POLITECNICO DI TORINO

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATIONS

MASTER'S DEGREE IN ELECTRONIC ENGINEERING

EDFA noise figure and WSS DGD modeling

SUPERVISOR:
Prof. Vittorio Curri

CANDIDATE:
Alberto Castronovo

CO-SUPERVISORS:
Dr. Rocco D'Ingillo
Dr. Renato Ambrosone

ACADEMIC YEAR 2023 - 2024

Index

Introduction

The surge in demand for applications such as virtual and augmented reality, expansive cloud services, and high-definition video streaming has significantly increased the need for greater capacity in optical and wireless networks [1]. As these technologies advance, they continually increase their requirements for high bandwidth and low latency, necessitating ongoing improvements in network infrastructure. This is a significant challenge for 5G networks, as the backhaul infrastructure used for previous generations is inadequate for meeting the stringent requirements in terms of very low latency and efficient management of dense traffic [2].

Nowadays, optical systems are approaching their theoretical capacity limits due to significant advancements like probabilistic constellation shaping and forward error correction. These innovations have helped maximize the efficiency of data transmission, pushing towards the Shannon limit. To move forward, the focus must shift to optimizing the physical layer and enhancing the control layer with more efficient resource allocation and management [3].

Innovative approaches such as software-defined networking (SDN), elastic optical networks (EON) and self-driving optical networks have emerged to push performance further. SDN decouples network control from hardware, enabling dynamic management and reconfiguration of resources. EON allows flexible allocation of spectrum based on traffic demands, improving bandwidth efficiency. Self-driving optical networks leverage AI and automation to optimize network operations, improving overall efficiency and reliability [4].

The modeling of Erbium-Doped Fiber Amplifiers (EDFA) is crucial for optimizing optical networks, as these components define the transmission bandwidth and play a crucial role in determining the Optical Signal-to-Noise Ratio (OSNR) of the signal [5]. EDFAs are a well-established technology, yet certain parameters, such as gain, amplified spontaneous emission (ASE) and noise figure, can vary significantly depending on signal attributes such as input power and channel frequency, but these variations do not conform to simple analytical formulas [6]. All of these considerations suggest that the development of accurate predictive models for EDFA parameters could be considered a potentially rewarding challenge.

Numerous models for EDFA parameters already exist in the literature [7] [8] , primarily focusing on the estimation of gain [9]. Explicit models have the lowest cost because they can be built based primarily or solely on a-priori knowledge of the devices, but often fall short of high accuracy requirements. Data-driven models, based on neural networks, instead offer greater accuracy [10] but are more difficult to employ in real-world applications due to the large datasets needed for training, which are often unavailable or expensive to acquire.

In this thesis work, a polynomial model for the noise figure of EDFAs is proposed. The design emphasizes high-speed implementation, to integrate seamlessly with more comprehensive machine learning EDFA models without imposing significant computational delays. Additionally, the model is designed for flexibility, allowing users to adjust the size of the training datasets to balance between accuracy and cost of measurements.

In addition to the noise figure model, measurements and tests are conducted to study the behavior of the Differential Group Delay (DGD), a parameter related to distortion. The objective is to determine whether the DGD behavior in more complex devices, such as optical multiplexers and demultiplexers, aligns with the linear behavior observed in optical fibers or if other effects are present.

Chapter 1

Open optical networks

1.1 Optical communications: an overview

1.1.1 History of optical communications

Optical communication has ancient roots, with early civilizations using smoke signals, fire, and mirrors to convey simple messages like battlefield victories. These rudimentary methods had significant limitations, such as predetermined messages and short transmission distances [11]. The evolution of human communication can be measured by two key metrics: transmission rate (the number of distinct symbols sent per unit time) and transmission distance. A major breakthrough in optical communication came in the late 18th century with Claude Chappe's optical telegraph. This system used metal bars atop towers to encode a basic alphabet, allowing transmission over distances of 10-15 km between towers, as shown in Figure 1.1. Chappe also introduced the concept of repeaters, with each tower replicating the symbol from the previous one. In modern terms, this system's transmission rate would be less than $1\,\mathrm{bit\,s^{-1}}$, but for the time it was a significant achievement. However, it was limited to daylight operation and required simultaneous coordination among all tower operators along the transmission line. The early 19th century, instead, saw optical communication give way to electrical methods. The electrical telegraph, employing new coding schemes like Morse code, offered higher transmission rates. Analog electrical techniques, including the transatlantic telegraph cable and the telephone, dominated long-distance communication for over a century.

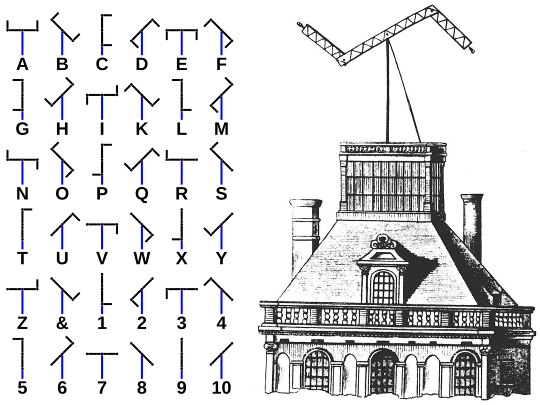


Figure 1.1: Optical telegraph coding scheme and signal tower.

The 20th century witnessed the establishment of a global telephone network. The introduction of coaxial cables increased capacity, but signal losses remained a critical issue, limiting operating frequencies to a few MHz [12]. This constraint drove innovators and researchers towards radio and microwave communication systems. These alternatives achieved similar performance to coaxial cables with fewer repeaters, but still faced bandwidth limitations of about $100\,\mathrm{Mbit\,s^{-1}}$. Optical communications regained prominence after 1960 with the invention of lasers. However, optical fibers at that time suffered from very high losses, making them unsuitable for long-distance communication. This sparked a concerted effort to reduce fiber losses. A significant breakthrough came in 1970 when Corning, an American company, achieved losses below 20 dB/km at wavelengths near 630 nm. Two years later, the same team further reduced losses to $4\,\mathrm{dB\,km^{-1}}$ by using germanium instead of titanium as the core dopant. The race to minimize losses continued until 1979 when a Japanese group achieved $0.2\,\mathrm{dB\,km^{-1}}$ in the wavelength region near 1550 nm, approaching the theoretical limit set by Rayleigh scattering. This value has remained largely unchanged to the present day. The first *GaAs* lasers emitted light at around 850 nm, establishing the "first window" used in optical communications, with transmission rates of about $45\,\mathrm{Mbit\,s^{-1}}$. Losses at this wavelength were around $3\,\mathrm{dB\,km^{-1}}$, requiring signal regeneration after approximately 10 km - a significant improvement over coaxial cables, which needed regeneration every 1 km. This marked the dawn of a new era in long-range communications, which became increasingly evident with the advent of the Internet at the end of the 20th century. The ability to transmit data over greater distances with fewer interruptions and higher bandwidths paved the way for the global connectivity we experience today.

A very important milestone that has impacted on optical communications in recent times is the employment of coherent receivers and, consequently, the modulation formats they support. Overlooked up to early 1980s in favor of intensity modulated direct-detection (IMDD) systems, coherent technology regained interest after a demonstration of precise frequency stabilization of semiconductor lasers [13], essential for heterodyne detection due to the sensitivity to phase and state of polarization (SOP) variations. This same sensitivity to the phase of the incoming signal, not just to the amplitude, is one of the key benefits of this technology, because it enables detection of phase-encoded modulation formats like phase shift keying (PSK) and quadrature amplitude modulation (QAM) [14]. Coherent receivers inherently offer a very narrow filtering capability, needed in today's strive towards high density Wavelength Division Multiplexing (DWDM), and Digital Signal Processors (DSPs) can be effectively used to equalize impairments such as chromatic dispersion and polarization-mode dispersion, leading to a zero net penalty, impossible with direct detection receivers [15].

1.1.2 Modern challenges of optical networks

A 2010 study [16] introduced a method for calculating the theoretical capacity limit of single-mode fibers. A crucial finding was the relationship between spectral efficiency (measured in $\mathrm{bit\,s^{-1}\,Hz^{-1}}$) and signal-to-noise ratio (SNR) for various transmission distances. Based on these results and assuming the use of polarization-division multiplexing, the maximum capacity of a single-mode fiber was estimated to be around $200\,\mathrm{Tbit\,s^{-1}}$. However, this prediction has since been surpassed, also due to more advanced multiplexing techniques and the advancements in technology allowing to use a larger bandwidth. Current research has pushed beyond $400\,\mathrm{Tbit\,s^{-1}}$utilizing more than 1500 Dense Wavelength Division Multiplexing (DWDM) channels distributed across multiple bands in a single optical fiber [17]. The demand for data traffic has grown steadily since the commercial introduction of the Internet in 1994. This growth has accelerated in recent years due to the proliferation of high-bandwidth services such as high-definition video streaming, virtual reality applications, and cloud computing. This increasing demand presents several challenges for the field of optical communications. Researchers are exploring new modulation formats, such as Space-Division Multiplexing (SDM), which shows promise for further increasing fiber capacity. Additionally, improving energy efficiency has become a critical focus in the development of optical communication systems [11].

A 2008 study [18] revealed that energy consumption in optical networks was primarily driven by access networks - the infrastructure connecting subscribers to their service providers. In contrast, optical transport consumed relatively little energy, accounting for less than 5% of total consumption. At higher bit rates, network routers became the dominant energy consumers. This energy distribution pattern suggests several areas for improvement. Reducing the hop count (the number of devices, typically routers, through which a data packet travels) is crucial. Additionally, enhancing router efficiency, implementing caching strategies, and utilizing content distribution networks can contribute to overall energy reduction. Another challenge in the optical network landscape is the prevalence of closed systems. Most currently deployed optical networks rely on components, both hardware and software, provided by a single vendor or a small group of tightly integrated vendors. This closed approach presents several drawbacks, including limited flexibility, vendor lock-in for future upgrades, higher upgrade costs, and more cumbersome innovation cycles. Recognizing these limitations, many industry experts are advocating for a transition towards open optical networking approaches. This shift aims to create more flexible, cost-effective, and innovative optical network infrastructures. As the demand for data transmission continues to grow, addressing both energy efficiency and network openness will be crucial in developing sustainable and adaptable optical communication systems for the future.

1.2 Optical network structure and technology

1.2.1 Elements of an optical network

The key components within an optical network include:

1. **Transceivers:** used to transmit and receive data, converting electrical signals into optical pulses and vice-versa.

2. **Transponders:** similar to transceivers, they are used to extend the tranmission distance by converting the signal to a different frequency and amplifying it, without changing the data content.

3. **ROADMs:** short name for Reconfigurable Optical Add/Drop Multiplexers, they allow for specific wavelengths to be dropped or added at a location, while allowing other wavelengths to traverse the device untouched.

4. **Optical Line Systems (OLS):** these systems form the backbone of optical networks, providing the necessary infrastructure for optical connections between network nodes (typically ROADMs). OLSs, as shown in Figure 1.2 [19], consist of two primary components:

   - **Optical fiber:** this is the fundamental medium for light transmission in optical networks. Engineered to guide light signals efficiently, optical fibers can maintain signal integrity over exceptionally long distances;
   - **Optical amplifiers:** placed at strategic intervals along the optical fiber route, these devices serve to periodically regenerate the optical signal. Their primary function is to compensate for signal attenuation that occurs naturally as light travels through the fiber, ensuring that the signal remains strong and clear over extended distances.

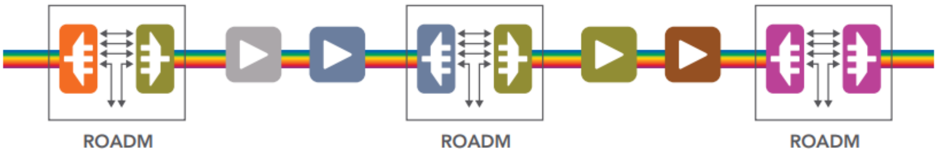


Figure 1.2: Example of an open OLS structure for disaggregated networks [19].

Transceiver

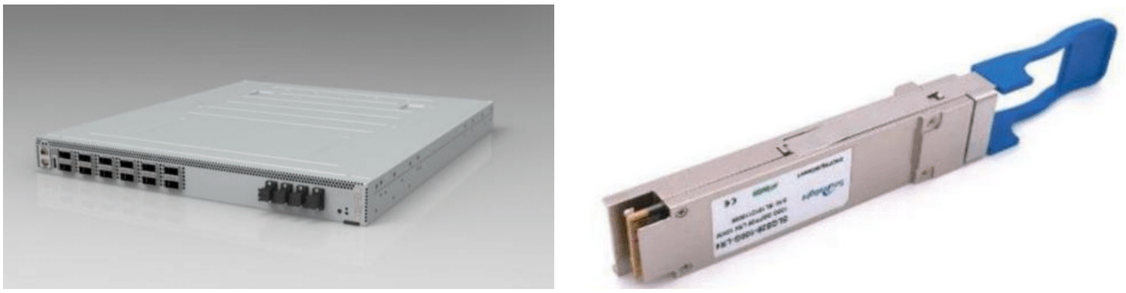
A transceiver consists of two primary components: the transmitter and the receiver. The transmitter utilizes a wavelength-specific laser to convert an input electrical signal into an optical signal. Conversely, the receiver employs a photodiode to perform the reverse operation, transforming optical signals back into electrical ones.

These devices can handle various signal types, including wideband signals at 850 nm, 1310 nm, or 1550 nm. They're also capable of processing various multiplexing techniques like Coarse Wavelength Division-Multiplexed (CWDM) or Dense Wavelength Division-Multiplexed (DWDM) signals. In essence, transceivers serve as the crucial interface between optical fibers and electronic devices.

The transceiver's design, featuring separate circuits for transmission and reception, enables independent operation of these functions and therefore full-duplex communication. Optical transceivers are available in a variety of form factors, with their size primarily determined by modulation speed. Despite this variety, they all adhere to the Multisource Agreement (MSA) standard, ensuring compatibility across different manufacturers and systems. Different form factors offer distinct advantages. For instance, the SFP-DD (Small Form-factor Pluggable - Double Density) is one of the most compact options, enabling high port density in network equipment. On the other hand, form factors like QSFP-DD (Quad Small Form-factor Pluggable - Double Density) and OSFP (Octal Small Form-factor Pluggable) support the highest speeds, currently up to $800\,\mathrm{Gbit\,s^{-1}}$. These high-speed options are designed to meet the demands of today's data-intensive applications and network infrastructures. Figure 1.3b shows a QSFP-DD transceiver.

Transponder

Optical transponders, also known as optical-electrical-optical (OEO) devices, share structural similarities with transceivers, comprising both a transmitter and a responder. However, their function is more specialized. These devices are designed to automatically receive, amplify, and retransmit an optical signal on a different wavelength, all while preserving the integrity of the transmitted data. The wavelength switching operation is made in the electrical domain. It is used to shift the input signal into a WDM channel of choice. Transponders are needed in case devices that work with different wavelengths need to communicate with each other, or when a system employs both single-mode and multi-mode fibers and an interface between the two is needed. An example device is shown in Figure 1.3a.



(a) A transponder.

(b) A transceiver.

Figure 1.3: Examples of optical network elements.

ROADM

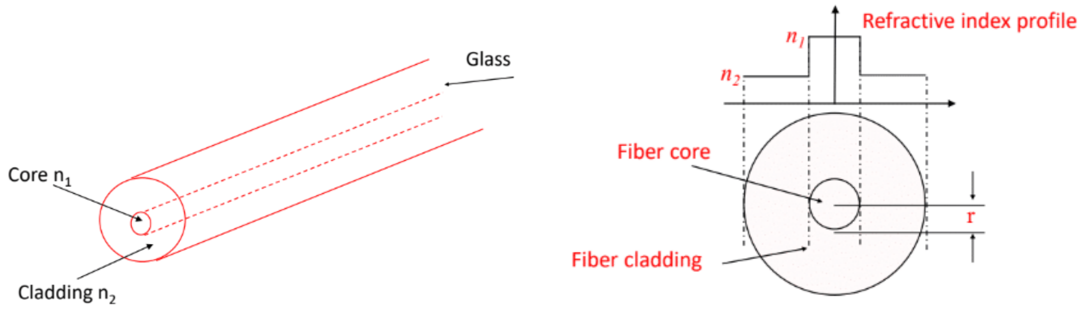
ROADMs are WDM networking devices that function analogously to a road intersection for optical fiber paths. Their primary role is to selectively add or drop specific wavelengths while allowing others to pass through unaffected, effectively managing the flow of optical signals within a network. They are an evolution of OADMs, which were devices with fixed configuration, meaning that it was not possible to configure them after deployment, and were limited to two directions. ROADMs, instead, can be remotely configured and support multiple directions, allowing higher freedom in creating the optimal network connection structure. ROADMs rely primarily on Wavelength Selective Switches (WSS) for their core functionality, which are 1-input, multiple-output devices that take as input a signal with potentially multiple WDM channels, and re-route each channel to a specific output port. Initially, ROADMs utilized fixed-grid WSS with a predetermined channel pattern and spacing. This rigid pattern meant that any wavelengths added to the network had to conform precisely to these fixed channels to traverse the ROADM effectively. However, with the growing demand for higher baud rates and therefore wider channel sizes, ROADMs have evolved to support flexible channel grids. Recent advancements in ROADM technology have introduced even greater levels of flexibility [20]. One significant improvement is the ability to assign wavelengths with more freedom, often referred to as "colorless" operation. This feature allows any wavelength to be assigned to any port, enhancing network adaptability. Another key development is the removal of directional constraints on add/drop wavelengths, known as "directionless" operation. This capability allows wavelengths to be added or dropped from any direction, increasing network routing options and efficiency. Furthermore, state-of-the-art ROADMs now offer the ability to re-route any wavelength to any direction without restrictions, a feature often termed "contentionless" operation. This advancement eliminates potential conflicts when multiple signals vie for the same resources, thereby optimizing network performance and flexibility.

