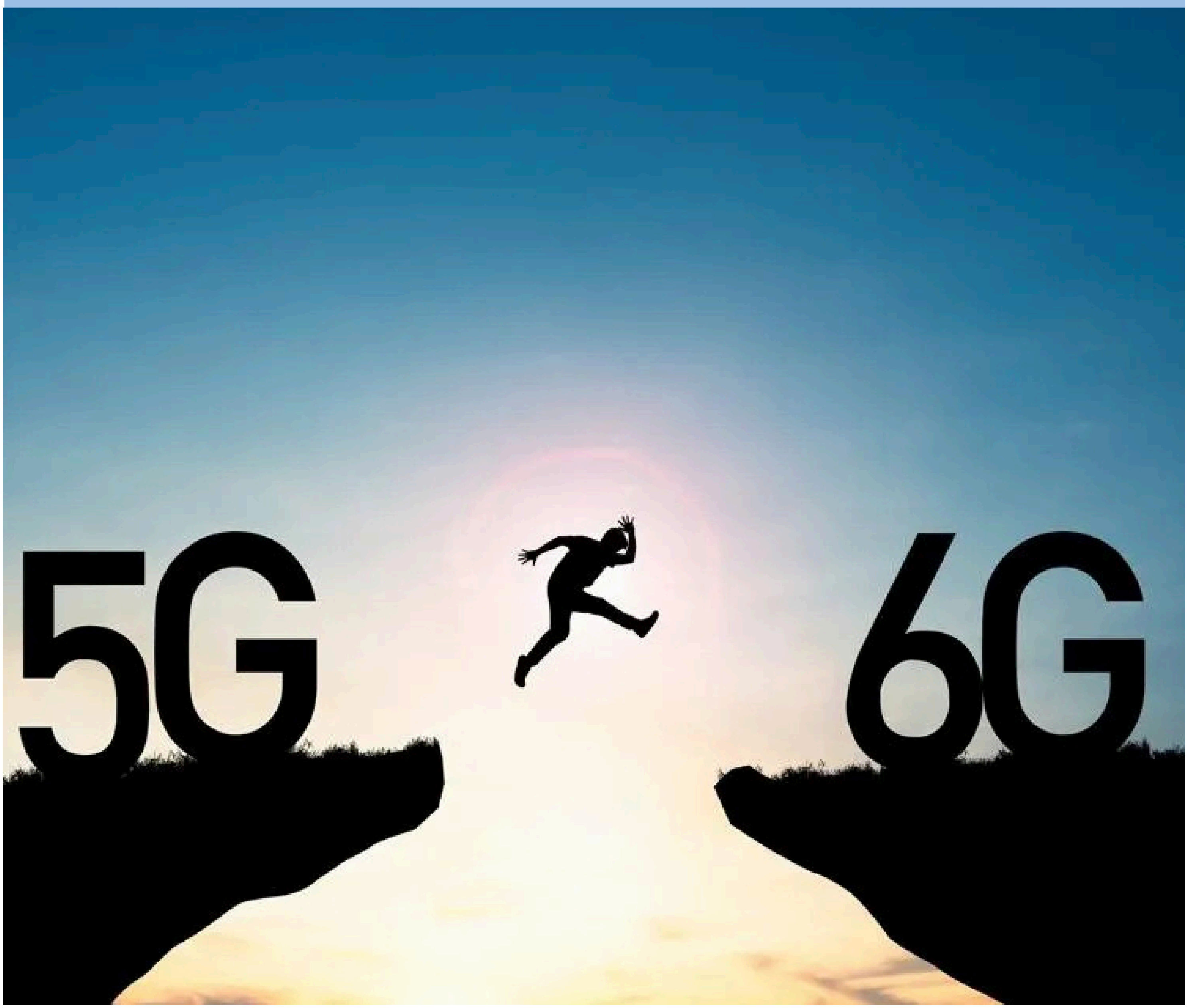
University of Stavanger

BRAGE SOLHIEM

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

StreamLit based web user interface for NVIDIA Sionna based 6G Physical-Layer Simulation

Master's Thesis - Data Science - December 2023



University of Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study programme/specialisation: Master's in engineering / Data Science	Autumn semester, 2023 Open / ~~Confidential~~
Author: Brage Solheim Brage Solheim……………………………	
Programme coordinator: Ferhat Özgur Catak Supervisor(s): Ferhat Özgur Catak	
Title of master's thesis: StreamLit based web user interface for NVIDIA Sionna based 6G Physical-Layer Simulation	
Credits (ECTS): 30	
Keywords: Physical Layer Simulation, Wireless Communication Systems, Signal Processing, Web User Interface	Number of pages: 69 + Appendix: 1 page Oslo, 15. december 2023

Abstract

This thesis presents the development of a web-based user interface for the NVIDIA Sionna-based 6G Physical-Layer Simulation, leveraging the Streamlit framework. The primary objective of this project is to create an accessible, interactive platform that facilitates easy configuration, execution, and monitoring of complex 6G physical layer simulations.

The implementation of this interface utilizes the capabilities of Python, alongside integration with TensorFlow, Keras, and NVIDIA Sionna, ensuring a robust and efficient simulation environment. Key features of the application include dynamic parameter adjustment for simulations, real-time visualization of outcomes, and the ability to interactively explore the effects of different physical layer configurations on the overall system performance.

Through this work, it was demonstrated that the integration of advanced simulation tools with user-friendly web interfaces could significantly enhance the understanding and accessibility of complex communication systems. The system's performance, validated through various simulation scenarios, illustrates the effectiveness of the interface in simplifying the complexities involved in 6G physical layer simulations.

Acknowledgements

I would like to express my profound gratitude to Professor Ferhat Özgur Catak. I am deeply thankful for the opportunity to work under his guidance and for the wisdom he has imparted.

I would also like to extend my deepest appreciation to my family for their unwavering support and endless love throughout my years of study. The journey to this point has been filled with challenges and triumphs, and through it all, my family has been my constant source of strength and encouragement.

Lastly, but certainly not least, I want to express my love to my girlfriend, Maria, for her constant encouragement, patience, and unwavering support.

Contents

List of Acronyms

Chapter 1

Introduction

In the rapidly advancing realm of communication technologies, 6G emerges as the forthcoming frontier, promising unprecedented capabilities and applications that extend well beyond the current limitations of 5G. As these next-generation technologies evolve, so does the complexity of their underlying systems, particularly at the physical layer where data modulation, transmission, and reception occur. Simulating these processes, thus, becomes essential to understanding, analyzing, and refining the technology before real-world deployment.

NVIDIA's Sionna is a state-of-the-art GPU-accelerated open-source library designed explicitly for link-level simulations of communication systems. Its ability lies not only in simulating complex 6G communication architectures but also in its native support for integrating machine learning. However, with great power comes the necessity for an interface that can seamlessly unlock this potential, catering to researchers, developers, and enthusiasts without delving into the intricacies of backend code each time a simulation needs to be run or modified.

This master thesis is dedicated to bridging this gap. It acknowledges the need for an intuitive and effective web user interface and aims to harness the capabilities of StreamLit and Sionna, to develop a dynamic, interactive, and user-friendly platform. This interface will enable users to effortlessly configure physical layer simulations, initiate and monitor their progress in real time.

Furthermore, this thesis encompasses a comprehensive review and analysis of the relevant background and theory. It will explore the theoretical underpinnings and advancements in the field, providing a solid academic foundation for the development of the web interface. This exploration will not only aid in understanding the functionalities and requirements of NVIDIA Sionna and StreamLit but also in identifying best practices and approaches in user interface design for complex simulation systems. Ultimately, the thesis aim to present a well-rounded Streamlit- and Sionna-based web user interface for physical layer simulations, built on theoretical insights.

1.1 Objectives

1.1.1 Understand the Sionna library

- Gain familiarity with the Sionna library for GPU-accelerated link-level simulations.
- Execute and understand the example tutorials provided by the Sionna library, focusing on key features such as MIMO, OFDM, and Channel Models.
- Comprehend the architecture of the Sionna library, including its various components and how they interact within the library ecosystem.

1.1.2 Design and implementation

- Develop a web user interface using Streamlit to facilitate easy configuration, execution, and monitoring of the physical layer simulations.
- Implement controls within the interface to start and stop the simulation processes as required.

1.1.3 Enhanced Features

- Extend the web user interface to include additional features such as visualizations of the simulation results for enhanced user interaction and data interpretation.

1.2 Outline

This thesis is organized into six main chapters, each focusing on different aspects of developing a web user interface for NVIDIA Sionna based 6G Physical-Layer

Simulation using StreamLit. Below is an overview of each chapter and how they interconnect to form a cohesive study.

Chapter 1: Introduction

- *1.1 Objectives*: Defines the goals and aims of the thesis.
- *1.2 Outline*.

Chapter 2: Background and Theory

- *2.1 Evolution of Communication Systems*: Traces the development of communication technologies leading up to 6G.
- *2.2 6G Wireless Communication*: Introduces the concept of 6G and its potential advancements over previous generations.
- *2.3 Characteristics of 6G*: Looks at specific aspects of 6G, including spectrum bands, AI integration, sustainability, and improved transfer speeds.
- *2.4 Challenges of 6G*: Discusses challenges in developing and implementing 6G technology, including THz propagation loss, 6G infrastructure and signal security concerns.
- *2.5 Role of Simulations in 6G Development*: Discusses the importance of simulations in the development of 6G technologies.
- *2.6 Link-Level Simulation*: Covers the key components and the significance of GPU-accelerated simulations.
- *2.7 The Physical Layer (OSI)*: Explains the role of the physical layer in communication systems.
- *2.8 Modulation Schemes*: Reviews various modulation schemes used in wireless communication.
- *2.9 MIMO*: Details MIMO technologies and their applications in wireless communication.
- *2.10 Multiplexing*: Explores different types of multiplexing techniques, including OFDM.

Chapter 3: Technology Stack

- *3.1 Python*: Discusses the use of Python in the project.
- *3.2 NVIDIA Sionna*: Provides an overview of NVIDIA and the Sionna library.
- *3.3 Streamlit*: Introduces Streamlit as the framework for the web interface.
- *3.4 Tensorflow*: Explains the role of Tensorflow in the project.
- *3.5 Keras*: Covers the use of Keras in the context of the project.
- *3.6 GitHub*: Discusses the use of GitHub for version control.

Chapter 4: Implementation and Demonstration

- *4.1 Interactive QAM Constellation Simulation Interface*: Describes the design and implementation of the QAM interface.
- *4.2 OFDM Uplink Transmission in the Frequency Domain*: Details the development of the OFDM transmission interface.
- *4.3 Web Applications Summary*: Summarizes the two developed web interfaces.

Chapter 5: Experimental Evaluation

- *5.1 Experiment 1*: BER Performance with QPSK Modulation
- *5.2 Experiment 2*: BER Performance with 64-QAM Modulation
- *5.3 Results and Discussion*: Explains the results from the experiments.

Chapter 6: Conclusion

- Engages in a critical analysis of the work completed, discussing the challenges, successes, and implications of the research.

Chapter 2

Background and Theory

2.1 Evolution of Communication Systems

Mobile communication has always been at the forefront of technological innovation, shaping the way society functions and interacts. From the introduction of 1G, which was primarily voice-based, we've transitioned to data-centric technologies where the internet, multimedia, and instant communication have become essential. With each generational shift, technology has striven to address the limitations of its predecessor. The 3rd Generation Partnership Project (3GPP) has been a central entity in the evolution of mobile communication systems. Established in 1998, this collaborative project between various telecommunications standards organizations has set the foundation for mobile telephony standards across the globe. [1] The timeline for the development of communication technologies is as follows [1, 2]:

- **1980's 1G**: The introduction of 1G marked the beginning of mobile communication, primarily focused on voice transmission.
- **1990's 2G**: The era of 2G introduced digital networks, significantly enhancing voice quality and security. It also marked the advent of Short Message Service (SMS), transforming text-based communication.
- **2000's 3G**: With the onset of the 3G era in the early 2000s, 3GPP introduced the Universal Mobile Telecommunications System (UMTS), enhancing mobile data and multimedia services, like video calling, spurred further by the emergence of smartphones, notably the first iPhone in 2007 [3].

- **2010's 4G**: The evolution continued with the introduction of 4G and Long-Term Evolution (LTE) standards by 3GPP, focusing on high-speed data transmission and improved efficiency. High-definition video streaming, online gaming, and seamless internet browsing became a norm.
- **2020's 5G**: The ongoing 5G era extends beyond conventional telephony, focusing on machine-type communication, ultra-reliable low-latency connectivity, and massive Internet of Things (IoT), autonomous vehicles, and smart city infrastructures [4].

2.2 6G Wireless Communication

With the foundation laid by these technologies, the next evolution of mobile communication will set the bar even higher. 6G is still a concept on the horizon, but its potential capabilities are promising.

The world continues to become increasingly interconnected, not only person-to-person, but also machine-to-machine and machine-to-person. This vast network of interconnected entities requires a strong, reliable and high-speed connection.

Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality (MR) are no longer concepts. Major tech companies around the world are already launching products in this category of Extended Reality (ER), such as Meta's Oculus Rift, Apple's Vision Pro VR headsets and Microsoft's HoloLens AR glasses. As XR technologies continues to evolve, we can expect it to be incorporated into the realm of remote work and home offices. Traditional video calls and conference meetings online could soon be supplemented or even replaced by virtual meeting rooms. These technologies requires extremely good connectivity to work as intended, i.e. high bandwith and ultra-low latency.

Moreover, the notion of smart cites, once a futuristic ideal, has now become a main focus within the domain of urban planning [5]. Smart cities are cities where technological advancements are integrated to make traditional networks and services more efficient, more safe and overall make for a better quality of life for its residents. From smarter urban transport networks, to upgraded water supply and waste disposal, lighting and heating of buildings [5]. A robust and reliable communication standard is essential for the transition to smart cities.

Furthermore, in coherence with technological advancements within robotics and manufacturing; businesses around the world have started to accept the inevitable transmission to more automated processes in manufacturing. The industrial automation technology has revolutionized the manufacturing business and is already proving to be more cost-effective, precise and also improves quality of production compared to traditional manual labor[6]. This shift not only optimizes production timelines but also minimizes human error, ensuring consistent product quality and fostering sustainable growth in various industrial sectors [6].

The convergence of these technological innovations underscores the undeniable importance of advancing our communication infrastructures. The blueprint of our future society is relying on the potential of 6G, offering greater speeds, reliability and capabilities that surpass the limitations of 5G. As industries move towards a future of automation, as cities become smarter, and as extended reality becomes more integrated in our daily routines, the need for a next-generation communication system is right around the corner. This transition depends heavily on the research and adoption of 6G.

The convergence of these technological innovations – from extended reality in everyday applications to the emergence of smart cities and the increasing reliance on industrial automation – underscores the critical importance of advancing our communication infrastructures. 6G emerges as a foundational element of our future society, not merely as an incremental improvement over 5G, but as a transformative leap forward. It promises to deliver greater speeds and more reliable connections, which are fundamental for the high-bandwidth and ultra-low latency demands of extended reality technologies. Moreover, the comprehensive connectivity offered by 6G is crucial for the seamless operation of smart city infrastructures. While smart cities today primarily utilize 4G and 5G networks, the integration of 6G could significantly enhance their capabilities and efficiency. In industrial contexts, 6G is essential for the real-time control and high reliability required in increasingly automated environments. As industries move towards a future of automation, as cities become smarter, and as extended reality becomes more integrated in our daily routines, the need for a next-generation communication system is right around the corner. This transition depends heavily on the research and adoption of 6G.

2.3 Characteristics of 6G

The sixth generation of wireless communications is aiming to stretch the capabilities of 5G to a new level. This section will delve into what can be expected in this new technology leap.

2.3.1 Terahertz Spectrum Bands

Wireless communication operates through the radio spectrum, a segment of the electromagnetic spectrum. Theoretically, the 5G spectrum encompasses frequencies up to 100 GHz, yet in practical applications, the usage extends to a maximum of 39 GHz [7]. Current advancements in 6G research are pushing the boundaries, focusing on using frequencies in the range of 100 GHz to 1000 GHz, also known as the Terahertz (THz) spectrum. This exploration positions 6G to revolutionize wireless communication by offering significantly enhanced data rates and reduced latency compared to its predecessors. Figure 2.1 illustrates the spectrum ranges.

