

but in this case, $\hat{q}_1^{(i)}$, $\hat{q}_2^{(i)}$, $\hat{q}_3^{(i)} \in \mathbb{R}_{\geq 0}$ are the electrical signals associated to the i-th symbol received at the red, green and blue photodiode respectively.

3.3.5.1. Least squares

In the first place, subsequent symbols in the known channel sequence are averaged, resulting in a new symbol $\mathbf{v}^{(i)}$, given by

$$\mathbf{v}^{(i)} = \frac{\mathbf{q}^{(i)} + \mathbf{q}^{(i+1)}}{2}, \tag{3.60}$$

for $i = 1, \ldots, n_s/2$. $\hat{\mathbf{v}}^{(i)}$ is defined for the received channel sequence equivalently. Two matrices can be formed with this, namely

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}^{(1)} & \cdots & \mathbf{v}^{(n_s/2)} \end{bmatrix} \in \mathbb{R}_{\geq 0}^{3 \times n_s/2} \tag{3.61}$$

and

$$\hat{\mathbf{V}} = \begin{bmatrix} \hat{\mathbf{v}}^{(1)} & \cdots & \hat{\mathbf{v}}^{(n_s/2)} \end{bmatrix} \in \mathbb{R}_{\geq 0}^{3 \times n_s/2}. \tag{3.62}$$

Note that one can define the linear transformation $\mathbf{H} \in \mathbb{R}^{3\times 3}$ such that $||\hat{\mathbf{V}} - \mathbf{H}\mathbf{V}||^2$ is as small as possible. $\mathbf{H}$ is the solution to the least squares problem, and it can be found with the following formula

$$\mathbf{H} = \hat{\mathbf{V}}\mathbf{V}^{\dagger}, \tag{3.63}$$

where $\mathbf{V}^{\dagger}$ denotes the Moore-Penrose pseudoinverse of the matrix $\mathbf{V}$. The equalization method consists of applying the inverse linear transformation of $\mathbf{H}$ to the received modulated data $\hat{\mathbf{d}}$, as in

$$\mathbf{z} = \mathbf{H}^{-1}\hat{\mathbf{d}}, \tag{3.64}$$

where $\mathbf{z}$ is the equalized modulated data. This signal is then demodulated using the conventional M-CSK demodulation scheme.

3.3.5.2. Supervised and semi-supervised ELM

The equalization method using ELM can be seen as a classification problem, as each received symbol is classified by the ELM algorithm into M classes, each class corresponding to each of the pre-defined M-CSK constellation symbols. We can assign each of the M constellation symbols a class label, simply given by its index in the sequence $(\mathbf{s}_1, \ldots, \mathbf{s}_M)$, formed by all the constellation symbols. In order to train the algorithm, the received channel sequence $\hat{\mathbf{q}}$ is paired with a sequence of labels $\mathcal{L}_{\mathbf{q}}$ known a priori at the receiver. From Equation 3.28, it is deduced that the label associated with $\hat{\mathbf{q}}^{(i)}$ is given by

$$\mathcal{L}_{\mathbf{q}}^{(i)} = i - M \left\lfloor \frac{i-1}{M} \right\rfloor, \tag{3.65}$$

for $i = 1, \ldots, n_s$ and with $\mathcal{L}_{\mathbf{q}}^{(i)} \in \{1, \ldots, M\}$. The one-hot encoded version label $\mathcal{L}_{\mathbf{q}}^{(i)}$ will be denoted as $\hat{\mathcal{L}}_{\mathbf{q}}^{(i)}$. The supervised training set $\mathcal{S}$ can be expressed as

$$\mathcal{S} = \{(\hat{\mathbf{q}}^{(k)}, \hat{\mathcal{L}}_{\mathbf{q}}^{(k)})\}_{k=1}^{n_s} . \tag{3.66}$$

For the unsupervised training used in SS-ELM, a sequence of received symbols corresponding to actual data, are used. This sequence can be referred to as the unsupervised training set $\mathcal{U}$, and it can be expressed as

$$\mathcal{U} = \{\hat{\mathbf{d}}^{(k)}\}_{k=1}^{n_t} , \tag{3.67}$$

where n_t is the number of data symbols to be considered in the unsupervised training phase. The output of the ELM algorithms are the estimated labels of each of the received modulated symbols $\hat{\mathbf{d}}$, which can be expressed as

$$\text{SELM}(\hat{\mathbf{d}}, \mathcal{S}, n_h) = \{\hat{\mathcal{L}}^{(k)}\}_{k=1}^{n_d} , \tag{3.68}$$

and

$$\text{SSELM}(\hat{\mathbf{d}}, \mathcal{S}, \mathcal{U}, n_h, C_0, \gamma) = \{\hat{\mathcal{L}}^{(k)}\}_{k=1}^{n_d} , \tag{3.69}$$

for the supervised and semi-supervised ELM algorithm respectively, where n_h is the number of hidden neurons, C_0 and γ are hyperparameters exclusive to semi-supervised ELM, and $\hat{\mathcal{L}}^{(i)}$ is the predicted class label, in one-hot encoding, of the received symbol $\hat{\mathbf{d}}^{(i)}$, for $i = 1, \ldots, n_d$. By converting $\hat{\mathcal{L}}^{(i)}$ from one-hot encoding into its numerical value, $\mathcal{L}^{(i)} \in \{1, \ldots, M\}$, one can find the i-th predicted data symbol, denoted by $\mathbf{z}^{(i)}$, via the following equation

$$\mathbf{z}^{(i)} = \mathbf{s}_{\mathcal{L}^{(i)}} \tag{3.70}$$

3.3.5.3. Unsupervised ELM

In the unsupervised version of ELM, only the unsupervised training set is used as an input, defined in the same way as in Equation 3.67. The application of the unsupervised ELM algorithm onto the received modulated data $\hat{\mathbf{d}}$ can be expressed as

$$\text{USELM}(\hat{\mathbf{d}}, \mathcal{U}, n_h, \zeta, \eta) = \{\mathcal{C}^{(i)}\}_{i=1}^{n_d} , \tag{3.71}$$

where n_h is the number of hidden neurons, ζ and η are hyperparameters and $\mathcal{C}^{(i)} \in \{1, \ldots, M\}$ is the class label of the i-th received data symbol, $\hat{\mathbf{d}}^{(i)}$, for $i = 1, \ldots, n_d$. In order to associate each class with a particular symbol in $\{\mathbf{s}_1, \ldots, \mathbf{s}_M\}$, the channel estimation sequence will be used. First, one defines each class set $\mathbf{C}_k$, for the class label $k = 1, \ldots, M$, as

$$\mathbf{C}_k = \{\hat{\mathbf{d}}^{(i)} \in \mathbb{R}^3_{\geq 0} \mid i = 1, \ldots, n_d, \, \mathcal{C}^{(i)} = k\} . \tag{3.72}$$

Then, one defines the centroid of the class set $\mathbf{C}_k$ as the mean value of all elements in the set, namely,

$$\mathbf{c}_k = \frac{1}{|\mathbf{C}_k|} \sum_{\mathbf{g} \in \mathbf{C}_k} \mathbf{g} , \tag{3.73}$$

and associates each centroid to one of the symbols in the received channel estimation sequence $\{\hat{\mathbf{q}}_1, \ldots, \hat{\mathbf{q}}_M\}$, specifically, to whichever is closest in the CIE 1931 color space, in terms of

Euclidean distance. The symbol label $\mathcal{L}_k$ associated to class k, is then

$$\mathcal{L}_k = \underset{u=1,\ldots,M}{\operatorname{argmin}} \|\hat{\mathbf{q}}_u^* - \mathbf{c}_k^*\| \,, \tag{3.74}$$

where $\hat{\mathbf{q}}_u^*$ is the u-th symbol in the received channel estimation sequence $(\hat{\mathbf{q}}_1, \ldots, \hat{\mathbf{q}}_M)$ and $\hat{\mathbf{c}}_k^*$ is the centroid of class k, both expressed in the CIE 1931 xy color space. With this, one can find the i-th predicted symbol label, denoted by $\mathcal{L}^{(i)}$, with

$$\mathcal{L}^{(i)} = \mathcal{L}_{\mathcal{C}^{(i)}} \,, \tag{3.75}$$

for $i = 1, \ldots, n_d$. Likewise, the i-th predicted symbol $\mathbf{z}^{(i)}$, is given by

$$\mathbf{z}^{(i)} = \mathbf{s}_{\mathcal{L}^{(i)}} \,. \tag{3.76}$$

3.3.6. Demodulation and BER calculation

The equalization step results in N_f sequences of n_d symbols. Each sequence is the equalized data signal from an specific frame. The k-th frame equalized data signal, denoted by $\mathbf{z}_{(k)} \in \mathbb{R}^{3\times n_d}$, for $k = 1, \ldots, N_f$, can be expressed as

$$\mathbf{z}_{(k)} = \begin{bmatrix} \mathbf{z}_{(k)}^{(1)} & \cdots & \mathbf{z}_{(k)}^{(n_d)} \end{bmatrix} \,. \tag{3.77}$$

$\mathbf{z}_{(k)}$ is demodulated using the standard M-CSK demodulation method, as explained in Section 3.1.4, resulting in the binary sequence $\mathbf{b}_{(k)} \in \{0,1\}^{1\times \ell_b}$ of length $\ell_b = m \cdot n_d$, also expressed as

$$\mathbf{b}_{(k)} = \begin{bmatrix} b_{(k)}^{(1)} & \cdots & b_{(k)}^{(\ell_b)} \end{bmatrix} \,. \tag{3.78}$$

Note that due to the eventual padding performed, the length of this sequence might be greater than the length of the binary sequence originally transmitted in the k-th frame, i.e. $\ell_b \geq \ell_a$. The complete received binary sequence $\hat{\mathbf{a}} \in \{0,1\}^{1\times n_a}$ is given by

$$\hat{\mathbf{a}} = \begin{bmatrix} \hat{a}^{(1)} & \cdots & \hat{a}^{(n_a)} \end{bmatrix} = \begin{bmatrix} b_{(1)}^{(1)} & \cdots & b_{(1)}^{(\ell_a)} & b_{(2)}^{(1)} & \cdots & b_{(2)}^{(\ell_a)} & \cdots & b_{(N_f)}^{(1)} & \cdots & b_{(N_f)}^{(\ell_a)} \end{bmatrix} \,. \tag{3.79}$$

The BER is then given by

$$\text{BER} = \frac{1}{n_a} \sum_{i=1}^{n_a} e_i \,, \tag{3.80}$$

where e_i is given by

$$e_i = \begin{cases} 0, & \text{if } \hat{a}^{(i)} = a^{(i)} \\ 1, & \text{otherwise} \,. \end{cases} \tag{3.81}$$

3.3.7. Simulation parameters

In terms of channel parameters, the simulation is identical to the one presented in [4]. Consequently, the simulated CIRs considers the following scenario: a tunnel of dimensions 6 m × 3 m × 5 m and three LEDs located in position $\mathbf{T}$ and three PDs located in position $\mathbf{R}$. In order to have some degree of spatial variability in the results, five $\mathbf{R}$ positions are evaluated, the same positions used in [4]. These five positions are denoted by $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_5$.

The CIR will be considered to be the same across the RGB wavelengths, meaning that

for any k-th frame,

$$h^{(k)}_{\text{mine, }1}(t) = h^{(k)}_{\text{mine, }2}(t) = h^{(k)}_{\text{mine, }3}(t)\,. \tag{3.82}$$

This is due to the assumption that the three LEDs have the same position $\mathbf{T}$ and the three PDs are in the same position $\mathbf{R}$. This is assumed because they are so close to each other that their spatial difference is negligible. Furthermore, the three LEDs have the same tilt and rotation angle, and the three PDs have the same tilt and rotation angle. This means that there is no spatial difference whatsoever between the three channels in the simulation scenario. Furthermore, that the underground mine CIR is the same for the three bands implies that the walls of the underground mine and the dust particles have the same reflectance across the three RGB wavelengths.

Additionally, the LED impulse response is modeled by a low pass filter, with impulse response

$$h_{\text{LED}}(t) = 10(e^{-t/T_{\text{fall}}} - e^{-t/T_{\text{rise}}})\,, \tag{3.83}$$

where $T_{\text{fall}} = 10^{-9}$ s and $T_{\text{rise}} = 0.5 \cdot 10^{-9}$ s. In this work, it is assumed that the LED impulse response is the same over the three RGB wavelengths, i.e.

$$h_{\text{LED}}(t) = h_{\text{LED, }1}(t) = h_{\text{LED, }2}(t) = h_{\text{LED, }3}(t)\,. \tag{3.84}$$

Relevant simulation parameters are presented in Table 3.1. Note that the data rates used are the maximum data rates allowed in the IEEE standard for each modulation order. Simulation parameters not defined here can either be calculated or they can be found in [4]. ELM hyperparameters are discussed in Section 3.5.

3.3.7.1. CSK constellation

There are multiple CSK constellations defined by the IEEE standard. Furthermore, there are methods to optimize CSK constellations, improving the performance of CSK links, which are not considered in the IEEE standard. In this work, an IEEE standard-compliant constellation is used, defined in [24] with center of band symbols

$$\begin{aligned}(x_{\text{R}}, y_{\text{R}}) &= (0.73, 0.27)\\ (x_{\text{G}}, y_{\text{G}}) &= (0.19, 0.78)\\ (x_{\text{B}}, y_{\text{B}}) &= (0.09, 0.13)\,,\end{aligned} \tag{3.85}$$

which roughly correspond to the following wavelengths [25]

$$\begin{aligned}\lambda_{\text{R}} &= 660 \quad (\text{nm})\\ \lambda_{\text{G}} &= 535 \quad (\text{nm})\\ \lambda_{\text{B}} &= 480 \quad (\text{nm})\,.\end{aligned} \tag{3.86}$$

The M-CSK constellation symbols $\{\mathbf{s}_1, \ldots, \mathbf{s}_M\}$ used in this work is presented in Table 3.2, along with their respective binary sequence correspondence. The net responsivity values R_{kj}, represent the j-th PD net responsivity (i.e. including the optical filters) for light coming from the k-th LED, where $j, k = 1, 2, 3$, correspond to red, green and blue wavelengths respectively. The net responsivity can be expressed as the multiplication of the optical filter gain and the PD responsivity for each wavelength, as follows

Table 3.1: Simulation parameters.