A particular ROADM type is Wavelength Crossconnect (WXC). The goal of this structure is to be able to route any wavelength from any input port to the same wavelength on any output port. For a WXC of n-th degree that supports m wavelengths, meaning it consists of n input and n output ports, we need n optical multiplexers at the input side, n optical demultiplexers at the output, and m optical switches with $n \times n$ ports in between. Figure 1.4 shows an example of a commercial ROADM from Lumentum [21].

Figure 1.4: Picture of a commercial ROADM [21].

Optical fiber

Optical fibers are thin, flexible circular cables, primarily made of high-quality silica glass, though other materials can be used. These fibers excel at transporting light signals, typically from lasers, over vast distances with minimal signal loss. The fiber's structure, illustrated in Figure 1.5, is key to its function. It consists of a core, where light travels, surrounded by a cladding. The light is confined within the core through total internal reflection, a phenomenon caused by the difference in refractive index between the core and cladding materials. This design allows optical fibers to guide light efficiently over long distances, making them crucial components in modern communication networks. The fibers that are employed over long distances are single-mode fibers (SMF), with a core diameter of around 8 µm and a cladding diameter of 125 µm. As the fiber is very thin and fragile, plastic coatings are used to enhance their mechanical properties.



(a) (b)

Figure 1.5: Diagrams showing the internal structure of an optical fiber. (a) general shape of the fiber, showing the core and cladding; (b) refractive index difference between core and cladding, which enables light confinement.

Optical amplifier

Optical amplifiers are specialized devices designed to boost the power of optical signals as they propagate through a network. Unlike transponders, which require optical-electrical-optical (OEO) conversion, optical amplifiers operate directly in the optical domain. Various types of optical amplifiers exist, each leveraging different physical effects to achieve signal amplification. One prominent category is Doped Fiber Amplifiers (DFAs). These devices utilize a specialized optical fiber segment with a core doped with rare-earth materials. When excited by a pump laser, these doped materials release additional photons at the same wavelength as the incoming optical signal, resulting in a stronger light pulse. An example of an Erbium Doped Fiber Amplifier (EDFA) from Cisco is reported in Figure 1.6 [22].

Multiple devices exist, with different doping materials, because each one is different in terms of the range of wavelengths that it can effectively amplify. The most employed in long-range and metropolitan optical networks are EDFAs, which work in C and L optical bands, respectively between 1530 nm and 1565 nm and between 1565 nm and 1625 nm. Semiconductor Optical Amplifiers (SOAs) amplify signals by injecting an electric current into a single-mode semiconductor fiber. The peak gain of these devices is determined by the semiconductor material and occurs for

Figure 1.6: Picture of a commercial EDFA [22].

photon energies just above the semiconductor band gap. Raman amplifiers, on the other hand, rely on the Stimulated Raman Scattering (SRS) phenomenon. They amplify the signal through interaction with the medium's vibrational modes. This process can be cascaded, allowing amplification across a broad wavelength range.

Measurement and laboratory devices

The following devices have been used throughout the measurements conducted for the realization of this work:

1. **OSA:** Optical Spectrum Analyzers are precision instruments used to measure and display the power spectral distribution of an optical signal across a defined wavelength range.

2. **VOA:** Variable Optical Attenuators are devices used to reduce the optical power of an input signal, with the attenuation level adjustable manually or controlled by an electrical input signal.

3. **Polarization scrambler:** a device that accepts an input optical signal and returns it with its State of Polarization (SOP) rapidly randomized, aiming to achieve a time-averaged depolarization effect.

4. **Polarization analyzer:** a device used to analyze polarization properties of an optical signals, including the representation of the State of Polarization (SOP) on the Poincaré sphere.

1.2.2 Transmission technology

Wavelength Division Multiplexing

In communication technologies, and specifically in optical communication infrastructures, the highest interest goes into maximizing the bandwidth of the medium, meaning the amount of data that can be transferred in the time unit over each single piece of fiber. This is done to reduce costs related to material, deployment and management of networks infrastructures, especially today, with an ever-increasing demand of bandwidth from internet applications. Multiplexing technologies allow multiple independent signals to share the same transmission medium and be transmitted in the same time or space, without interfering with each other. Multiple multiplexing technologies exist, and have been used in the past, like Time Division Multiplexing (TDM), or in other applications such as TV and radio transmission, like Frequency Division Multiplexing (FDM), or are being experimented for future usage, like Space Division Multiplexing (SDM). Currently, the most employed technique in optical communications is Wavelength Division Multiplexing (WDM), further divided into Coarse-WDM and Dense-WDM, depending on the number of supported simultaneous channels. This technique consists in propagating multiple different signals on the same piece of fiber, by using a different wavelength range for each channel. This also allows bidirectional communications over a single optical fiber, provided that the two directions utilize different independent channels. Figure 1.7 shows the basic diagram of a WDM setup, including optical multiplexer and demultiplexer and an optical fiber that allows the simultaneous transport of all the channels on different wavelengths.

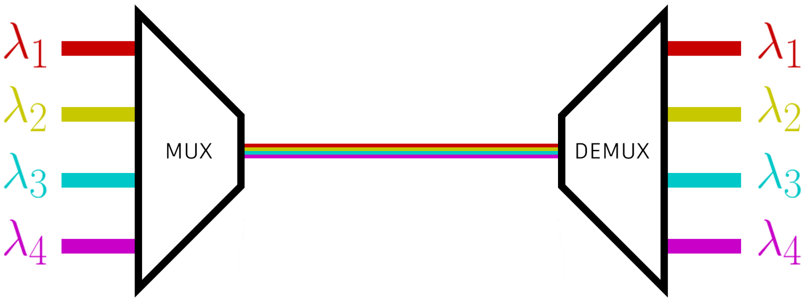


Figure 1.7: Representation of a WDM setup.

CWDM and DWDM differ in the spacing between adjacent channels and in the ability to amplify the signals while remaining in the optical domain, without

OEO conversion. These differences make CWDM a less expensive solution for shorter distances, up to around 70 km, while DWDM is suitable for much longer distances, as it supports also signal amplification. This technique requires an optical multiplexer at the transmitter side, which combines all the input signals into the same fiber, and an optical demultiplexer at the receiver side, which performs the inverse operation, and separates the content of the fiber into multiple outputs, one channel per line. This technique, with current technology, easily allows for a capacity of 800 Gbit s^{-1}, although this number is progressively growing up.

Modulation format

The modulation technologies and techniques used to transmit symbols in optical fibers can also greatly influence medium capacity. The first 10 Gbit s^{-1} transceivers used Intensity-Modulation with Direct-Detection (IMDD), which is a simple modulation format, used until 2010 and now limited to access networks and short distance transmissions. The advent of coherent, DSP-based optical receivers in 2007 enabled the use of multilevel modulation formats, which means that multiple bits were transmitted with each symbol, thus multiplying the capacity. The transmission rate for each wavelength rapidly increased from 10 to 100 Gbit s^{-1} and more, and recently it reached 800 Gbit s^{-1}.

1.3 Towards open networks and software solutions

1.3.1 The transition to open optical networks

Recently, there has been a growing push towards more open and automated optical networks [23]. This approach offers benefits such as vendor interoperability, reduced total cost of ownership (TCO), accelerated network innovation, and improved service delivery. Many companies are backing this trend, as evidenced by numerous white papers on the subject [24] [25], by contributing to the development of open standards and interfaces like YANG data models, Open APIs, and the Open ROADM Multi-Source Agreement (MSA).

In short, open optical networking involves applying open architectures, interfaces, and systems to optical networks [24]. This typically involves deploying network infrastructure with Open Line Systems (OLS) and open transponders. The main idea is to disaggregate the network, allowing OLS, transponders, and terminal systems from different vendors to work together. In the simplest scenario, a single supplier provides the entire OLS, while transponders and end-to-end systems come from other vendors. However, some network operators propose further disaggregation, with multiple vendors' components integrated within the OLS, mixed and matched as needed [19].

1.3.2 Software Defined Networks

Traditional networking devices like repeaters, bridges, routers, and switches have effectively managed network traffic for decades. However, modern network environments, particularly data centers, are growing too complex and costly to handle with these conventional technologies. This escalating complexity and expense, along with the demand for faster innovation, are driving a shift from traditional methods to a more open, flexible, and innovation-driven approach called Software-Defined Networking (SDN) [26].

SDN originally emerged from the OpenFlow project at Stanford University. It consists in a shift in network architecture, fundamentally characterized by four aspects:

1. **Decoupling of Control and Data Planes:** control functions are separated from network devices, which are simplified to merely forward packets.

2. **Flow-Based Forwarding:** Instead of relying on destination-based routing, SDN uses flow-based forwarding. A flow is defined by a set of packet characteristics and corresponding actions, ensuring uniform treatment of all packets within a flow.

3. **Centralized Control Logic:** The control logic is relocated to an SDN controller or Network Operating System (NOS), a software platform on commodity servers that centralizes and abstracts network management, much like an operating system.

4. **Programmability:** The network is controlled by software applications interacting with the data plane through the NOS, making SDN highly adaptable and allowing dynamic changes to network policies.

This architecture simplifies network management by centralizing control, which enhances the ease of policy modification and automatic adaptation to network state changes. It also facilitates the development of advanced networking functions. Abstractions in SDN, particularly those related to forwarding, distribution, and specification, simplify programming and control of the network, with OpenFlow serving as a foundational abstraction akin to a device driver in computing. The distribution abstraction within the NOS provides a global network view and handles the dissemination of control commands to forwarding devices [27].

1.3.3 Digital twin and machine learning

A digital twin (DT) is the virtual replica of an object, process, or an entire system, created for simulation, testing and optimization purposes, allowing these operations without the need to physically interact with the subject. DTs are increasingly applied to telecommunications, particularly optical networks. The concept of a Digital Twin Optical Network (DTON) involves creating a digital replica of the physical network for simulation and real-time management. Unlike traditional optical networks, which suffer from limited sensing and static modeling capabilities, DTON enhances network operations through accurate predictions, dynamic simulations, and optimization strategies [28]. DTON facilitates monitoring, modeling, and control of optical networks, significantly improving deployment planning and operation. It supports applications such as fault detection, quality estimation, power control, and configuration prediction, primarily focusing on the physical layer. However, integrating the network layer as well is crucial for comprehensive global control and management. Several architectures to deploy and operate a DTON are proposed in literature [29]. In general, DTONs are deployed using a cloud infrastructure and are composed of different parts, which may include a data collector acting as a broker to gather and distribute data, a data repository that maintains and updates the topology and connectivity of the network, Service Mapping Models (SMM) which provide data modes for various network applications, and the Digital Twin Entity Manager (DTEM), which manages the DTON as a whole, tracking its lifecycle and interacting with the virtualization platform.

In recent times, the integration of Machine Learning (ML) into the field of optical networks has been a recurring interest [30]. Some providers are beginning to incorporate ML to improve resource management, monitoring, traffic prediction and classification and fault handling. Traditional methods such as Bayesian estimation or heuristic solutions usually only take into account the current state of the network and ignore historical information. ML techniques would rely on models trained on historical data, and therefore be more immune to noise. Moreover, especially due to 5G networks, data traffic has become increasingly dynamic and heterogeneous, and flexible reconfiguration is getting more and more critical to meet QoS levels. Traditional methods present a large computational effort, and ML could offer a more efficient real-time solution.

Chapter 2

EDFA Noise Figure modeling

2.1 An overview on EDFA devices

An Erbium-Doped Fiber Amplifier (EDFA) is a device used to amplify input signals in optical fibers. It is extensively used in fiber-optic telecommunications due to its high gain, low noise capabilities, and ability to support a wide range of wavelengths. EDFAs consist of several components, the most crucial of which is a length of fiber with a core uniformly doped with erbium ions. The amplification process is achieved by propagating a pump laser alongside the input signal through the same Erbium-Doped Fiber (EDF). This excites the erbium ions, allowing for the stimulated emission of additional photons [3].

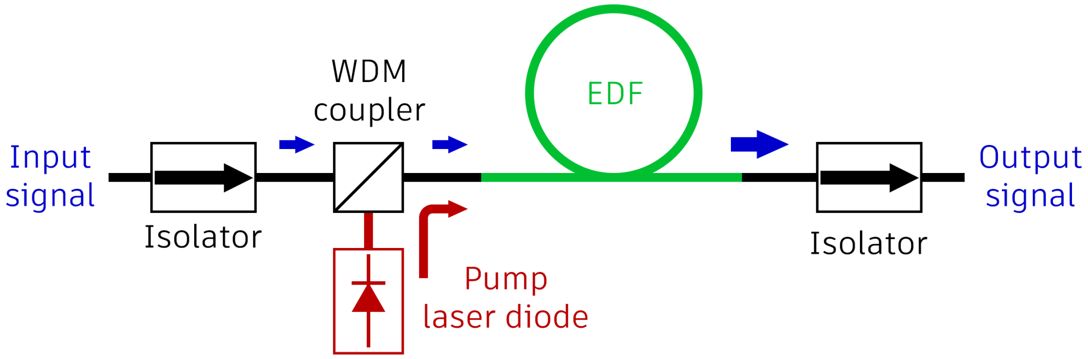


Figure 2.1: A basic EDFA block diagram.

Figure 2.1 illustrates the block diagram of an elementary internal EDFA structure. More advanced architectures, such as multi-stage structures, are used to improve performance but are beyond the scope of this analysis. The input signal at optical wavelengths is coupled into the EDF along with a shorter-wavelength pump laser (typically 980 nm or 1480 nm), commonly via a Wavelength Division Multiplexing (WDM) coupler. Optical isolators are employed at both the signal and pump sources, as well as at the output of the EDF, to prevent backpropagation and unwanted reflections. The pump laser light can propagate in the same direction as the signal (co-propagation) or in the opposite direction (counter-propagation). For a preliminary analysis, the Er^{3+} ions present in the fiber core can be considered as simple systems with only two discrete energy levels. However, in this case, a three-level model is presented, which is also used in the section on noise sources. A representation of those levels and the energy transitions of electrons are reported in Figure 2.2. The levels, labeled s_1, s_2, and s_3 correspond to the energy levels $I_{15/2}$, $I_{13/2}$, and $I_{11/2}$ of erbium [31]. A fourth energy level, higher than s_3, could be added to model signal absorption due to photons in s_3. However, this contribution is neglected in the present analysis. It is important to note that energy levels are not single discrete values, but rather very dense ranges of multiple values. Consequently, transitions involving photons of similar frequency (and, therefore, energy) occur between the same two levels.