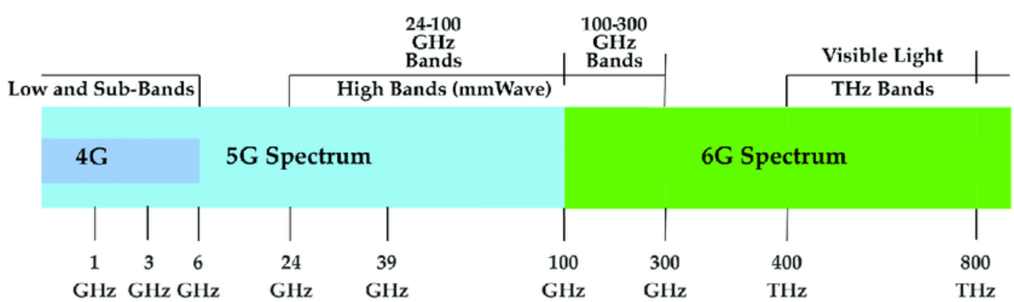


Figure 2.1: Spectrum Bands[8]

2.3.2 AI and 6G URLLC

Ultra Reliable Low Latency Communication (URLLC) was introduced with 5G [9]. This technology is designed to support applications that require highly reliable data transmission with minimal delay such as live video calls, self-driving cars, industrial automation, XR and remote remote surgery. These delay-sensitive applications need a very reliable transmission of data at very low latencies (0.1 milliseconds). 6G is expected to improve URLLC to support latencies of less that

0.1 milliseconds. This will improve the capabilities of the applications mentioned above. Furthermore, AI and ML technologies are set to play an important role in achieving superior efficiency at reduced computational complexity, further optimizing the functionality and responsiveness of these applications.

2.3.3 Green and sustainable communication

A considerable part of the research on 6G also includes concerns regarding the environment [10]. It is widely recognized that the environment is in desperate needs of attention and protection. The advancements in 6G technology is set to reduce our carbon footprints by providing sectors like transportation, manufacturing, agriculture and energy with more efficient communication tools. By leveraging the different characteristics of 6G, industries can optimize their operations which can lead to reduced emissions and reduce energy consumption [10]. The vision for 6G isn't just about faster connectivity, but also about creating a more sustainable and environmentally conscious future.

2.3.4 Improved Transfer Speeds

Utilizing THz spectrum bands, 6G intends to provide much greater throughput and data rates than its predecessor. The transition from 5G to 6G is expected to boost throughput and data rates from 100 Mbps to 1 Gbps and from 20 Gbps to 1 Tbps, respectively [7].

2.4 Challenges of 6G

The development and implementation of 6G face several significant challenges, some of which are discussed in this section.

- **THz Propagation Loss:** When it comes to the Terahertz spectrum utilization of 6G, one of the most prominent challenges is the issue of propagation loss [11]. The THz waves are characterized by significantly higher propagation losses, primarily due to their shorter wavelengths. These shorter wavelengths inherently lead to a decrease in the transmission range, presenting a significant limitation in the effective distance over which these high-frequency signals can be communicated compared to lower frequencies.

- **Updated Infrastructure:** The transition to 6G requires a upgrade of the existing telecommunications infrastructure, a process that includes several key elements. Firstly, it involves the deployment of new antenna types that are able to manage the higher frequencies and increased data rates characteristic of 6G [11]. Additionally, the introduction of advanced transceivers is required, which are sophisticated enough to operate efficiently within the 6G spectrum while managing both high-speed data transmission and reception. Another critical aspect is the implementation of small cells, which are essential for maintaining network coverage and capacity in densely populated areas. These small cells will help overcome the limitations of high-frequency signals, which have higher propagation losses.

- **Enhanced Signal Security:** As we advance into the era of 6G, ensuring the security of the physical layer of telecommunications becomes increasingly crucial. This encompasses safeguarding the transmission and reception of signals against unauthorized interception and tampering. 6G technology is being developed with a strong focus on enhancing communication and interaction experiences for individuals. This human-centric approach prioritizes the needs and experiences of users in the design and functionality of 6G networks, ensuring that these systems are designed to support a wide range of personal and social communication applications [12]. Hence, enhanced signal security in 6G is not just a technical necessity but also a commitment to user-centric privacy and data protection. This involves the development of sophisticated encryption methods, secure communication protocols, and advanced security measures to protect users from potential cyber threats and ensure the integrity and confidentiality of their communications. Incorporating machine learning into the automated security aspects of network virtualization and software-based networks will significantly enhance the detection and prevention of cyber threats within 6G environments [13].

2.5 Role of Simulations in 6G Development

Before real-world implementation of any new technology, simulations serve as a bridge between theoretical research and practical implementations. It provides a virtual environment where one can test and replicate different scenarios.

2.5.1 Performance Analysis

Simulations allows for testing the system performance under different conditions. All though the conditions will never be exactly the same as real-world scenarios, simulations provide a controlled environment to predict and analyse potential outcomes. This allows researches to anticipate how the system behaves under different conditions. For 6G performance analysis, metrics like throughput, latency and Bit Error Rate (BER) can be analyzed.

2.6 Link-Level Simulation

Link-level simulations is often used in the context of wireless communications, specifically focus on modeling and simulating the behavior and performance of individual communication links, such as the link between a transmitter and a receiver [14]. Link-level simulations can help understand the behavior of communication algorithms and systems over communication channels, especially in the presence of noise, interference, and other factors that can influence the performance.

2.6.1 Link-level Simulations: Key Components

I. **Transmitter (Tx):** The transmitter opens the communication by sending data. In simulations, this would involve:

   a. Data source: Could be random or specific data patterns.

   b. Modulation: Converts data into signals suitable for transmission, e.g., Quadrature Amplitude Modulation Quadrature Amplitude Modulation (QAM) etc.

   c. Coding: Error correction codes might be added to protect the data, e.g., Low-Density Parity Check Low-Density Parity Check (LDPC) codes.

II. **Channel:** Represents the medium over which the signal travels from Tx to Rx. Usually the channel involves disturbances. In wireless communications, this could be:

   a. Path loss: Attenuation of signal from the transmitter to the receiver.

   b. Multipath fading: A propagation phenomenon which makes radio signals travel multiple paths to reach the transceiver. This could happen due to atmospheric ducting, reflection, refraction, and diffraction [15].

   c. Doppler effect: This can occur if there is change in motion between Tx and Rx.

   d. Noise: Typically modeled as Additive White Gaussian Noise Additive White Gaussian Noise (AWGN) but could also involve other types of noise.

III. **Receiver (Rx):** Captures the transmitted signal, processes it, and tries to extract the original data. The processing might involve:

   a. Equalization: Compensates for the effects of multipath fading.

   b. Demodulation: Converts the received signal back to data.

   c. Decoding: Error correction decoding to retrieve the original data.

To analyse the performance of a link-level simulation there are some performance metrics to investigate:

I. **Bit Error Rate (BER):** This metric represents the error bits received divided by the total number of bits sent. It gives an idea about the quality of the link and is usually expressed as a percentage [16].

II. **Signal-to-Noise Ratio (SNR):** Represents the ratio of the power of a signal to the power of background noise. Higher SNR usually indicates a better link quality [17].

III. **Throughput:** The rate at which data is successfully transmitted over the link.

IV. **Effects of Different Algorithms:** How different modulation schemes, coding techniques, or equalization methods impact the performance of the communication link.

2.6.2 GPU accelerates simulations

Graphical Processing Unit (GPU) was initially designed to process graphics, which requires excessive computational capabilities. Over time, researchers have recognized the potential of GPUs for general-purpose computing due to its architecture. This has led to the evolution of General-Purpose Graphics Processing Units (GPGPUs). GPGPU computing refers to the use of GPUs for computationally heavy applications beyond graphic rendering, such as Machine Learning (ML), data analysis and simulations [18]. A big portion of 5G and 6G research involves testing different systems, which often require very excessive computational workloads. By utilizing a the computational power of GPUs or GPGPUs, researches can significantly speed up the training process.

2.7 The Physical Layer

This section explains the Physical Layer and how it functions as the foundational aspect of network architecture.

The OSI Model

The Open Systems Interconnection (OSI) Model is a conceptual framework that divides network communication into seven distinct layers [19]. These seven layers can be seen in Figure 2.2. The layers encompasses various functions that collaborate to exchange data between devices across a network. Essentially, the OSI Model serves as a guide to the processes required for end users to connect with each other, even when they are situated on opposite sides of the world.

The Physical Layer

The first and bottom-most layer of the OSI Model is known as the Physical Layer (PHY). It is responsible for transmitting raw bits of data (0's and 1's) across actual physical components, hence the name physical layer. A physical component, or medium, could be copper wires, fiber-optic cables, wireless radio-waves, light pulses from a laser or anything that can be used to establish a connection. The bit representation in the physical layer is important, as it signifies the most fundamental form of data, where each bit represents a binary choice of 0 or 1. The

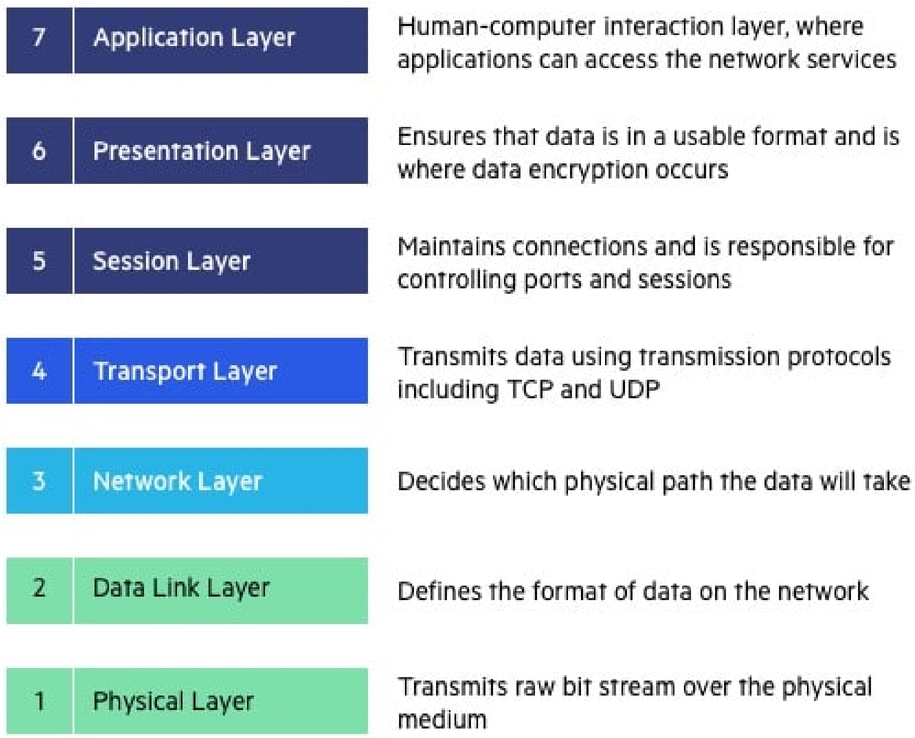


Figure 2.2: The OSI Model[20]

Physical Layer is responsible for converting these digital signals into 'physical' signals that can traverse the selected medium. It does this by establishing rules for which frequencies, signal voltages or modulation schemes should represent 0 and 1 and translate this into waveforms.

The Physical Layer is the foundation for the other layers in the OSI Model. It is responsible for transmitting data recieved from the higher layers and for recovering data before passing it on to the higher layers [21]. A solid physical layer is essential for ensuring the robust functionality and reliability of the entire network. However, physical implementations are often both costly and time-consuming with regards to the acquisition, installation, and maintenance of hardware components, as well as the physical infrastructure required for connectivity.

Simulating the Physical Layer is an effective method for testing and analyzing the network under various conditions and scenarios. It allows for performance evaluations and optimization of the network's efficiency prior to deployment.

2.8 Modulation Schemes

A central part of Information and Communications Technology (ICT) is to transmit information, which often originates from a baseband signal. A baseband signal is a signal that has not been modulated to higher frequencies [22], for instance the frequency of a voice message. Transmitting this baseband signal over a wireless medium is not feasible. This is because this baseband signal might interfere with other signals, similar to the difficulty one might experience in trying to communicate with a friend at a concert. Just as your voice can be drowned out or confused with the surrounding music and crowd noise, the unmodified baseband signal can similarly interfere with or be disrupted by other nearby signals in the communication spectrum. To make the signal more robust and unique, it needs to be modulated onto a carrier signal. Modulation serves several purposes, like making a signal robust. However, mainly its about compressing as much data as possible into the least amount of spectrum possible. In technical terms modulation is about maximizing the spectral efficiency [23]. In addition to efficiently use of the frequency spectrum, avoiding overlapping of other signals, modulation is also needed for transmitting signals over long distances and to improve the overall quality of the transmission. There are several techniques for modulation, often referred to as modulation schemes.

Modulation is accomplished by modifying a carrier signal (or carrier wave) to carry the information in the baseband signal. This modification involves changing the properties of the carrier wave, i.e. its *amplitude*, *frequency* or *phase*, in accordance with the baseband signal. Modulation serves several purposes in communication technology, such as:

- **Antenna sizes** – Traditional 4G/LTE and earlier communication systems primarily relied on large cell towers for data transmission, necessitated by the use of lower frequency signals which require larger antennas [24]. High-frequency modulation, as employed in 5G and potentially in 6G systems, necessitates the use of smaller antennas due to the shorter travel range of higher frequency signals. These smaller antennas offer the advantage of denser deployment, enhancing network capacity and speed through their ability to operate at higher frequencies [25].

- **Signal Integrity** – Modulation enhances signal integrity, making signals more resilient to noise and interference, thereby making transmission more reliable over longer distances, depending on the modulation technique used [26].

- **Bandwidth Utilization** – Higher frequencies allow for a greater number of signal variations within a given time period. This translates to the ability to carry more data, making higher frequency waves suitable for higher data rate transmissions, a key factor in advanced communication systems like 5G and 6G. However, higher frequency signals often have shorter ranges and are more susceptible to attenuation and physical barriers like buildings and trees.

Modulation can be classified into two primary types: Analog Modulation and Digital Modulation. Analog Modulation is used when the information signal being transmitted is of analog form, i.e a continous signal such as voice waveforms. Digital Modulation is used when the information signal is digital, characterized by binary data, typically represented as sequences of 0's and 1's. Both analog and digital modulation make use of an analog carrier waves for transmission, however there are multiple ways of doing so.

2.8.1 Analog Modulation

Analog modulation includes techniques for transmitting audio and visual information by encoding it onto higher frequency waves (carrier waves), that are capable of reaching longer distances. This encoding can be accomplished through three primary methods:

- **Amplitude Modulation (AM)** modifies the height, or *amplitude*, of the carrier wave to match the original information's intensity [27].

- **Frequency Modulation (FM)** Varies the speed, or *frequency*, at which the carrier wave sways, corresponding to the information's amplitude [27].

- **Phase Modulation (PM)** Alters the starting point, or *phase*, of the carrier wave in a way that reflects the information's changing values [27].

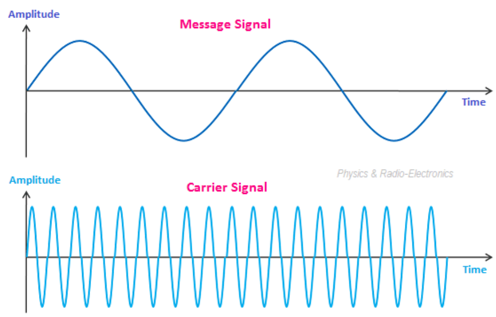


Figure 2.3: Baseband and carrier signal

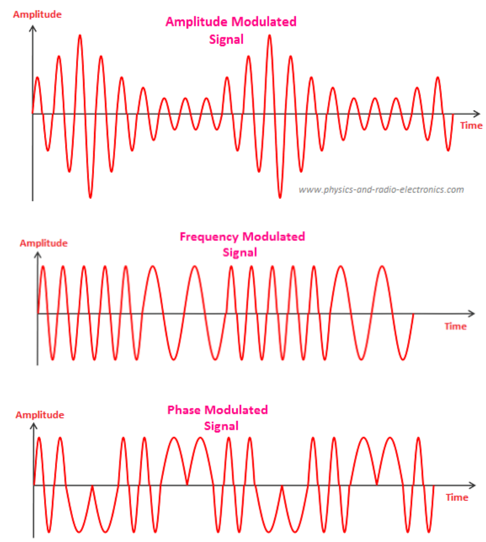


Figure 2.4: Effects of AM, FM and PM [28]

In Figure 2.3 and Figure 2.4, we can observe the effects of the different Analog Modulation techniques.