Parameter	**Value**
Binary sequence length, n_a (bits)	1048464
Data bits per frame, ℓ_a (bits)	524232 [9]
# of frames, N_f	2
# of optical clock periods in interframe period, N_{opt}	400 [9]
Optical clock frequency, f_{opt} (MHz)	24 [9]
Data rate, R_b (Mbps)	**4-CSK:** 24 [9] **8-CSK:** 72 [9] **16-CSK:** 96 [9]
CIR sampling frequency, f_h (GHz)	4 [4]
CIR vector length, N_h	140 [4]
Samples per symbol after receiver, N_s	2
Average optical transmitted power, $P_{\text{T,avg}}$ (W)	1
ELM activation function, $g(\cdot)$	$g(x) = \frac{1}{1+e^{-x}}$
LED position, $\mathbf{T}$ (x, y, z) (m)	$(3, 0.5, 4.5)$
PD position, $\mathbf{R}$ (x, y, z) (m)	$\mathbf{R}_1$**:** $(3, 1, 1.8)$ $\mathbf{R}_2$**:** $(3, 1.5, 1.8)$ $\mathbf{R}_3$**:** $(4, 2, 1.8)$ $\mathbf{R}_4$**:** $(2.2, 2.5, 1.8)$ $\mathbf{R}_5$**:** $(1, 2.5, 1.6)$

$$R_{kj} = G_{kj} \cdot p_k \,, \tag{3.87}$$

where $\mathbf{G} = [G_{kj}]_{3\times 3}$ is the optical filter gain matrix, given by [26]

$$\mathbf{G} = \frac{1}{100}\begin{bmatrix} 80.351 & 1.428 & 1.908 \\ 11.769 & 75.148 & 13.076 \\ 0.279 & 27.244 & 70.684 \end{bmatrix}, \tag{3.88}$$

and $\mathbf{p} = [p_1, p_2, p_3]^{\text{T}}$ contains the PD responsivity for the red, green and blue wavelength, respectively, and it is given by [27]

$$\mathbf{p} = \begin{bmatrix} 0.3518 \\ 0.2063 \\ 0.1474 \end{bmatrix} \quad \text{(A/W)}. \tag{3.89}$$

Consequently, the net responsivity values R_{kj} to be considered in this simulation are

$$[R_{kj}]_{3\times 3} = \frac{1}{100}\begin{bmatrix} 28.2662 & 0.5023 & 0.6712 \\ 2.4274 & 15.4997 & 2.6970 \\ 0.0411 & 4.0147 & 10.4160 \end{bmatrix} \quad \text{(A/W)}. \tag{3.90}$$

Table 3.2: CSK symbols used in the simulation.

4-CSK			8-CSK			16-CSK		
Symbol	**Data**	**(x, y)**	**Symbol**	**Data**	**(x, y)**	**Symbol**	**Data**	**(x, y)**
$\mathbf{s}_1$	[0 0]	(0.19, 0.78)	$\mathbf{s}_1$	[0 0 0]	(0.19, 0.78)	$\mathbf{s}_1$	[0 0 0 0]	(0.19 0.78)
$\mathbf{s}_2$	[0 1]	(0.337, 0.393)	$\mathbf{s}_2$	[0 0 1]	(0.157, 0.563)	$\mathbf{s}_2$	[0 0 0 1]	(0.239 0.651)
$\mathbf{s}_3$	[1 0]	(0.09, 0.13)	$\mathbf{s}_3$	[0 1 0]	(0.37, 0.61)	$\mathbf{s}_3$	[0 0 1 0]	(0.157 0.563)
$\mathbf{s}_4$	[1 1]	(0.73, 0.27)	$\mathbf{s}_4$	[0 1 1]	(0.509, 0.396)	$\mathbf{s}_4$	[0 0 1 1]	(0.37 0.61)
			$\mathbf{s}_5$	[1 0 0]	(0.09, 0.13)	$\mathbf{s}_5$	[0 1 0 0]	(0.206 0.434)
			$\mathbf{s}_6$	[1 0 1]	(0.189, 0.326)	$\mathbf{s}_6$	[0 1 0 1]	(0.337 0.393)
			$\mathbf{s}_7$	[1 1 0]	(0.41, 0.2)	$\mathbf{s}_7$	[0 1 1 0]	(0.419 0.481)
			$\mathbf{s}_8$	[1 1 1]	(0.73, 0.27)	$\mathbf{s}_8$	[0 1 1 1]	(0.386 0.481)
						$\mathbf{s}_9$	[1 0 0 0]	(0.123 0.347)
						$\mathbf{s}_{10}$	[1 0 0 1]	(0.172 0.218)
						$\mathbf{s}_{11}$	[1 0 1 0]	(0.09 0.13)
						$\mathbf{s}_{12}$	[1 0 1 1]	(0.303 0.177)
						$\mathbf{s}_{13}$	[1 1 0 0]	(0.55 0.44)
						$\mathbf{s}_{14}$	[1 1 0 1]	(0.599 0.311)
						$\mathbf{s}_{15}$	[1 1 1 0]	(0.517 0.223)
						$\mathbf{s}_{16}$	[1 1 1 1]	(0.73 0.27)

3.4. Equalization Method Proposal: LS + ELM

The application of $[R_{kj}]_{3\times 3}$ to the received optical signal adds plenty of distortion, but in a linear capacity. Since LS finds the optimal inverse linear transformation to reverse this linear distortion, it would be interesting to study what happens when applying LS as a first, linear equalizer (in order to remove the linear distortion), and then ELM as a second, non-linear equalizer. This means that the output of LS passed as the input to the ELM equalizer. In order to accommodate both LS and ELM, the channel estimation sequence for the combined equalization types, denoted LS+SELM, LS+SSELM and LS+USELM, is $\mathbf{q}_{\text{ELM}}$, i.e.

$$\mathbf{q}_{\text{LS+SELM}} = \mathbf{q}_{\text{LS+SSELM}} = \mathbf{q}_{\text{LS+USELM}} = \mathbf{q}_{\text{ELM}}\,. \tag{3.91}$$

Let us consider a frame with received modulated data $\hat{\mathbf{d}} \in \mathbb{R}_{\geq 0}^{3\times n_d}$, received channel estimation sequence $\hat{\mathbf{q}} \in \mathbb{R}_{\geq 0}^{3\times n_s}$ and transmitted channel estimation sequence $\mathbf{q} \in \mathbb{R}_{\geq 0}^{3\times n_s}$. In the first place, the data is equalized with LS as follows

$$\mathbf{z}_{\text{LS}} = (\hat{\mathbf{q}}\mathbf{q}^{\dagger})^{-1}\hat{\mathbf{d}}\,. \tag{3.92}$$

The received channel estimation sequence is also equalized with LS,

$$\hat{\mathbf{p}} = (\hat{\mathbf{q}}\mathbf{q}^{\dagger})^{-1}\hat{\mathbf{q}}\,. \tag{3.93}$$

Now that the LS equalization is complete, one can define the supervised training set as

$$\mathcal{S}_{\text{LS}} = \{(\hat{\mathbf{p}}^{(k)}, \hat{\mathcal{L}}_{\mathbf{q}}^{(k)})\}_{k=1}^{n_s}\,, \tag{3.94}$$

and the unsupervised training set as

$$\mathcal{U}_{\text{LS}} = \{\mathbf{z}_{\text{LS}}{}^{(k)}\}_{k=1}^{n_t}\,. \tag{3.95}$$

With these definitions, one can express

$$\text{LS+SELM}(\hat{\mathbf{d}}, \mathcal{S}, n_h) = \text{SELM}(\mathbf{z}_{\text{LS}}, \mathcal{S}_{\text{LS}}, n_h)\,, \tag{3.96}$$

$$\text{LS+SSELM}(\hat{\mathbf{d}}, \mathcal{S}, \mathcal{U}, n_h, C_0, \gamma) = \text{SSELM}(\mathbf{z}_{\text{LS}}, \mathcal{S}_{\text{LS}}, \mathcal{U}_{\text{LS}}, n_h, C_0, \gamma)\,, \tag{3.97}$$

$$\text{LS+USELM}(\hat{\mathbf{d}}, \mathcal{U}, n_h, \zeta, \eta) = \text{USELM}(\mathbf{z}_{\text{LS}}, \mathcal{U}_{\text{LS}}, n_h, \zeta, \eta) \tag{3.98}$$

as the outputs of the combined equalization methods. The rest of the equalization process is the same as with the regular versions.

3.5. Hyperparameter Tuning

Table 3.3: Candidate hyperparameters for grid search.

Equalization type	**Parameter**	**Values**
SELM	n_h	$\{20, 35, 50, 65, 80, 95\}$
SSELM	n_h	$\{20, 35, 50, 65, 80, 95\}$
	C_0	$\{10^1, 10^2, \ldots, 10^6\}$
	γ	$\{10^1, 10^2, \ldots, 10^6\}$
USELM	n_h	$\{20, 35, 50, 65, 80, 95\}$
	ζ	$\{2^2, 2^3, \ldots, 2^7\}$
	η	$\{10^{-4}, 10^{-3}, \ldots, 10^4\}$

For the ELM algorithm to perform at its best, appropriate hyperparameters must be chosen. In order to find appropriate hyperparameters, grid search will be used. In the case of ELM, the objective function to be minimized through grid search is the BER, but since the BER depends so heavily on the SNR, it is convenient to work at a fixed noise level. The candidate hyperparameter values are presented in Table 3.3, mostly taken from [23]. Consider $\Theta(\boldsymbol{\lambda}, \mathbf{R})$ to be the equalization model for a given equalization method and modulation order, trained with hyperparameters $\boldsymbol{\lambda} = \{\lambda_1, \ldots, \lambda_n\}$ and PD position $\mathbf{R}$. In this work, the optimal hyperparameters for a particular equalization method and modulation order were chosen such that, given a noise level SNR $= \sigma$, the BER calculated after one simulation execution, averaged over a set of positions $\mathcal{R}$, is minimized. In mathematical notation, the optimal hyperparameters, for a particular equalization method and modulation order, $\boldsymbol{\lambda}^*$, are given by

$$\boldsymbol{\lambda}^* = \underset{\boldsymbol{\lambda}}{\text{argmin}} \frac{1}{|\mathcal{R}|} \sum_{\mathbf{R} \in \mathcal{R}} \text{BER}[\Theta(\boldsymbol{\lambda}, \mathbf{R}), \sigma]\,. \tag{3.99}$$

BER$[\cdot]$ is obtained from the simulations outlined in previous sections. In this work, $\mathcal{R} = \{\mathbf{R}_1, \mathbf{R}_5\}$ and $\sigma = 20$ dB were used. Those two positions are the most apart from each other out of the five positions in consideration, granting the results some level of spatial variability. The chosen noise level is considered a middle ground between a very noisy channel and a noiseless channel, the assumption being that hyperparameters that do well in a mild noise level do reasonably well in the extreme cases. The optimal hyperparameters for each equalization method and modulation order are presented in Table 3.4.

Table 3.4: Optimal hyperparameters found by grid search.

Equalization type	**Parameter**	**Values**		
		4-CSK	**8-CSK**	**16-CSK**
SELM	n_h	20	80	5
SSELM	n_h	95	65	20
	C_0	10^6	10^6	10^3
	γ	10	10	10^4
USELM	n_h	80	20	35
	ζ	4	64	64
	η	100	0.1	10
LS+SELM	n_h	95	80	4
LS+SSELM	n_h	80	95	95
	C_0	10^6	10^6	10^5
	γ	10^4	10^5	10^6
LS+USELM	n_h	65	20	95
	ζ	4	128	4
	η	100	10	10^4

3.6. Experiment Description

Since the simulations make use of random processes, such as the channel transmission, and the initialization of the weights and biases for ELM, in order for the BER curves to have statistical significance, multiple iterations of the same experiment must be performed. The repetition of the same experiment when stochastic processes are involved, is called the Monte Carlo method; each repetition of the experiment is called a Monte Carlo iteration.

In this work, the BER curves are calculated for the range of electrical SNR $= \{0, 2, 4, \ldots, 38, 40\}$ (dB). At each SNR value, 50 Monte Carlo iterations will be performed, with the BER being their output. Each iteration consists in running one instance of the simulation, with the parameters given in Table 3.1 and the hyperparameters given in Table 3.4, for all 21 combinations of seven equalization methods $\{\text{LS}, \text{SELM}, \text{SSELM}, \text{USELM}, \text{LS+SELM}, \text{LS+SSELM}, \text{LS+USELM}\}$ and three modulation orders $\{\text{4-CSK}, \text{8-CSK}, \text{16-CSK}\}$. Additionally, the PD position is not fixed throughout the 50 iterations, but rather 10 iterations are performed considering each $\mathbf{R}_i$, with $i = 1, \ldots, 5$, as the PD position.

Chapter 4

Results and Discussion

4.1. Histograms

In this section, 21 histograms for each equalization method and modulation order combination are presented. Each histogram contains the frequency distribution of the BER at a specific SNR level, calculated from 50 data points (since each of the 50 Monte Carlo iteration outputs a BER value). When possible, a Gaussian fit is overlaid, with its mean plotted in red. Figures 4.1, 4.2, 4.3, 4.4, 4.5, 4.6 and 4.7 present the histograms obtained for 4-CSK modulation paired with LS, SELM, SSELM, USELM, LS+SELM, LS+SSELM and LS+USELM equalization, respectively. Figures 4.8, 4.9, 4.10, 4.11, 4.12, 4.13 and 4.14 do the same but for 8-CSK modulation, while Figures 4.15, 4.16, 4.17, 4.18, 4.19, 4.20 and 4.21 do it for 16-CSK modulation.