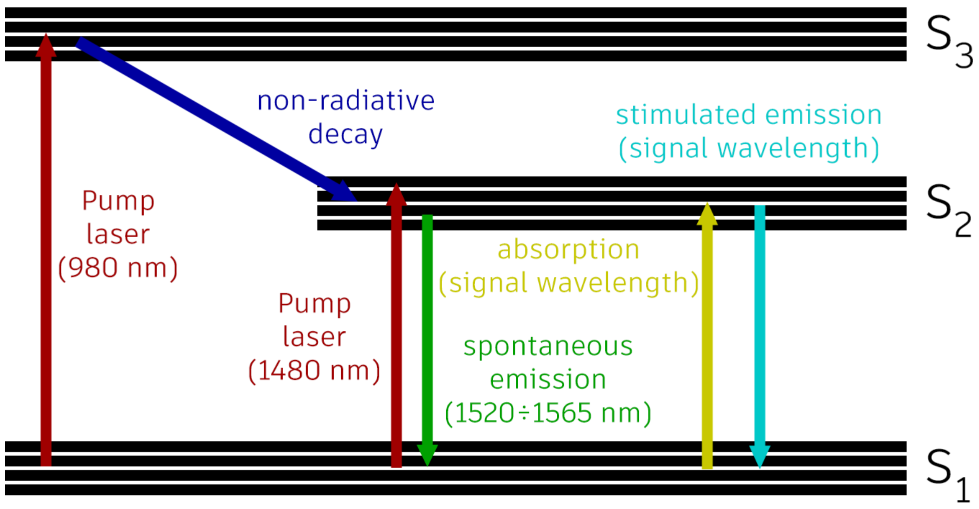

S3
non-radiative decay
stimulated emission (signal wavelength)
S2
Pump laser (980 nm)
absorption (signal wavelength)
Pump laser (1480 nm)
spontaneous emission (1520÷1565 nm)
S1

Figure 2.2: The three-bands model of Er^{3+} ions energy levels.

The propagation of the 980 nm pump laser light in the fiber excites erbium ions from the low energy state s_1 to the pump level state s_3. Ions in s_3 have a very short lifetime and thus immediately relax to a metastable state, s_2, releasing photons in a non-radiative transition. When a 1480 nm laser is used, the ions are directly excited to s_2. This process creates a population inversion between the low energy state and the metastable state, enabling amplification of the input signal power in the range of 1520 to 1565 nm. Signal photons can also be absorbed by the material, and spontaneous emission can occur. Both these effects happen between s_1 and s_2, if we neglect the transition of an excited electron from s_3 to a fourth level s_4 due to absorption.

EDFAs can be used in three different operating modes [3]:

1. **Automatic Gain Control (AGC)**, the most commonly used, which regulates the laser source current in order to achieve a target gain;

2. **Automatic Power Control (APC)**, which works the same way but the target is an output power;

3. **Automatic Current Control (ACC)**, which delivers a constant current to the laser source.

Commercial EDFAs allow to set the target gain, usually expressed in dB, and the target tilt, which indicates the slope of the peak power of the active channels in the output power spectrum and is usually evaluated as the difference between the power levels of the first and last channel inside the bandwidth of interest (usually the optical C-band or L-band). This quantity is also expressed in dB.

2.1.1 Noise sources in amplifiers

Multiple sources contribute to the noise generated and propagated inside an optical amplifier [32]. The main elements are:

1. **Signal-spontaneous (sig-sp) beating:** originates from the mixing of the coherent signal with the incoherent Amplified Spontaneous Emission (ASE) in the same polarization. It exhibits a flat frequency distribution, proportional in amplitude to gain and input signal power.

2. **Spontaneous-spontaneous (sp-sp) beating:** similar to sig-sp beating, but involves copolarized spectral components of ASE. It is independent of

input signal power and gain, and is more pronounced around the central frequency.

3. **Multipath interference (MPI):** phase or frequency noise already affecting the input signal is converted to intensity noise in the output signal. It has low bandwidth and increases with input power and gain.

4. **shot noise:** random fluctuation in the photo-generated current in the photodetector, due to random and independent carrier generation. It is frequency-independent, resembling white noise, and is proportional to the average photo-generated current.

For simplicity, in this analysis, we will refer to the signal-spontaneous (sig-sp) beating contribution as 'ASE noise'.
The most significant noise contribution in EDFAs is Amplified Spontaneous Emission (ASE), as both signal-spontaneous (sig-sp) and spontaneous-spontaneous (sp-sp) beating depend on it, with the former generally being dominant. ASE occurs when excited erbium ions decay to the low-energy state before interacting with photons from the input signal. This results in the spontaneous emission of photons with random phase and direction. Some of these spontaneously emitted photons are aligned with the propagation axis. Consequently, these random-phase photons mix with the input signal and are amplified alongside it, producing a noise component proportional to the amplifier's gain. Equations describing ASE noise can be derived using formulae for carrier populations within energy levels and for pump and signal powers. The parameters for these equations are obtained from complex experimental measurements. However, these parameters are subject to change (e.g., due to aging), making a one-time measurement unreliable over long periods. Consequently, periodic measurements are required to maintain accurate parameter values, which is both expensive and impractical for devices in use. For this reason, alternative, simpler models for ASE noise and other parameters are often preferred.

2.1.2 Noise Figure of EDFAs

EDFAs introduce unwanted optical power fluctuations to the amplified output signal, with ASE noise being the most dominant contribution, as described in the previous section. This undesired addition reduces the Optical Signal-to-Noise Ratio (OSNR) of the transmitted signal, making its measurement crucial. To characterize this effect, a parameter known as the noise figure is introduced, analogous to the approach used with electronic amplifiers [31]. The noise figure NF is defined as the noise factor F expressed in dB units.

$$NF = 10\, log_{10}\, F \tag{2.1}$$

The noise factor F is the ratio of the signal-to-noise ratio (SNR) values at the input and output of the amplifier. It is a function of the optical frequency ν and the baseband frequency f of the electrical output from the photodetector employed in the receiver.

$$F\left(\nu, f\right) = \frac{SNR_{IN}}{SNR_{OUT}\left(\nu, f\right)} \tag{2.2}$$

Essentially, the noise figure indicates the degradation of signal quality due to the insertion of the amplifier into the optical line. For a long time, from the 1980s up to the beginning of the 2000s, the noise figure of a high-gain phase-insensitive linear amplifier was thought to have a lower bound for its noise figure of 3 dB [33], due to reasons related to the Heisenberg uncertainty principle. While it is now thought not to be a fundamental limit for optical amplifiers [34], the value of 3 dB still holds as a very good, near-perfect result to achieve, and researchers are trying to overcome this limit [35]. The best result that was obtained in the measurements used in this work is around 3.7 dB.

To evaluate the noise figure, we consider a setup as shown in Figure 2.3. The receiver is assumed to be ideal, and the laser source is shot noise limited, meaning it has a high SNR and that the primary contribution to source noise is shot noise.

The SNR for a shot noise-limited source is described by the following formula:

$$SNR_{IN} = \frac{\eta P_{IN}}{2h\nu B_e} \tag{2.3}$$

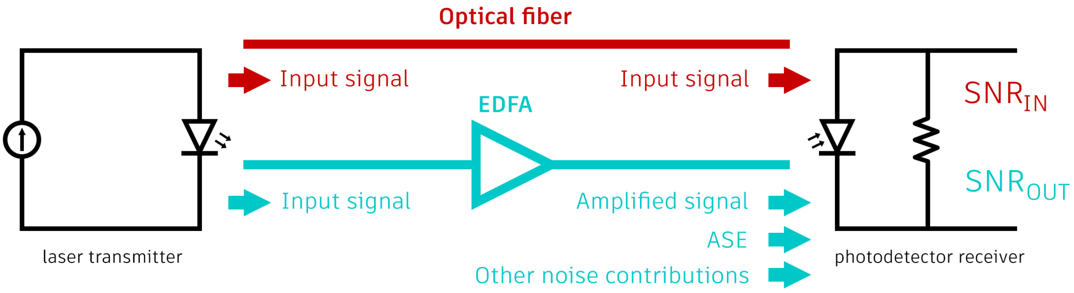


Figure 2.3: Input and output SNR measurement setups.

where η is the quantum efficiency of the photodetector, h is Planck's constant expressed in $\mathrm{J \cdot s}$ and B_e is the noise bandwidth, in Hz. In the case of an ideal receiver, η equals 1.

The output SNR of the amplifier can be written as:

$$SNR_{OUT} = \frac{G^2 P_{IN}^2}{B_e \left[S_e (\nu, f) + \eta^{-1} S_{shot} \right]} \tag{2.4}$$

where the noise spectral density S_e, expressed in $\mathrm{W^2\,Hz^{-1}}$, takes into account all noise contributions except shot noise. The formula for the noise factor is then:

$$F(\nu, f) = \frac{S_{total}(\nu, f)}{2hfG^2 P_{IN}} \tag{2.5}$$

The spectral density of signal-spontaneous beating noise and shot noise can be expressed as:

$$S_{sig-sp} = 4\rho_{ASE} G P_{IN}, \; S_{shot} = 2hfGP_{IN} \tag{2.6}$$

where ρ_{ASE} is the ASE power density. All other noise contributions can be neglected if the input channel power is above the effective input noise of the amplifier (typically around $-50\,\mathrm{dBm}$) [31].

The formula for the noise factor can now be rewritten as:

$$F = \frac{4\rho_{ASE} G P_{IN} + 2hfGP_{IN}}{2hfG^2 P_{IN}} = \frac{2\rho_{ASE}}{hfG} + \frac{1}{G} \tag{2.7}$$

The term $1/G$, corresponding to the shot noise, can usually be neglected. The signal-spontaneous contribution typically ranges from 3 to 10, while $1/G$ is in the order of 10^{-2} or less, since G is typically above $15\,\mathrm{dBm}$.

Finally, the expression can be rewritten by replacing ρ_{ASE} with the corresponding power, a metric more commonly obtained from spectrum analyzer measurements. This is achieved by multiplying the ASE noise power density by the corresponding noise bandwidth, B_0. The final expression for the noise factor is:

$$F = \frac{2\rho_{ASE}B_0}{hfGB_0} = \frac{P_{ASE_{AMP}}}{hfGB_0} \tag{2.8}$$

The term $P_{ASE_{AMP}}$ represents the ASE power introduced by the amplifier. It is calculated by measuring the ASE power at the output of the amplifier and subtracting the Source Spontaneous Emission (SSE) power multiplied by the gain of the amplifier (in linear units). This procedure aims to exclude the contribution of pre-existing signal noise, which is also amplified. Figure 2.4 illustrates this process.

The final expression for the noise figure is:

$$NF\,|_{dB} = P_{ASE_{AMP}}\,|_{dBm} - G\,|_{dB} - 10\,log_{10}\,(hfB_0) \tag{2.9}$$

Note that, in this expression, Planck's constant is expressed in mJ · s instead of J · s to match the unit of $P_{ASE_{AMP}}$, which is expressed in dBm. It should also be noted that the formula explicitly depends on the ASE noise power and therefore on the input signal power, the amplifier gain, and the channel frequency.

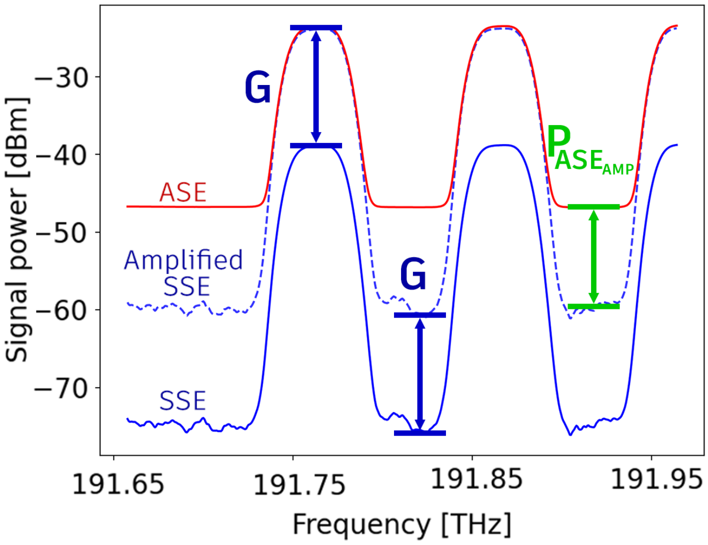


Figure 2.4: Graphical example of the meaning of $P_{ASE_{AMP}}$. The output spectrum (red) is compared with the input spectrum (blue) increased by the amplifier gain (blue dashed). $P_{ASE_{AMP}}$ is the amount of noise added to the amplified SSE level by the EDFA.

2.2 Comparison of different EDFA models

Estimating the parameters of network elements, such as EDFAs, using specialized models is crucial for the optimal design, operation, and maintenance of optical communication systems. This estimation allows for accurate performance predictions of network elements under various conditions without the need to deploy the network initially, thereby greatly aiding in network design. Various models for EDFAs have been proposed in numerous articles [7] [8]. Most of the early models focus on gain estimation, which is notoriously difficult to analyze due to two primary factors: static-state inter-channel gain variation and dynamic gain excursion [31]. The static-state variation arises from uneven gain across channels due to the dependence of the fiber's absorption and gain spectra on the signal wavelength. Dynamic gain excursion refers to rapid gain fluctuations in response to changes in input power or channel configurations, which are common in optical networks [3].

Analytical models

It is possible to derive formulas for the theoretical estimation of EDFA parameters, such as gain and noise figure. By starting from the fundamental physical equations, such as those governing the carrier populations in the energy levels of erbium ions, a comprehensive analytical expression for these parameters can be obtained. The main advantage of analytical models is that they depend solely on a-priori knowledge of the equipment and a restricted amount of measurements, with values possibly provided by manufacturers, and do not require any deployment or input-output characterization. While this is the simplest method, it does come with several challenges. Firstly, these models are typically very complex and rely on several assumptions, such as the two-energy-level approximation for erbium ions, which can compromise their accuracy. Additionally, they do not consider dynamic effects, such as those previously mentioned for gain estimation, nor manufacturing defects. Another issue is the inconsistency between different devices. EDFAs can be constructed with various architectures and include different components in their schematics. Consequently, an analytical model based on the underlying physical equations must be adapted for each device architecture, making it impractical to apply the same model to different brands and devices. This issue is directly related to another challenge with commercial devices: manufacturers often keep the details of their device architectures confidential, making it difficult to develop accurate models for consumers. Some simpler analytical models for gain estimation are available, such as those based on a center-of-mass function, which enable evaluation of gain in Automatic Gain Control (AGC) mode using a straightforward expression. These models can be easily adapted to multiple devices. However, their accuracy is limited because they overlook various internal effects of the devices. Specifically, they assume an ideally constant gain spectrum shape, which does not hold true for real devices, and they focus on the impact of the loaded channel configuration, disregarding gain variations due to channel power fluctuations. Finally, there is the issue of aging. The physical parameters used in these models change over time, necessitating continuous updates to maintain their reliability. This process can be very costly, as these parameters are determined through complex experimental measurements. While it is possible to incorporate time-dependent factors into the equations, doing so would further increase the already high complexity of the model.

Machine Learning models

With all the problems associated with analytical models of EDFA parameters, research has been undertaken to explore more solutions and technologies to provide better alternatives. The most promising alternatives are machine learning models, especially those based on neural networks. These models offer two major advantages, namely accuracy and flexibility. There are several examples in the literature of models that consistently estimate gain with an RMSE of around 0.15 dB [8], and these models can be easily adapted or extended to many different devices and brands with little modification or training. The primary drawback of these models, and a significant barrier to their practical implementation, is their typically high requirement for large training datasets. Acquiring such datasets is often expensive or impractical. While smaller datasets can be utilized, they pose challenges in ensuring performance reliability, particularly in complex systems.

One of the goals of the model proposed in this thesis is to offer a rapid and adaptable solution capable of being trained with datasets of varying sizes, aiming to strike a balance between achieving target accuracy and minimizing the costs associated with required measurements.

2.3 Model requisites, expectations, limitations

The proposed EDFA model aims to provide an accurate noise figure estimate, but it is not designed to be a stand-alone model. Instead, it is intended to be used as part of a more comprehensive digital twin of the network and to be completely brand independent. Therefore, the model should meet the following requirements:

1. **Accuracy:** the model should achieve a reasonably high level of accuracy, with errors comparable to the uncertainty of measurements in the training datasets;

2. **Computational speed:** it should process data efficiently to avoid becoming a bottleneck when handling large training and test datasets for the complete network model;

3. **Brand independence:** the model should be easily adaptable to multiple brands and devices with minimal modification;

4. **Usability and portability:** while not critical, the model should ideally be user-friendly, intuitive, and portable.

Regarding the input parameters of the model, Equation 2.9 indicates that the noise figure depends on the parameters: $P_{ASE_{AMP}}$, G, and f. The EDFAs employed in the model enable the target gain and tilt to be adjusted in AGC mode. Thus, G, the channel gain, will primarily be a function of these two parameters. $P_{ASE_{AMP}}$ depends not only on the gain but also on the input signal power. For these reasons, the four input parameters chosen for this model are:

1. **input signal power,** in dBm;

2. **gain,** in dB;

3. **tilt,** in dB;

4. **channel frequency,** in THz.

The primary goal of this work is to achieve an accuracy such that the maximum errors in the test datasets are within or below the theoretical uncertainty of the noise figure derived analytically from the measurement datasets.