2.8.2 Digital Modulation

Digital modulation involves the process of encoding binary information signals onto an analog carrier wave. While similar in principle to methods like AM, where the amplitude, frequency, or phase of a carrier wave is modified, digital modulation specifically encodes binary data onto the carrier wave [29]. This technique is often refereed to as *keying*. An examination of the three most common keying techniques follows:

- **Amplitude Shift Keying (ASK)** – ASK is a form of keying that varies the amplitude of the carrier wave to represent binary data, with distinct amplitude levels for the binary digits 0 and 1. In Figure 2.5, the ASK technique uses On and Off (OOF) amplitudes. Alternatively, this keying method could be employed using two distinct amplitude levels, where one level represents a binary '1' (On) and the other represents a binary '0' (Off) [29].

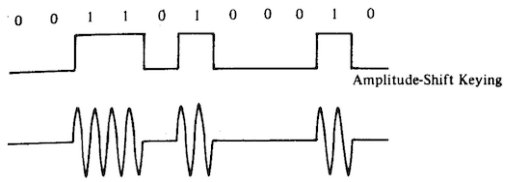


Figure 2.5: ASK Modulation [30]

- **Frequency Shift Keying (FSK)** – FSK changes the frequency of the carrier wave to convey information, assigning specific frequencies to each binary digit. Figure 2.6 shows an example of FSK modulation.

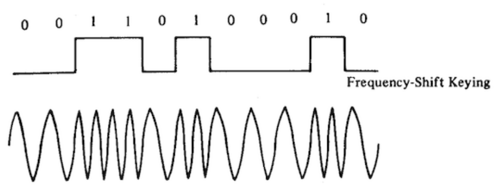


Figure 2.6: FSK Modulation [30]

- **Phase Shift Keying (PSK)** – In PSK, the phase of the carrier wave is varied to represent binary data, with each binary digit having a unique phase value. Figure 2.6 shows an example of FSK modulation.

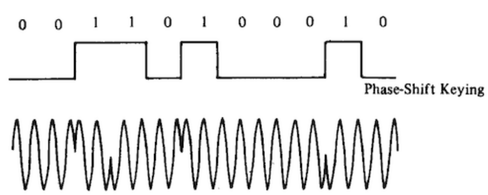


Figure 2.7: PSK Modulation [30]

The following section presents how some of these modulation techniques can be combined to create more sophisticated forms of modulation, which are used in modern communication technologies like 4G, 5G and are expected to be equally significant in the evolution of 6G networks.

2.8.3 Quadrature Amplitude Modulation (QAM)

Quadrature Amplitude Modulation (QAM), also known as IQ modulation, is a sophisticated digital modulation technique that involves both *amplitude* and *phase* variations [31]. QAM allows for efficient data transmission (often digital) by modulating two carrier signals which are phase-shifted by 90°relative to each other, typically represented as a sine (sin) and a cosine (cos) wave [32]. One of the carriers is referred to as the In-phase or "I" signal, while the other is known as the Quadrature or "Q" signal.

The core principle of QAM lies in its capacity to modulate these two signals independently and then combine them to form a single composite signal, as illustrated in Figure 2.8. This composite signal is characterized by its unique *constellation points*, which represent various combinations of amplitude and phase shifts. Each constellation point corresponds to a specific symbol, which encodes a specific combination of bits.

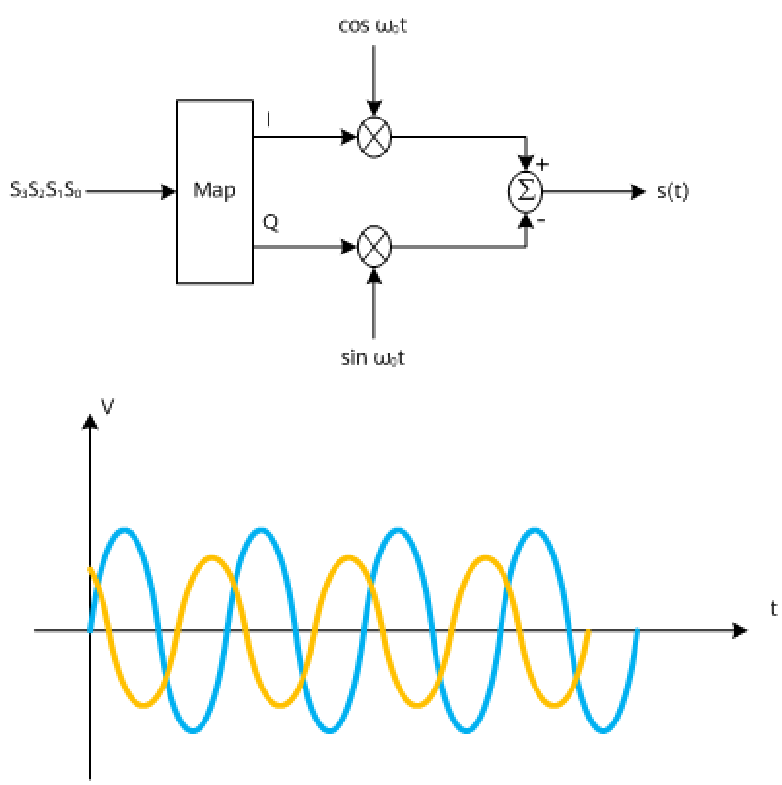


Figure 2.8: QAM modulation and the two carrier waves [31]

The number of bits represented by each symbol depends on the specific QAM order. In QAM, the complexity of the modulation is reflected in the order of QAM, often designated as M-QAM. In Equation 2.1, 'M' represents the total number of unique symbols in the modulation scheme and 'N' represents the number of bits .

$$M = 2^N \tag{2.1}$$

The number of bits encoded by each symbol is calculated using the logarithm, as shown in Equation 2.2:

$$N = \log_2(M) \tag{2.2}$$

As an example, a 16-QAM system includes 16 distinct symbols in its constellation. According to the formula, each symbol encodes $\log_2(16)$, which equals 4 bits per symbol. In a 64-QAM system, the constellation expands to 64 unique symbols, with each symbol encoding $\log_2(64)$, thus representing 6 bits per symbol. The higher the QAM order, the more bits each symbol carries, which enhances the rate of data transmission. This characteristic is particularly advantageous in bandwidth-intensive applications. However, higher QAM orders also have a denser packing of symbols. Consequently, the spacing between the symbols is reduced, leading to a lower tolerance for noise within the system. Essentially, higher QAM order can offer faster data rates and higher level of spectral efficiency, but also requires high Signal-to-Noice ratio (SNR) levels to maintain a reliable connection [33].

Examples of QAM orders

- 4-QAM or Quadrature Phase Shift Keying (QPSK) (2 bits per symbol)
- 16-QAM (4 bits per symbol)
- 64-QAM (6 bits per symbol)
- 256-QAM (8 bits per symbol)
- 1024-QAM (10 bits per symbol)

2.8.4 Constellation Diagrams

Digital radio frequency (RF) signals, such as PSK or QAM signals, can be visualized in a constellation diagram, which is a graphical representation of the complex plane. Furthermore, the constellation diagram shows all possible amplitude and phase combinations that can be used to represent the data [34].

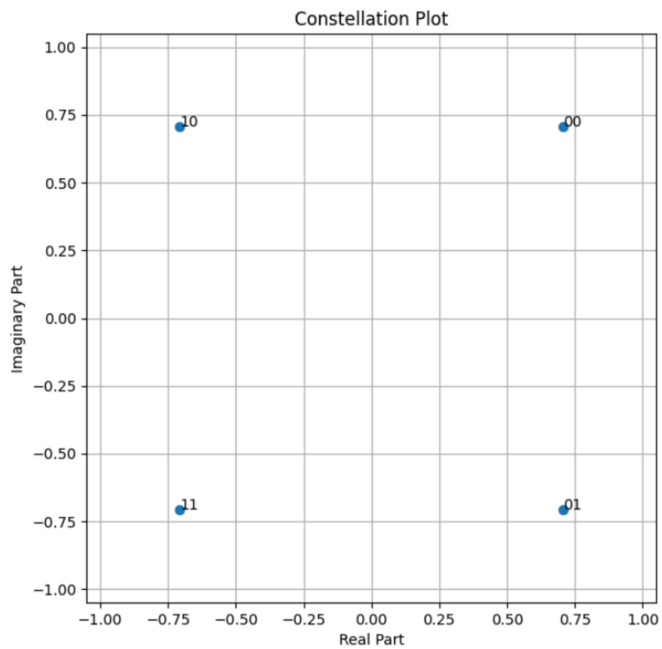


Figure 2.9: 4-QAM (QPSK) Constellation Diagram from web app

In a constellation diagram, such as in Figure 2.9:

- The **horizontal axis** (Real Part) represents the in-phase component of the signal, often denoted as I. This component affects the amplitude of the cosine aspect of the carrier wave.
- The **vertical axis** (Imaginary Part) represents the quadrature component of the signal, often denoted as Q. This component is orthogonal to the in-phase component and affects the amplitude of the sine aspect of the carrier wave.

The points, or dots, in Figure 2.9 are the constellation points discussed earlier. A constellation point represents a unique symbol that encodes a specific number of bits to be transmitted. These points are used as reference points when decoding the transmitted data. In the demodulation process, the received signal is compared against these constellation points to decode the transmitted data. The demodulator determines which constellation point the received signal is closest to and decodes the corresponding bits.

The distances between the constellation points in the modulation phase (before transmission, see Figure 2.9) are important, as they illustrate the signal's resilience to noise and interference. A larger distance between points implies better noise tolerance, as the points can be more easily distinguished from one another when the signal is demodulated. Conversely, points that are closer together can convey more data (since there are more of them within a given area). The following list describes the key aspects of the distances between constellation points:

- **Error Tolerance**: Greater distances between constellation points enhance the system's tolerance to noise and interference, making it easier for the receiver to distinguish between them despite signal distortions.

- **Signal-to-Noise Ratio (SNR)**: The distance between points is related to the SNR. A higher SNR means that the signal strength significantly exceeds the noise level, allowing for clearer distinction of the points.

- **Bit Error Rate (BER)**: The distance between points affects the BER, which measures the rate at which errors occur in transmitted data. Larger distances generally lead to a lower BER. However, there is a trade-off between the error rate and throughput.

- **Throughput vs. Error Rate Trade-off**: Higher-order modulation schemes increase throughput but can also elevate the error rate. The choice of scheme often depends on channel conditions, with higher-order schemes favored in stable conditions for maximum data rate and lower-order schemes in unstable conditions for enhanced reliability.

- **Power Efficiency**: The trade-off between power efficiency and point spacing is significant. Enhancing the transmission power broadens the spacing between these points, effectively reducing the Bit Error Rate (BER). However, power consumption is an important factor in battery-dependent devices such as mobile phones.

- **Adaptive Modulation**: This approach dynamically adjusts point spacing according to channel conditions, altering the modulation scheme as needed. In optimal conditions, a complex scheme with closer points is used, whereas a simpler scheme with points further apart is preferred under challenging conditions. Adaptive modulation effectively balances power efficiency, data throughput, and error performance. Adaptive modulation is compatible with advanced technologies like Multiple Input Multiple Output (MIMO) [35].

2.9 Multiple Input Multiple Output (MIMO)

MIMO is an antenna technology in wireless communications that employs multiple antennas both at the transmitting source and the receiving end [36]. Communication systems, where both the transmitter and receiver are equipped with a single antenna, often face challenges due to signal distortion. Such distortions, arising from fading factors like reflection, refraction, path loss, and ambient noise, can significantly degrade communication quality. In single-antenna systems, such as Single Input Single Output (SISO) systems, these issues not only limit the reliability of the transmission but also constrain the data throughput and coverage.

MIMO, on the other hand, is a technology that fundamentally alters this scenario by employing multiple antennas at both the transmitter and receiver ends, as illustrated in Figure 2.10. The 3rd Generation Partnership Project (3GPP) introduced MIMO technology in the Mobile Broadband Standard with its Release 8 in Q4 of 2008 [37]. This multi-antenna strategy was not merely an incremental improvement but a major leap forward. By using multiple transmission and reception paths, MIMO effectively mitigates the issues associated with signal fading and interference. The result is a significant improvement in the throughput, coverage, and reliability of wireless links.

Furthermore, MIMO technology is an essential part of increasing the capacity of communication channels without requiring additional spectrum. It achieves this through techniques like **spatial multiplexing**, which allows for the simultaneous transmission of multiple data streams over the same frequency band, effectively multiplying the channel's capacity [38]. This makes MIMO an indispensable component of modern wireless communication standards, including LTE, Wi-Fi, the emerging 5G networks, as well as 6G, where the demand for high data rates and reliable connections is ever-growing. Additionally, MIMO systems can utilize **spatial diversity**, a technique that provides multiple paths for the signal, thereby greatly reducing the likelihood of fading effects and improving overall reliability [38]. In this section presents how MIMO technology works and discusses the essential approaches for MIMO systems; spatial multiplexing and spatial diversity.

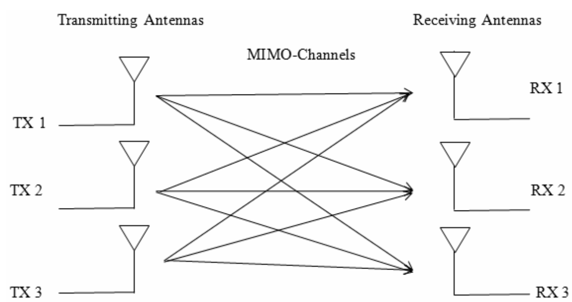


Figure 2.10: MIMO Illustration

2.9.1 Spatial Diversity and Spatial Multiplexing in MIMO

Spatial Diversity

Spatial Diversity is a technique in MIMO systems that reduces signal fading [38]. It involves the use of multiple antennas at the transmitter, the receiver, or both, to provide a diverse set of paths that a radio wave can travel from the transmitter to the receiver.

How Spatial Diversity works The key idea behind spatial diversity is to send copies of the input data (symbols). These copies may be modified (in a mixer) by changing the phase and giving it a different amplitude from an amplifier. But, essentially the same symbol is being sent from each of the transmitter antennas, just with a different amplitude and phase. By sending the same information over multiple paths, the probability that at least one of the paths will provide a good quality signal at the receiver is increased. This redundancy reduces the effects of fading and improves overall signal robustness. This is one of the approaches of using the multiple antennas in a MIMO communication system.

Benefits of Spatial Diversity and Beamforming

The primary benefit of spatial diversity is the significant improvement in signal reliability and a reduction in the Bit Error Rate (BER). It enhances the link's robustness to channel impairments, making wireless communication more reliable over a range of conditions. Spatial diversity is commonly used with beamforming. Beamforming involves adjusting the phase and amplitude of the signals at each transmitter antenna in such a way that the signals constructively interfere at a particular point or direction in space, enhancing the signal power in that direction [39]. Conversely, the signals can be adjusted to destructively interfere, reducing the power of the signal in undesired directions, which helps in minimizing interference to other devices or networks. The illustration in Figure 2.11 displays how beamforming directs a wireless signal towards a targeted direction, as opposed to dispersing the signal over a broad area.

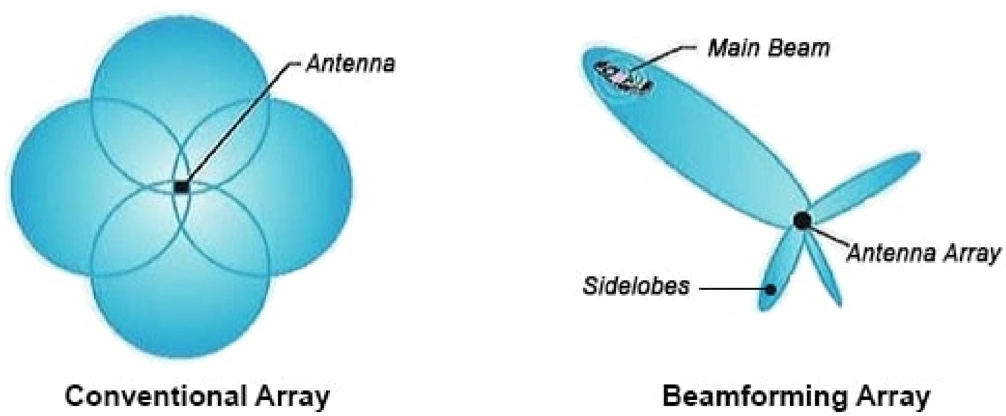


Figure 2.11: Illustration of Beamforming [40]

Spatial Multiplexing

Spatial Multiplexing, another key technique in MIMO systems, is primarily focused on *increasing data rates*. Due to this, the technique is considered the primary reason that 4G/LTE and 5G networks to employ MIMO systems, including Massive-MIMO [38].