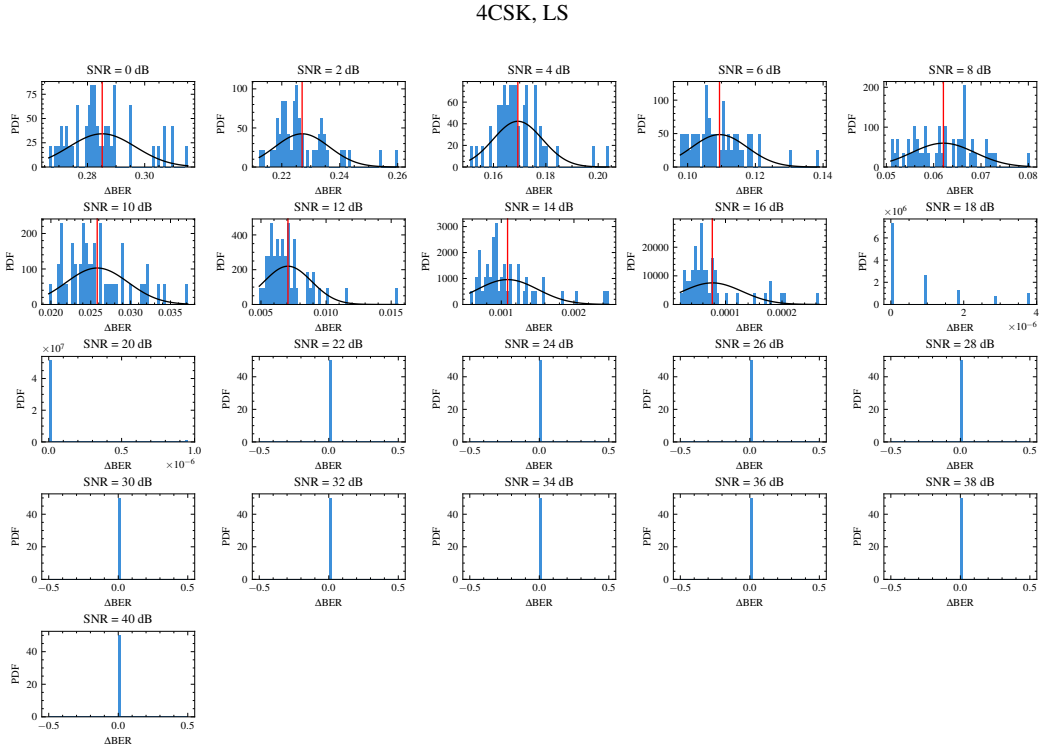


Figure 4.1: Histograms for LS with 4-CSK.

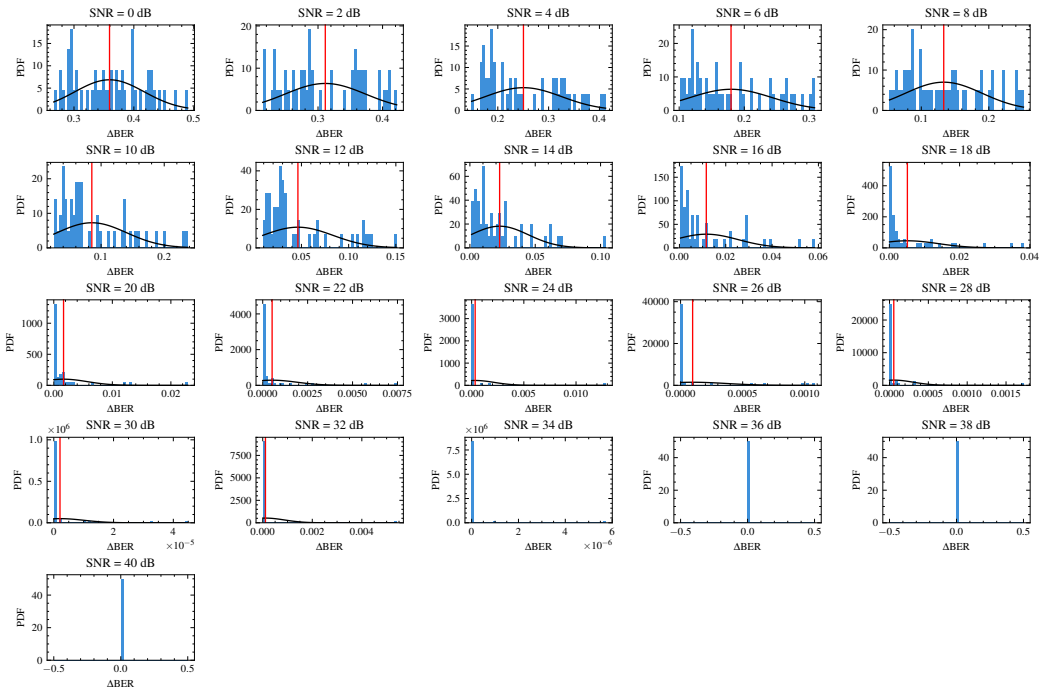


Figure 4.2: Histograms for SELM with 4-CSK.

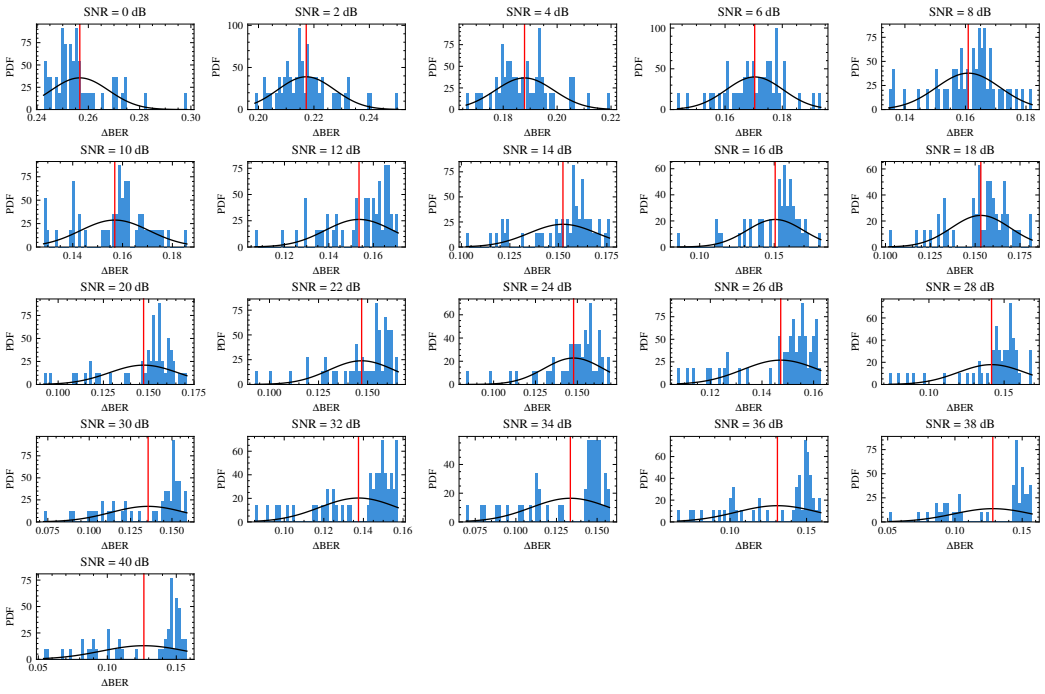


Figure 4.3: Histograms for SSELM with 4-CSK.

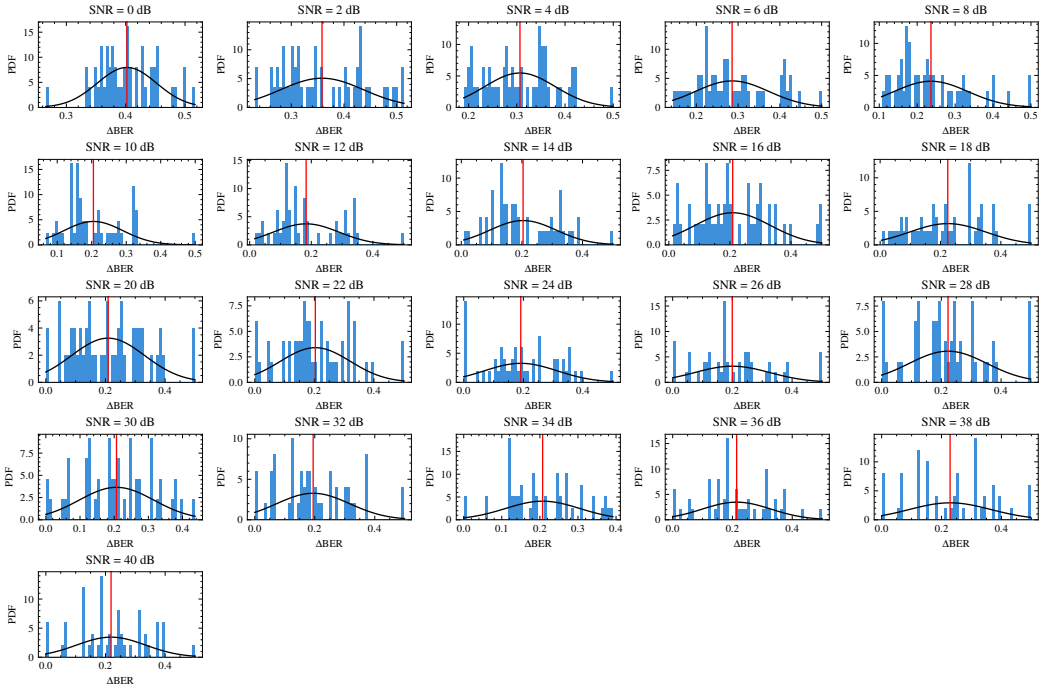


Figure 4.4: Histograms for USELM with 4-CSK.

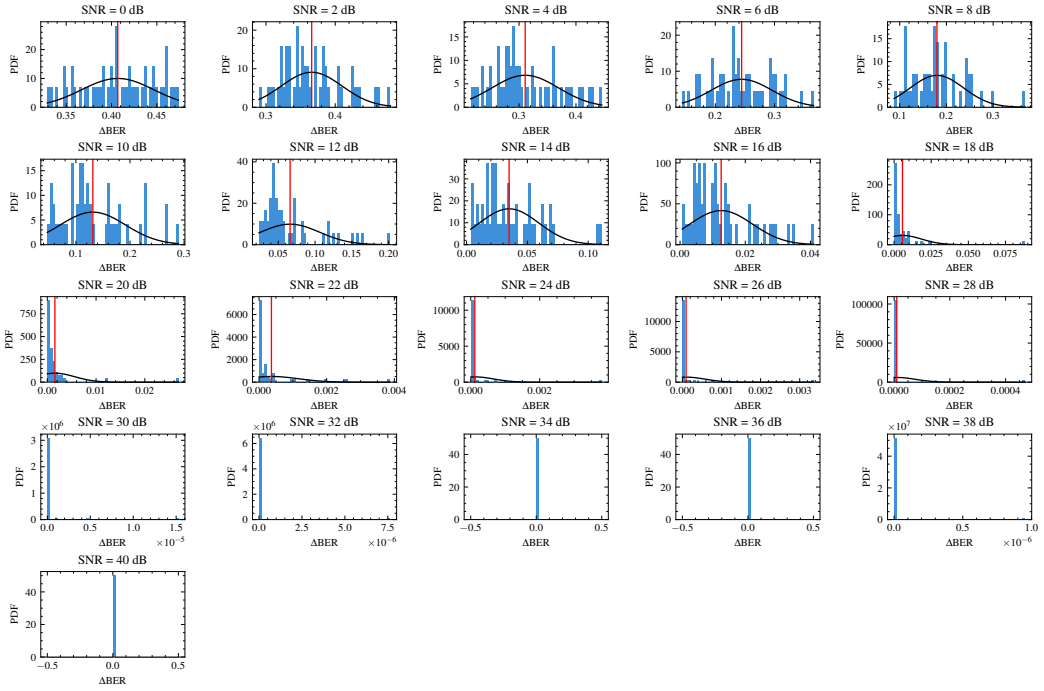


Figure 4.5: Histograms for LS+SELM with 4-CSK.

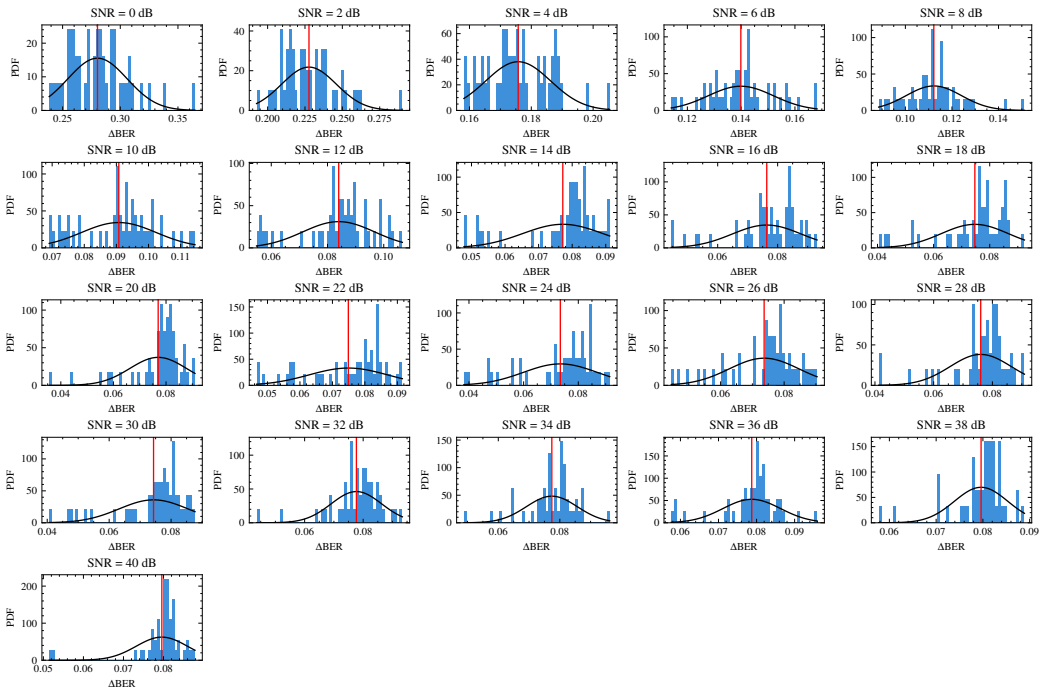


Figure 4.6: Histograms for LS+SSELM with 4-CSK.