Chapter 3

Polynomial model construction

After evaluating various potential solutions for implementing the model, the chosen approach was a polynomial structure. This model takes the form of a multivariate polynomial expression, which is efficiently evaluated using Horner's rule, explained in Section 3.4.2. The coefficients are determined through linear regression using the Generalized Least Squares (GLS) algorithm, as explained in Section 3.3.5. While technically categorized as a machine learning model, it differs from traditional ML or neural network-based models in its simplicity, computational speed, and portability.

3.1 Employed tools

The entire project was conducted using the Python programming language within the PyCharm IDE by JetBrains. Python was selected because it is a very powerful open-source solution that provides flexibility, rapid development capabilities, and an extensive array of well-documented, powerful libraries for data manipulation and visualization, including NumPy, Pandas, Matplotlib, and SciPy.

3.2 From measurements to noise figure evaluation

The initial step in creating the model involves evaluating the noise figure from the power spectra of the signal before and after amplification by the EDFA. The procedure for evaluating the noise figure is illustrated by various OSA user manuals [36], and Sections 3.2.4 and 3.2.5 follow those steps. Once the frequencies corresponding to the active channels are isolated, a dataset must be generated. Each row in this dataset provides the noise figure value for a unique combination of four input parameters: input power, target gain, target tilt, and channel central frequency.

3.2.1 Test setup

Figure 3.1 shows the setup used for obtaining the measurement datasets.

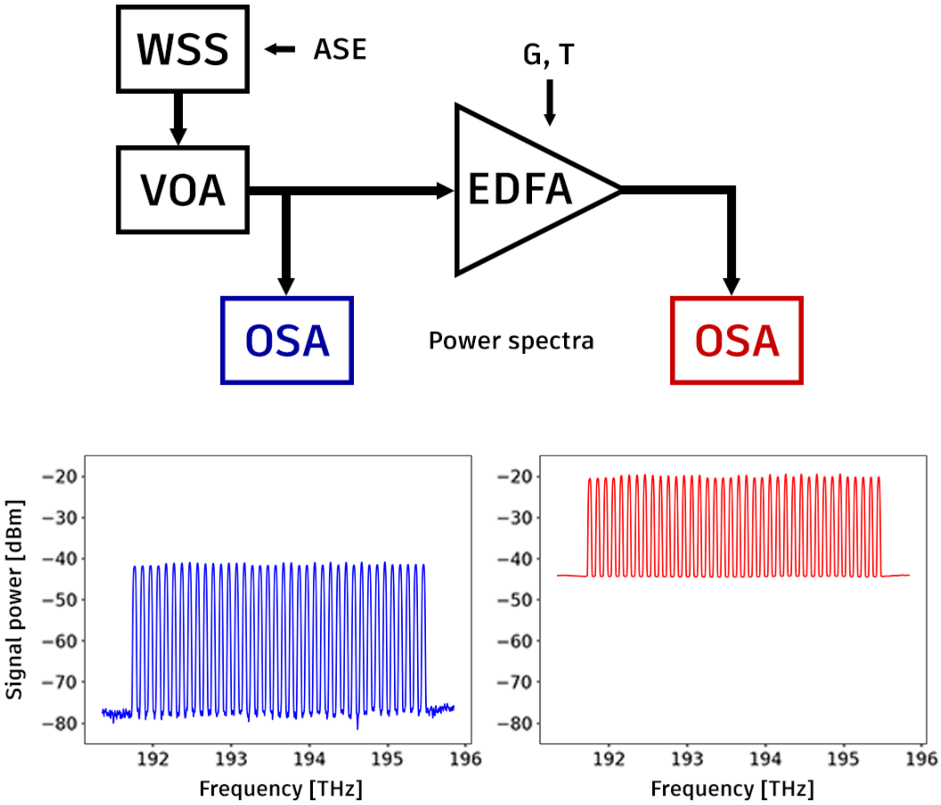


Figure 3.1: Block diagram of the test setup for the measurement of input and output power spectral densities.

This setup allows for the measurement of the input and output power spectral densities (PSD) of the signal passing through the amplifier. The signal is delivered to the amplifier using a variable optical attenuator (VOA) to achieve the desired input power, while the target gain and tilt are set programmatically on the EDFA.

Devices from different vendors, and even different device types from the same vendor, will require their own characterization and final model, because the differences in the internal EDFA architecture will lead to different performance among those devices. Devices of the same type from the same vendor, however, will be grouped together, because the only relevant distinctions among them will be related to manufacturing processes. Most of these devices can operate in high or low gain range, which are two separate modes. It is important to differentiate between them when creating the models, because manufacturers often optimize the noise figure of the two modes independently and therefore their results can be completely uncorrelated with each other.

The characterization depends on the spectral load employed. These measurements were conducted under an even spectral load, ranging from 38 to 48 active channels depending on the device model, with a flat input Wavelength Division Multiplexing (WDM) comb power profile. The channels are spaced at 100 GHz, and each channel has a bandwidth of approximately 30 GHz. The resolution bandwidth (RBW) of the optical spectrum analyzer (OSA) used for all measurements was set to 10 GHz.
For the creation of the datasets, it is crucial to establish reasonable steps for the input parameters. This approach aims to gather comprehensive insights into their impact on the noise figure while ensuring the resulting datasets remain manageable in size. Therefore, the parameters are chosen as follows:

1. Total input power: about $-10\,\text{dBm}$ to $6\,\text{dBm}$ for low gain range, about $-18\,\text{dBm}$ to $2\,\text{dBm}$ for high gain range, with a step of $2\,\text{dB}$.

2. Gain: depending on EDFA model and gain range, values between $10\,\text{dB}$ and $35\,\text{dB}$, with a step of $1\,\text{dB}$.

3. Tilt: between $-5\,\text{dB}$ to $5\,\text{dB}$ or $-3\,\text{dB}$ to $3\,\text{dB}$, depending on EDFA model, with a step of $1\,\text{dB}$.

In section 3.5.6, considerations are made to establish whether these initial assumptions regarding the step sizes for the parameters are reasonable. This analysis will explore whether finer steps are necessary to achieve similar accuracy levels or if larger steps could suffice, potentially reducing the amount of data required.

3.2.2 The measurement datasets

The datasets obtained from the OSA measurements are stored as MATLAB (.mat) files. Each device and input power combination has its own file, structured with the following internal data:

1. **Gain_target:** a 1-D array, which contains all the values of target gain, in dB, that were set on the device;

2. **Gain_real:** a 1-D array, which contains all the values of gain that were actually achieved by the device, in dB (these numbers act as feedback values against the target gain);

3. **Tilt_target**: a 1-D array, which contains all the values of target tilt, in dB, that were set on the device;

4. **Tilt_real:** a 2-D array (one row for each gain value), which contains all the values of tilt that were actually achieved by the device, in dB (these numbers act as feedback values against the target tilt);

5. **spectrum_freq**: a 1-D array, which contains all the frequency points at which the OSA evaluated the signal power, in THz;

6. **spectrum_TX_power:** a 1-D array, which contains all the measurements corresponding to the signal power at the amplifier input over the whole frequency range, in dBm;

7. **spectrum_RX_power:** a 3-D array (one row for each pair of target gain and tilt), which contains all the measurements corresponding to the signal power at the amplifier output over the whole frequency range, in dBm;

8. **TOT_Power_IN:** a floating-point number, indicating the total input power of the amplifier, in dBm;

9. **TOT_Power_OUT:** a 2-D array (one floating-point value for each pair of target gain and tilt), indicating the total output power of the amplifier, in dBm.

10. **OSA_PARAMS:** a data structure containing various sub-parameters, including information on the device vendor, the fiber connector type (flat or angled), the Resolution Bandwidth (RBW) in GHz, the center frequency and span in THz.

3.2.3 Spectrum normalization

An intermediate step involves adjusting the offset of the power spectra. Due to the processing methods of the Optical Spectrum Analyzer (OSA), the total power indicated by TOT_Power_IN and TOT_Power_OUT differs from that evaluated from the corresponding power spectra. Therefore, a normalization factor is required to align these values. This factor can either be additive in decibels (dB) or multiplicative in linear units.

The total signal power, in dBm, can be evaluated from the power spectral density using the following equation:

$$P_{spectrum}|_{\mathrm{dBm}} = 10\, log_{10}\left(\sum_{n=1}^{N} p_n|_{\mathrm{\mu W}} \cdot \frac{\Delta f}{RBW}\right) - 30 \tag{3.1}$$

where p_n is the power spectral density value at each frequency point f_n, RBW is the resolution bandwidth of the spectrum analyzer, and Δf is the frequency step, the spacing between each pair of frequency points, which, if uniform, is simply equal to:

$$\Delta f|_{\mathrm{GHz}} = \frac{f_N - f_1}{N} \cdot 1000 \tag{3.2}$$

where the factor 1000 is used to convert it from THz to GHz, which is the same unit as the RBW. The factor of -30 in equation 3.1 is used to express P_{signal} in dBm.

After evaluating the total signal power from the spectrum, the offset can be obtained with the formula:

$$P_{offset}|_{\mathrm{dB}} = P_{tot}|_{\mathrm{dBm}} - P_{spectrum}|_{\mathrm{dBm}} \tag{3.3}$$

where P_{tot} represents the signal power indicated in the measurements (stored in the parameters named TOT_Power_IN and TOT_Power_OUT).

Finally, the offset is applied to each element of the *spectrum_TX_power* and *spectrum_RX_power* arrays. It is important to note that there is one offset for the input power spectrum, whereas for the output power, there exists a distinct offset for each unique pair of target gain and tilt.

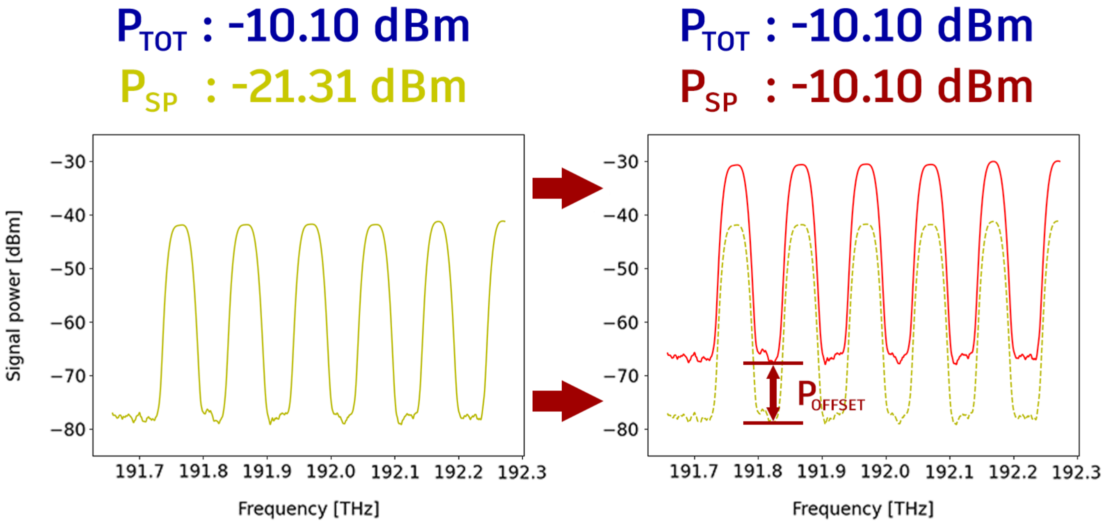


Figure 3.2: Example of spectrum normalization with an offset.

3.2.4 Locating the active channels

The initial step in manipulating the measurement data involves determining the central frequency of each active channel in the spectrum. This is achieved using a filtered copy of the power spectrum to reduce data fluctuations and exclude erroneous samples. The filtering process combines clamping the spectrum values within a reasonable range to exclude outliers and convolving it with a constant function. The convolution parameters required fine-tuning to ensure suitability across all devices. The central frequency of the active channels is determined using the *find_peaks* function provided in *SciPy*, which identifies the indices of all the local maxima within the spectrum. To mitigate issues stemming from slight variations in channel central frequencies, they are determined solely from the input power spectrum, and these values are subsequently applied to the output power spectra. While it's feasible to utilize a-priori knowledge regarding channel central frequencies — typically available in test setups — this program prioritizes flexibility.

Thus, it aims to operate with minimal reliance on external information. Therefore, the program does not require any input regarding the number of active channels or their central frequencies.

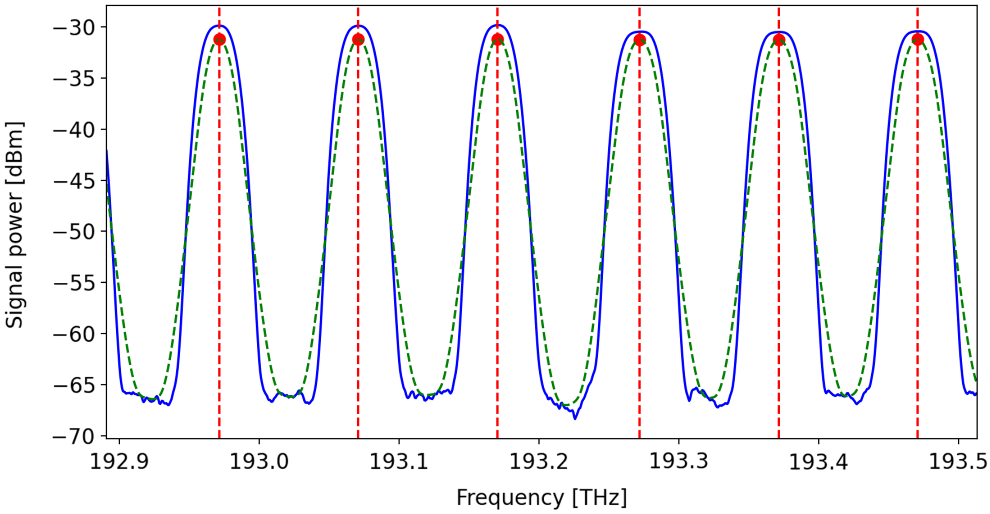


Figure 3.3: Visualization of the process of finding the central frequency of active channels. The power density spectrum (blue) is convoluted with a constant function to obtain a smooth curve (green). Then, the *find_peaks* function is used on this curve to obtain the points corresponding to the channel peaks (red). The difference between red and blue curves on the y-axis is not a concern, as this method only aims at finding the correct values on the x-axis.

3.2.5 Evaluating SSE and ASE

Source Spontaneous Emission (SSE) and Amplified Spontaneous Emission (ASE) are critical parameters for noise figure evaluation, representing the noise floor in the input and output power spectra, respectively. While named after spontaneous emission, these values encompass other minor noise contributions, which are disregarded for simplicity. The process of determining the noise floor level is illustrated in Figure 3.4. The approach involves identifying points between each pair of adjacent channels where the power value equals the noise floor. Subsequently, the noise values at frequencies corresponding to the center frequencies of active channels are derived through linear interpolation. For the first and last channels, the first and last frequency points are used accordingly.

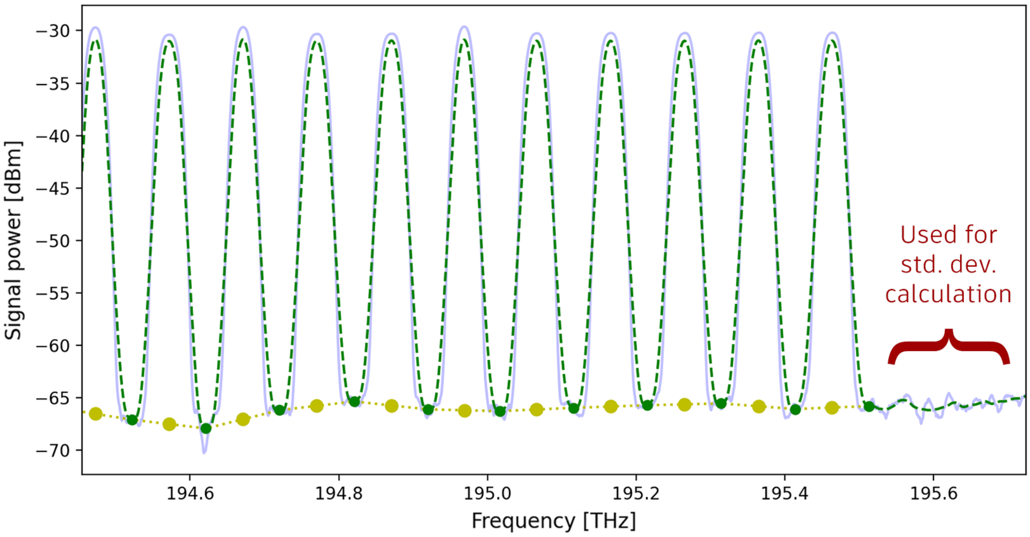


Figure 3.4: Evaluation of SSE (yellow dots). The power density spectrum is convoluted to obtain a smooth curve (green line). The middle points between two adjacent channels (green dots) are used for the piecewise linear interpolation, to evaluate the noise at the channel frequencies (yellow dots).