How Spatial Multiplexing works In Spatial Multiplexing, the high rate input data sequence is divided into multiple lower rate data sequences. This process is known as multiplexing. Each of these lower rate data sequences is then transmitted simultaneously but independently from separate antennas. This is a crucial aspect of spatial multiplexing – using each antenna to transmit a different stream of data. By transmitting multiple data streams simultaneously over the same frequency channel, but through different antennas, it effectively multiplies the capacity of the channel.

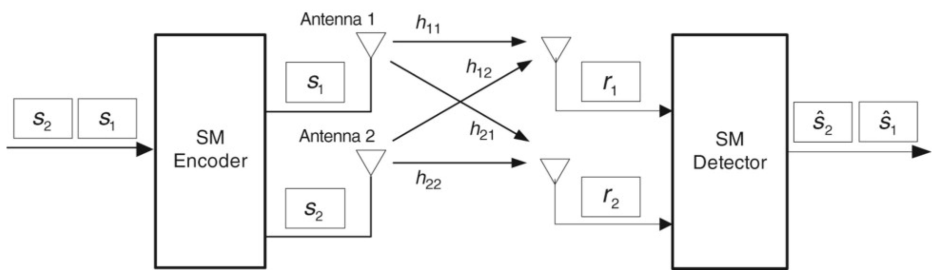


Figure 2.12: Spatial Multiplexing 2.12

The diagram in Figure 2.12 illustrates the concept of spatial multiplexing (SM) in a MIMO system.

1. **SM Encoder**: The process begins with the SM encoder, where the input data stream is divided into two separate data sequences S_1 and S_2. These sequences are lower rate versions of the original data, prepared to be transmitted over different antennas.

2. **Transmission via Antennas**: Each of these sequences is then transmitted through its own antenna. In this diagram, sequence S_1 is sent through Antenna 1, and sequence S_2 is sent through Antenna 2. These antennas transmit the signals simultaneously.

3. **Channel Propagation**: The signals S_1 and S_2 pass through the wireless channel, characterized by the channel coefficients h_{11}, h_{12}, h_{21}, and h_{22}.

   - The coefficient h_{11} represents the channel from Antenna 1 to Receiver 1, h_{12} from Antenna 1 to Receiver 2, h_{21} from Antenna 2 to Receiver 1 and h_{22} from Antenna 2 to Receiver 2.

4. **Reception of Signals**: Upon arrival at the receiver's end, each antenna receives a composite signal that includes contributions from both transmitted sequences. These signals, denoted as r_1 and r_2, contain the mixed information that needs to be decoded.

5. **SM Detector**: At the receiver end, a SM detector processes the received signals. The detector's function is to separate the combined signals back into the original streams $\hat{S}_1$ and $\hat{S}_2$ (where the hat notation indicates these are estimates of the transmitted signals). This separation is based on knowledge (or estimates) of the channel coefficients and the use of signal processing algorithms.

Benefits: The most significant advantage of spatial multiplexing is its ability to increase the data throughput linearly with the number of antennas, without needing extra bandwidth or increased transmit power. This makes it an efficient method for high data rate transmission, especially in environments with rich scattering.

Requirements and limitations: The performance of spatial multiplexing is highly dependent on the channel conditions. Spatial multiplexing thrives in environments with rich scattering. Such conditions ensure that the multiple transmitted signals can take different paths to the receiver, which helps in separating the signals at the receiver end due to their distinct spatial signatures. The scattering objects in the environment, such as buildings, vehicles, and other structures, provide multiple paths for the signals, which leads to a rich scattering scenario. For spatial multiplexing to be effective, each transmitted signal should optimally have a distinct path. This allows the receiver to distinguish between each of the incoming signals based on the differences in how they have been affected by the channel. If the signal paths are highly correlated, which means the signals experience similar fading and delay as they travel from the transmitter to the receiver, it becomes much harder for the receiver to separate the signals. This is because the signals would appear similar at the receiver, making it challenging to distinguish one from the other. In cases where there is low scattering and high correlation between signal paths, the effectiveness of spatial multiplexing diminishes. The system may then fail to achieve the expected increase in capacity that would normally be possible under ideal scattering conditions.

2.9.2 Massive MIMO

Massive MIMO is an advanced wireless communication technology that significantly expands upon the concepts of traditional MIMO systems. Massive MIMO involves equipping wireless base stations with a large number of antennas, typically in the order of tens or even hundreds, hence the naming ”massive” [41].

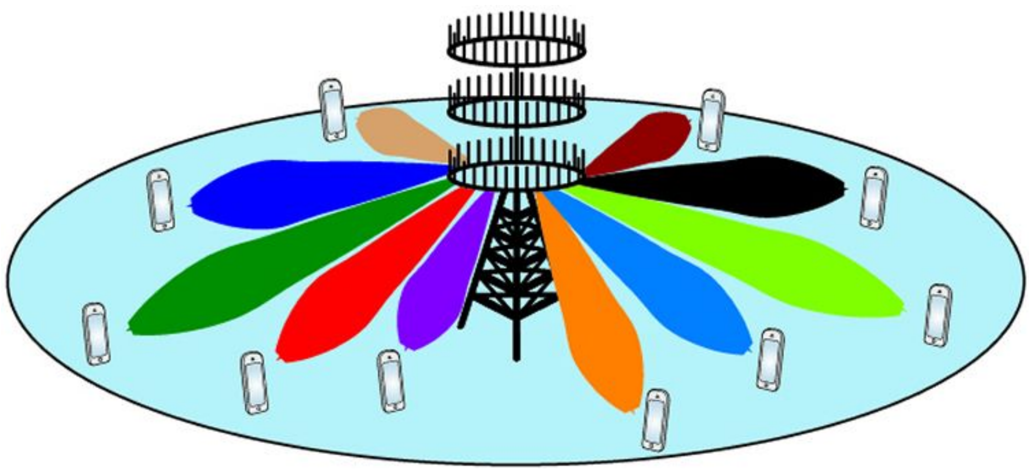


Figure 2.13: Illustration of Massive MIMO [41]

The illustration in Figure 2.13 displays the concept of the Massive MIMO system. At the center, there’s a cell tower equipped with a large array of antennas. This array is indicative of the ”massive” aspect of MIMO, showing that the tower is designed to handle numerous simultaneous connections through spatial multiplexing. The various colored beams fanning out from the tower represent different signal paths. Each color represents a separate beamformed signal directed toward a specific user or sector.

Beamforming in Massive MIMO

Beamforming operates jointly with Massive MIMO to improve network throughput and overall capacity. By utilizing the magnitude of antenna arrays in Massive MIMO to concentrate the direction of signals, beamforming minimizes interference among beams aimed in different directions. Massive MIMO's extensive antenna setup also facilitates 3D beamforming. 3D beamforming concentrate signals in both horizontal and vertical orientations to users, thus optimizing data transfer rates and network capacity, a particularly advantageous feature in densely populated urban regions with high-rise structures [42].

2.9.3 Other types of MIMO technology systems

While conventional MIMO and Massive MIMO are well-established in enhancing wireless communication systems, there are several other MIMO configurations and technologies that have been developed to address specific challenges and requirements. This subsection explores some of these varied MIMO technology systems.

Multi-User MIMO (MU-MIMO): MU-MIMO extends the concept of MIMO to allow multiple users to access the same network simultaneously. Unlike single-user MIMO (SU-MIMO), where all the antennas are dedicated to a single user, MU-MIMO divides the antenna resources among multiple users. This division is done in both the downlink and uplink, allowing for more efficient spectrum use and improving user throughput in multi-user environments [43].

Cooperative MIMO (Co-MIMO) and Network MIMO: Co-MIMO, also known as Network MIMO, involves multiple base stations coordinating to serve users simultaneously. By sharing information and resources, these base stations can create a virtual MIMO array that extends over a larger geographic area, providing improved coverage and capacity, especially at the cell edges where users typically experience lower signal quality [44].

MIMO-OFDM: MIMO-OFDM systems have become the standard in modern wireless communications such as 5G NR, LTE, and WLAN, primarily due to their exceptional ability to handle frequency-selective channels and enable high data rates. The robust combination of MIMO's multiple antenna elements and OFDM's efficient use of subcarriers allows these systems to meet the ever-growing demands for faster and more reliable data transmission [45].

As the need for higher data rates continues to escalate, MIMO-OFDM systems are evolving to become more complex and expansive. This evolution is characterized by an increase in the number of antenna elements used in MIMO configurations, enhancing the system's capacity to transmit and receive more data simultaneously. Additionally, there's a greater allocation of resources in terms of subcarriers in OFDM, which allows for more efficient use of the available spectrum, making these systems integral to the current and future landscape of wireless technology.

2.9.4 Other antenna configurations in Wireless Communication

In wireless communication systems, the configuration of antennas plays a crucial role in determining the efficiency, capacity, and performance of the communication link. This section explores different antenna technologies, from simple single-antenna systems to complex multiple-antenna systems, and their implications for wireless communication.

Single Input Single Output (SISO)

SISO represents the most basic antenna configuration, where both the transmitter and receiver are equipped with a single antenna. The simplicity of SISO makes it easy to implement but limits the system's ability to exploit multipath propagation for improved performance and does not offer any diversity or multiplexing gains.

Single Input Multiple Output (SIMO)

In SIMO configurations, the transmitter uses a single antenna while the receiver employs multiple antennas. This setup allows the receiver to capitalize on spatial diversity by combining the multiple versions of the signal received via different paths to enhance signal reliability and reduce the likelihood of dropouts.

Multiple Input Single Output (MISO)

Conversely, MISO systems use multiple antennas at the transmitter and a single antenna at the receiver. MISO can be used for transmit diversity and beamforming, directing the signal strength towards the receiver to improve link reliability and potentially enhance signal reach.

Advanced Developments

Emerging technologies like adaptive antenna systems and smart antennas are pushing the boundaries of what is possible with antenna configurations. By dynamically adjusting their parameters, these systems can react in real-time to changing channel conditions, user movement, and varying traffic patterns to optimize the communication link [46].

Each antenna technology offers unique benefits and trade-offs, and the choice of configuration depends on the specific requirements of the communication system, such as range, data rate, and robustness to interference. As wireless communication demands continue to grow, the evolution of antenna technologies remains a critical area of research and development.

2.10 Multiplexing

Multiplexing is the technique of transmitting multiple signals simultaneously over the same transmission medium (cable, optical fiber, wireless etc.) [47]. This approach increases the efficiency of communication systems, allowing for simultaneous transmission of various data types, such as voice, video, and text, without the need for additional physical infrastructure. Inherently, multiplexing optimizes the utilization of the available transmission capacity. It employs a device known as a Multiplexer (MUX) at the transmitter side to combine multiple input signals into a single line and a Demultiplexer (DEMUX) at the receiver side to separate the combined signal back into its original components, as illustrated in Figure 2.14.

Multiplexing has been and still is, one of the most important concepts regarding the evolution of communication technologies. Initially, its application was seen in early telegraph and telephone systems in the 1870's, where it helped to maximize the use of existing wiring [47].

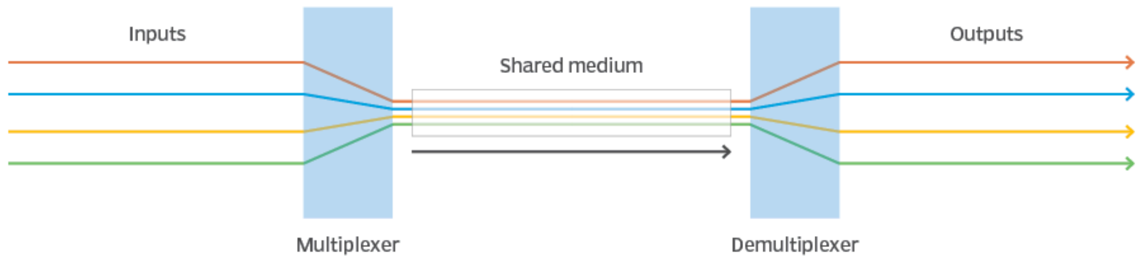


Figure 2.14: Multiplexer, Shared medium, Demultiplexer [47]

2.10.1 Types of Multiplexing

In the early 20th century, multiplexing primarily involved simple techniques like Frequency Division Multiplexing (FDM), which is often used in radio and television broadcasting [48]. This technique allocates different frequency bands to different channels, allowing multiple signals to travel simultaneously through the same medium. Further technology advancements in the 20th century led to Time Division Multiplexing (TDM), in which different signals are transmitted in separate time slots.

When 2G and 3G technologies came along in the 2000's, Code Division Multiple Access (CDMA) was introduced. CDMA is a form of multiplexing where different codes are used for different channels, increasing the capacity of wireless networks.

As we transition from 4G to 5G and look towards 6G, multiplexing continues to be a cornerstone technology. Advanced multiplexing techniques, such as Orthogonal Frequency Division Multiplexing (OFDM) and Massive MIMO, are at the forefront of this transition. They are not only crucial for handling the ever-increasing data traffic but also for enabling the diverse magnitude of services and applications expected in the next generation of wireless networks.

Frequency-division Multiplexing (FDM)

This technique divides the total bandwidth available into a series of non-overlapping frequency bands, each used for a separate channel [49]. Each individual signal is modulated onto a different carrier frequency; these carrier frequencies are then combined into a single composite signal for transmission. FDM is commonly used in radio and TV broadcasting, as well as telephone systems.

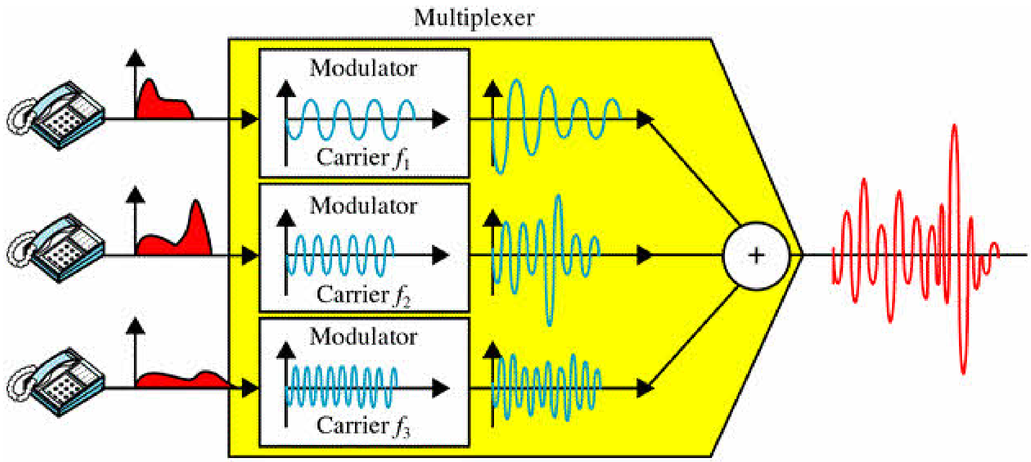


Figure 2.15: FDM [49]

Time-division Multiplexing (TDM)

TDM is an efficient method for transmitting multiple data streams over a single communication channel by allocating distinct time intervals, or slots, for each data stream. This technique is versatile, applicable to both digital and analog signals, and plays a crucial role in optimizing the use of channel capacity in various communication systems. In TDM, the time available on a communication channel is segmented into frames, and each frame is further divided into a set of time slots [49]. Each user or data stream is assigned a specific time slot within the frame. During its allotted time slot, a user can transmit its data. The data from all users are then interleaved and sent sequentially over the channel in this structured format.

Space-division Multiplexing (SDM)

Space Division Multiplexing (SDM) enhances the capacity of wireless communication systems by utilizing the spatial separation of users [50]. This method employs multiple antennas at both the transmitter and receiver side to establish distinct communication channels. These channels operate independently, enabling simultaneous data transmissions by multiple users within the same fre-

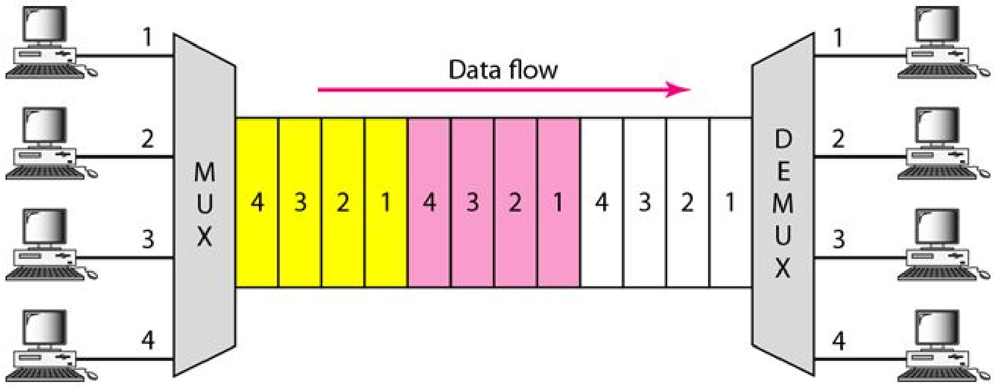


Figure 2.16: Time-division Multiplexing [49]

quency band, without interference. Adding more antennas further increases the system's capacity by creating additional independent channels.