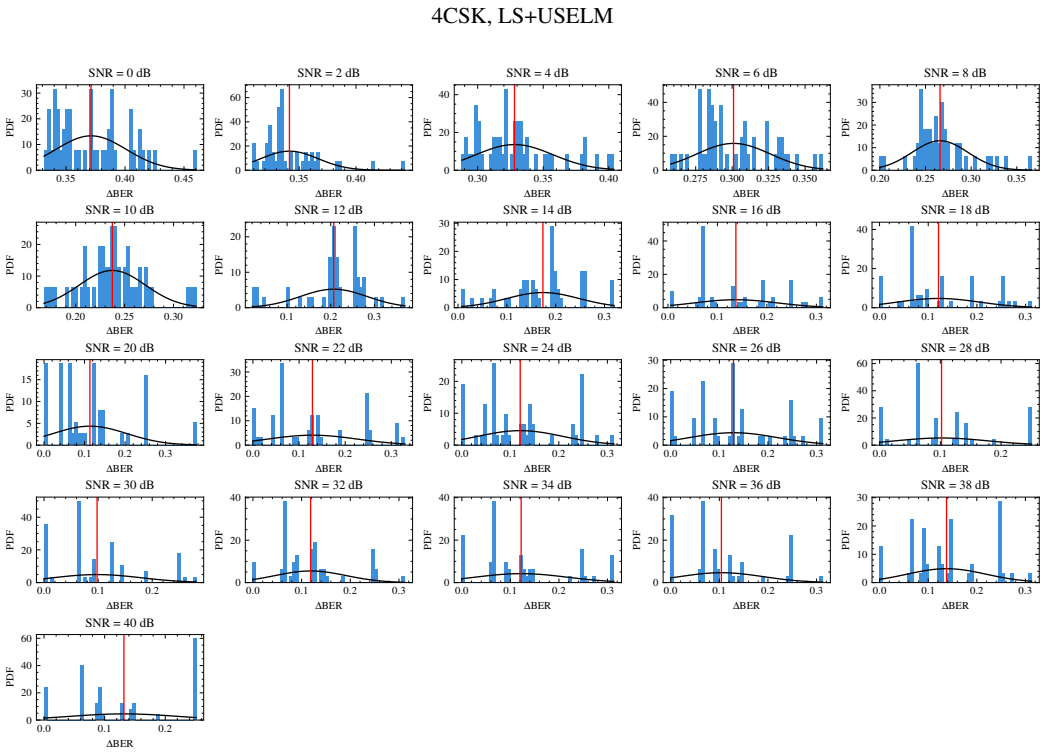


Figure 4.7: Histograms for LS+USELM with 4-CSK.

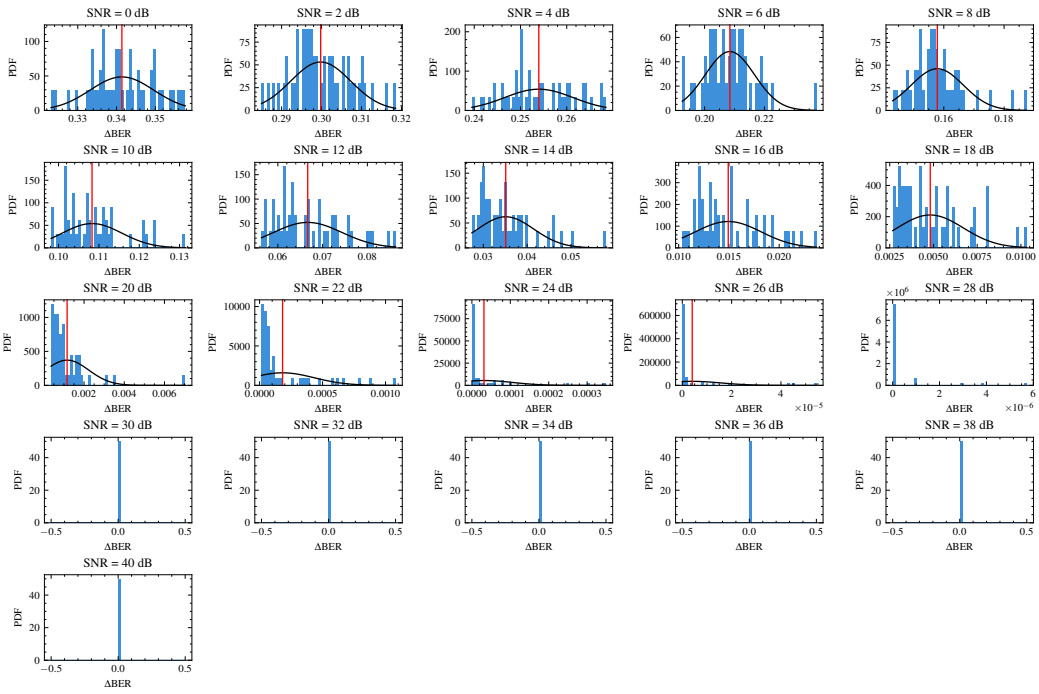


Figure 4.8: Histograms for LS with 8-CSK.

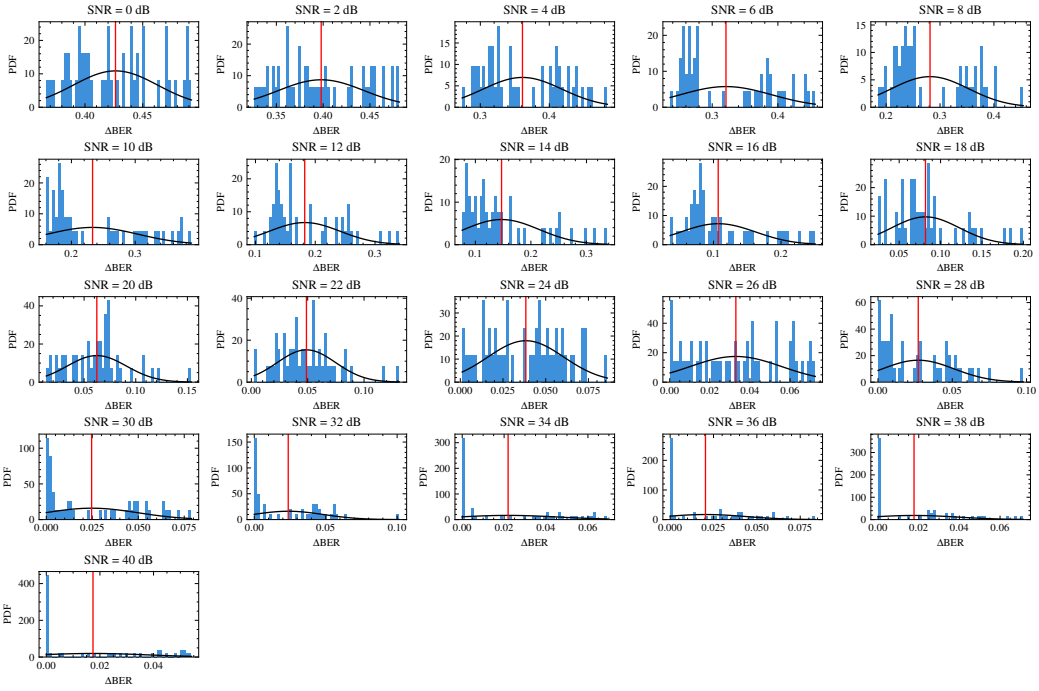


Figure 4.9: Histograms for SELM with 8-CSK.

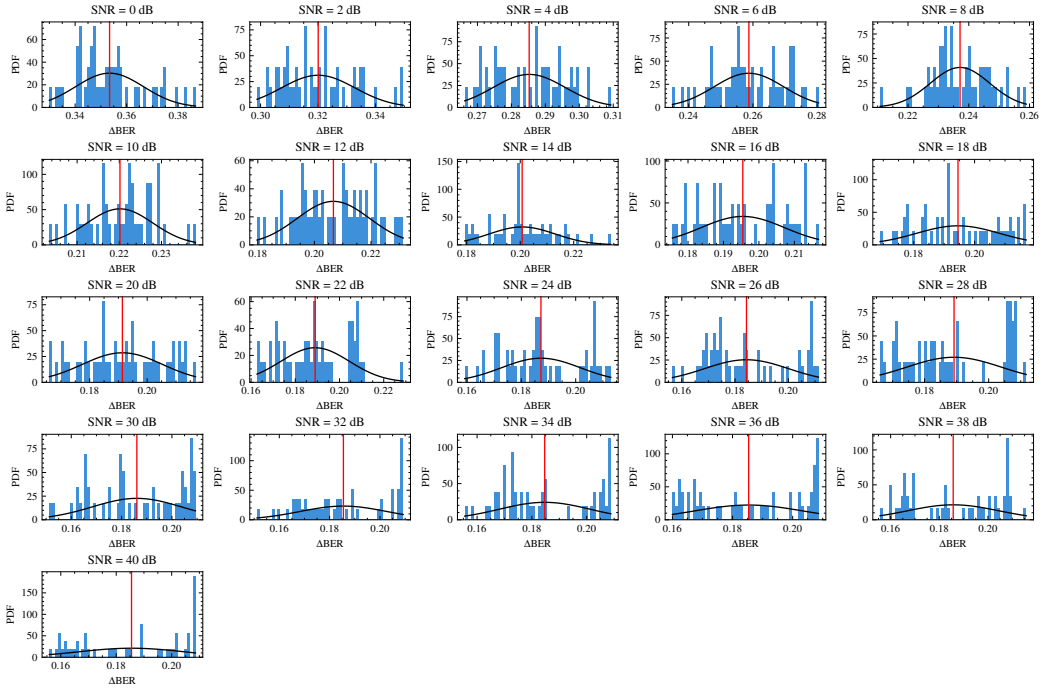


Figure 4.10: Histograms for SSELM with 8-CSK.

8CSK, USELM

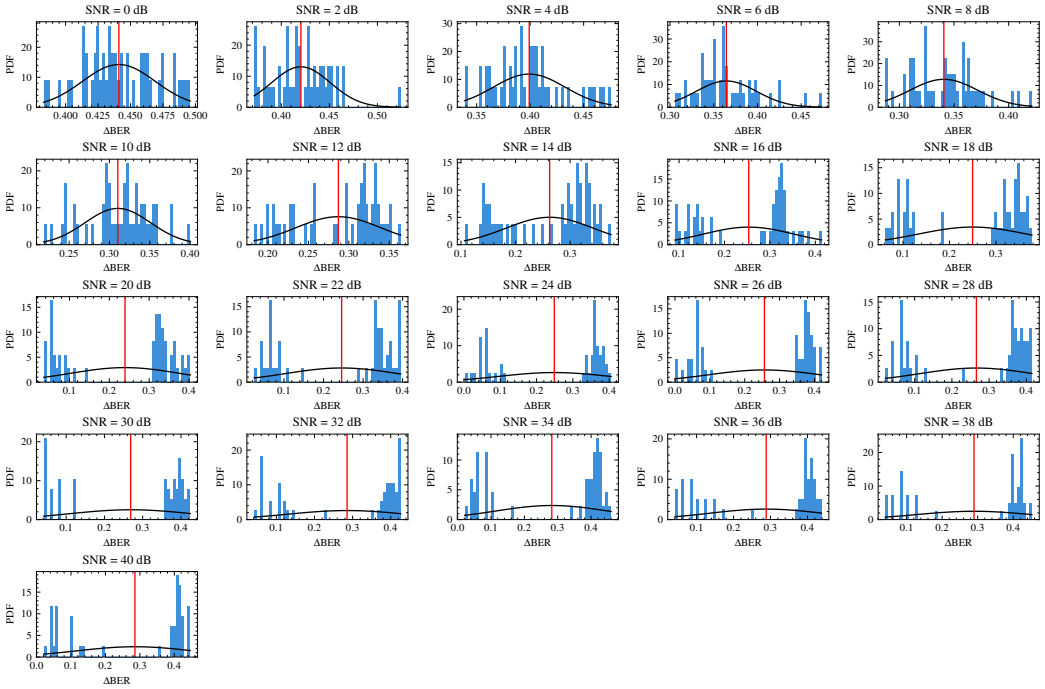


Figure 4.11: Histograms for USELM with 8-CSK.

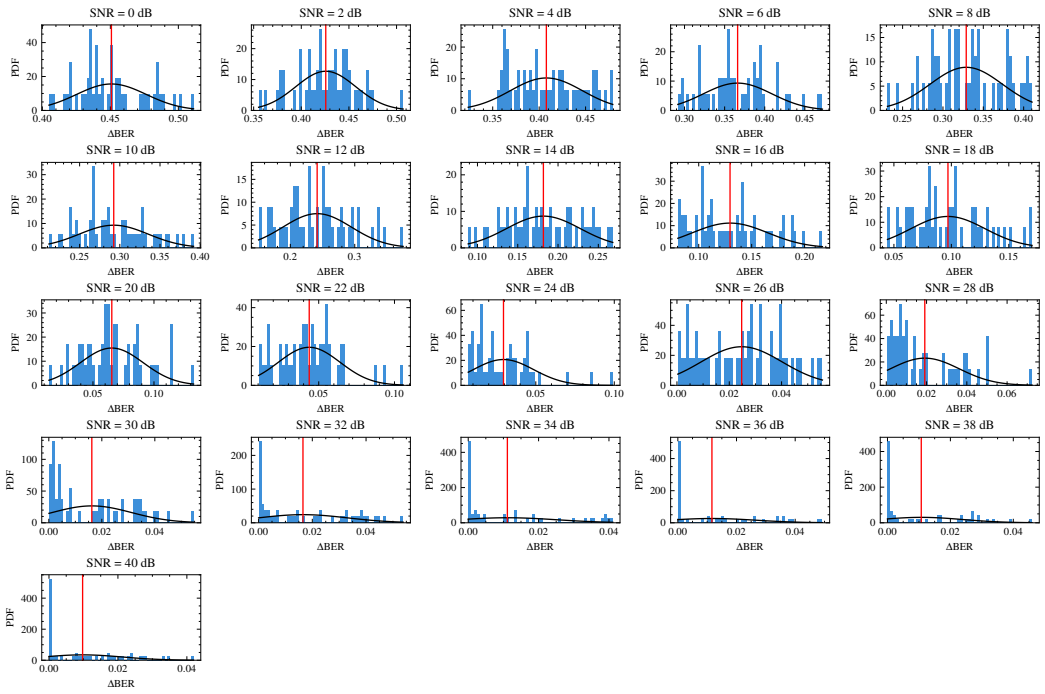


Figure 4.12: Histograms for LS+SELM with 8-CSK.