In many datasets, the input power values exhibit a noisy, wavy curve for the noise floor. Consequently, an additional convolution step is performed. This second convolution necessitates fine-tuning of the parameters, which depend on factors such as the number of channels and the standard deviation of the noise level at the band edges. SSE and ASE are used to compute the $P_{ASE_{AMP}}$ parameter shown in equation 2.9. This value is equal to the difference between the ASE noise and the amplified SSE noise, and can be evaluated as:

$$P_{ASE_{AMP}}|_{\mathrm{dBm}} = 10\,log_{10}\left(10^{\frac{P_{ASE,\mathrm{dBm}}}{10}} - 10^{\frac{P_{SSE,\mathrm{dBm}}+G}{10}}\right) \tag{3.4}$$

This process is the mathematical formulation of the quantity shown in Figure 2.4.

3.2.6 Creating the noise figure dataset

After obtaining all the necessary parameters, a dataset is generated for each device. Additionally, two separate datasets are created for the low and high gain ranges. This differentiation is made because, as previously discussed, the EDFAs analyzed in this study are designed to optimize noise figure independently in each range, resulting in distinct behaviors. Each dataset includes the evaluated noise figure of the device for every unique combination of total input power, target gain and tilt, and channel frequency. These datasets will be utilized for the polynomial regression. Each dataset comprises tens of thousands of entries, with a total storage size of a few megabytes each. Every dataset is saved twice, in separate files. One is used for efficient storage and I/O operations, while the second is in a human-readable format.

3.3 Polynomial regression

In the previous section, the noise figure as a function of the four input parameters was evaluated. In this section it is explored how the noise figure varies with respect to each parameter individually, aiming to justify the use of a polynomial structure. Following this analysis, an appropriate degree for each of the four polynomials will be estimated. The coefficients will be determined using the Generalized Least Squares (GLS) algorithm. Each model is constructed by combining datasets from multiple devices of the same brand, type, and gain range.

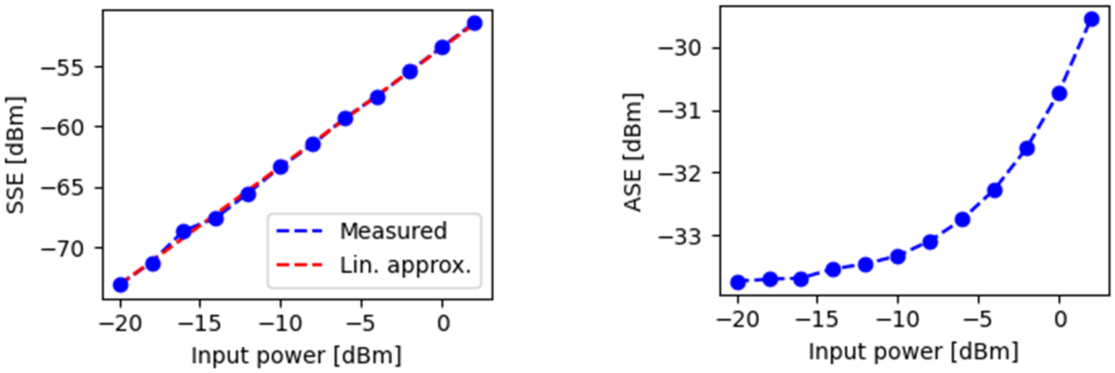
3.3.1 Noise figure and input power

From the formula shown in equation 2.9, it is evident that the noise figure is primarily influenced by three variables $P_{ASE_{AMP}}$, G and f. The input power predominantly affects the noise levels, including ASE and SSE values, while channel gain is less affected by this parameter. SSE corresponds to the noise affecting the input signal before passing through the amplifier. Figure 3.5a illustrates that the SSE value, measured in dBm, is a linear function of the input power, This linearity is anticipated under the assumption that all measurements were conducted using a test signal with a consistent SNR, which remains stable throughout all the measurements.

ASE, on the other hand, behaves differently. At higher input powers, it follows a linear trend similar to SSE. However, at lower powers, it saturates to a minimum floor. This behavior is attributed to additional noise contributions that are independent of the input power, as well as the RBW of the OSA used for the measurements. This characteristic shape can be observed in Figure 3.5b.

Finally, the behavior of the parameter $P_{ASE_{AMP}}$, which combines the other two noise contributions, is depicted in Figure 3.6. According to equation 3.4, at low input powers, the amplified SSE noise is approximately 12 dB lower than ASE, resulting in $P_{ASE_{AMP}}$ being nearly equal to the ASE level. As the input power increases, the difference between the two parameters diminishes, causing the SSE contribution to become more significant in the formula.

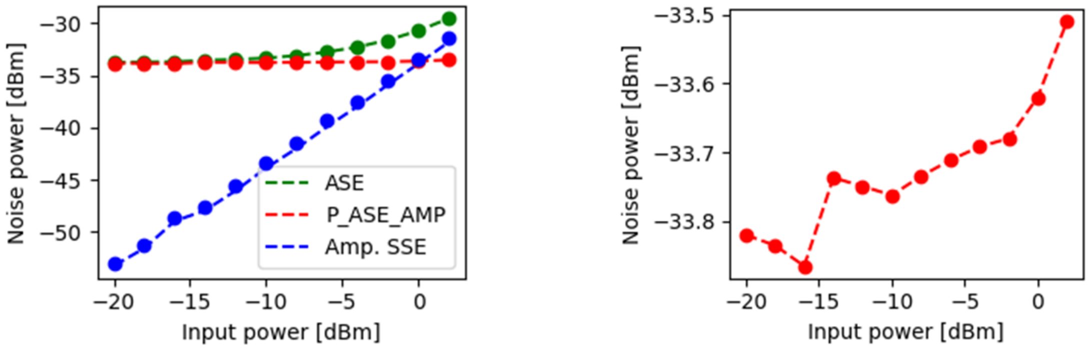
The curve of $P_{ASE_{AMP}}$ remains notably stable with respect to input power, showing a maximum excursion of around 0.5 dB across the entire input power range. Figure 3.7 presents the noise figure results as a function of input power for various



(a) (b)

Figure 3.5: Noise levels vs. input poewr for a Cisco EDFA-35 device. (a) SSE; (b) ASE.

values of target gain, tilt, and frequency (or channel number). It is evident that the noise figure generally slightly increases with input power when a low target gain is configured, but this trend is reversed with high gain settings. This shift can be attributed to the increased influence of SSE on $P_{ASE_{AMP}}$ at higher gain, resulting in a negative trend due to the steeper slope of SSE compared to that of ASE. For low gain, in contrast, $P_{ASE_{AMP}}$ closely follows the positive trend of ASE because the contribution of SSE is minimal. A 4th degree polynomial is considered suitable for modeling the fluctuations of noise figure with input power.



(a) (b)

Figure 3.6: $P_{ASE_{AMP}}$ vs. input power for a Cisco EDFA-35 device. (a) $P_{ASE_{AMP}}$, ASE and amplified SSE; (b) $P_{ASE_{AMP}}$ only.

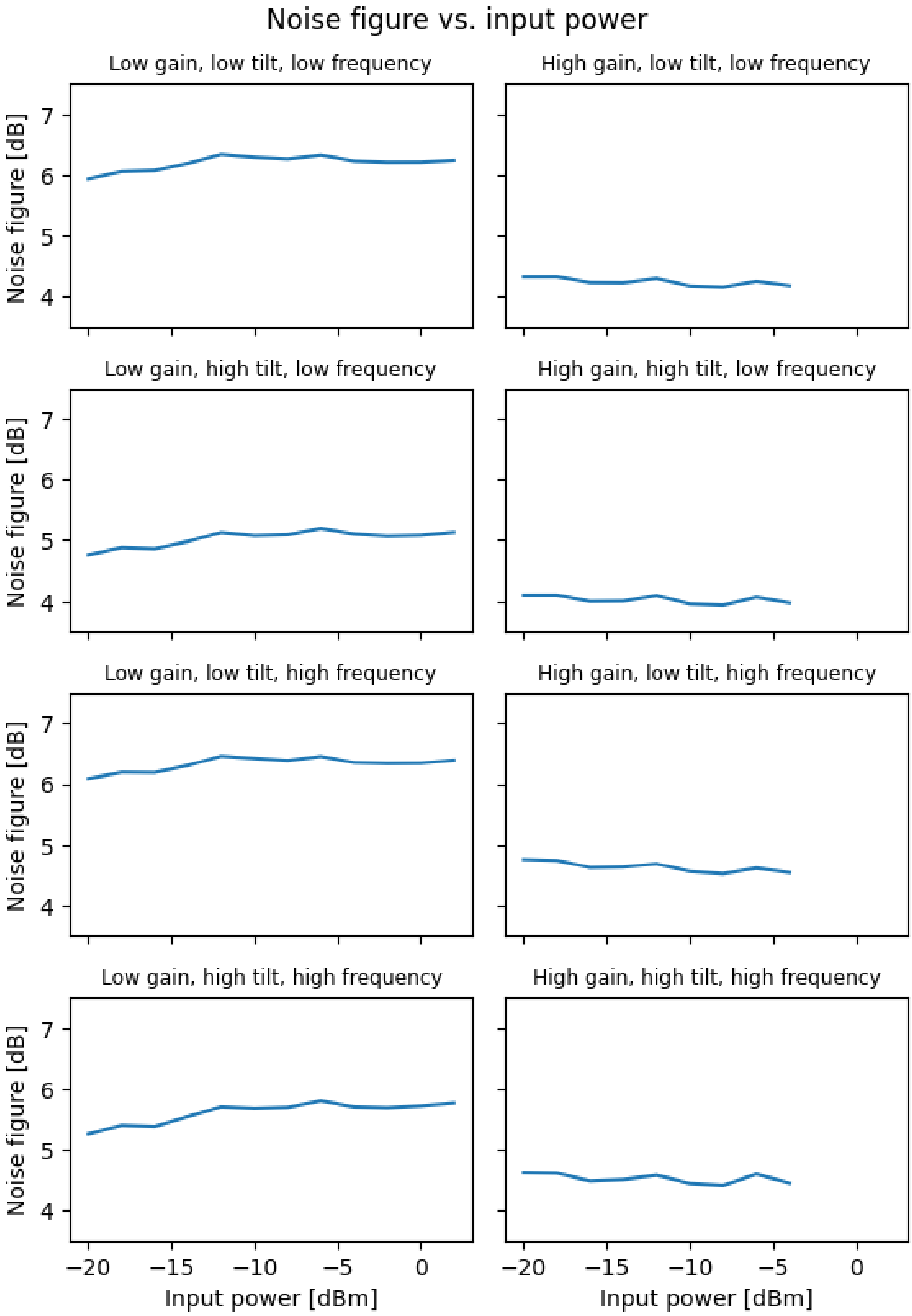


Figure 3.7: NF vs. input power, various input parameters configurations.

3.3.2 Noise figure and gain

The target gain is the central parameter of an EDFA. As evident from equations 2.9 and 3.4, channel gain significantly impacts the noise figure. It appears explicitly in the formulas and also influences the value of $P_{ASE_{AMP}}$. Specifically, SSE remains constant with respect to channel gain since it depends solely on input signal parameters and not on the amplifier's characteristics. In contrast, ASE noise demonstrates a linear increase (in dBm units, which implies an exponential growth) with respect to gain, as expected.

This behavior is illustrated in Figure 3.8, which also includes a linear fit of the curve.

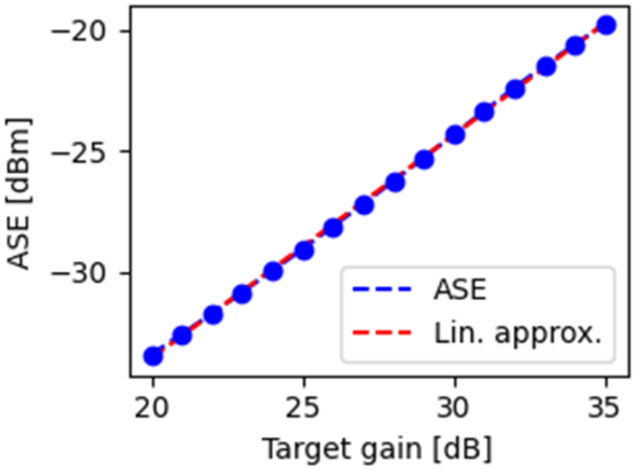
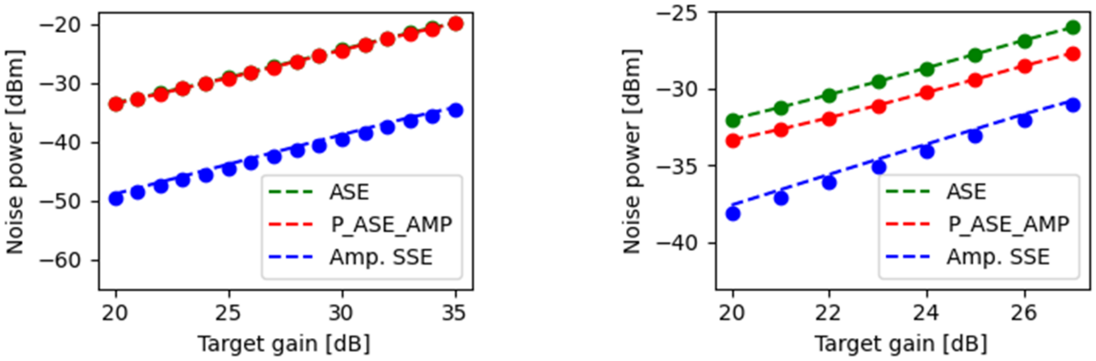


Figure 3.8: ASE vs. gain, with linear fitting.

In Figure 3.9, it is possible to observe that both ASE and amplified SSE exhibit linear behaviors with respect to gain, albeit with similar slopes. Consequently, $P_{ASE_{AMP}}$ also demonstrates a linear behavior, with values lying between ASE and amplified SSE. At lower input power levels, the curve closely aligns with ASE, reflecting the significantly lower level of amplified SSE. As input power increases, the impact of $P_{ASE_{AMP}}$ becomes more pronounced.

In all test cases, $P_{ASE_{AMP}}$ clearly increases with channel gain, with its linear approximation slope always below 1, approaching this limit as gain increases. Consequently, the difference $P_{ASE_{AMP}} - G$, shown in equation 2.9, decreases as channel gain rises. Since the third term in the equation is constant with respect to gain (depending only on frequency), we can infer that the noise figure will follow a similar



(a) (b)

Figure 3.9: $P_{ASE_{AMP}}$ vs. channel gain. (a) low input power; (b) high input power.

trend, exhibiting lower values at higher target gains. These results are illustrated in Figure 3.10. The noise figure shows a steeper decline for lower gains because $P_{ASE_{AMP}}$ varies less with gain, but the slope diminishes as gain increases. A third-degree polynomial adequately fits the tested curves, yielding a very low Root Mean Square Error (RMSE).