SDM is a common technology in various wireless communication systems, including cellular networks, Wi-Fi, and satellite communications. In the context of cellular networks, SDM finds application in the form of MIMO technology.

2.10.2 Orthogonal Frequency-Division Multiplexing (OFDM)

Orthogonal Frequency-Division Multiplexing (OFDM) is the radio technology behind the great bandwidth capabilities of Long-Term Evolution (LTE) [51]. Its core principle involves dividing a data stream into multiple smaller sub-streams that are transmitted simultaneously over different frequencies. The overlapping, equally spaced complex sinusoids seen in Figure 2.17 are the sub-carriers. Furthermore, the 5 peaking sub-carriers illustrates the orthogonal part of OFDM. At the center of each sub-carrier, the signal contribution from the other carriers is 0. In essence, when a sub-carrier signal is at its peak, all others are null, which allows for more efficient data transfer.

Orthogonality in OFDM

Orthogonality refers to the property where the sub-carriers in an OFDM system are mathematically arranged so that they are perfectly out of phase with each other. This means that at the peak of one sub-carrier, all other sub-carriers have

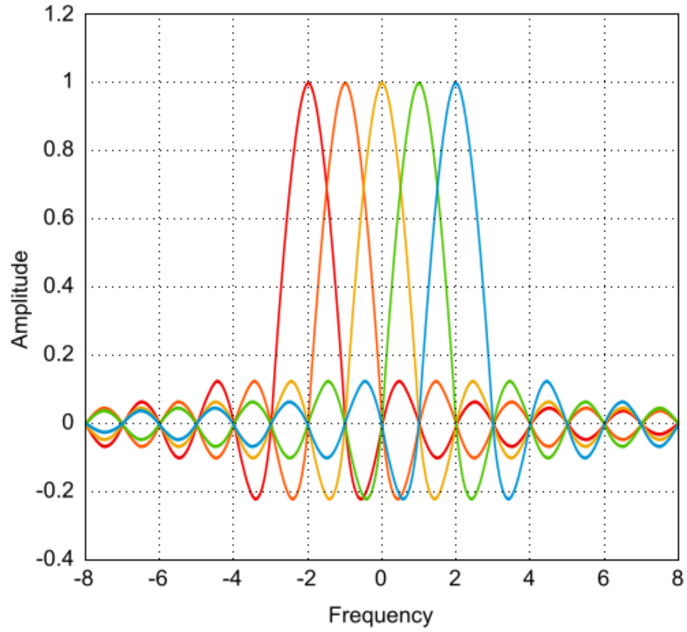


Figure 2.17: The orthogonal property of OFDM in a single channel [51]

a value of zero (null) at that particular point in time and frequency.

Elimination of Interference

In a non-orthogonal system, overlapping carriers can interfere with each other, a phenomenon known as Inter-Carrier Interference (ICI) [52]. This interference distorts the signal and reduces the accuracy of the data received. As we have seen in Figure 2.17, because of the orthogonality, the ICI is eliminated. This allows each sub-carrier to be packed closely together without fear of interference, maximizing the use of the available spectrum.

Increased Spectral Efficiency

By ensuring that sub-carriers do not interfere with each other, OFDM can utilize the available bandwidth more efficiently. More sub-carriers can be packed into a given bandwidth, each carrying its own stream of data. This density of sub-carriers translates to a higher data transfer rate within the same spectral space compared to non-orthogonal methods.

Efficient Transmitter and Receiver Design

The orthogonality also simplifies the transmitter and receiver design. Since the sub-carriers do not interfere, the transmitter and receiver can easily isolate and encode/decode each sub-carrier's signal independently using the Inverse Fast Fourier Transform (IFFT) at the transmitter side and Fast Fourier Transform (FFT) at the reciever side [52].

Chapter 3

Technology Stack

In this chapter, we take a look at the technologies that was used in the development of the web applications. These technologies include advanced machine learning libraries, simulation tools, and efficient web development frameworks.

3.1 Python

Python, a high-level programming language, was the preferred choice for this thesis due to its integration with TensorFlow, Keras, NVIDIA Sionna, NumPy, and Matplotlib. Its versatility enabled the effective implementation of simulations and user interfaces. The development of the web application was facilitated using Streamlit, while NVIDIA Sionna was utilized for specific simulations. Additionally, Python's NumPy and Matplotlib libraries further improved the data processing and visualization aspects.

3.2 NVIDIA Sionna

3.2.1 NVIDIA

Founded in 1993, NVIDIA has long been a leader in the graphics processing industry. Their pioneering efforts began with the launch of the world's first Graphics Processing Unit (GPU), the GeForce 256, in 1999 [53]. This groundbreaking development significantly enhanced the capabilities of computer graphics, enabling video game developers and animators to create more realistic and dynamic animations than before.

Over time, NVIDIA's focus has expanded beyond the realm of gaming. The exceptional computational power of their GPUs has positioned NVIDIA as a key player in the field of artificial intelligence (AI) hardware and software. This expansion has significantly influenced the evolution of communication systems, offering powerful tools like NVIDIA's Sionna library.

3.2.2 Sionna library

The Sionna library is an advanced, GPU-accelerated open-source library developed for simulating link-level simulation of communication systems. Sionna enables fast prototyping of complex communication system frameworks and supports the integration of machine learning in 6G signal processing [54]. In this project, Sionna is used to simulate diverse physical-layer conditions reflective of real-world scenarios. Its integration into the StreamLit-based web interfaces developed allows for interactive configuration, monitoring, and analysis of these simulations. This thesis aims to showcase the utilization of Sionna's advanced channel modeling capabilities. It will demonstrate how these capabilities can be effectively applied to simulate next-generation wireless communication systems with a high degree of accuracy and efficiency. Sionna's state-of-the-art simulation capabilities are not just central for this thesis, but also signify a broader impact on 6G research and development. Its ability to model advanced communication environments also drives understanding of 6G potentials and challenges.

3.3 Streamlit

Streamlit is an open source Python library used for creating web applications for machine learning and data science [55]. The library allows developers to create web applications using Python code. Streamlit also supports various data formats and visualization libraries such as Matplotlib, Pandas, TensorFlow, Keras and NumPy. Given its ability to efficiently render data visualizations and Python compatibility makes it a great choice for NVIDIA Sionna simulations.

3.4 Tensorflow

Tensorflow is an open source machine learning library developed by Google. It provides a flexible platform for building and deploying machine learning models. Tensorflow was initially developed for extensive numerical operations, which made it very useful for deep learning projects. Hence, Google decided to make it open-source for the community to use [56].

Tensorflow enables developers to create dataflow graphs, where each node in the graph represent a mathematical operation and each connection is a tensor. A tensor is mathematical object that consits of a multi-dimentional relationship between other objects, like scalars, vectors or higher dimensional arrays [56].

One of the key features of Sionna's integration with TensorFlow is the use of differentiable layers. These layers enable the backpropagation of gradients through an entire system when encorporating neural networks into a communication system design.

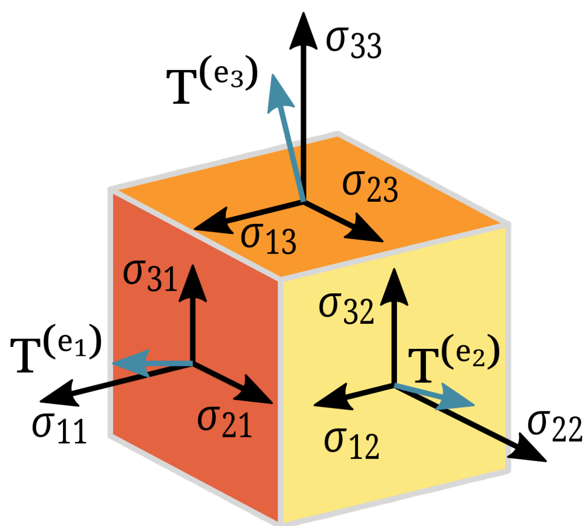


Figure 3.1: Tensor[57]

- 0-D Tensor: Scalar (a single number)
- 1-D Tensor: Vector (an array of numbers)
- 2-D Tensor: Matrix (a 2D array of numbers)

- 3-D Tensor and above: Higher-dimensional arrays

Tensors are fundamental to many fields, especially in physics and engineering, as they can represent a wide range of properties. In context of deep learning and TensorFlow, a tensor is a multi-dimensional data-array.

3.5 Keras

Keras is a deep learning API which runs on top of TensorFlow. It offers fundamental tools and components for creating and deploying machine learning solutions rapidly. It allows for designing neural networks by connecting building blocks of layers, loss functions and optimizers [58].

3.6 GitHub

GitHub is a widely-used platform for version control and collaborative software development [59]. GitHub provides a web-based interface that facilitates efficient management of projects, particularly in coding and programming. In this thesis, GitHub was used to enhance the code management, and facilitates the sharing and documentation of the web application.

Chapter 4

Implementation and Demonstration

In the pursuit of advancing the understanding and accessibility of complex communication systems, this chapter presents two web-based interactive interfaces designed to simulate the physical layer of 6G networks. These applications serves as educational and research tools that allows users to configure and visualize various aspects of physical layer simulations with an emphasis on modulations schemes and signal processing. The web applications are QAM Constellation Simulator in Streamlit and OFDM Uplink Transmission in the Frequency Domain.

4.1 Interactive QAM Constellation Simulation Interface in Streamlit

This interface is for visualizing and understanding the impact of noise on digital signals and the performance of different modulation schemes. It demonstrates the fundamental concepts of signal constellation mapping, the introduction of noise in the communication channel, and the demapping process that attempts to recover the original bits from the noisy received signal.

The foundational concepts and examples outlined in "Sionna's Tutorial Part 1 [60]" have been instrumental in guiding the design and functionality of the simulation platform. The tutorial's thorough approach to explaining the basics of a communication system's physical layer, including the creation of a QAM constellation, has been adapted to create a more interactive and user-friendly experience. By leveraging Streamlit's capabilities, the theory has been transformed into an engaging application that allows users to dynamically explore and manipulate the QAM constellation model.

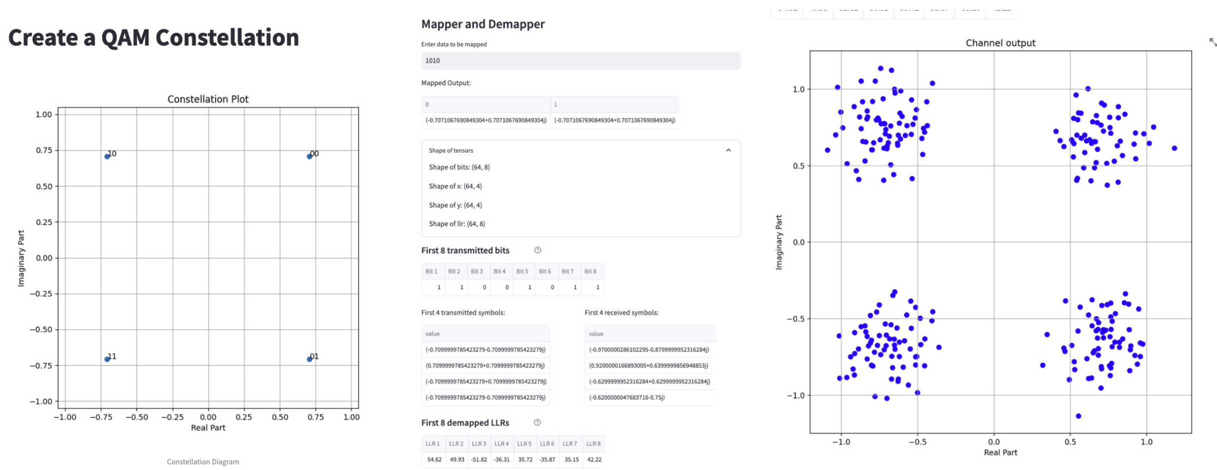


Figure 4.1: Constellation Diagram app

Features

- Modulation Scheme Configurations: Select different modulation schemes to observe how they affect signal transmission.
- Eb/No Configurations: Dynamically change the 'Energy per Bit to Noise Power Spectral Density Ratio' to simulate different Signal to Noise scenarios.
- Adjustable parameters like batch size, block length and number of samples to display.
- Real-time Constellation Diagram: Visualize the modulation constellation in real-time, reflecting the adjustments in simulation parameters.
- Data mapping and demapping: Observe how the data is mapped to the constellation points, and how it is demapped back to the original data.

- AWGN channel: Explore the effects of the AWGN channel on the transmitted signal.
- Tensor Shape Visualization: Inspect the effects of shape of the tensors involved in the simulation.

4.1.1 Layout Description of the Constellation Diagram Application

The constellation diagram application, as shown in Figure 4.1, presents a user-friendly and intuitive interface, designed to facilitate the visualization and analysis of communication signals. It includes interactive elements for parameter configuration, such as modulation scheme selection and signal-to-noise ratio adjustments, which dynamically influence the displayed constellation diagram. Additionally, the application provides real-time visual feedback through the diagram, enabling users to immediately observe the effects of their parameter adjustments. This design not only aids in the practical exploration of signal processing theories but also serves as an educational tool for demonstrating the intricacies of digital signal modulation in communication systems.

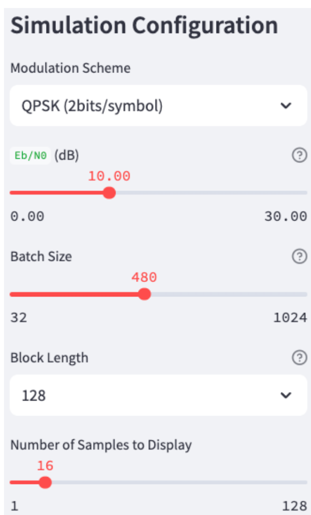


Figure 4.2: Simulation Configuration

Simulation Configuration

This subsection explains the configuration layout for the simulation as shown in Figure 4.2, where the user can customize and control the simulation parameters. It provides an interface for the user to adjust various settings that dictate the behavior of the simulation, such as the number of bits in each block of data and the specific modulation scheme to be employed. The adjustments made by the user are reflected instantly, enabling real-time exploration of different simulation scenarios and immediate visual feedback.

- **Modulation Scheme Dropdown**: As shown in Figure 4.3, the dropdown menu allows users to select a modulation scheme (e.g., QPSK, 16-QAM, etc.), which determines how data is encoded onto carrier waves.

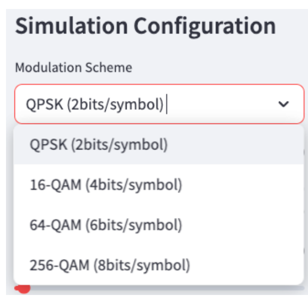


Figure 4.3: Modulation mapping

- **Eb/No Slider**: Represents the signal-to-noise ratio (Energy per bit to Noise power spectral density ratio) in dB. This slider allows users to adjust the ratio, which is a critical parameter in assessing the performance of a communication system.
- **Batch Size Slider**: Determines the number of transmissions (or batches of data) that are processed simultaneously. This can be adjusted for performance trade-offs between speed and accuracy.
- **Block Length Dropdown**: Sets the size of each data block.
- **Information Tooltip (? icon)**: Provides additional information or context for each parameter when clicked or hovered over.

Data Mapping and Bit Transmission Details

This subsection provides an in-depth explanation of 'Mapper and Demapper' functionality within the web application in Figure 4.4.