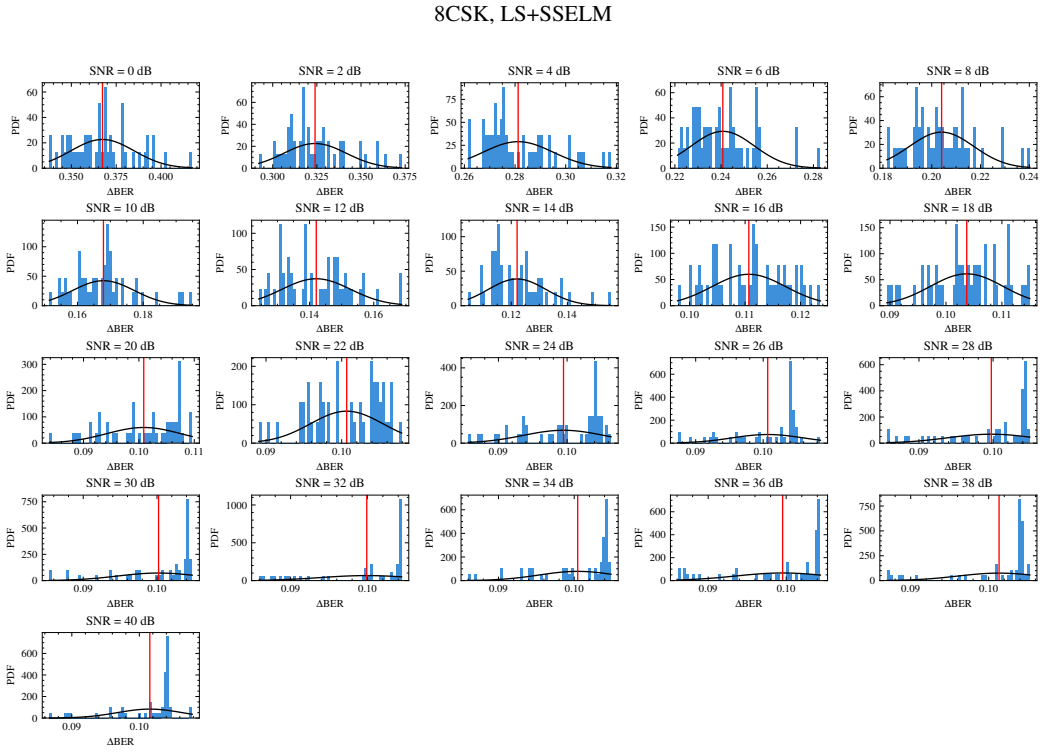


Figure 4.13: Histograms for LS+SSELM with 8-CSK.

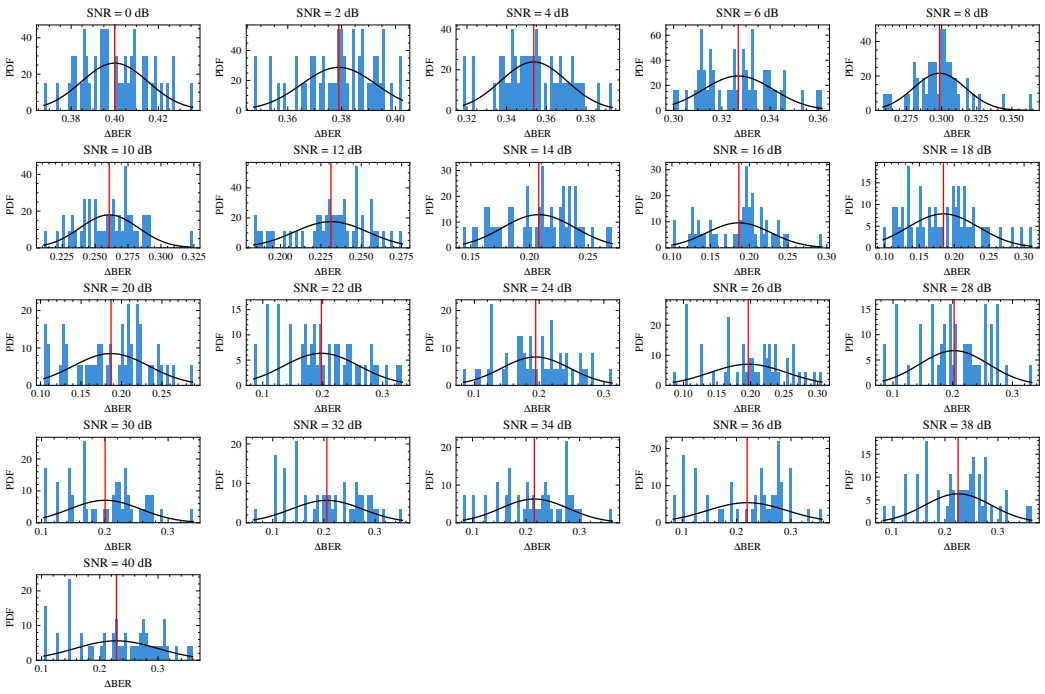


Figure 4.14: Histograms for LS+USELM with 8-CSK.

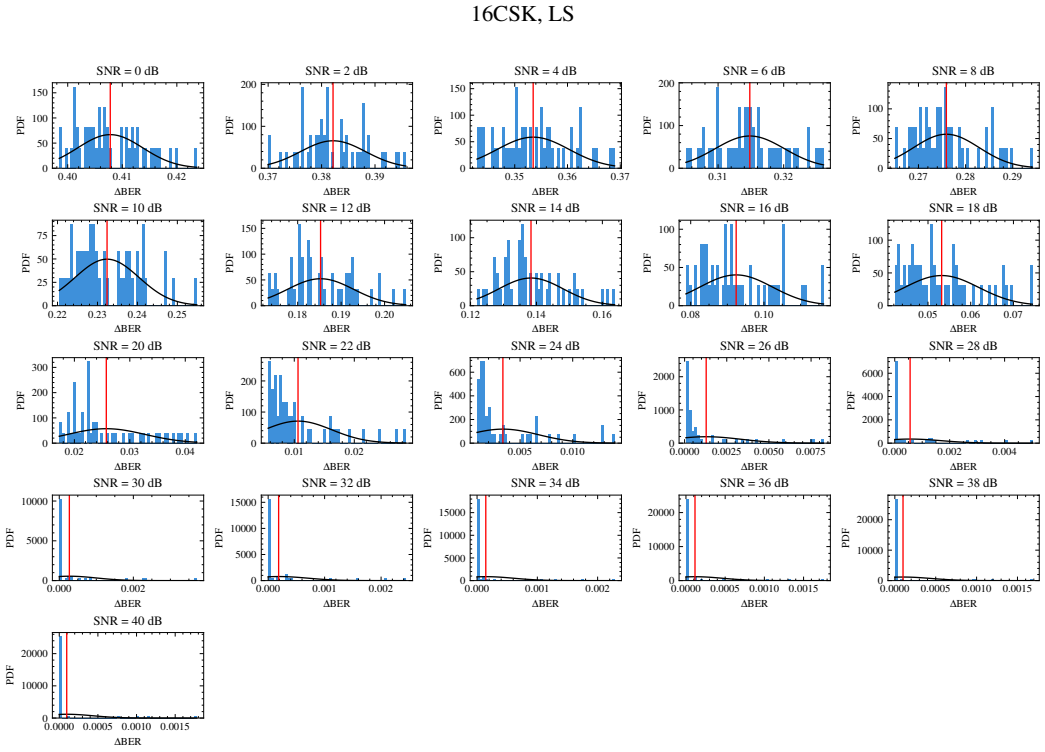


Figure 4.15: Histograms for LS with 16-CSK.

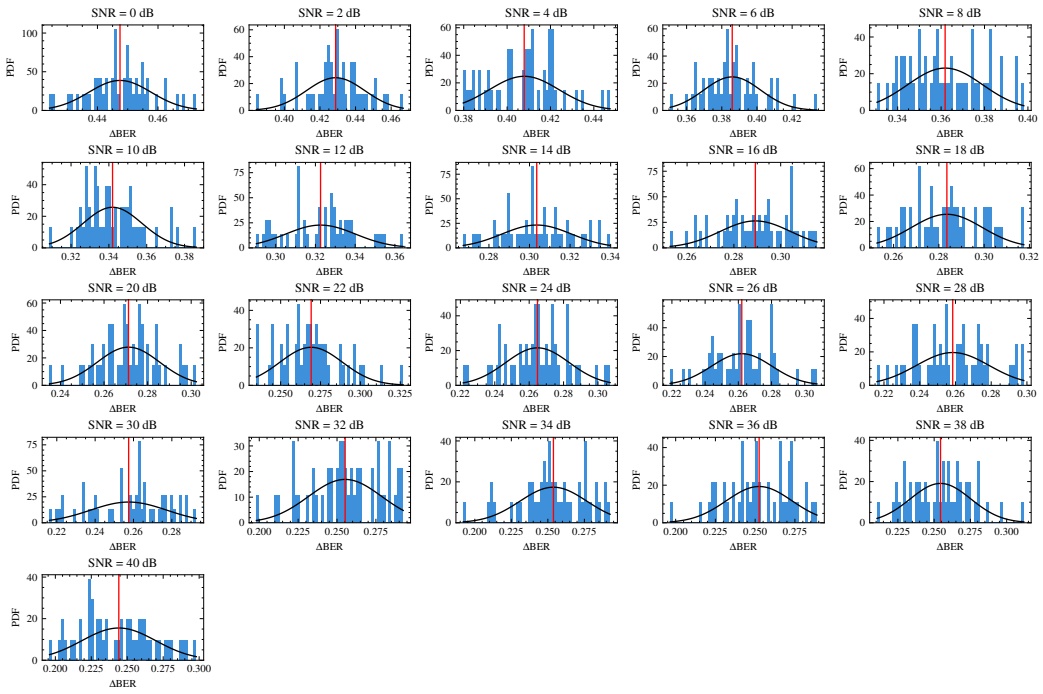


Figure 4.16: Histograms for SELM with 16-CSK.

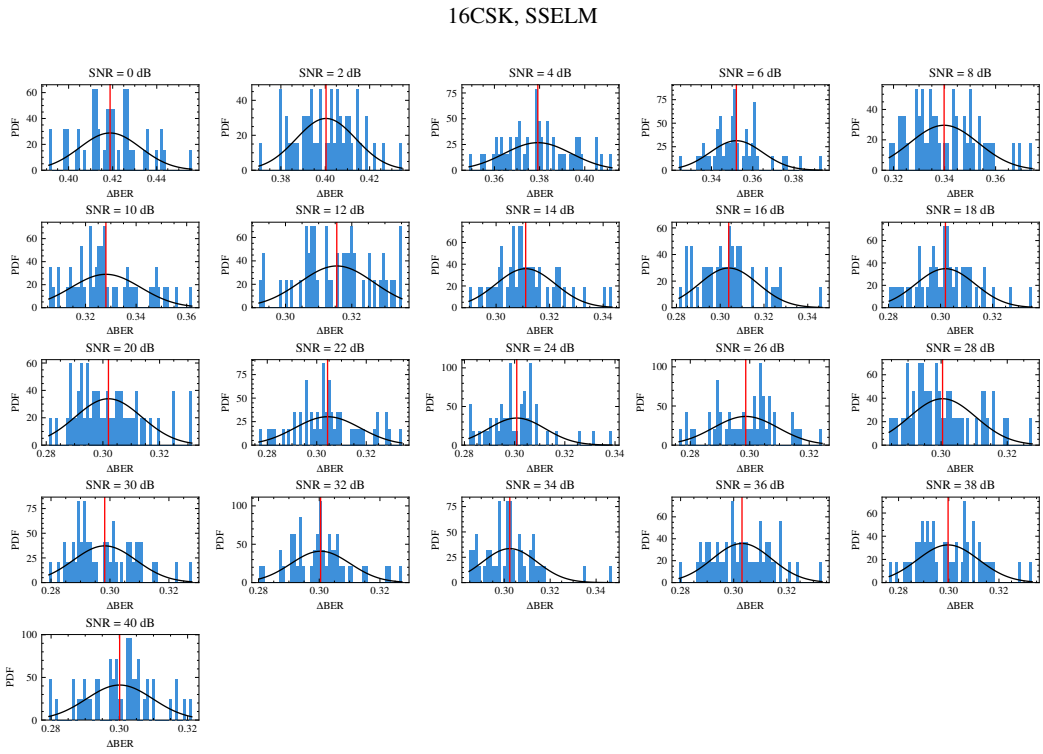


Figure 4.17: Histograms for SSELM with 16-CSK.

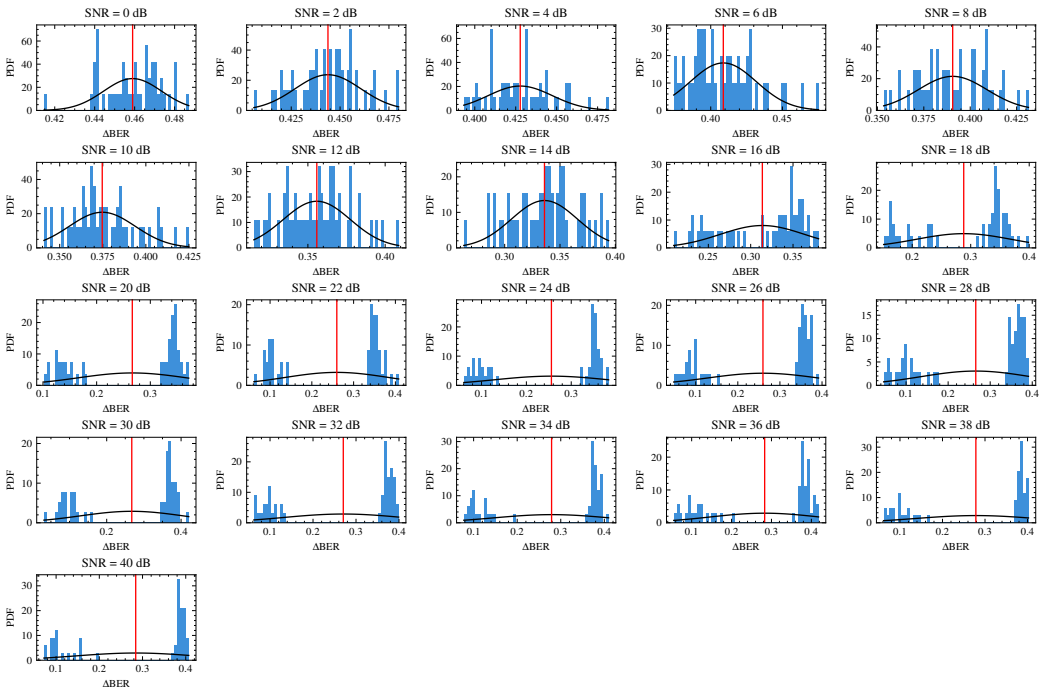


Figure 4.18: Histograms for USELM with 16-CSK.