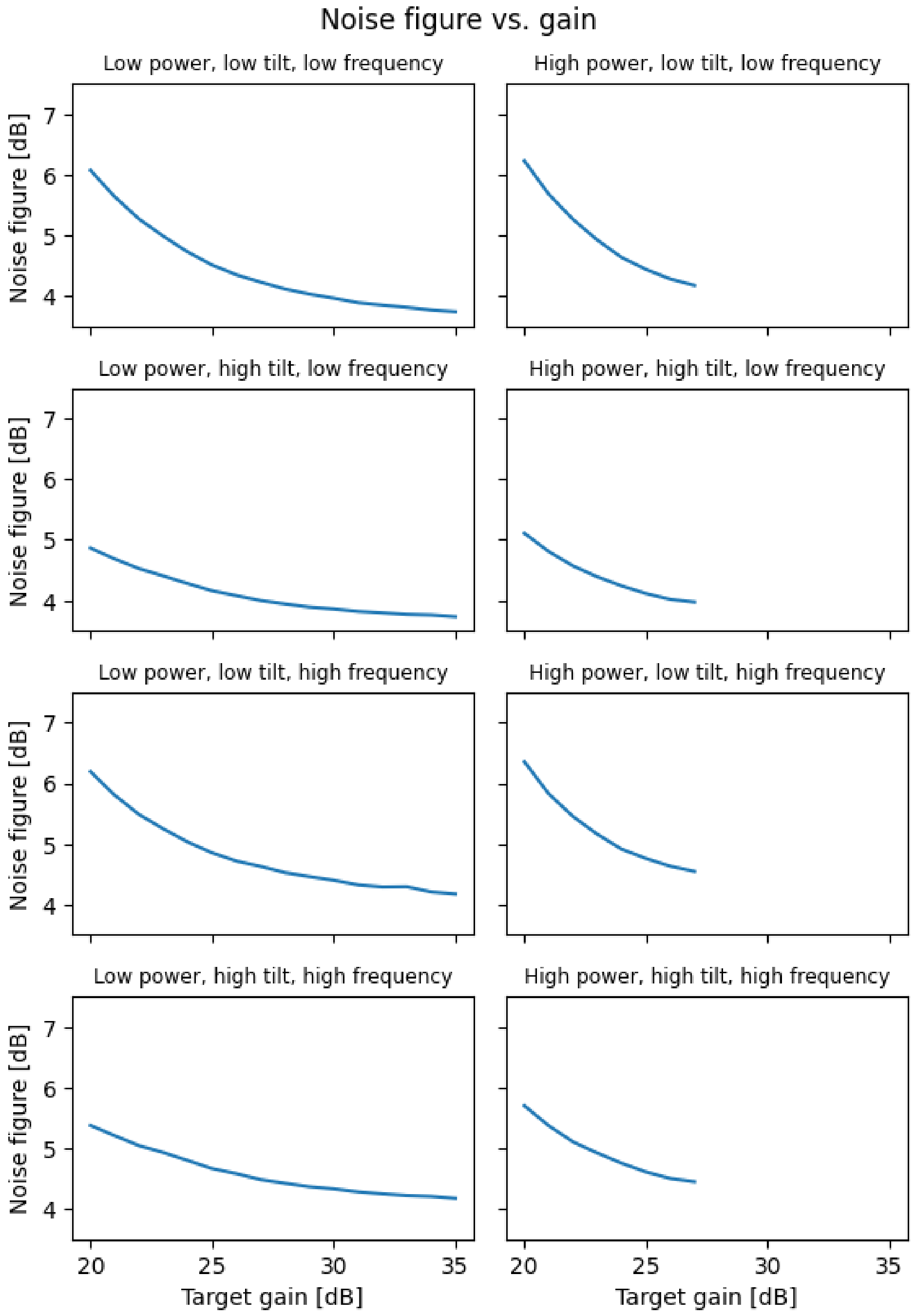


Figure 3.10: NF vs. channel gain, various input parameters configurations.

3.3.3 Noise figure and tilt

In the context of EDFAs, 'tilt' refers to the gradient of the gain curve, with respect to frequency. This parameter can be adjusted to counteract wavelength-dependent variations in gain, which lead to varying power levels among channels propagating through the fiber at different frequencies. Several factors contribute to this phenomenon, including the amplifier's finite gain bandwidth and potential gain saturation. Tilt control provides a means to shape the amplifier's gain profile, thereby mitigating these undesired effects and ensuring more uniform amplification across the range of frequencies. Gain in an EDFA can be modeled as follows [9]:

$$G(f) = G + \frac{T}{B}(f - f_0) + g(f) \tag{3.5}$$

where G is the target gain, T the target tilt, B the amplifier bandwidth, f_0 the center frequency and $g(f)$ the gain ripple. This model is represented in Figure 3.11 [9].

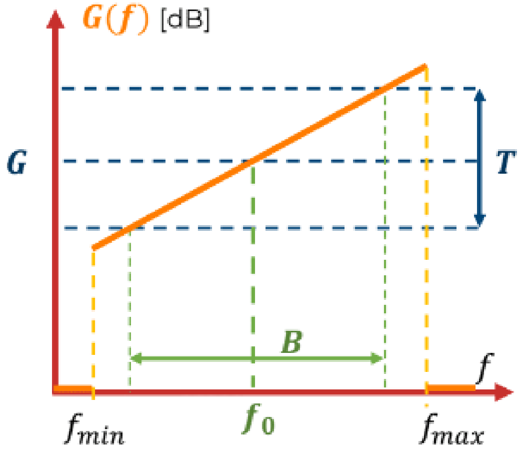


Figure 3.11: Gain model of the EDFA, including tilt and ripple [9].

Tilt primarily influences channel gain and its impact varies with frequency. Consequently, different outcomes for cases involving low and high channel indices can be anticipated.

Similarly to what was concluded in the preceding section, the results illustrated in Figure 3.12 demonstrate that overall the noise figure tends to decrease with increasing tilt. This effect is more pronounced at lower target gains, while it diminishes at higher gains. In the previous section, it was observed that higher gain generally corresponds to a lower noise figure. For lower channel frequencies, gain increases with increasing tilt, whereas for higher channel frequencies, the opposite trend is observed. Typically, the target gain value is achieved with tilt values close to 0, as expected, or slightly negative at most. The same trend also applies to $P_{ASE_{AMP}}$. The variation in channel gain due to tilt remains consistent across different input powers, target gains, and channel frequencies. However, $P_{ASE_{AMP}}$ exhibits a more negative slope with respect to tilt when the target gain is low. This results in minimal variation for low frequencies, where the positive slope gets flattened, and a more pronounced negative variation for high frequencies, where the slope would already be negative. Therefore, the difference with gain is more distinct at low gains, whereas the contributions tend to balance each other more at higher gains. A third-degree polynomial was found to be a suitable fitting function for the target tilt dependency of the noise figure.

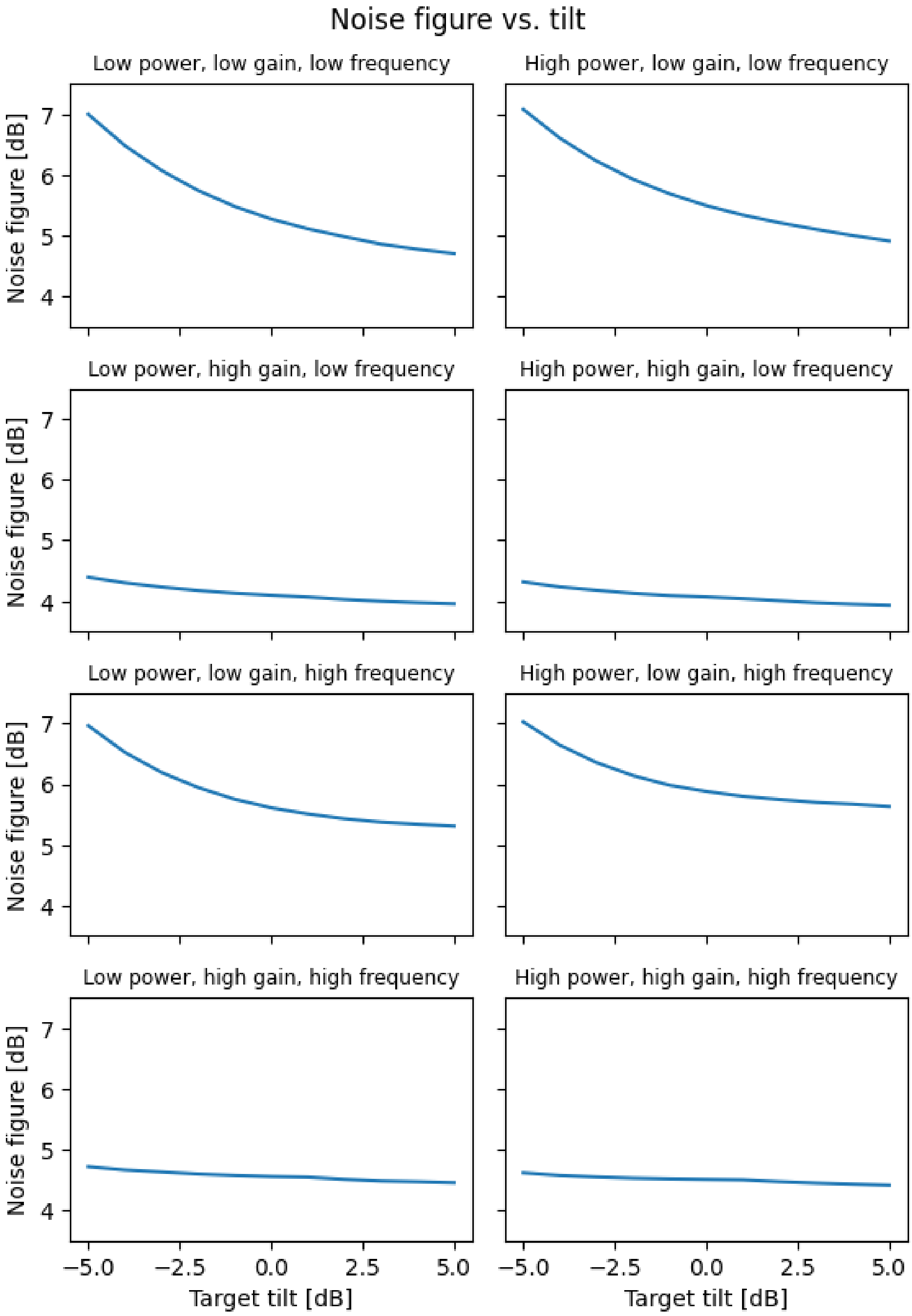


Figure 3.12: NF vs. target tilt, various input parameters configurations.

3.3.4 Noise figure and frequency

Finally, the impact of channel frequency on the noise figure is assessed. As shown in Equation 2.9, frequency appears explicitly as a term affecting the noise figure. Additionally, from Equation 3.5, it is evident that frequency influences channel gain.

A distinction from other cases is the term $10\,log_{10}\,(hfB_0)$, which was previously a constant but now represents an explicit dependency on frequency. However, this term exhibits nearly negligible variations relative to frequency. If we evaluate the maximum excursion of this term in the frequency range of interest with a differential approximation:

$$\Delta\,NF = 10\,log_{10}\,(hf_{MAX}\,B_0) - 10\,log_{10}\,(hf_{MIN}\,B_0) \tag{3.6}$$

$$\Delta\,NF = \Delta\,f \cdot \frac{\delta}{\delta f}(10\,log_{10}\,(hf_{MID}\,B_0)),\ f_{MID} = \frac{f_{MIN}\,+f_{MAX}}{2} \tag{3.7}$$

$$\Delta\,NF = (f_{MAX} - f_{MIN})\frac{10}{f_{MID}\,ln(10)} \tag{3.8}$$

Substituting values corresponding to the C-band for f_{MIN} and f_{MAX},the resultant change is about 0.08 dB. Given that the noise figure typically has a value above 4 dB, this corresponds to a relative contribution of most 2%. Therefore, this term alone does not significantly influence the trend of noise figure as a function of frequency.

As previously discussed, channel gain decreases with increasing channel frequency when the target tilt is positive, and the opposite occurs with negative tilt. A similar trend applies to ASE noise, which, for positive tilts, exhibits a "plateau" in the middle and varies more at the edges of the band. SSE is noisier at lower powers, but this is most likely due to the sensitivity of the instrument. $P_{ASE_{AMP}}$ follows a trend similar to channel gain, so the resulting noise figure will depend on the actual slopes and variations of each parameter.

Results are summarized in Figure 3.14. It is immediately noticeable that, compared to other parameters, the variation of the noise figure with frequency exhibits significantly greater variance, making it more challenging to fit into a polynomial curve. This behavior is detailed in Figure 3.13. A helpful aspect is that, unlike other mathematical models, overfitting is not a concern because the C-band remains constant, so it is unnecessary for the model to make accurate predictions outside the tested frequency range. Thus, increasing the degree of the polynomial to achieve higher accuracy is a viable strategy. The only caveat is that, by raising the polynomial's degree, we must keep the degrees of other polynomials low to maintain a small dataset size and therefore a short training time for our models. An 8th-degree polynomial was found to provide the optimal balance among training speed, numerical precision, and model accuracy.

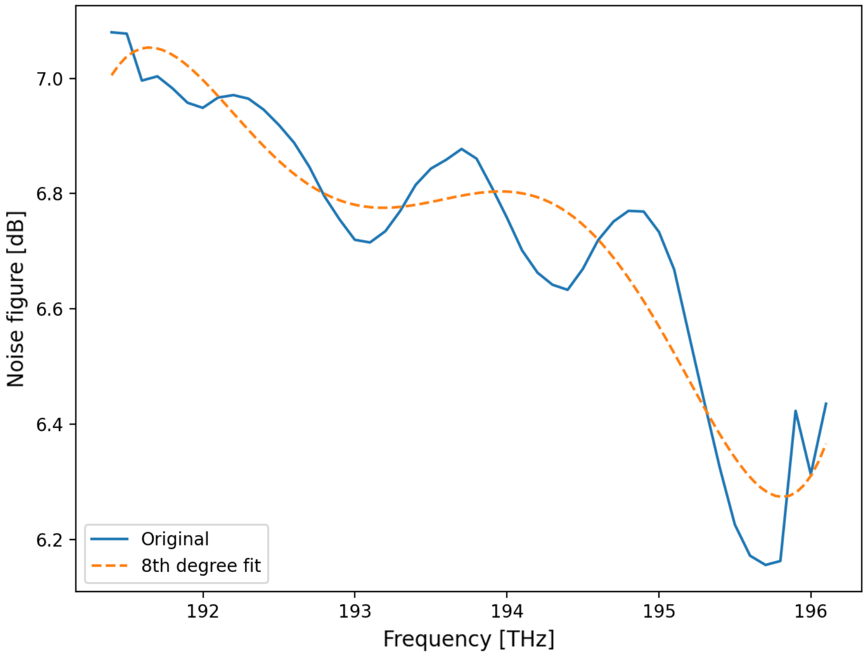


Figure 3.13: Example of the high variance of noise figure vs. frequency under a worst-case scenario, for a Juniper device. The curve is hard to fit effectively, even with high order polynomials, but luckily only few cases exhibit a similar behavior.

3.3.5 The GLS regression

The first part of the analysis involved obtaining various datasets for each device within each gain range. Each dataset contains the noise figure values for every unique combination of input power, target gain, tilt, and channel frequency. The current objective is to build the mathematical model that describes each type of EDFA under analysis.

The polynomial model can be mathematically expressed as follows:

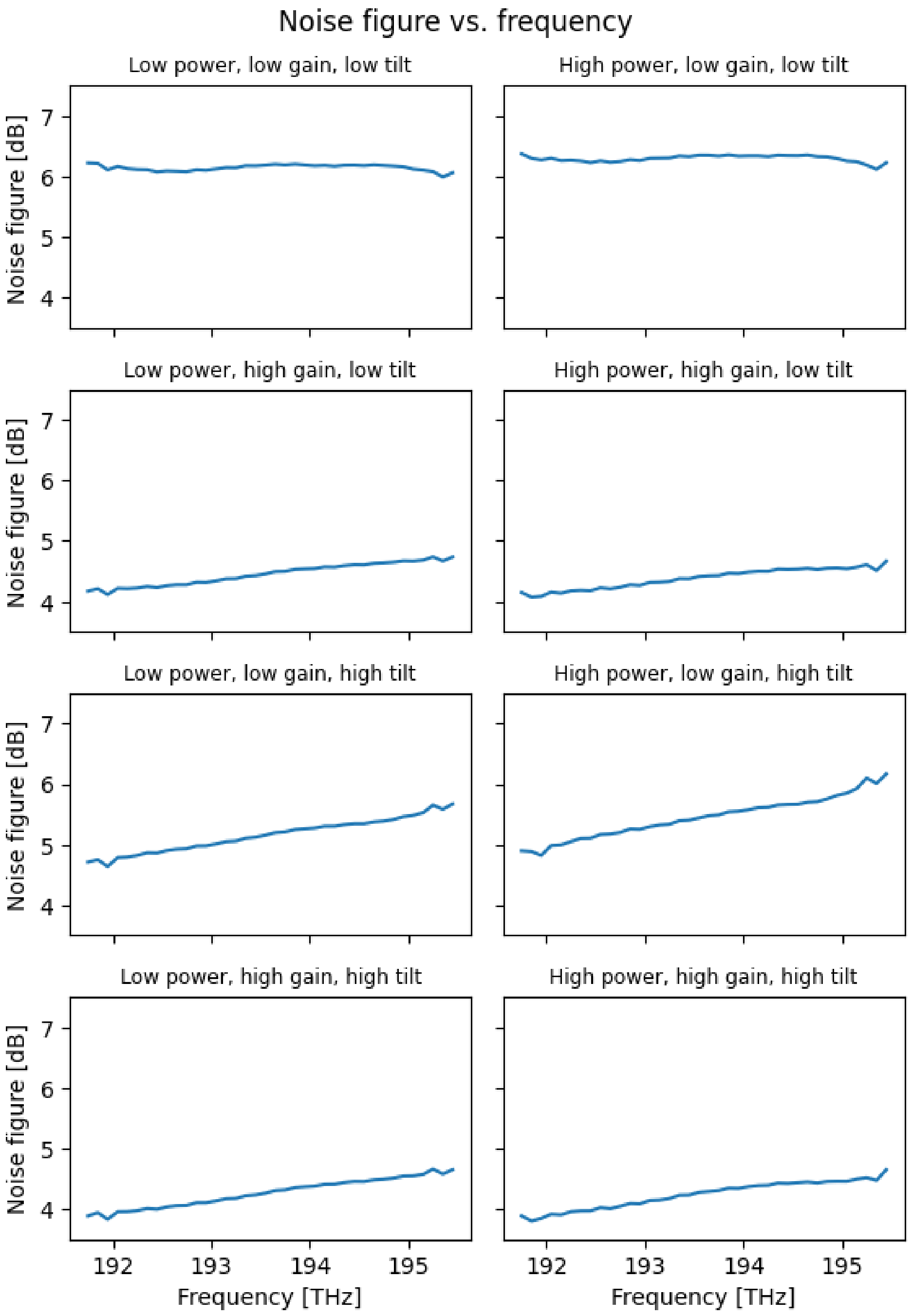


Figure 3.14: NF vs. channel frequency, various input parameters configurations.

$$NF(P, G, T, f) = \sum_{a=0}^{4} \sum_{b=0}^{3} \sum_{c=0}^{3} \sum_{d=0}^{8} m_{abcd} \, P^a \, G^b \, T^c \, f^d \tag{3.9}$$

where m_{abcd} represents the coefficient of each monomial. Note that this is not the formula used for polynomial evaluation, as discussed in section 3.4.2.