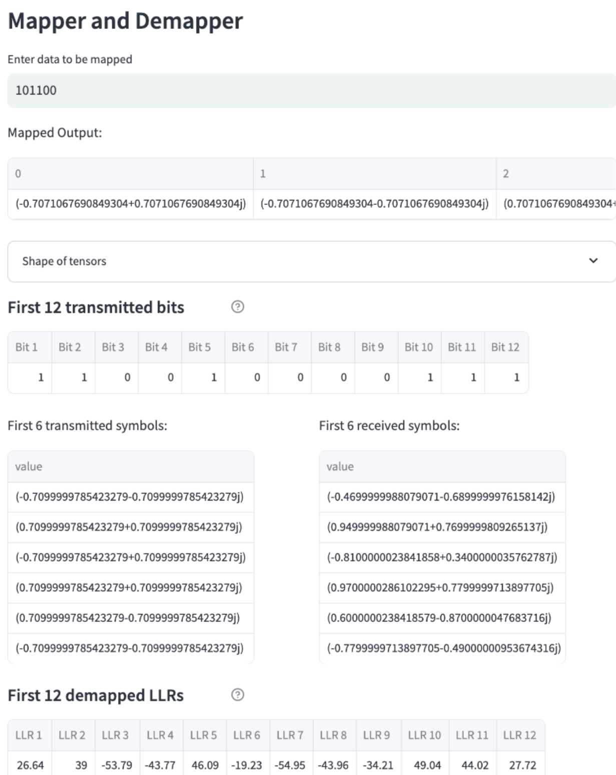


Figure 4.4: Constellation information

- **User Input for Mapping**: A user inputs a binary sequence, in this case, ”101100”.
- **Mapped Output**: The binary input data is mapped to complex symbols, which are then used for transmission. Here, two complex numbers are shown, which correspond to the binary input ”101100”. These complex numbers are the positions on the QAM constellation diagram, where the real part is represented on the x-axis and the imaginary part on the y-axis.
- **Shape of Tensors**: An expandable section provides information on the shapes of the data structures (tensors) used in the simulation.

- **First `X` Transmitted Bits**: A table shows the first `x` bits that are transmitted. In this case, there are 12 bits: "1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1".

- **First `X` Transmitted Symbols**: This part shows the complex numbers (symbols) that the bits are mapped to before being transmitted. These are the ideal symbol locations on the constellation diagram before any channel effects like noise are applied.

- **First `X` Received Symbols**: After transmission through the channel (with noise and other impairments), the received symbols are slightly different from the transmitted ones. This section shows the complex numbers received, which are used to evaluate how the channel has affected the signal.

- **First `X` Demapped LLRs**: The Log-Likelihood Ratios (LLRs) for the first X demapped bits are displayed. The LLR values indicate the confidence in the bit decisions; higher magnitudes (positive or negative) suggest higher confidence. Positive values suggest a higher likelihood of the bit being '1', and negative values suggest a higher likelihood of the bit being '0'. These LLR values is used by the decoder to decide on the most likely transmitted bits.

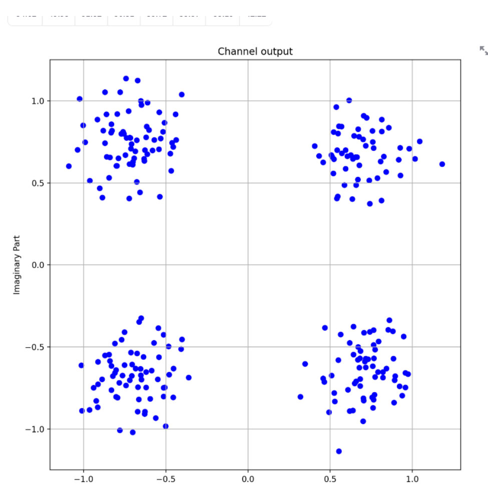


Figure 4.5: Constellation output

Channel output

The diagram in Figure 4.5 displays the received symbols as they appear after being passed through a communication channel.

- **Real Part**: This axis represents the in-phase component of the complex signal.
- **Imaginary Part**: This axis represents the quadrature component of the complex signal.
- **Blue Dots**: Each dot represents a symbol that has been received. The position of the dot corresponds to its value in the complex plane.

The clustering of the dots into distinct groups indicates the different symbol locations that were originally intended in the transmitted signal. However, the spread or dispersion of the dots within each cluster shows the effect of noise and other channel impairments, such as fading or interference, which cause the symbols to deviate from their original positions.

4.1.2 Code Structure and Flow

- **Library Imports**: The code begins by appending the system path to ensure that the custom imports module is accessible.
- **Modulation Mapping**: A dictionary `modulation_mapping` was created to map the number of bits per symbol to the corresponding modulation scheme's name. This way, the user can select between modulation schemes. `NUM_BITS_PER_SYMBOL` is selected by the user via the dropdown menu in the sidebar, as shown in Figure 4.3, which dictates the modulation scheme. The code for this process is shown in Code Listing 4.1.

```
# Mapping of bits per symbol to modulation names
modulation_mapping = {
    2: "QPSK (2bits/symbol)",
    4: "16-QAM (4bits/symbol)",
    6: "64-QAM (6bits/symbol)",
    8: "256-QAM (8bits/symbol)"

NUM_BITS_PER_SYMBOL = st.sidebar.selectbox(
```

```
        'Modulation Scheme',
        options=list(modulation_mapping.keys()),
        format_func=lambda x: modulation_mapping[x]  # Displaying
                                                    the modulation name in
                                                     the select box
    )
}
```

Code Listing 4.1: Modulation mapping process

- **Streamlit UI Components**: 'st.title()' sets the title of the app displayed on the page. 'st.sidebar.title()' sets the title of the sidebar where configuration settings are input.

- **Constellation Object**: An instance of the `Constellation` class from the `sn.mapping` module of the Sionna library is created, as show in Code Listing 4.2.

```
constellation = sn.mapping.Constellation("qam",
                                         NUM_BITS_PER_SYMBOL)
```

Code Listing 4.2: Instance of Constellation class

- **Constellation Plot**: A buffer `buf` is created to temporarily hold the constellation plot. The plot is generated using the `constellation.show()` method and saved into `buf`.

```
# BytesIO buffer to capture the plot
buf = io.BytesIO()
constellation.show()
plt.savefig(buf, format='png')  # Save the plot to the buffer
buf.seek(0)  # Rewind the buffer to the beginning
```

Code Listing 4.3: Capturing constellation diagram

`BytesIO` is a class from Python's `io` module. An object of this class acts as a file-like buffer that stores data in memory instead of writing it to disk. Here, `buf` is created as a BytesIO object which will be used to temporarily hold the binary data for the plot. An `st.image()` function then renders the plot in the Streamlit interface with the caption 'Constellation Diagram'. Instead of saving the generated plot to a file, it is saved directly into the BytesIO buffer `buf`. The `plt.savefig` function is part of matplotlib and is used to save figures. Here, the figure is saved as a PNG image into the in-memory buffer. The format='png' argument specifies the format of the saved image.

- **Mapper and Demapper Initialization**: Mapper and demapper objects are initialized, which are responsible for converting between bits and symbols according to the constellation scheme.
- **User Input for Mapping**: Binary data to be mapped is collected from the user via `st.text_input()`. The data is then processed and checked to ensure it's compatible with the number of bits per symbol for the chosen modulation scheme, as shown in Code Listing 4.4

```
if binary_data.shape[1] % constellation.num_bits_per_symbol == 0:
    mapped_output = mapper(binary_data)

    # Convert the output tensor to a numpy array for display
    mapped_output_array = mapped_output.numpy()

    st.write("Mapped Output:", mapped_output_array)
else:
    st.write("Input length is not a multiple of the number of bits
                                            per symbol")
    st.stop()
```

Code Listing 4.4: Check input length and map data

The first line in Code Listing 4.4 checks if the length of the binary data `binary_data.shape[1]` is appropriate for the selected modulation scheme `constellation.num_bits_per_symbol`. The % operator is the modulo operation which returns the remainder of the division of the two numbers. If the remainder is 0, it means that the binary data length is an exact multiple of the number of bits per symbol required by the modulation scheme, which is necessary for correct mapping.

If the input length is correct, the binary data is passed to the mapper function. The mapper converts the series of binary digits into complex symbol values according to the rules of the modulation scheme, which will then be represented on a constellation diagram.

After the binary data is mapped to symbols, the result is a tensor. Since tensors are not as straightforward to display, the tensor is converted into a NumPy array. The `.numpy()` method is called on the `mapped_output` tensor to make this conversion.

If the input length is not correct (i.e., not a multiple of the number of bits per symbol), the else block executes. It displays an error message informing the user that the input length is invalid, and then `st.stop()` is called to halt the execution of the app to prevent any further processing or displaying of incorrect data.

- **Sionna Components for Simulation**:
  - `binary_source`: This component is responsible for generating random binary sequences. The sequences are created using a uniform distribution, meaning each bit (0 or 1) has an equal probability of occurrence. Such sequences are essential for simulating the input data in communication systems.
  - `awgn_channel`: Stands for Additive White Gaussian Noise channel. It simulates a common type of noise in communication channels, characterized by its statistical properties — it has a mean of zero (hence 'white') and a normal (Gaussian) distribution. This component is needed for testing how well a communication system can perform under typical noise conditions.
  - `mapper` and `demapper`: These components are used for the modulation and demodulation processes, respectively. The 'mapper' converts binary sequences into symbols according to a specific modulation scheme (like QPSK or QAM), while the 'demapper' performs the inverse operation — it converts received symbols back into binary sequences.
- **Simulation Configuration via Sidebar**:
  - `ebno_db`: This parameter allows users to adjust the 'Eb/N0' ratio, which stands for the energy per bit to noise power spectral density ratio. 'Eb/N0' is a critical parameter that signifies the signal-to-noise ratio (SNR) per bit and is expressed in decibels (dB). This slider in the sidebar ranges from 0.0 to 30.0 dB, with a default value of 10.0 dB. Adjusting this

value helps simulate different channel conditions and assess the performance of the communication system under various levels of noise.

– `batch_size`: This configuration sets the number of samples that are processed simultaneously in the simulation. It is adjustable via a slider, allowing a range from 32 to 1024 with increments of 64.

– `block_length`: This setting determines the number of bits in each transmitted message block. It is configurable through a select box, with options generated based on powers of 2, ranging from 2^3 to 2^{10}. The block length choice is important in simulations as it influences the granularity of the data processing and can affect the error rates and throughput of the communication system.

- **Noise Variance Calculation**: In the simulation, the noise variance, represented by the variable `no`, is an essential parameter for setting up the AWGN (Additive White Gaussian Noise) channel. This variance is calculated based on the 'Eb/N0' ratio (denoted as `ebno_db` in the code), which is a measure of the signal strength relative to the noise level. The function `sn.utils.ebnodb2no` from the Sionna library is used for this calculation, where `ebno_db` is the 'Eb/N0' value in decibels, `num_bits_per_symbol` represents the number of bits per symbol for the modulation scheme, and `coderate`[1] is set to 1.0, indicating an uncoded transmission.

- **Simulation Process**: The simulation process is shown involves several key steps. Initially, binary data bits are generated using the `binary_source`, which is configured with the specified `batch_size` and `block_length`. These bits are then mapped to symbols, represented by the variable `x`, using the `mapper`. This mapping process converts the binary data into symbols as per the selected modulation scheme. Subsequently, an Additive White Gaussian Noise (AWGN) channel is simulated, where noise is added to the mapped symbols `x`, resulting in the variable `y`. This step mimics real-world signal distortion due to noise in communication channels. Finally, the received noisy symbols `y` are passed through the `demapper`, which attempts to re-

[1] Signifying the ratio between the number of input data bits and the total number of encoded bits. A 'coderate' value of 1.0 indicates that there is no extra redundancy added for error correction, a characteristic of uncoded transmission systems. Used here because the primary focus is to show the fundamental characteristics of the transmission channel.

construct the original bits from the noisy symbols, producing log-likelihood ratios (llr). These ratios indicate the probability of each bit being a '0' or a '1'.

- **Tensor Shapes**: To provide insights into the internal workings of the simulation, the application includes an expandable section created with Streamlit's `st.expander()` function. This section, titled "Shape of tensors", displays the dimensions of various tensors used in the simulation. It shows the shape of the `bits` tensor, which represents the generated binary bits, the `x` tensor for the mapped symbols, the `y` tensor for the symbols after passing through the AWGN channel, and the `llr` tensor for the log-likelihood ratios.

- **User-Defined Sample Selection**: The application provides a UI component to select the desired number of samples to be visualized. The variable `max_samples` is dynamically set to equal the `block_length`, ensuring that the user cannot select more samples than the simulation block contains. A slider in the Streamlit sidebar, labeled "Number of Samples to Display," allows for an interactive selection ranging from 1 to `max_samples`. The slider's default position is the lesser of 8 or `max_samples`, which accounts for scenarios where the block length is less than 8. Upon selection, `num_samples` determines the actual number of samples to display, while `num_symbols` calculates the corresponding number of symbols by dividing `num_samples` by the number of bits per symbol (`NUM_BITS_PER_SYMBOL`). The code for this is shown in Code Listing 4.5.

```
# Select how many samples to print
max_samples = block_length  # The maximum number of samples is
                              limited by the block length
num_samples = st.sidebar.slider("Number of Samples to Display", 1,
                                max_samples, min(8,
                               max_samples))
num_symbols = int(num_samples / NUM_BITS_PER_SYMBOL)
```

Code Listing 4.5: User Configuration of Sample Display

- **Channel Output Visualization**: The real and imaginary parts of the received symbols y are extracted and plotted using 'matplotlib'. The plot is displayed using 'st.pyplot()'.

4.2 OFDM Uplink Transmission in the Frequency Domain

This web application serves as an interactive platform to simulate and visualize the BER performance of an Orthogonal Frequency-Division Multiplexing (OFDM) system under various channel conditions, following the structure of Sionna's tutorial part 3: Advanced Link-level Simulations [61]. The web interface enables users to manipulate simulation parameters and observe the resulting Bit Error Rate (BER) over a 3GPP Clustered Delay Line (CDL) channel model. The primary objective of this application is to provide a user-friendly interface for configuring and running physical layer simulations. It allows users, even those without extensive knowledge of Python or Sionna's, API to engage with communication system simulations.

Features The web application allows users to:

- Modulation Scheme Configurations: Select different modulation schemes to observe how they affect signal transmission.
- Code Rate Configurations: Select different code rates to observe how they affect signal transmission.
- Eb/No Configurations: Dynamically change the 'Energy per Bit to Noise Power Spectral Density Ratio' to simulate different Signal to Noise scenarios.
- Batch Size Configurations: Dynamically change the batch size to simulate different batch sizes.
- User Terminal Configurations: Dynamically change the number of user terminals.
- Antenna Configurations: Dynamically change the number of antennas for UT and BS to simulate different scenarios.

4.2.1 Application Layout descripton

The application demonstrates a realistic Single-Input Multiple-Output (SIMO) point-to-point link between a mobile user terminal (UT) and a base station (BS).

The application leverages the Sionna library to model key components of an OFDM system, such as encoding, mapping, channel estimation, equalization, and decoding. Furthermore, the web interface allows users change parameters and to simulate and compare the BER performance of least squares (LS) channel estimation with the ideal case of perfect channel state information (CSI).

The Streamlit-based web interface is designed with simplicity in mind, employing sliders and input fields for parameter configuration and a 'Run Simulation' button to initiate the simulation. The results are graphically displayed, offering an intuitive understanding of the simulation outcomes.

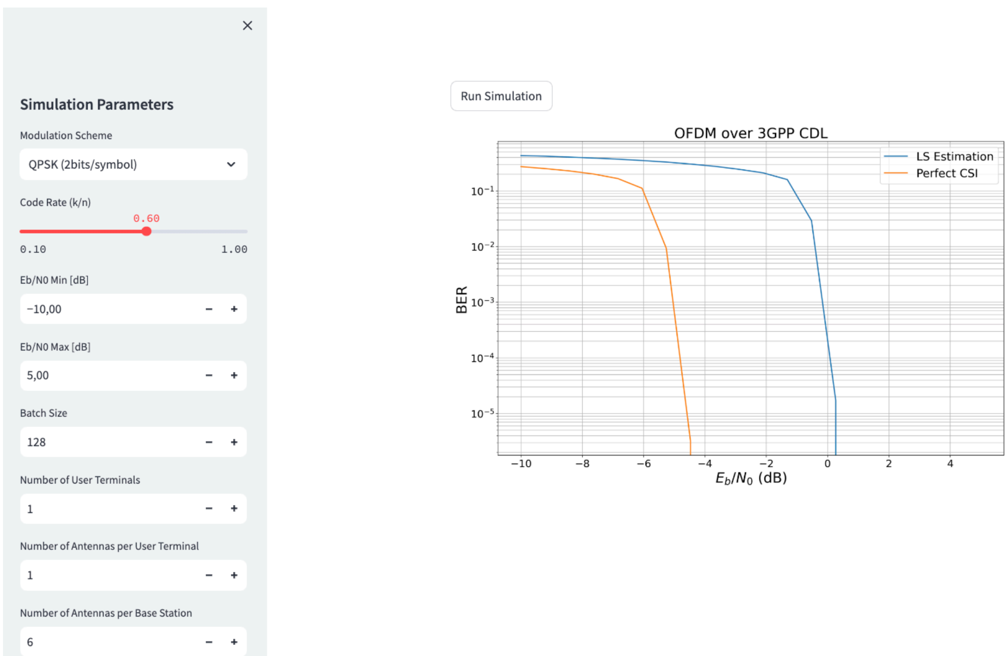


Figure 4.6: OFDM simulation screen

Simulation Configuration

- **Modulation Scheme**: Users can select the modulation scheme from a predefined list, impacting the number of bits per symbol in the simulation.