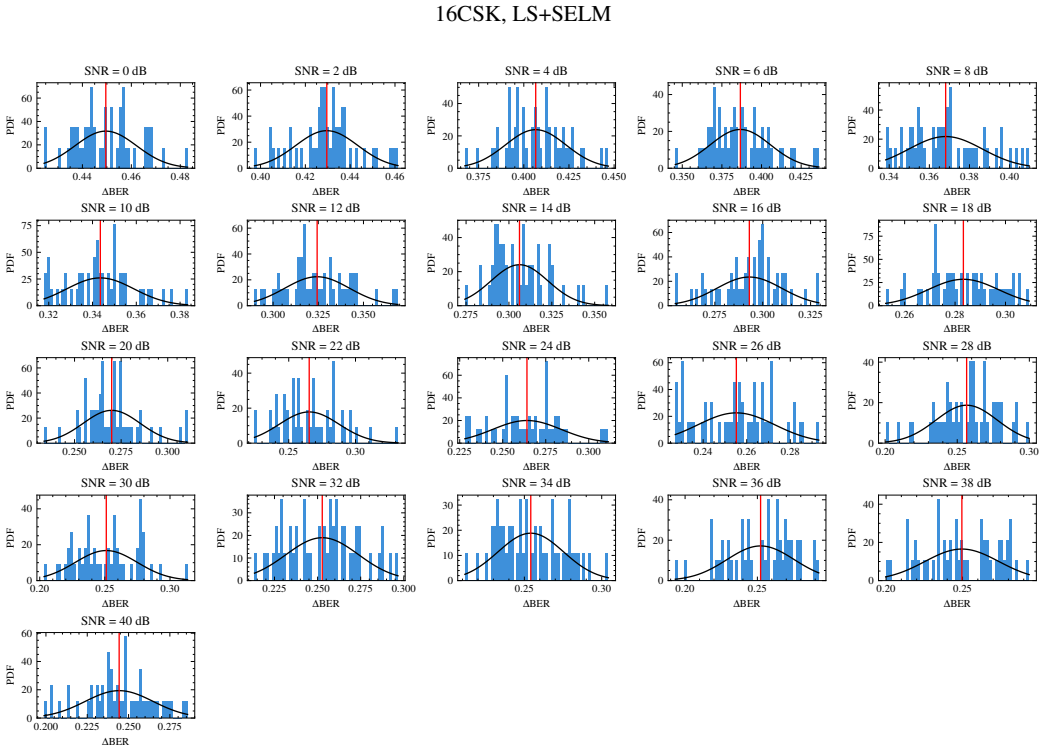


Figure 4.19: Histograms for LS+SELM with 16-CSK.

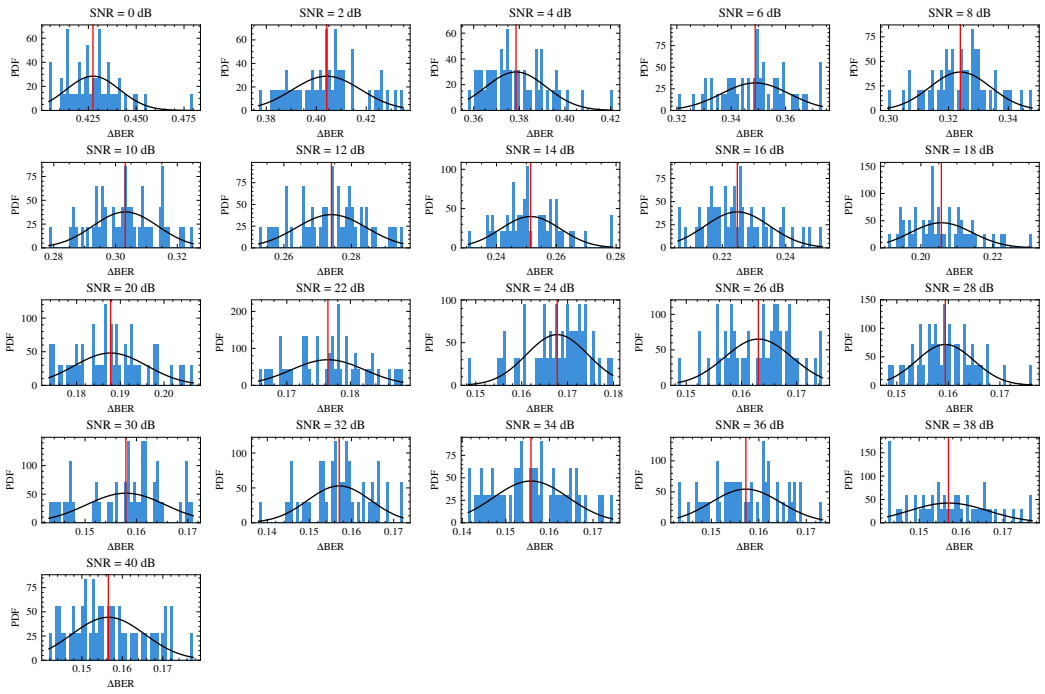


Figure 4.20: Histograms for LS+SSELM with 16-CSK.

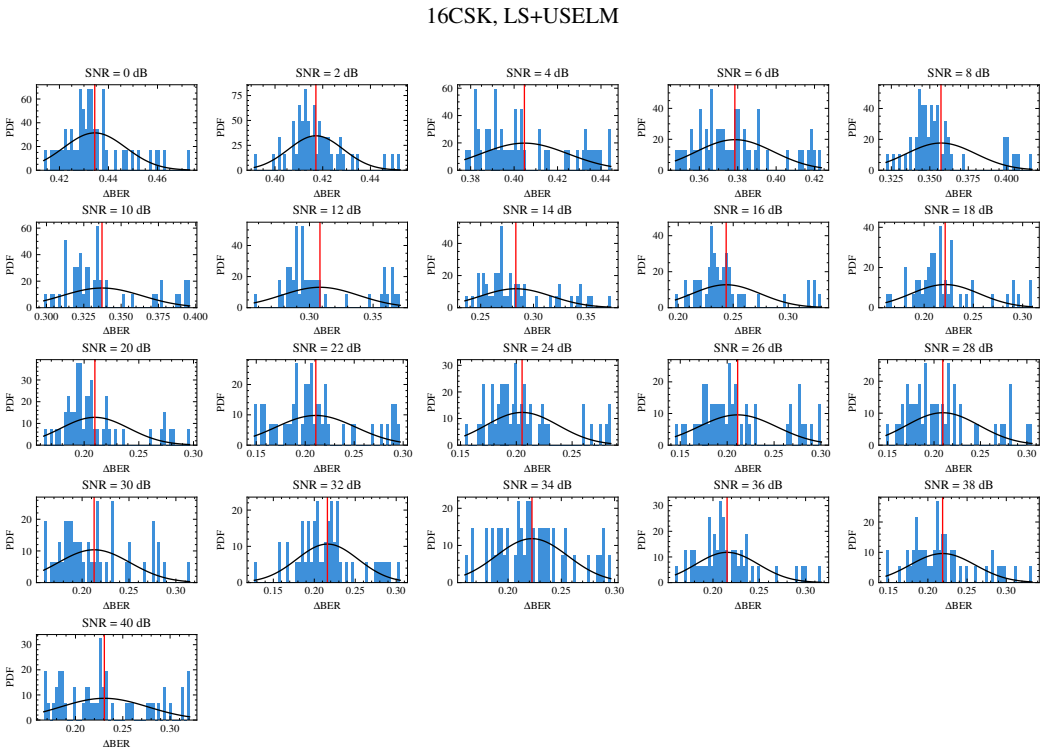
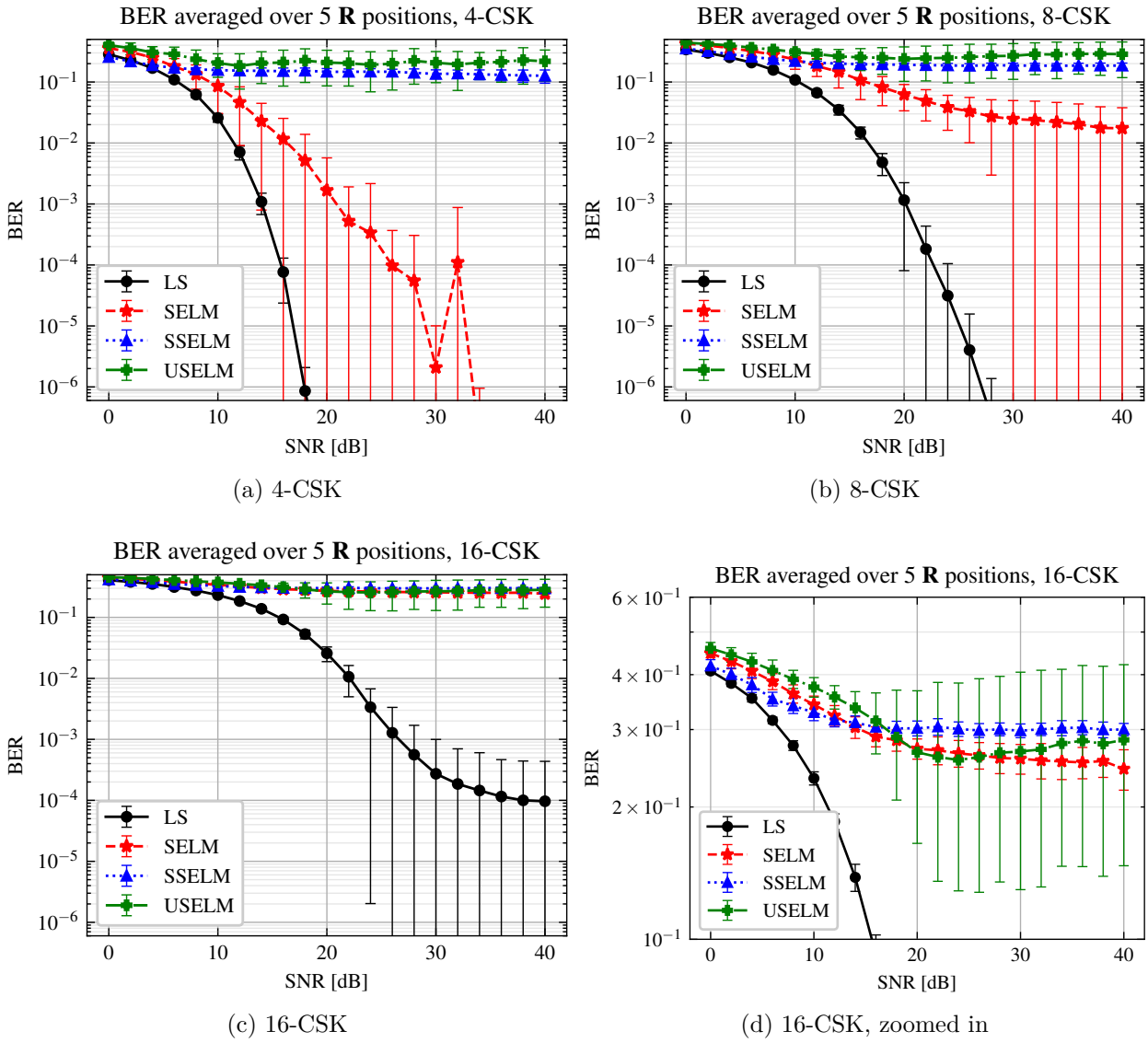


Figure 4.21: Histograms for LS+USELM with 16-CSK.

4.2. BER Curves

From the same data used to plot the histograms, BER curves can be plotted. The mean and standard deviation calculated from the 50 BER values, for each SNR value, can be plotted in a BER versus SNR graph, resulting in the BER curves. Figure 4.22 shows the BER curves for all combinations of the equalization methods {LS, SELM, SSELM, USELM} and the three modulation orders {4-CSK, 8-CSK, 16-CSK}, while Figure 4.23 shows the BER curves for all combinations of the equalization methods {LS, LS+SELM, LS+SSELM, LS+USELM} and the three modulation orders {4-CSK, 8-CSK, 16-CSK}.



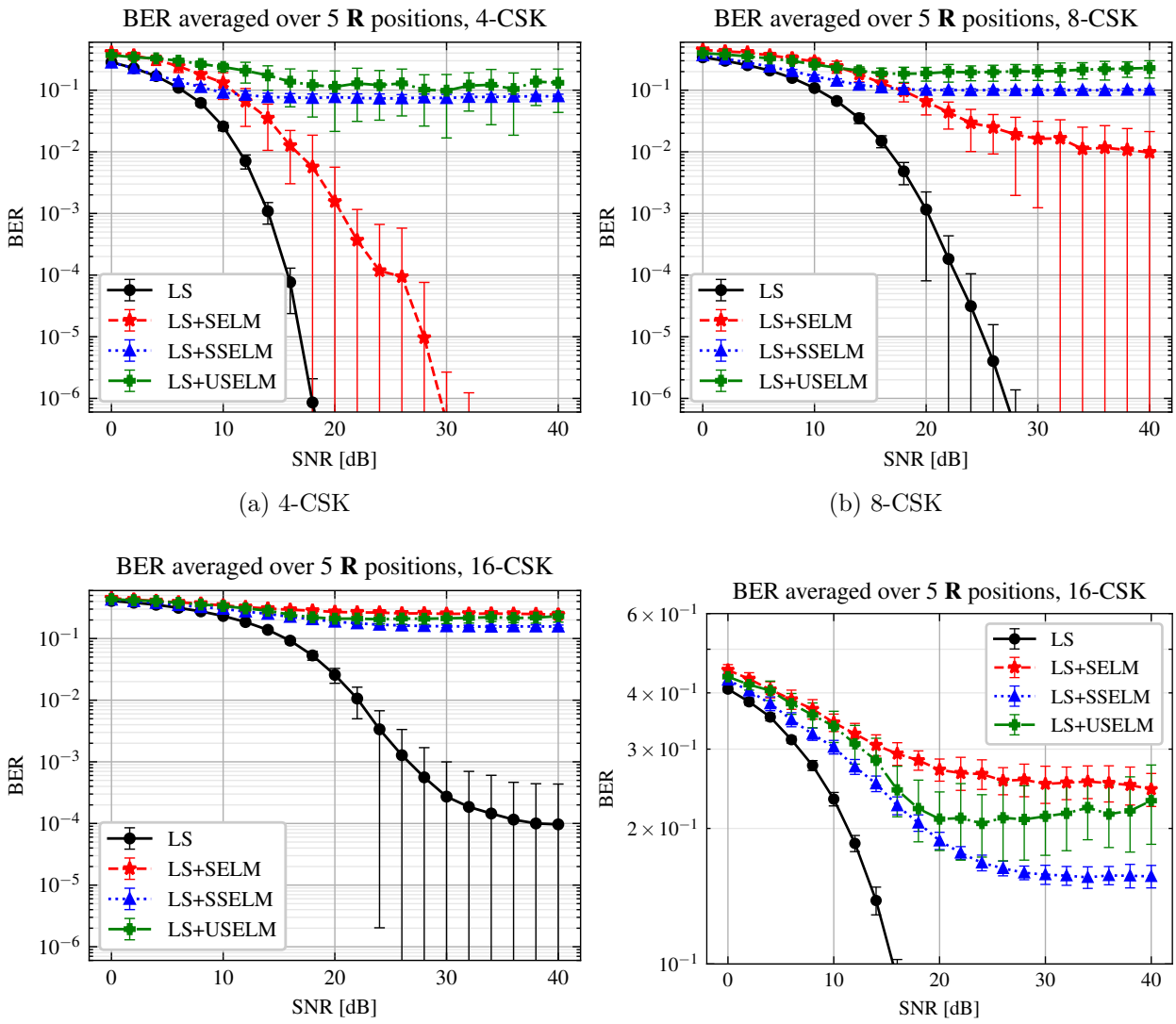
(a) 4-CSK

(b) 8-CSK

(c) 16-CSK

(d) 16-CSK, zoomed in

Figure 4.22: BER curves, LS and ELM equalization methods. Error bars represent one standard deviation.