The idea is to evaluate the polynomial coefficients using the Generalized Least Squares (GLS) method, an algorithm for estimating the parameters of a linear regression model. To achieve polynomial regression, a new dataset is required. This dataset will not only include the previously mentioned five columns but also a column for the noise figure and an additional column for each monomial, resulting in a total of 721 columns (5 terms for the polynomial related to input power, 4 for gain, 4 for tilt and 9 for frequency).

The GLS method requires a training set to build our model and a test set to evaluate its performance. A model for each device within each gain range is obtained, and multiple datasets from different devices of the same brand and model are available, which will be joined together to make each model more accurate.

Two types of analysis will be conducted. First, the models will be created with a 70% training and 30% test split to assess performance. After this, the models will be trained on the complete datasets, which will be the final models for use inside other models with different datasets.

Essentially, the fitting operation is performed using the *GLS* function from the *statsmodels* Python library. The polynomial coefficients are then saved in a new dataset with a single column and stored as *.parquet* files.

3.4 Notes on code and algorithms

3.4.1 Python class: an easy interface

Although the entire model creation process was conducted using Python programs and libraries, the results are simply represented by coefficients for a multivariate polynomial, stored in column-oriented *.parquet* files. Consequently, the evaluation of the noise figure can be performed using any programming language or tool capable of reading these files and executing basic mathematical operations. This provides excellent portability for the model, offering maximum flexibility in creating an interface. Polynomial computations can potentially be done very quickly, as shown in section 3.5.7. A Python interface was selected for the model due to the popularity of the language, and its seamless integration with powerful libraries such as *Pandas* and *NumPy*. This interface is implemented as a Python class, exposing attributes and methods to enhance ease of use and allowing integration into more comprehensive EDFA models. The class requires Python 3.8 or newer, for reasons due to libraries and syntax. The core of the library consists in the following files and folders:

1. **edfa.py**: a Python file which describes the **EDFA_NF** class and the *Enum* class **EDFA_limits**;
2. **poly_eval.py**: a Python file which contains methods for polynomial evaluation;
3. **models**: a folder containing all the models supported by the library, which has to be placed in the same directory as the other two files.

In order to use the library, the first step is to install the required dependencies, which are listed in **requirements.txt**. This can be made by issuing the following command in the Python environment:

```
pip install pandas numpy fastparquet
```

The *Pandas* and *NumPy* libraries are used for data manipulation, while the *fastparquet* library provides the engine used by *Pandas* to open the *.parquet* files efficiently.

To use the library, the EDFA_NF class has to be instantiated as follows:

```
from edfa import EDFA_NF

e = EDFA_NF()
```

Once the class has been instantiated, the init_model() method has to be called. This function opens the model file corresponding to the given parameter, and loads the coefficients into the class attributes. This approach eliminates the need to repeatedly access the file for each invocation of the noise figure evaluation function.

```
e.init_model("EDFA17_LOW_38")
```

The models are named with the pattern **X_Y_Z**, where X is the device brand and specific model, Y is the gain range, and Z is the number of active channels in the training datasets of the model. This number is important for two reasons. First, for more accurate results, it is ideal to select a model obtained with a spectral load similar to the one being estimated, to mitigate potential effects from different channel configurations. Second, the input parameter that the model accepts is the total power of all active channels. Therefore, when estimating the noise figure for a spectrum with a different number of channels, this difference needs to be taken into consideration, as shown further in the section. Once the initialization method has been called, there are two possible ways to estimate the noise figure. The first one is by calling estimate_NF(), which performs a single estimation with float arguments. If, instead, multiple estimations at once are required, which is the most common case, it is better to call estimate_NF_array(), which requires four 1-D numpy.ndarray input parameters, with the same length.

```
e = EDFA_NF()
e.init_model("EDFA17_LOW_38")
p_in: float = -8.       # dBm
gain: float = 18.       # dB
tilt: float = 1.        # dB
freq: float = 193.      # THz
nf = e.estimate_NF(p_in, gain, tilt, freq)
# 7.264683934336477

p_in = numpy.array([-4., 0., 2.])
```

```
gain = numpy.array([18., 16., 14.])
tilt = numpy.array([0., 3., 0.])
freq = numpy.array([192.5, 193., 193.5])
nf = e.estimate_NF_array(p_in, gain, tilt, freq)
# [7.26939002 7.32205182 7.80681984]
```

The two functions exhibit better computational speed with low and high amounts of samples, respectively. For more details, refer to Section 3.5.7.
It is important to remind that the input power parameter refers to the total power of all the active channels. Therefore, a conversion is needed if the data we are using refer to a different number of channels:

$$P_{IN_{ADJ}} = P_{IN} - 10\,log_{10}\left(N_{current}\right) + 10\,log_{10}\left(N_{model}\right) \tag{3.10}$$

where $N_{current}$ and N_{model} are the numbers of active channels in the configuration to be estimated and in the model training datasets, respectively.

The library includes an *Enum* class called `EDFA_limits` which stores the minimum and maximum values of all input parameters for each supported model. When the estimation methods are called, the user will receive a warning if any input parameters fall outside these boundaries. Additionally, the class provides automatic clamping of input parameters to ensure they remain within the operating range, if enabled, in addition to issuing warnings.

```
e.clamp_all()
# sets all the clamping flags to True. This will enable clipping
                                 for:
# - input power
# - gain
# - tilt
# - frequency
# - output power, by reducing gain if input power + gain > max.
                                 output power
e.clamp_none()
# disables clamping for all 5 metrics
e.set_clamp("gain", True)
# sets one specific flag to the given boolean value. The allowed
                                 keys, to be passed as strings,
                                 are:
# - "power_in"
# - "gain"
# - "tilt"
# - "frequency"
# - "power_out"
# the method will silently ignore the command if the key is not valid.
```

3.4.2 Polynomial evaluation: Horner's rule

As previously discussed, speed is a major concern for this model, as it is intended to be integrated into more complex EDFA or network models. Therefore, optimizing the computation speed of polynomial evaluation is crucial. To achieve this, we utilize Horner's method, an algorithm designed to reduce the number of operations required for polynomial evaluation by computing it in a nested form. This method can be generalized for any polynomial [37].

$$\begin{aligned} p(x) &= a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0 \\ &= a_0 + x(a_1 + x(a_2 + ... + x(a_{n-1} + a_n x)...)) \end{aligned} \tag{3.11}$$

In this form, an nth degree polynomial can be evaluated by performing only n multiplications and n additions. Although any polynomial can be evaluated using this algorithm, and a function that performs this computation for any polynomial could easily be derived, Python loops are known for being relatively slow, as is typical for interpreted programming languages. For this reason, to further accelerate computation, the complete expression of the polynomial is unrolled into Horner's form and explicitly implemented within the interface library.

3.5 Analysis of results

This section summarizes the results regarding the absolute and relative errors observed in the estimation of the noise figure by means of the polynomial models compared to values derived from the datasets. Additionally, it illustrates the computation times for noise figure calculations from the datasets and the creation of the model itself. Initially, considerations are made regarding the uncertainty associated with Optical Spectrum Analyzer (OSA) measurements to establish realistic expectations for the accuracy of the models. Subsequently, the performance of each model is evaluated in two scenarios: first, using a 70-30 split of the datasets, and second, with complete datasets intended for broader application. Further analysis investigates the correlation between dataset size or number and accuracy. This assessment aims to determine whether further improvements are possible, if accuracy has reached its limit, or if it's feasible to reduce the number or size of datasets while maintaining a comparable level of accuracy. Performance metrics such as model speed and memory usage are then evaluated. Finally, a test is conducted where this model is integrated into a deep-learning framework aimed at estimating the power spectral density at the output of an EDFA, based on configuration parameters and input PSD.

3.5.1 Uncertainty of OSA measurements

Of course, the measurements provided by the OSA have some degree of uncertainty, which establishes the lower bound for model accuracy. It is futile to strive for a model that achieves a maximum error lower than that of the measurements, as it would mean accurately estimating an inherently inaccurate value. Instead, it is prudent to estimate this uncertainty and aim for similar accuracy levels with the models. Performing an accurate analytical error analysis presents several challenges. Firstly, multiple OSAs were used to acquire the training datasets, each potentially introducing different (and uncorrelated) amounts of error. Secondly, accurately estimating these errors would require detailed knowledge of the interdependencies between variables (e.g., gain as a function of frequency), which cannot be represented by simple analytical relationships. Therefore, experimental estimation of these errors is more suitable. A contribution to the uncertainty is the sensitivity of the total input and output power measurements. OSAs represented those values with one decimal place, therefore the absolute uncertainty due to quantization of P_{IN} and P_{OUT} is:

$$\delta P = \frac{0.1\,\mathrm{dBm}}{2} = 0.05\,\mathrm{dBm} \tag{3.12}$$

This variation of both P_{IN} and P_{OUT} results in a noise figure error of up to 0.1 dB, representing the absolute minimum uncertainty, assuming that all the other quantities are obtained with ideal infinite accuracy.

To establish a reference for the experimental error in measurements, four distinct datasets from identical devices are examined: EDFA-35 from Cisco in the low gain range. Measurements are compared for identical values of total input power, gain, tilt, and channel index, and the maximum difference among them is assessed. Figure 3.15 displays the cumulative distribution of measurement errors. Here, the y-axis represents the percentage of dataset measurements with errors within the value indicated on the x-axis. A curve that is more shifted to the left indicates better performance, as it signifies that a larger portion of the dataset has errors lower than those shown on the x-axis.

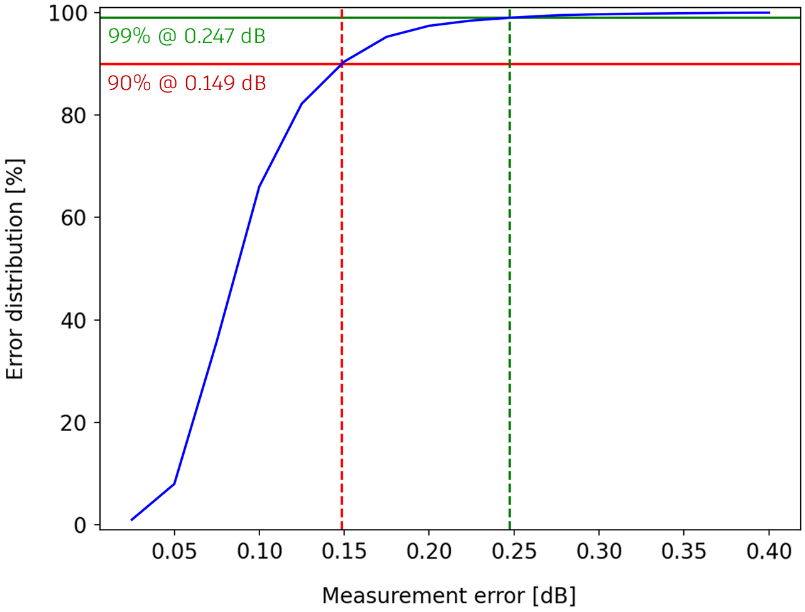


Figure 3.15: Cumulative distribution of the measurement errors for Cisco EDFA-35 devices in low gain range.

From this figure, it is evident that 90% of the measurements of Cisco EDFA-35 devices in low gain range have errors within 0.149 dBm, while selecting an absolute error of 0.247 dBm ensures that 99% of the measurement errors are safely included. In later sections, these two error values will be referred to as the '90% threshold' and the '99% threshold'.

Hence, it is reasonable to select 0.25 dBm as a plausible lower limit for model accuracy. However, this value may vary for different devices, given its experimental nature.

3.5.2 70-30 split models

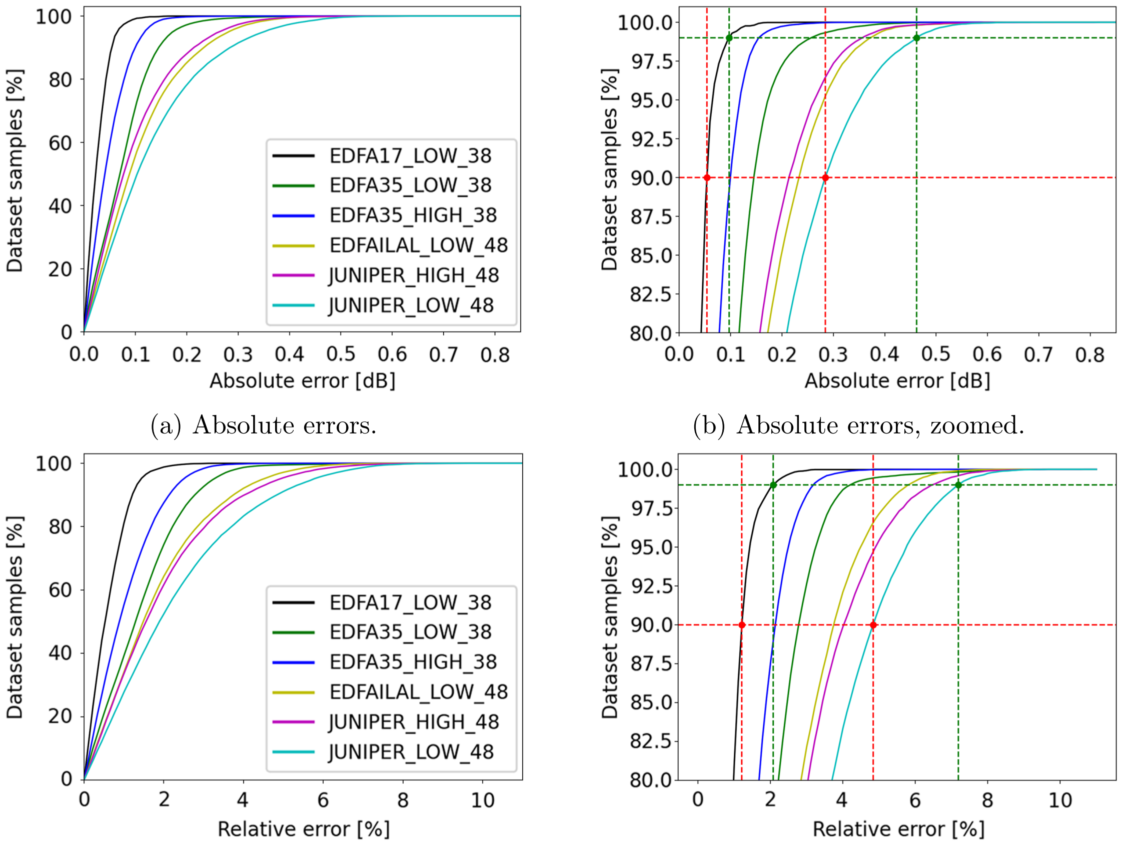
These initial models are created by concatenating all datasets from various devices of the same type—same brand, model, and gain range. These datasets are split into a training set, used for model construction, and a test set, employed to assess result accuracy. Six different models are created:

1. Cisco EDFA-17, low gain range, 1 dataset;
2. Cisco EDFA-35, high gain range, 2 datasets;
3. Cisco EDFA-35, low gain range, 4 datasets;
4. Cisco L-band EDFA, low gain range, 6 datasets;
5. Juniper, high gain range, 4 datasets;
6. Juniper, low gain range, 4 datasets.