- **Code Rate Configuration**: A slider allows users to set the code rate (k/n), defining the proportion of information bits to coded bits.

- **Eb/No Min and Max (dB)**: Represents the signal-to-noise ratio (Energy per bit to Noise power spectral density ratio) in dB. This slider allows users to adjust the ratio, which is a critical parameter in assessing the performance of a communication system.
- **Batch Size Slider**: Determines the number of transmissions (or batches of data) that are processed simultaneously. This can be adjusted for performance trade-offs computational speed and the precision of simulation outcomes.
- **Number of User Terminals**: This parameter sets the number of user terminals in the simulation.
- **Antenna Configuration**: Users can specify the number of antennas per user terminal and per base station.
- **Simulation Execution**: A 'Run Simulation' button triggers the simulation process, displaying the resulting BER performance plot.

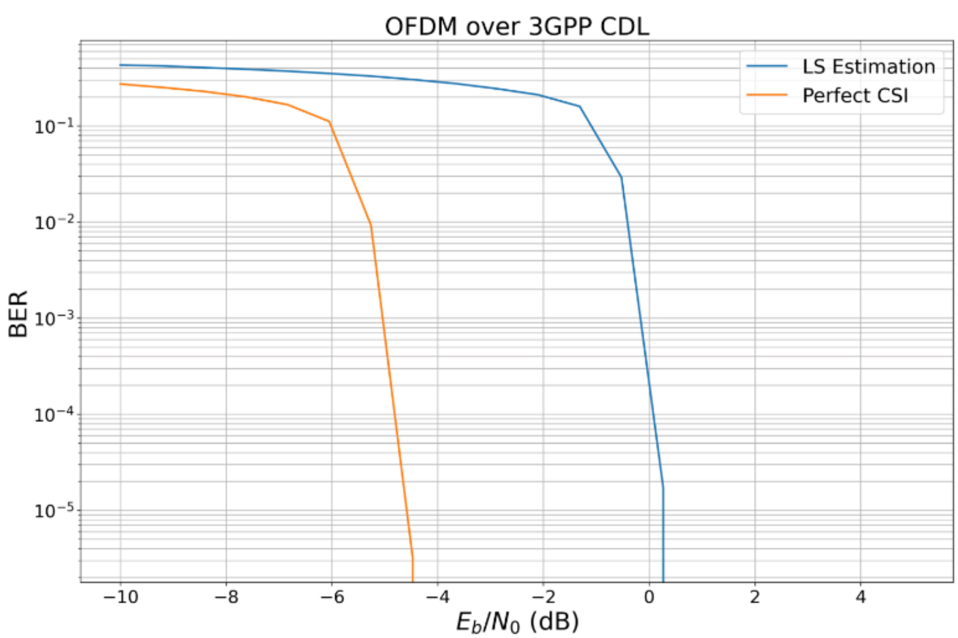


Figure 4.7: LS Estimation and Perfect CSI

The plot shown in Figure 4.7 is a Bit Error Rate (BER) performance graph for an OFDM system over a 3GPP CDL channel model, which is a standard model used to represent typical urban radio propagation environments. The graph plots BER against Eb/N0. The horizontal axis represents Eb/N0 in dB. This is a logarithmic scale that indicates the quality of the communication link; higher values mean better link quality. The vertical axis represents the Bit Error Rate, with lower values indicating better performance. Typically, the BER is expected to decrease as Eb/N0 increases, as confirmed in the graph in Figure 4.7. The blue curve (LS Estimation) represents the BER performance when Least Squares (LS) channel estimation is used. LS estimation is a simple form of channel estimation that is often used in practice, but it is not the most accurate. The performance is generally worse than with perfect knowledge of the channel. The orange curve (Perfect CSI) represents the BER performance with Perfect Channel State Information (CSI). Perfect CSI assumes that the transmitter and receiver have exact knowledge of the channel characteristics, which is an idealized scenario. It serves as a benchmark for the best possible performance that the system can achieve. The gap between the blue and orange curves represents the performance loss due to the imperfect channel estimation of the LS method compared to the ideal case of perfect CSI. At lower Eb/N0 values, the BER is high for both curves, indicating a high error rate due to the noise level. As the Eb/N0 increases, the BER improves (decreases) for both scenarios. However, the BER with LS estimation remains consistently higher than with perfect CSI across the entire Eb/N0 range, as expected, because the LS estimation method, while computationally simpler, doesn't account for certain channel characteristics and noise as effectively as a model with Perfect CSI. The imperfections in channel estimation inherent in LS lead to less accurate symbol recovery, thereby resulting in a higher BER under identical signal-to-noise conditions.

4.2.2 Code Structure and Flow

The code is organized to facilitate the simulation of an OFDM system with user-configurable parameters via a web interface built with Streamlit.

- **Library Imports and System Path Configuration** The code begins by appending the system path to ensure that all custom modules are accessible for the simulation.

- **Streamlit UI Configuration** The web application utilizes Streamlit to create a UI for configuring simulation parameters. This interface includes a sidebar that allows users to input and adjust various parameters of the OFDM system simulation.

- **Modulation Scheme Selection:** A dictionary `modulation_mapping` maps the number of bits per symbol to modulation scheme names. The `st.sidebar.selectbox` function creates a dropdown menu, enabling users to select a modulation scheme. This choice determines the value of `NUM_BITS_PER_SYMBOL`, which is critical for the modulation process in the simulation.

- **Code Rate Configuration:** The code rate (k/n) is adjustable via a slider created with `st.sidebar.slider`, as shown in Code Listing 4.6. This slider allows users to select values between 0.1 and 1.0, with the default set at 0.5, facilitating control over the proportion of information bits to coded bits.

```
CODERATE = st.sidebar.slider(
    'Code Rate (k/n)',
    min_value=0.1,
    max_value=1.0,
    value=0.5,  # Default value of the slider
    step=0.01,
    format="%.2f"  # Format to display the float with two
                                          decimal places
)
```

Code Listing 4.6: Code rate slider

- **Eb/No Range Input:** Number input fields for `EBNO_DB_MIN` and `EBNO_DB_MAX` allow users to specify the minimum and maximum values of the Eb/No ratio in dB.

- **Additional Parameter Configuration:** Users can also configure the batch size (`BATCH_SIZE`), the number of user terminals (`NUM_UT`), and the number of antennas per user terminal (`NUM_UT_ANTENNAS`) and base station (`NUM_BS_ANTENNAS`). These inputs enable the customization of the simulation's complexity and the specific setup of the communication system.

Stream and Antenna Configuration

- NUM_STREAMS_PER_TX: This variable, set equal to the number of antennas per user terminal (NUM_UT_ANTENNAS), defines the number of transmission streams in the system.
- RX_TX_ASSOCIATION and STREAM_MANAGEMENT: These configurations establish the stream management for the system. The RX_TX_ASSOCIATION array, set to a single-element array with a value of 1, indicates a one-to-one association between receive and transmit streams. The StreamManagement module in Sionna is then used to manage these streams.

Resource Grid Setup

- **RESOURCE_GRID**: This object is instantiated from Sionna's OFDM ResourceGrid class. It defines the OFDM resource grid with parameters such as the number of OFDM symbols, FFT size, subcarrier spacing, number of transmitters (user terminals), and the cyclic prefix length. It also specifies the pilot pattern and indices for OFDM symbols used for pilots, essential for channel estimation in OFDM systems.

Antenna and Channel Configuration

- **CARRIER_FREQUENCY**: This variable sets the carrier frequency of the system, which is crucial for defining the antenna element spacing.
- **UT_ARRAY and BS_ARRAY**: These variables define the antenna arrays for the user terminal and the base station, respectively. The configuration includes polarization, antenna type, and carrier frequency. The base station array's size is determined by the number of antennas per base station.

Channel Model Setup

- **DELAY_SPREAD, DIRECTION, and CDL_MODEL**: These parameters configure the channel model. DELAY_SPREAD specifies the nominal delay spread, DIRECTION sets whether the user terminal or base station is transmitting, and CDL_MODEL selects the specific CDL (Clustered Delay Line) channel model variant.

- **SPEED**: This parameter sets the speed of the user terminal, assuming that base stations are fixed and do not move.
- **CDL**: This is the instantiation of the CDL channel model using the above parameters, along with the defined antenna arrays. It models the channel impulse response, which is crucial for simulating realistic communication channel conditions.

OFDMSystem Class

The `OFDMSystem` class, inheriting from Keras' Model class, encapsulates the entire process of an OFDM communication system. This class integrates different functionalities from the Sionna library.

Class Initialization

- **Constructor**: The `__init__` method initializes the class with a boolean parameter `perfect_csi`. This parameter indicates whether the system should assume perfect channel state information during the simulation.
- **Calculating n and k**: The number of coded bits (`n`) and the number of information bits (`k`) are calculated based on the resource grid, number of bits per symbol, and the code rate. These are essential for setting up the encoding and modulation processes.

Sionna Component Initializations

- **Binary Source**: The `binary_source` from Sionna's utility module generates batches of binary data, representing the information bits to be transmitted.
- **Encoder**: The encoder, implemented as an LDPC (Low-Density Parity-Check) encoder (`sn.fec.ldpc.LDPC5GEncoder`), is an importent part of the error correction coding process. This encoder transforms the original information bits (data to be transmitted) into codewords. A codeword consists of the original information bits supplemented with additional parity bits. These parity bits are calculated based on the LDPC algorithm. This

encoding is an important step in preparing the data for transmission over a communication channel.

- **Mapper**: The mapper, created as an instance of Sionna's `sn.mapping.Mapper`, is responsible for the modulation process in the communication system. It converts blocks of binary information bits into symbols that can be transmitted over a communication channel. The mapper is set to use Quadrature Amplitude Modulation (QAM).The user selects the order of the QAM by the dropdown menu discussed in 4.2.2. Hence, the mapper's role in the `OFDMSystem` class is to perform the modulation of the encoded data. The arrangement of symbols on the resource grid can affect various aspects of system performance, including data throughput, error rates, and resilience to channel impairments. The resource grid mapper's role is to optimize this arrangement for the best possible performance under the given system and channel conditions.

- **Resource Grid Mapper**: The `rg_mapper`, an instance of `sn.ofdm.ResourceGridMapper`, takes the output from the mapper (the modulated symbols) and assigns them to specific locations on the resource grid. This includes allocating symbols to the appropriate subcarriers and OFDM symbols (time slots) as per the defined structure of the grid. Besides data symbols, the resource grid also includes pilot symbols, which are known symbols used for channel estimation at the receiver. The `rg_mapper` is responsible for inserting these pilot symbols into the grid at predefined positions.

- **Channel**: The `channel` object simulates the frequency domain channel, including effects like noise and fading. The channel uses the Clustered Delay Line (CDL) model, specified by the `CDL` parameter. The `RESOURCE_GRID` parameter is passed to the channel, linking it with the OFDM resource grid. This integration allows the channel model to consider how each symbol on the resource grid is affected during transmission. Furthermore, by setting `add_awgn=True`, the channel includes the effect of AWGN. The `normalize_channel=True` option ensures that the channel's power is normalized. Normalization is crucial for maintaining consistent signal levels and preventing the signal from either diminishing or overpowering as it passes through the channel. The `return_channel=True` parameter indicates that the channel model also returns the channel state information (CSI). The channel simulation accounts for the effects that would be encountered in a real transmission environment, providing insights into how the system would behave in actual deployment.

- **Channel Estimator**: The `ls_est` (least squares estimator) provides channel estimates. It calculates the channel estimate by minimizing the mean square error between the received signal and the signal expected based on the channel model. The parameter `interpolation_type="nn"` specifies the interpolation method used in the estimator. 'nn' stands for 'nearest neighbor', and so the channel estimate for a given point on the resource grid is approximated as being equal to the nearest known estimate. Using the LS estimation is computationally efficient, but if there is much fading effects in the channel it might not always be the most accurate estimation.

- **Equalizer**: The `lmmse_equ` (Linear Minimum Mean Square Error equalizer) processes the received signal to mitigate channel effects. The equalizer's role is to correct the fading distortions in the channel and restore the signal to its original state as much as possible at the receiver's end. Alongside equalizing the signal, the LMMSE equalizer also provides estimates of the noise variance.

- **Demapper**: The `demapper` converts the received symbols, which have been affected by the channel and equalized, back into a bitstream. The demapper produces LLRs for all coded bits. LLRs provide a measure of confidence about whether a particular bit is a '0' or a '1'. These values are not just binary decisions but contain probabilistic information about each bit, which can be leveraged by the decoder for more effective error correction.

- **Decoder**: The primary role of the `decoder` is to interpret the received bitstream, which includes both data and parity bits, and correct any errors that may have occurred during transmission. This process involves using the parity bits to identify and rectify errors in the data bits. The parameter `hard_out=True` specifies that the decoder will provide 'hard' decisions on the information bits. This means the output will be definitive binary values (0 or 1) for each bit, as opposed to 'soft' decisions which provide probabilistic information.

Method for Simulation Execution

- **Call Method**: Annotated with `@tf.function` for efficient TensorFlow graph execution, this method simulates the transmission and reception of signals in the OFDM system. It takes parameters like batch size and Eb/N0 ratio, and returns the original and received bits.

Simulation Execution and Visualization

- Upon user request, the `run_simulation` function is called, executing the simulation over a range of Eb/N0 values and plotting the BER performance.
- The resulting plot is displayed within the web interface, providing visual feedback on the OFDM system's performance under the specified conditions.

4.3 Web applications summary

This chapter has detailed the development and functionality of two web-based interactive interface for simulating the physical layer of 6G networks using NVIDIA Sionna and Streamlit. Focusing on Constellation Diagrams, QAM modulation schemes and OFDM systems.

The Interactive QAM Constellation Simulation Interface offers users a dynamic platform to visualize and understand the impact of noise and modulation on digital signals. The inclusion of features such as real-time parameter adjustment, immediate visual feedback, and user-friendly controls making concepts more accessible.

The OFDM Uplink Transmission application enabling users to simulate and visualize the impact of different channel conditions and configurations on OFDM system performance. The tool facilitates experimentation with various parameters like Eb/N0 ratios, antenna configurations, and channel estimation methods, providing insights into their influence on the system's bit error rate (BER). The application's design ensures that even users with limited programming knowledge can interact with and learn from these simulations.

Overall, these web applications demonstrates the potential of integrating advanced simulation tools like Sionna with user-friendly interfaces by use of Streamlit.