(a) 4-CSK

(b) 8-CSK

(c) 16-CSK

(d) 16-CSK, zoomed in

Figure 4.23: BER curves, LS and LS+ELM equalization methods. Error bars represent one standard deviation.

4.3. Discussion

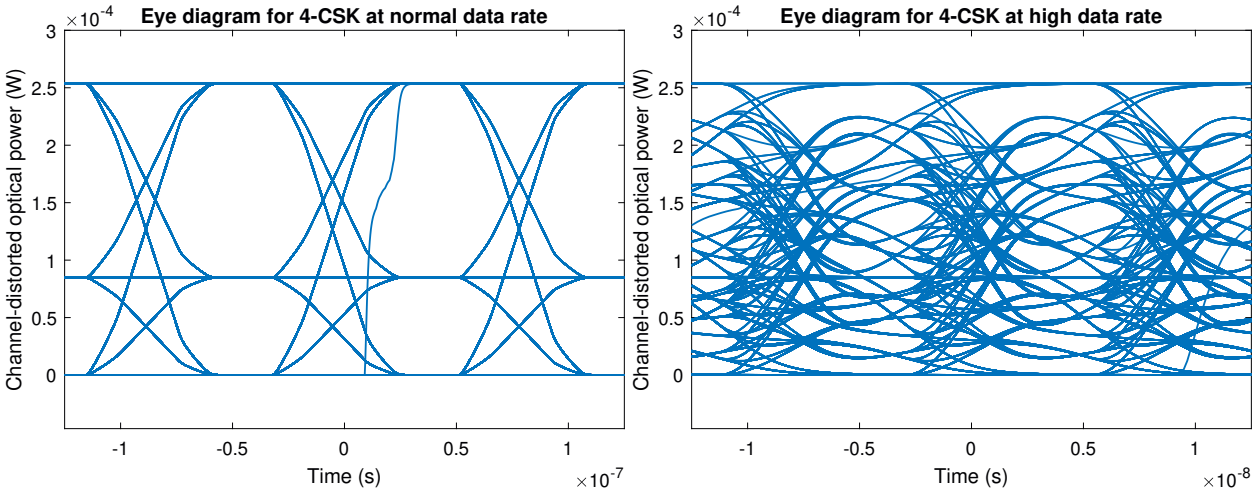
In the first place, analyzing the Figure 4.22, it is clear that the LS equalizer performs much better than any of the other equalizers. Of note is the fact that SSELM and USELM are consistently over 0.1 of BER, even when the noise is low. This implies that there is something inherent deficient in these two algorithms that make them not being able to estimate the received symbols correctly. Looking at the histograms for USELM sheds a light on what might be happening with this algorithm. For 8-CSK and 16-CSK, the USELM BER distribution looks like it either does very badly or relatively well, but nothing in between. This indicates that the clusters USELM is predicting are not getting better with less noise, implying that the USELM algorithm is not capable of clustering the symbols in a useful way. It is also interesting to see the impact of the modulation order, specially in 16-CSK, where

LS seems to plateau at a BER of 10^{-4} approximately, while SELM is very affected, joining SSELM and USELM well above 0.1 of BER.

Comparing the algorithms from Figure 4.22 with the ones from Figure 4.23, one can see small improvements in SSELM and USELM. This is noticeable, for example, in the zoomed-in BER curve for 16-CSK, where LS+SSELM and LS+USELM outperform SELM and LS+SELM. Looking at the histograms for LS+USELM, and comparing them to the histograms of USELM, one can see that the two-sidedness present in the USELM distribution, is gone from the LS+USELM distribution.

Unfortunately, the number of simulations is too small to extract significant statistical results from the histograms, like the probability distributions. The BER curves, however, tell a clear story: the simpler equalizer, LS, has outperformed the more sophisticated ELM in all experiments.

4.3.1. Further analysis



(a) 24 Mbps.

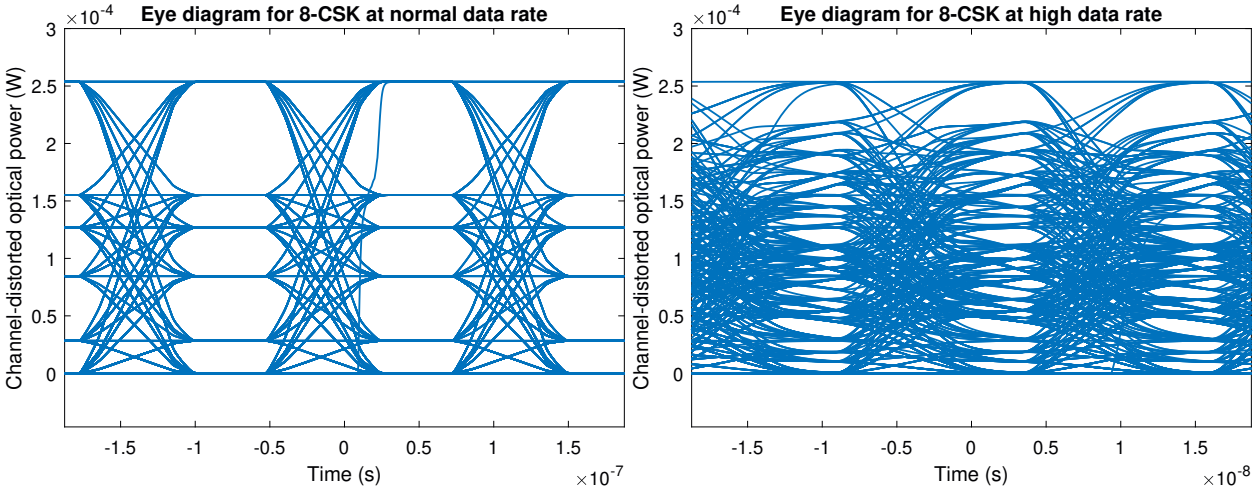
(b) 2.4 Gbps.

Figure 4.24: Eye diagram, 4-CSK.

ELM has been evaluated as an equalizer mainly in OFDM-based wireless communication channels, where it tends to outperform other equalization techniques. Wireless communication channels are very hostile, with ISI being one of the main concerns. On the other hand, the LS equalizer was proposed in the IEEE standard as a way to combat color mismatching stemming from optical filtering and the SPD of the LED. It explicitly does not consider ISI by estimating the channel as a 3×3 matrix.

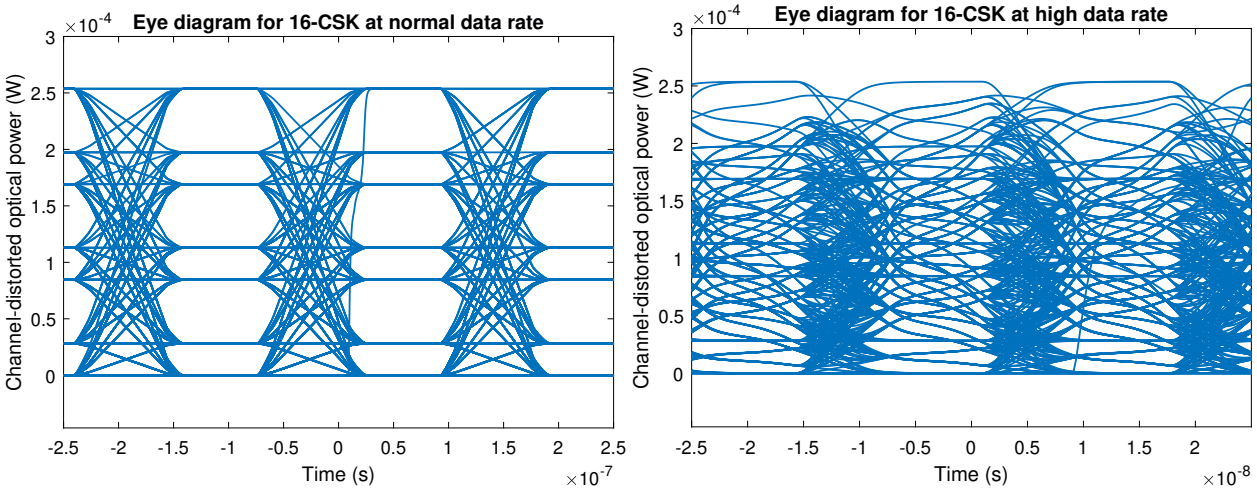
With this in mind, it might be worth considering the possibility that the VLC channel model, as was simulated in this work, simply did not have enough non-linear distortion for the ELM equalizer to be worth it. In other words, given that the channel model can be approximated so well by a linear transformation, it is only natural that the algorithm that actively optimizes for this matrix, even in closed form, performs better than the one who starts with random weights.

But, is there really no ISI in the underground mine channel simulations? As it turns out,



(a) 72 Mbps. (b) 7.2 Gbps.

Figure 4.25: Eye diagram, 8-CSK.



(a) 96 Mbps. (b) 9.6 Gbps.

Figure 4.26: Eye diagram, 16-CSK.

when not taking into consideration the chromacity mismatching, there is pretty much no ISI at these data rates. To show this, the eye diagrams for the channel-distorted optical signal (specifically, the red band) were calculated, considering the PD position to be $\mathbf{R}_1$. Figure 4.24 presents the eye diagram for 4-CSK, 4.25 for 8-CSK and 4.26 for 16-CSK. These signals do not consider the optical filtering nor the AWGN noise, and they were calculated for the maximum allowed data rate in the IEEE standard, shown to the left, and for 10 times that data rate, shown to the right. It can be seen, on the diagrams to the left, that there is a sampling period where there is virtually no ISI, at least considering that specific color band. In the case with the higher data rate, such ISI-free sampling periods do not exist.

What this shows is that the underground mine channel model used in this work is very

capable of inducing non-linear distortions, but the maximum data rates allowed in the IEEE standard are too low for the channel distortion to affect the signal significantly. As a result, the channel can be well approximated by a linear transformation, and consequently LS outperforms algorithms such as ELM. The former finds an optimal solution, while the latter tries to generalize.

4.3.2. Computational complexity

A common method of comparison between two algorithms is the number of floating point operations (flops) it requires. Given a linear equation, $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A} \in \mathbb{R}^{m \times n}$, MATLAB can solve for $\mathbf{x}$ very efficiently by using the function `mldivide`. This function can employ multiple algorithms depending on the characteristics of matrix $\mathbf{A}$, each with different required flops. For our purposes, MATLAB will use the LU solver when $\mathbf{A}$ is a square matrix, and the QR solver when it is not. The number of flops required for solving the linear system using each of the aforementioned algorithms is provided in Table 4.1. Both algorithms find an adecuate solution even when the system is overdetermined (by minimizing $||\mathbf{Ax}-\mathbf{b}||^2$) or underconstrained (by minimizing $||\mathbf{x}||^2$). In the LS equalizer case, the linear problem takes

Table 4.1: Number of flops for solving the system $\mathbf{Ax} = \mathbf{b}$, when using MATLAB's `mldivide` function.

Matrix dimensions, $\mathbf{A} \in \mathbb{R}^{p \times q}$	**Method**	**Number of flops**
$p = q$	LU solver [28]	$\frac{2}{3}q^3 + \frac{5}{6}q^2 - \frac{1}{6}q$ [29]
$p \neq q$	QR solver [28]	$2pq^2 - \frac{2}{3}q^3 + 2pq + q^2$ [22]

the form $\mathbf{HV} = \hat{\mathbf{V}}$, or equivalently, $\mathbf{V}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}} = \hat{\mathbf{V}}^{\mathrm{T}}$. In this case, $\mathbf{H}^{\mathrm{T}}$ is the unknown $\mathbf{x}$ and $\mathbf{V}^{\mathrm{T}}$ is the matrix $\mathbf{A}$, thus $p = n_s/2$ and $q = 3$, making the cost $12n_s - 9$ flops according to Table 4.1, when $p \neq q$. Note that the case $p = q$ is not worth considering in the LS equalizer since $q = 3$ is not a system parameter one can modify. Consider that the averaging of subsequent symbols in the channel estimation sequence must also be considered: the sum of $n_s/2$ pairs of symbols is $3n_s/2$ flops (each sum is one flop, and each symbol consists of three values) and the scaling by $1/2$ of $n_s/2$ symbols is $3n_s/2$ flops (each scalar division is one flop, and each symbol consists of three values). In total, the training of the LS equalization method takes $15n_s - 9$ flops.

The training of the SELM equalizer is very similar, since the problem is to solve the linear system $\mathbf{\Lambda}\beta = \mathbf{y}$. In this case, β is the unknown $\mathbf{x}$ and $\mathbf{\Lambda}$ is the matrix $\mathbf{A}$, thus $p = n_s$ and $q = n_h$, making the cost $\frac{2}{3}n_s^3 + \frac{5}{6}n_s^2 - \frac{1}{6}n_s$ flops if $n_s = n_h$, and $2n_s n_h^2 - \frac{2}{3}n_h^3 + 2n_h n_s + n_h^2$ if $n_s \neq n_h$, according to Table 4.1. Additionally, the cost of calculating each $\Lambda_{ij} = g(\boldsymbol{\omega}^T\mathbf{x}_i + \mu_j)$, for $i = 1, \ldots, n_s$ and $j = 1, \ldots, n_h$, when initializing the matrix $\mathbf{\Lambda}$ must be added to the total flop count. The cost of calculating each element Λ_{ij} is approximately 11 flops: 6 for the vector product $\boldsymbol{\omega}^T\mathbf{x}_i$, 1 for the sum $\boldsymbol{\omega}^T\mathbf{x}_i + \mu_j$, and at least 4 for the application of $g(\cdot)$ (a sigmoid function); considering there are $n_s n_h$ elements in $\mathbf{\Lambda}$, it makes the total cost of initializing the matrix $11n_s n_h$ flops. With this, the total flops required for training the SELM equalizer is approximately $\frac{2}{3}n_s^3 + \frac{71}{6}n_s^2 - \frac{1}{6}n_s$ when $n_s = n_h$, and $2n_s n_h^2 - \frac{2}{3}n_h^3 + 13n_h n_s + n_h^2$ when $n_s \neq n_h$.

The total flop count for both the LS and SELM equalizer is summarized in Table 4.2. It is clear from these results, that the computational complexity of SELM is much greater than LS. Even though both involve solving a linear system of equations, LS is concerned with the matrix $\mathbf{V}^{\mathrm{T}}$ whose columns are fixed at $q = 3$, while SELM is concerned with the matrix $\mathbf{\Lambda}$,

Table 4.2: Number of flops for LS and SELM in terms of number of channel estimation symbols (n_s) and number of hidden neurons (n_h).

Equalization method	**Assumptions**		**Number of flops**
LS	$p = \frac{1}{2}n_s$ $q = 3$		$15n_s - 9$
SELM	$p = n_s$ $q = n_h$	$n_s = n_h$	$\frac{2}{3}n_s^3 + \frac{71}{6}n_s^2 - \frac{1}{6}n_s$
		$n_s \neq n_h$	$2n_s n_h^2 - \frac{2}{3}n_h^3 + 13n_h n_s + n_h^2$
		$n_h = 3$	$57n_s - 9$

whose columns depend on the number of hidden neurons n_h, an hyperparameter that must be pre-set. The computational complexity for SELM grows in polynomial time with respect to n_h, making it very costly to add additional hidden neurons. Setting $n_h = 3$, so $\mathbf{\Lambda}$ and $\mathbf{V}^{\mathrm{T}}$ have the same number of rows, gives SELM a computational cost of $57n_s - 9$ flops. If we select $n_s = 24$ and $n_h = 3$, then LS requires around 351 flops, while SELM requires 1359 flops, making LS about 4 times faster than SELM with this parameters. Since SSELM and USELM are algorithms much more complex than SELM, and thus, more computationally expensive, they were not considered in this analysis.

Chapter 5

Conclusions and Future Work

5.1. Conclusions

In this work, a simulation scenario and computational experimental setup was designed with the objective of evaluating the performance of different equalization methods based in the ELM algorithm for a CSK-based VLC system in an underground mine. These equalization methods were compared to the IEEE VLC standard proposed equalization method for CSK. The simulation scenario consisted of an underground mine in which a single transmitter sent data to a single receiver through a VLC system using the CSK modulation. Phenomena inherent to underground mines such as optical signal blockage by heavy machinery, light scattering due to dust particles, irregular non-flat walls, tilted and rotated LEDs and PDs, were all considered in the simulation scenario by incorporating a state-of-the-art underground mine channel model.

The performance results for the LS equalizer confirm the hypothesis that LS is a feasible equalizer for the single-input single-output (SISO) VLC link based in CSK modulation, even under the harsh conditions of underground mines. On the other hand, the results overwhelmingly show that the ELM-based equalizer is not suited for this kind of system, since it is outperformed by the computationally cheaper LS equalizer. With these results, the general objective of evaluating the feasibility of the ELM equalizer in this particular scenario is achieved, and the result of this evaluation is that it is not feasible. All specific objectives were also achieved: the ELM-based equalizer and the equalizer proposed by the IEEE were both compared through the BER and their computational complexity. Furthermore, technical limitations set by the IEEE standard were taken into account, such as the length of the data frame, the length of the channel estimation sequences and most importantly, the data rate of the system. Additionally, the performance of the equalizers was done considering different communication system parameters, such as the modulation order and position of the receiver.

ELM has been used as equalizer successfully in other wireless communication systems. What appears to be different in VLC systems is that the data rates allowed by the IEEE VLC standard are too low for the multipath component to generate any significant non-linear distortion in the received signal. The result is that the CSK-VLC channel can easily be approximated by a 3×3 matrix. While the LS equalizer finds this optimal matrix easily, ELM-based equalizers struggle.

To the best of the author's knowledge, this is the first time a CSK-based VLC system has been simulated in an underground mine scenario. It is also the first time that a ELM-based equalizer has been tested in a simulated CSK-based VLC system at all.

5.2. Future Work

Any additional experiment where the signal is heavily distorted in a non-linear fashion is a good direction for a follow up. It would be interesting to see if the LS equalizer starts to struggle against the ELM-based equalizer or not, when the ISI is increased. This could be as easy as using a higher data rate. Another approach could be the study of a multiple-input multiple-output system, where co-channel interference becomes a problem. Lastly, it would be interesting to evaluate the ELM-based equalizer in other difficult VLC channels, such as underwater light communications or vehicle-to-vehicle light communications.

Modifying the system simulation and making it more complex (and thus more non-linear) is also an interesting line of inquiry to follow, since it appears that the simplifications done on the simulation, such as considering that the channel affects all wavelengths of light in the same fashion, underscores the non-linearity of the channel. Another option under this same line of inquiry, would be to actually build the system in real life, but this could take years of work.

Finally, in order to perform a more in-depth comparison of the computational cost of each equalizer, it would be a good idea to also obtain the time each equalizer takes to equalize the signal.

Bibliography

[1] S. Yarkan, S. Guzelgoz, H. Arslan and R. R. Murphy, "Underground Mine Communications: A Survey," in *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp. 125-142, 2009, doi: 10.1109/SURV.2009.090309.

[2] J. Wang, A. Al-Kinani, W. Zhang and C. -X. Wang, "A new VLC channel model for underground mining environments," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2017, pp. 2134-2139, doi: 10.1109/IWCMC.2017.7986613.

[3] P. A. Hoeher, *Visible light communications: theoretical and practical foundations*. Munich: Carl Hanser Verlag GmbH & Co KG, 2019.

[4] P. Palacios et al., "A VLC Channel Model for Underground Mining Environments With Scattering and Shadowing," in *IEEE Access*, vol. 8, pp. 185445-185464, 2020, doi: 10.1109/ACCESS.2020.3030615.

[5] J. P. Rivelli Malcó, D. Zabala-Blanco, R. D. Breslin, P. P. Játiva, A. D. Firoozabadi and M. Flores-Calero, "Wireless OFDM links with equalizers based on extreme learning machines," *2021 IEEE Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, 2021, pp. 1-6, doi: 10.1109/CHILECON54041.2021.9703008.

[6] D. Zabala-Blanco, M. Mora, C. A. Azurdia-Meza and A. Dehghan Firoozabadi. "Extreme learning machines to combat phase noise in RoF-OFDM schemes," in *Electronics*, 2019, vol. 8, no. 9, p. 921, doi: 10.3390/electronics8090921.

[7] J. Liu, K. Mei, X. Zhang, D. Ma and J. Wei, "Online Extreme Learning Machine-Based Channel Estimation and Equalization for OFDM Systems," in *IEEE Communications Letters*, vol. 23, no. 7, pp. 1276-1279, July 2019, doi: 10.1109/LCOMM.2019.2916797.

[8] D. Gao, Q. Guo, J. Tong, N. Wu, J. Xi and Y. Yu, "Extreme-Learning-Machine-Based Noniterative and Iterative Nonlinearity Mitigation for LED Communication Systems," in *IEEE Systems Journal*, vol. 14, no. 4, pp. 4674-4683, December 2020, doi: 10.1109/JSYST.2020.2978535.

[9] 802.15 WG - Wireless Specialty Networks (WSN) Working Group, "IEEE Standard for Local and metropolitan area networks - Part 15.7: Short-Range Optical Wireless Communications," in *IEEE Std 802.15.7-2018 (Revision of IEEE Std 802.15.7-2011*), pp. 1-407, 2019

[10] B. Sklar, *Digital communications*, 3rd ed., vol. 2. New Jersey: Prentice Hall, 2001.

[11] J. G. Proakis et al., *Communication systems engineering*, 2nd ed., vol. 2. New Jersey: Prentice Hall, 1994.

[12] C. M. Bishop, *Pattern recognition and machine learning*. New York: Springer, 2006.

[13] Z. Ghassemlooy, et al., Eds., *Visible light communications: theory and applications*. Boca Raton, FL: CRC Press, 2017.

[14] User:PAR, *"File:CIExy1931.png."* Wikimedia Commons, 2005. Accessed: Jun. 3, 2022. Available: https://commons.wikimedia.org/wiki/File:CIExy1931.png.

[15] E. Monteiro and S. Hranilovic, "Design and Implementation of Color-Shift Keying for Visible Light Communications," in *Journal of Lightwave Technology*, vol. 32, no. 10, pp. 2053-2060, May 15th, 2014, doi: 10.1109/JLT.2014.2314358.

[16] Z. Ghassemlooy, W. Popoola and S. Rajbhandari, *Optical wireless communications: system and channel modelling with MATLAB®*. Boca Raton, FL: CRC Press, 2019.

[17] P. H. Pathak et al., "Visible light communication, networking, and sensing: A survey, potential and challenges.," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, 2015.

[18] J. Wang, A. Al-Kinani, J. Sun, W. Zhang and C. -X. Wang, "A path loss channel model for visible light communications in underground mines," in *2017 IEEE/CIC International Conference on Communications in China (ICCC)*, 2017, pp. 1-5, doi: 10.1109/ICCChina.2017.8330479.

[19] M. M. Céspedes & A. García Armada, "Characterization of the Visible Light Communications during the Construction of Tunnels," in *2019 16th International Symposium on Wireless Communication Systems (ISWCS)*, 2019, pp. 356-360, doi: 10.1109/ISWCS.2019.8877189.

[20] Y. Zhai & S. Zhang, "Visible light communication channel modelsand simulation of coal workface energy coupling," in *Math. Problems Eng.*, vol. 2015, pp. 1-10, Dec. 2015.

[21] A. Tamayo. "Estudio del estándar 802.15. 7 del IEEE sobre sistemas de comunicación por luz visible." *Universidad de Sevilla*, 2016.

[22] D. Q Lee, "Numerically efficient methods for solving least squares problems", in *Research Experience for Undergraduates (REU) 2012*, University of Chicago, IL, Aug. 12, 2012. Available: https://math.uchicago.edu/~may/REU2012/REUPapers/Lee.pdf.

[23] G. Huang, S. Song, J. N. D. Gupta and C. Wu, "Semi-Supervised and Unsupervised Extreme Learning Machines," in *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2405-2417, December 2014, doi: 10.1109/TCYB.2014.2307349.

[24] A. Yokoi, J. Son, and T. Bae, "More description about CSK constellation.," *IEEE 802.15.7 standard*, 2010.

[25] C. R. Schoenenman, *Converting Frequency to RGB*, comp.graphics.algorithms (Usenet newsgroup), Nov. 18, 1991. Accessed: Jul. 5, 2022 [Online]. Available: http://steve.hollasch.net/cgindex/color/freq-rgb.html.

[26] LEE Filters, *Colour effect filters: Light the world your way*, leefilters.com. Accessed: Jul. 5, 2022. [Online]. Available: https://leefilters.com/lighting/colour-effect-lighting-filters/.

[27] Thorlabs, "PDA36A(-EC) Si Switchable Gain Detector: User Guide," 13053-D04 datasheet, July 2017.

[28] The MathWorks, Inc. *mldivide,* \, MathWorks Help Center, 2022. Accessed: Jun. 15, 2022. [Online]. Available: https://www.mathworks.com/help/matlab/ref/mldivide.html

[29] R. M. Prosser (2019). "LU Factorization," in *CS370 Numerical Computation*. University of Waterloo, ON, Canada. [Lecture notes]. Available: https://student.cs.uwaterloo.ca/~cs370/notes/LU.

[30] R. Becerra, "Source Code: Evaluation of Extreme Learning Machine as Channel Equalizer for Color-Shift Keying-Based Visible Light Communication Systems Employed in Underground Mining Scenarios", *GitHub*, 2022. [Source code]. Available: https://github.com/RaiBP/vlc_project.

[31] G. Huang, S. Song, J. N. D. Gupta and C. Wu, "SS-US-ELM", *GitHub*, 2015. [Source code]. Available: https://github.com/ExtremeLearningMachines/SS-US-ELM.

Annex

Source Code

The code used for implementing the VLC system simulation and obtaining the results presented in this work is open-source and can be found on [30]. Some of the most important files are:

- `monte_carlo_simulation.m`: Main script, used for performing the 50 Monte Carlo iterations. It consists of four nested for-loops, from the outermost to the innermost: loop over each Monte Carlo iteration, loop over each SNR point, loop over each equalization method and loop over each modulation order.
- `get_channel_model.m`: Given a LED and PD position, this function calculates the VLC channel model in the underground scenario. This code was provided by the authors of [4], and permission was given to use and modify it for this work.
- `form_frames.m`: Formats a random binary sequence (the message) into a series of frames, as described in Section 3.3.2.
- `modulate_frames.m`: Converts the series of frames into a modulated signal, as described in Section 3.3.3.
- `apply_channel.m`: Simulates the transmission and reception of the modulated signal as described in Section 3.3.4.
- `extract_frames.m`: Extracts the frames from the received modulated signal, as described in Section 3.3.4.3.
- `equalization.m`: Equalizes each received frame as described in Section 3.3.5.
- `selm_equalization.m`: Equalizes the received signal using the supervised ELM algorithm. This code is a modified version of the source code provided by the authors of [6], and permission was given to use and modify it for this work.
- `sselm_equalization.m`: Equalizes the received signal using the semi-supervised ELM algorithm. This code is a modified version of the source code uploaded by the authors of [23] to public GitHub repository [31], under the Apache 2.0 license, granting permission for use and modification in this work.
- `uselm_equalization.m`: Equalizes the received signal using the usupervised ELM algorithm. This code is a modified version of the source code uploaded by the authors of

[23] to public GitHub repository [31], under the Apache 2.0 license, granting permission for use and modification in this work.