Chapter 5

Experimental Evaluation

5.1 Experiment 1: BER Performance with QPSK Modulation

In the initial experiment, the BER performance of an OFDM system over a 3GPP Clustered Delay Line (CDL) channel model was evaluated. The system utilized a Single-Input Multiple-Output (SIMO) configuration with the following simulation parameters:

Simulation Parameters:

- Modulation Scheme: Quadrature Phase Shift Keying (QPSK)
- Code Rate: 0.50
- E_b/N_0 Range: -8 dB to 3 dB
- Batch Size: 128
- Number of User Terminals: 1
- Number of Antennas per User Terminal: 1
- Number of Antennas per Base Station: 4

The BER was assessed under two different channel estimation conditions:

1. Least Squares (LS) Channel Estimation

2. Perfect Channel State Information (CSI)

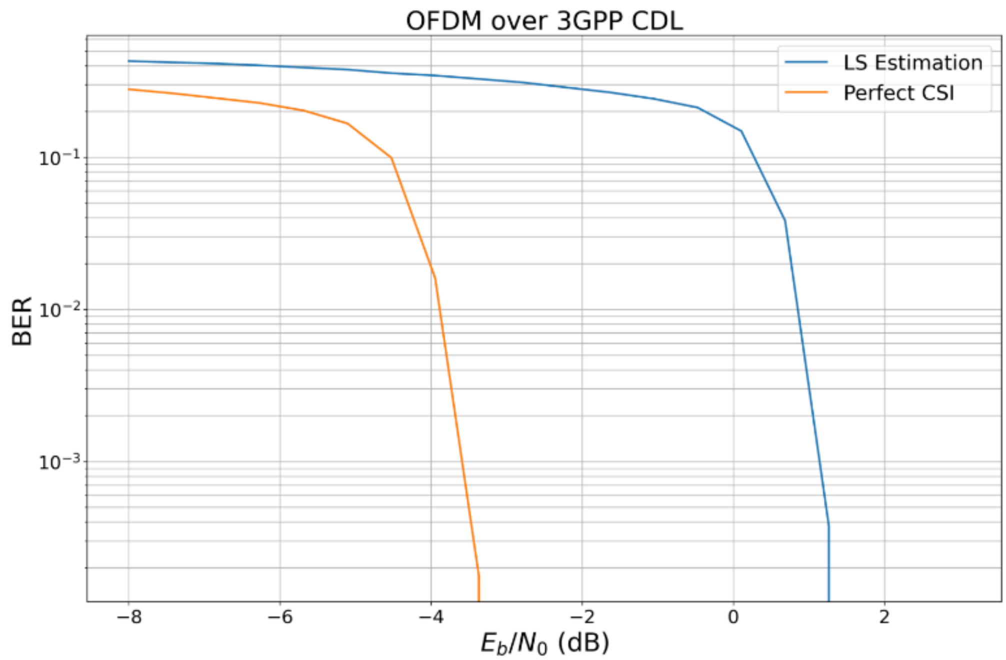


Figure 5.1: BER performance of the OFDM system with QPSK

As observed in Figure 5.1, the system with Perfect CSI outperforms the one with LS Estimation across the entire E_b/N_0 range. This discrepancy highlights the significance of precise channel estimation in OFDM systems. Notably, the BER for LS Estimation degrades as the E_b/N_0 level decreases, illustrating the impact of lower signal-to-noise ratios on system performance.

The results from this experiment establish a baseline for BER performance against the subsequent experiment.

5.2 Experiment 2: BER Performance with 64-QAM Modulation

In the second experiment, the modulation scheme was changed from QPSK to 64-QAM to investigate the effects of higher-order modulation on BER performance. The simulation parameters remained the same except for the modulation scheme and the E_b/N_0 range, which was extended to -2 dB to 10 dB to capture the full extent of the BER curve.

Simulation Parameters:

- Modulation Scheme: 64-QAM (6 bits/symbol)
- Code Rate: 0.50
- E_b/N_0 Range: -2 dB to 10 dB
- Batch Size: 128
- Number of User Terminals: 1
- Number of Antennas per User Terminal: 1
- Number of Antennas per Base Station: 4

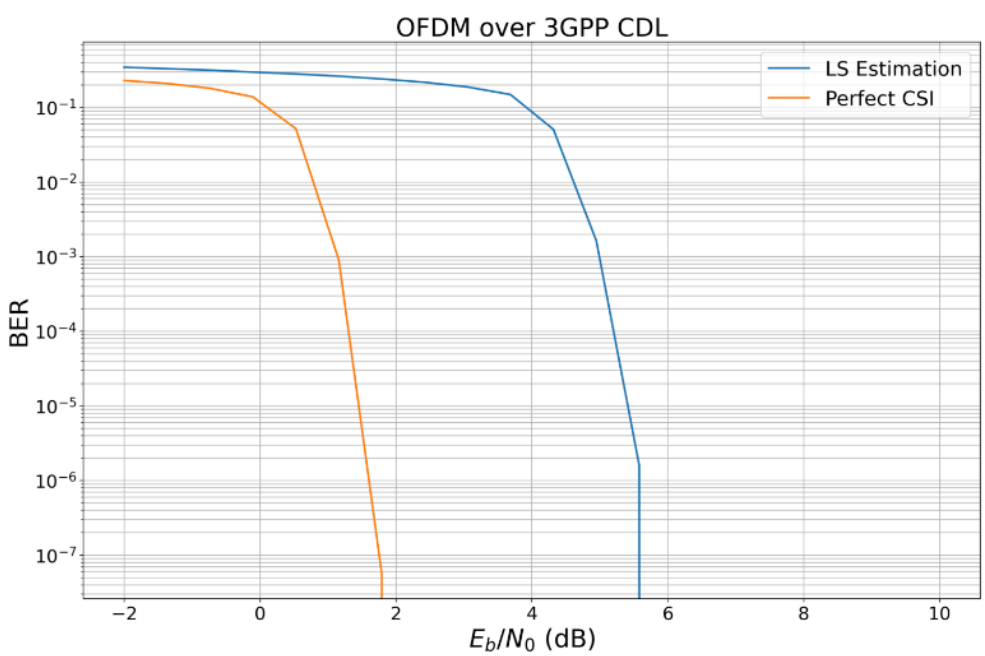


Figure 5.2: BER performance of the OFDM system with 64-QAM

5.3 Results and Discussion:

The results, illustrated in Figure 5.2, show a marked difference in the BER performance with 64-QAM compared to QPSK (as observed in Experiment 1). The BER curve for 64-QAM under perfect CSI conditions shows that to achieve similar BER levels as QPSK, a significantly higher E_b/N_0 is required, which is consistent with theoretical expectations due to the denser constellation of 64-QAM.

Comparatively, the LS Estimation's BER curve also requires a higher E_b/N_0 to achieve performance comparable to the perfect CSI case, which is even more pronounced than in Experiment 1. This is indicative of the increased sensitivity of higher-order modulations to imperfections in channel estimation and demonstrates the trade-off between spectral efficiency and robustness to noise in the choice of modulation schemes for OFDM systems. Higher-order modulation increases spectral efficiency but requires a better signal-to-noise ratio to maintain BER performance.

Chapter 6

Conclusions

This thesis journey began with an in-depth exploration of communication technologies in the *Background and Theory* chapter, revealing the important role of simulations in the evolution of these technologies. A comprehensive understanding of 6G characteristics, link-level simulations, constellation diagrams, modulation schemes, MIMO technologies, various types of multiplexing, and OFDM was developed.

Through hands-on engagement with the Sionna library, this project has provided a platform for deepening knowledge in GPU-accelerated link-level simulations and the complexity of modern communication systems. The Sionna tutorials served as a gateway to understanding the library's components and its functionalities.

The culmination of this research resulted in the development of two web applications. These applications were designed with the integration of Sionna and Streamlit. Their intuitive design facilitates easy configuration and execution of simulations, offering users the ability to fine-tune parameters and directly observe the impact on simulation outcomes through immediate visual feedback.

The first application, an Interactive QAM Constellation Simulation Interface, illustrates the effects of noise on digital signals across different modulation schemes. The second application demonstrates a SIMO link, enriched with adjustable parameters through Streamlit. Both applications are meant to serve as educational tools, enhancing the understanding of complex communication concepts.

While certain features such as the ability to halt simulations mid-execution were not realized, the overall experience has been profoundly educational and insightful. Engaging in this thesis work has deepened my appreciation for the ongoing advancements in the field of communication technologies.

Appendix A

Source Code

The GitHub repository can be accessed here:
`https://github.com/bragesolheim/sionna_streamlit`.

Bibliography

[1] "Introducing 3gpp." https://www.3gpp.org/about-us/introducing-3gpp. Accessed: 2023-09-10.

[2] "Timeline from 1g to 5g: A brief history on cell phones." https://www.cengn.ca/information-centre/innovation/timeline-from-1g-to-5g-a-brief-history-on-cell-phones/. Accessed: 2023-09-10.

[3] "The evolution of apple's iphone." https://www.computerworld.com/article/3692531/evolution-of-apple-iphone.html. Accessed: 2023-11-18.

[4] L. Chettri and R. Bera, "A comprehensive survey on internet of things (iot) toward 5g wireless systems," *IEEE Internet of Things Journal*, vol. 7, pp. 16–32, Jan 2020.

[5] "What is a smart city? – definition and examples." https://www.twi-global.com/technical-knowledge/faqs/what-is-a-smart-city. Accessed: 2023-09-10.

[6] "How automation helps reduce human error and improves data quality." https://trdsf.com/blogs/news/how-automation-helps-reduce-human-error-and-improves-data-quality. Accessed: 2023-09-16.

[7] "What is a 6g network? definition, features, and benefits." https://www.spiceworks.com/tech/networking/articles/what-is-6g/. Accessed: 2023-09-25.

[8] "Revolution or evolution? technical requirements and considerations towards 6g mobile communications." https://www.researchgate.net/figure/4G-5G-and-6G-spectrum-bands_fig2_358001785. Accessed: 2023-09-26.

[9] "Intelligent ultra-reliable and low-latency communications in 6g." https://www.comsoc.org/publications/magazines/ieee-wireless-communications/cfp/intelligent-ultra-reliable-and-low-latency. Accessed: 2023-09-17.

[10] "Building a more sustainable future with 6g." https://www.csrwire.com/press_releases/735836-building-more-sustainable-future-6g. Accessed: 2023-09-26.

[11] "Terahertz communication for vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 66, pp. 5617–5625, July 2017.

[12] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future intelligent and secure vehicular network toward 6g: Machine-learning approaches," *Proceedings of the IEEE*, vol. 108, pp. 292–307, Feb 2020.

[13] M. Z. Chowdhury, M. Shahjalal, S. Ahmed, and Y. M. Jang, "6g wireless communication systems: Applications, requirements, technologies, challenges, and research directions," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 957–975, 2020.

[14] "Link and system level simulators." https://www.surrey.ac.uk/institute-communication-systems/facilities/link-and-system-level-simulators. Accessed: 2023-09-26.

[15] "Multipath fading." https://www.electronics-notes.com/articles/antennas-propagation/propagation-overview/multipath-fading.php. Accessed: 2023-09-24.

[16] "Bit error rate." https://en.wikipedia.org/wiki/Bit_error_rate. Accessed: 2023-09-24.

[17] "Signal-to-noise ratio." https://en.wikipedia.org/wiki/Signal-to-noise_ratio. Accessed: 2023-09-24.

[18] "Gpgpu." https://www.heavy.ai/technical-glossary/gpgpu. Accessed: 2023-09-26.

[19] "Physical layer in osi model." https://shardeum.org/blog/physical-layer-in-osi-model/#What_is_the_Physical_Layer_in_OSI_Model. Accessed: 2023-09-14.

[20] "The osi model." https://www.cloudflare.com/learning/ddos/glossary/open-systems-interconnection-model-osi/. Accessed: 2023-09-14.

[21] "Link-level simulation." https://se.mathworks.com/help/comm/link-level-simulation.html. Accessed: 2023-09-13.

[22] "Baseband signal." https://www.sciencedirect.com/topics/computer-science/baseband-signal. Accessed: 2023-09-27.

[23] "Digital modulation." https://pysdr.org/content/digital_modulation.html. Accessed: 2023-11-02.

[24] "5g antenna vs 4g antenna: What's the difference?." https://www.raypcb.com/5g-antenna-vs-4g-antenna/. Accessed: 2023-09-27.

[25] "How small are 5g antennas." https://blog.antenova.com/how-small-are-5g-antennas. Accessed: 2023-10-29.

[26] "Principles of communication - modulation." https://www.tutorialspoint.com/principles_of_communication/principles_of_communication_modulation.htm. Accessed: 2023-09-24.

[27] "Analog communication - modulation." https://www.tutorialspoint.com/analog_communication/analog_communication_modulation.htm. Accessed: 2023-09-24.

[28] "Physics and radio-electronics: A journey of ideas." https://www.physics-and-radio-electronics.com/blog/frequency-modulation/. Accessed: 2023-09-30.

[29] L. Frenzel, "Understanding modern digital modulation techniques," *Electron. Des. Technol. Commun*, 2012.

[30] M. S. Rahman, "Digital signal processing," *Proceedings of the Short Course on "Microcontroller/Microprocessor Based Industrial Production"*, vol. 1, pp. DSP–1 – DSP, 12 2004.

[31] "What is qam?." https://info.support.huawei.com/info-finder/encyclopedia/en/QAM.html. Accessed: 2023-10-03.

[32] "Quadrature amplitude modulation." https://en.wikipedia.org/wiki/Quadrature_amplitude_modulation. Accessed: 2023-10-03.

[33] "Qam formats: 8-qam, 16-qam, 32-qam, 64-qam, 128-qam, 256-qam." https://www.electronics-notes.com/articles/radio/modulation/quadrature-amplitude-modulation-types-8qam-16qam-32qam-64qam-128qam-256qam.php. Accessed: 2023-10-03.

[34] "Constellation diagram." https://en.wikipedia.org/wiki/Constellation_diagram. Accessed: 2023-10-25.

[35] "What is adaptive modulation?." https://mimomax.com/about-us/our-technologies/what-is-adaptive-modulation/. Accessed: 2023-10-06.

[36] "Mimo (multiple input, multiple output)." https://www.techtarget.com/searchmobilecomputing/definition/MIMO. Accessed: 2023-10-06.

[37] "3gpp)." https://en.wikipedia.org/wiki/3GPP. Accessed: 2023-10-06.

[38] "Is spatial multiplexing different from spatial diversity?." https://commsbrief.com/difference-between-spatial-multiplexing-and-spatial-diversity/. Accessed: 2023-11-05.

[39] U. o. P. Kyle Jamieson, "Mimo i: Spatial diversity." https://www.cs.princeton.edu/courses/archive/spring18/cos463/lectures/L16-mimo.pdf. Accessed: 2023-11-05.

[40] everythingRF, "What is beamforming?." https://www.everythingrf.com/community/what-is-beamforming. Accessed: 2023-11-08.

[41] "What is massive mimo?." https://ma-mimo.ellintech.se/what-is-massive-mimo/. Accessed: 2023-11-07.

[42] "Understanding massive mimo technology." https://my.avnet.com/abacus/solutions/markets/communications/5g-solutions/understanding-massive-mimo-technology/. Accessed: 2023-11-08.

[43] "What is mu-mimo, and why is it essential for wi-fi 6 and 6e?)." https://www.networkworld.com/article/965066/what-is-mu-mimo-and-why-is-it-essential-for-wi-fi-6-and-6e.html. Accessed: 2023-11-06.

[44] "Cooperative mimo." https://en.wikipedia.org/wiki/Cooperative_MIMO. Accessed: 2023-11-10.

[45] "Mimo-ofdm precoding with phased arrays." https://se.mathworks.com/help/comm/ug/mimo-ofdm-precoding-with-phased-arrays.html. Accessed: 2023-11-10.

[46] "Smart antennas." https://en.wikipedia.org/wiki/Smart_antenna. Accessed: 2023-11-08.

[47] "What is multiplexing?." https://www.techtarget.com/searchnetworking/definition/multiplexing. Accessed: 2023-09-27.

[48] "Multiplexing." https://en.wikipedia.org/wiki/Multiplexing. Accessed: 2023-10-17.

[49] "Multiplexing techniques." https://mastersharvanprajapati.blogspot.com/2018/02/q.html. Accessed: 2023-10-19.

[50] "Types of multiplexing in data communications." https://www.geeksforgeeks.org/types-of-multiplexing-in-data-communications/?ref=lbp. Accessed: 2023-10-19.

[51] "Ofdm – the science behind lte." https://yatebts.com/ofdm-the-science-behind-lte/. Accessed: 2023-10-12.

[52] "Orthogonal frequency-division multiplexing." https://en.wikipedia.org/wiki/Orthogonal_frequency-division_multiplexing. Accessed: 2023-10-14.

[53] "Geforce 256." https://en.wikipedia.org/wiki/GeForce_256. Accessed: 2023-12-01.

[54] "Nvidia sionna: An open-source library for 6g physical-layer research." https://developer.nvidia.com/sionna. Accessed: 2023-09-16.

[55] "streamlit." https://streamlit.io/. Accessed: 2023-09-27.

[56] "What is tensorflow? deep learning libraries program elements." https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-tensorflow. Accessed: 2023-09-17.

[57] "Tensor." https://no.wikipedia.org/wiki/Tensor. Accessed: 2023-09-17.

[58] "Keras." https://keras.io/about/. Accessed: 2023-09-17.

[59] "Github." https://en.wikipedia.org/wiki/GitHub. Accessed: 2023-12-01.

[60] "Part 1: Getting started with sionna." https://nvlabs.github.io/sionna/examples/Sionna_tutorial_part1.html. Accessed: 2023-08-14.

[61] "Part 3: Advanced link-level simulations." https://nvlabs.github.io/sionna/examples/Sionna_tutorial_part3.html. Accessed: 2023-08-14.

University of Stavanger

0369 Oslo
E-mail: b.solheim@stud.uis.no

Cover Photo: GETTY