The training and test sets are divided using a 70/30 ratio, following common practice in machine learning applications. The results for the cumulative distributions of the absolute and relative errors are depicted in Figure 3.16.

Predictably, the EDFA-17 model exhibits better performance since it utilizes data from a single device, thereby avoiding potential errors introduced by manufacturing variations and resulting in higher overall measurement correlation. If additional datasets were available for this model, its performance might be negatively affected, but unfortunately, none were available.

For absolute errors, the 90% threshold is between 0.05 and 0.28 dB, depending on the model, while the 99% threshold is between 0.10 and 0.46 dB. For relative errors, instead, the 90% threshold is between 1.19 and 4.83%, depending on the



(a) Absolute errors.

(b) Absolute errors, zoomed.

(c) Relative errors.

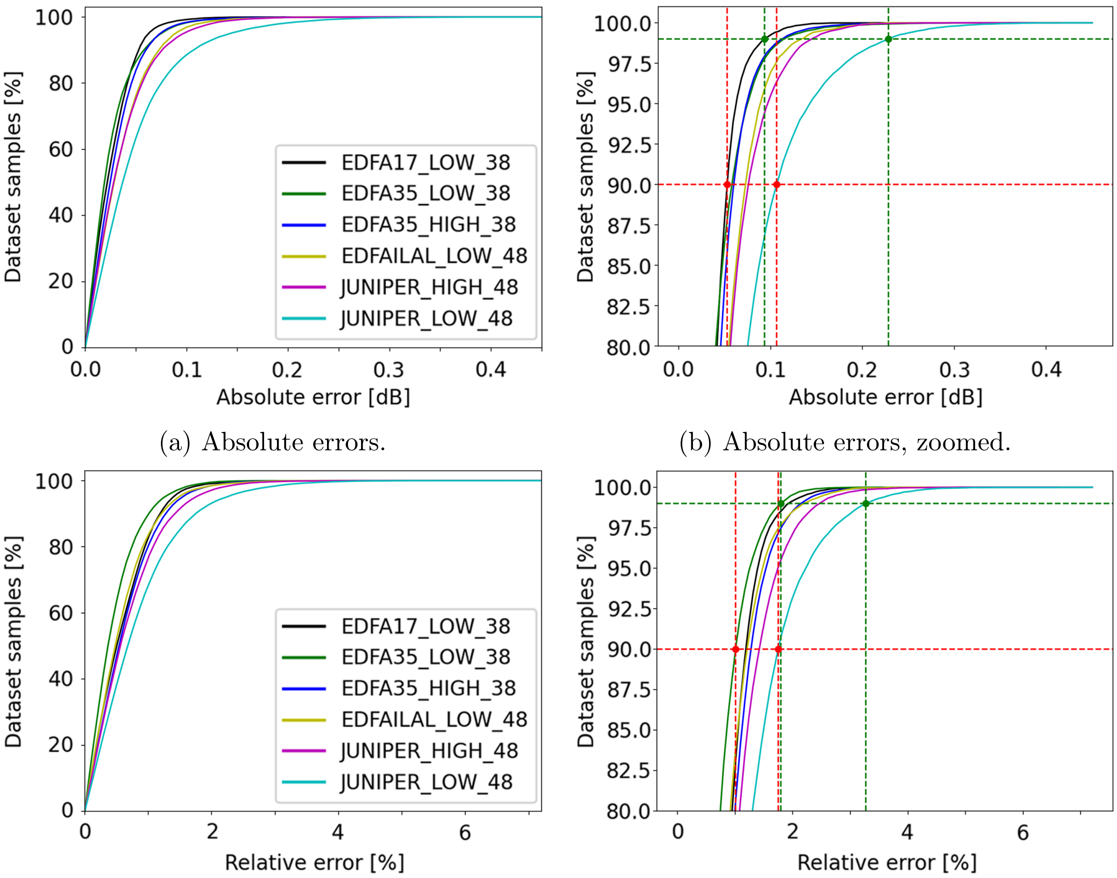
(d) Relative errors, zoomed.

Figure 3.16: Cumulative distribution of absolute and relative errors of 70-30 split models.

model, while the 99% threshold is between 2.08 and 7.19%.The maximum values for absolute and relative error are 0.81 dB and 10.56% and are both from the Juniper devices in high gain range.

3.5.3 Complete dataset models

Now, the same models are created, but with the entire datasets used for training instead of a random 70% of their entries. The test set consists of the average of all datasets for each device. This analysis aims to capture errors specific to the model functions, such as the polynomial fits, and to assess whether a polynomial fit is suitable for this model or if there is room for improvement in the regression function. The models created are identical to those in the previous case, in terms of brand, gain range and number of datasets. The results are reported in Figure 3.17.



(a) Absolute errors.

(b) Absolute errors, zoomed.

(c) Relative errors.

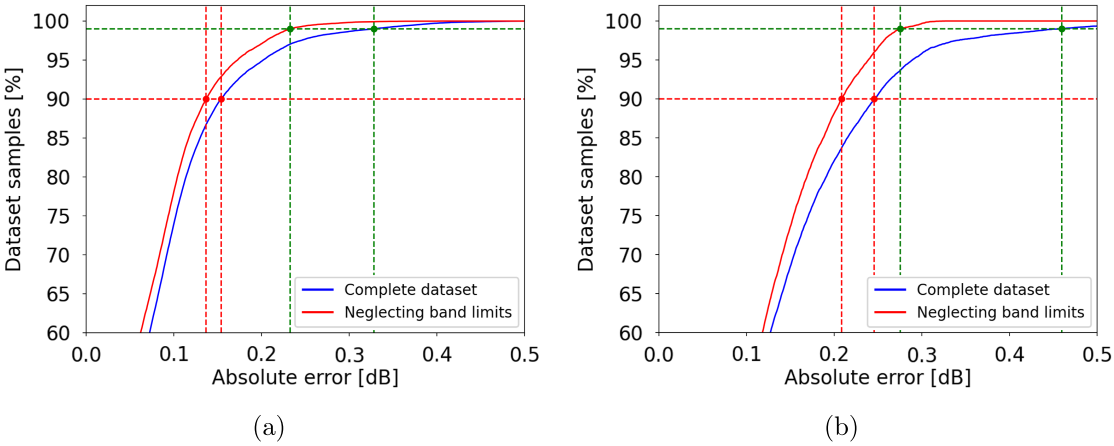
(d) Relative errors, zoomed.

Figure 3.17: Cumulative distribution of absolute and relative errors of complete dataset models.

For absolute errors, the 90% threshold is between 0.05 and 0.11 dB, depending on the model, while the 99% threshold is between 0.09 and 0.23 dB. For relative errors, instead, the 90% threshold is between 1.01 and 1.74%, depending on the model, while the 99% threshold is between 1.79 and 3.27%.The maximum values for absolute and relative error are 0.42 dB and 7.06% and are both from the Juniper devices in low gain. Interestingly, despite having only one dataset, the EDFA-17 shows higher relative errors compared to the EDFA-35 in the low gain range. The errors attributed to the model are approximately equivalent to those observed in the dataset measurements, as reported in Figure 3.15. This suggests that the polynomial form of the fitting function introduces an error comparable to the inherent uncertainty in the measurements. Thus, using a polynomial to describe the dependency of the noise figure on the four input parameters is appropriate.

3.5.4 Correlation of errors

An important aspect to investigate regarding model errors is the potential correlation between input configurations and higher or lower absolute or relative errors. This is crucial because if the highest errors are concentrated in specific input patterns, it could lead to an overestimation of the average accuracy of the model. With more information on error distribution, it might be possible to apply different error tolerances to amplifier models based on their operating ranges. This analysis examines the mean value of the noise figure absolute error as a function of each input parameter, considering the cumulative distribution of absolute error as in previous sections. By plotting the absolute error against each parameter, we can observe distinct behaviors among device brands. A common trend observed across almost all devices is the higher error near the frequency band limits. Examples of this trend can be seen in Figure 3.18, where focusing solely on central channels results in lower overall errors. This phenomenon may be attributed to over-fitting, as polynomial models typically exhibit less accuracy at the boundaries of their fitting intervals.

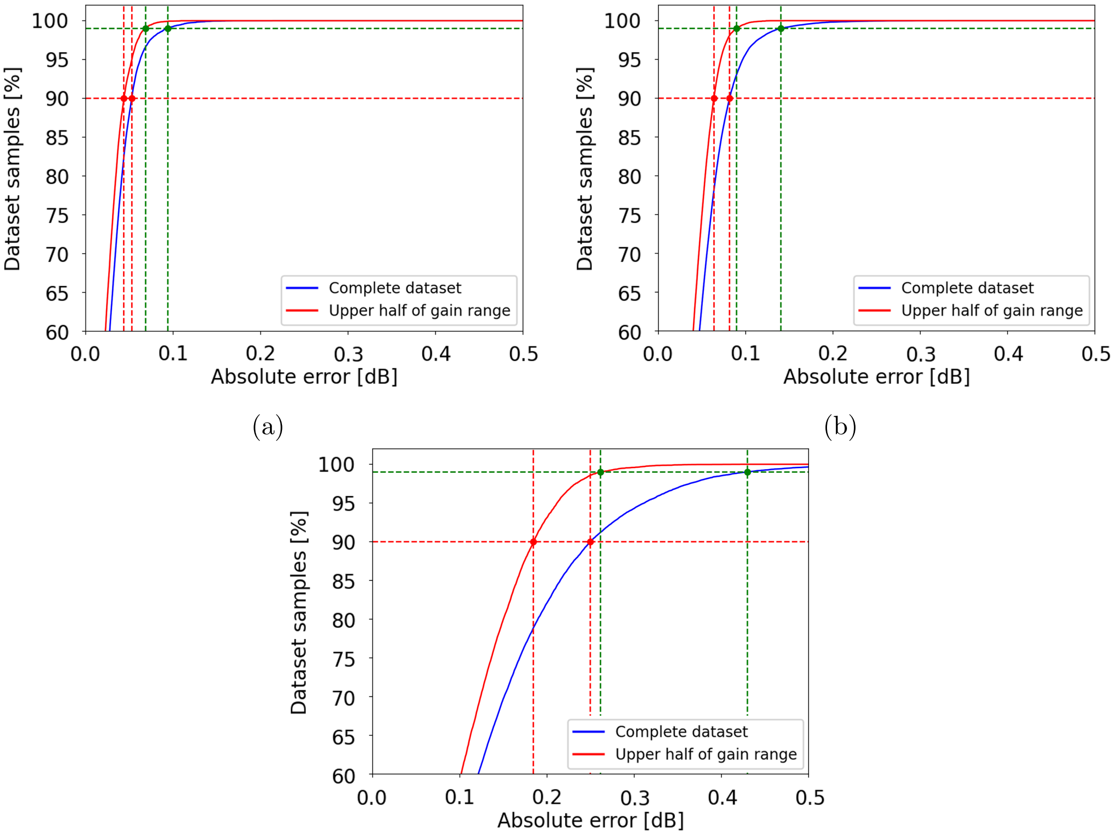


(a)

(b)

Figure 3.18: Cumulative distribution of absolute errors, correlation with frequency range. (a) EDFA-35, low gain range; (b) L-band EDFA, low gain range.

The total input power appears to have minimal effect on the error distribution, with inconsistent trends observed among datasets of the same device type. However, Cisco EDFA-17, EDFA-35, and Juniper devices consistently display lower errors at higher target gain values, as depicted in Figure 3.19.



(a)

(b)

(c)

Figure 3.19: Cumulative distribution of absolute errors, correlation with gain range. (a) EDFA-17, low gain range; (b) EDFA-35, high gain range; (c) Juniper, low gain range.

Juniper devices also show a decrease in errors with higher values of target tilt, as shown in Figure 3.20.

In conclusion, it is evident that the noise figure model exhibits varying levels of error across different intervals of the input data. These differences can be quite significant, as illustrated in Figure 3.18b, where the 99% threshold error is reduced from 0.46 to 0.28 dBm, or Figure 3.19c, where the same threshold is reduced from 0.43 to 0.27 dBm. Unfortunately, these intervals are heavily dependent on the device, so no general rule can be established for reducing the range of input parameters to achieve lower errors, except for the consistent behavior related to channel frequency observed across all devices.

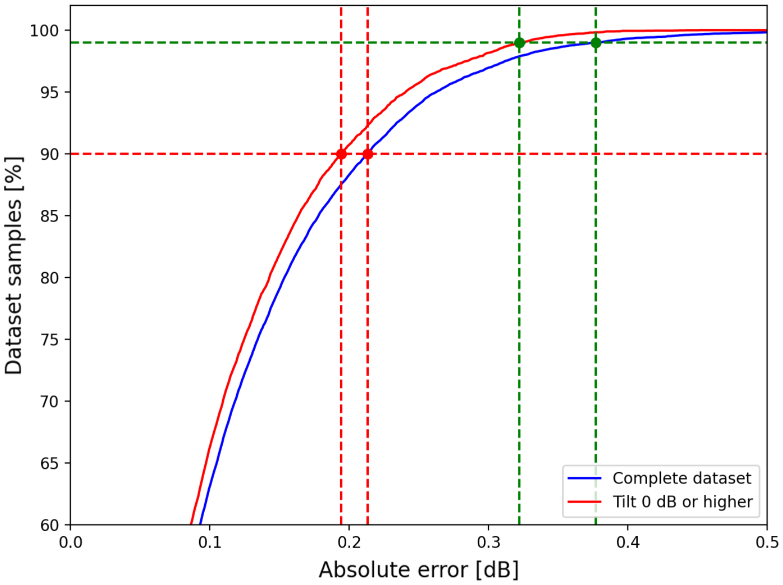


Figure 3.20: Cumulative distribution of absolute errors, correlation with tilt range in JUNIPER devices.

3.5.5 Code execution time

In this section, the execution time of the developed Python scripts will be evaluated. All tests are conducted on a machine equipped with an Intel Core i7-1250U processor. Firstly, the time required to compute the noise figure from the Matlab datasets and save the results into *.parquet* files is assessed. Subsequently, the time taken to create the device models is measured.

From Matlab files to noise figure datasets

The first part of the code is used to perform the following steps:

1. load each *.mat* EDFA dataset, containing the input and output spectral power densities of the amplifier as a function of total input power, target gain and tilt;

2. perform all the steps listed in section 3.2;

3. save the results in *.parquet* files to optimize speed and storage size, and also in *.xlsx* format for readability. This last step is not necessary for model creation, and the *.xlsx* files can be discarded or not created at all.

The results for the running time of the program are reported in table 3.1.

Device	Gain	Dataset no.	Time [s]	Row count	Time per row [ms]
EDFA-17	LOW	1	13.22	21318	0.62
EDFA-35	HIGH	1	32.95	56848	0.58
EDFA-35	HIGH	2	31.99	56848	0.56
EDFA-35	LOW	1	25.79	45144	0.57
EDFA-35	LOW	2	27.77	46816	0.59
EDFA-35	LOW	3	25.77	45144	0.57
EDFA-35	LOW	4	34.54	58608	0.59
EDFA-ILA-L	LOW	1	12.08	21360	0.57
EDFA-ILA-L	LOW	2	12.52	21360	0.59
EDFA-ILA-L	LOW	3	9.03	14976	0.60
EDFA-ILA-L	LOW	4	9.16	14976	0.61
EDFA-ILA-L	LOW	5	8.83	14976	0.59
EDFA-ILA-L	LOW	6	8.66	14976	0.58
JUNIPER	HIGH	1	10.79	18816	0.57
JUNIPER	HIGH	2	11.44	18816	0.61
JUNIPER	HIGH	3	12.15	18816	0.65
JUNIPER	HIGH	4	11.06	18480	0.60
JUNIPER	LOW	1	16.15	29232	0.55
JUNIPER	LOW	2	16.70	29232	0.57
JUNIPER	LOW	3	17.23	29232	0.59
JUNIPER	LOW	4	16.90	29232	0.58

Table 3.1: Computation time of the noise figure for each EDFA dataset.

Each row, representing a unique combination of total input power, gain, tilt, and channel index across each dataset, is processed in approximately 0.60 ms. This equates to processing about 1667 rows per second, or 100,000 rows per minute. This metric provides a quick estimation of computation time based on dataset size. It's important to note that this number is specific to the machine used and should be computed experimentally.

From noise figure datasets to models

In this section, the time taken to train the polynomial model is measured. The program under test executes the following tasks:

1. concatenate all the datasets with the noise figure calculations corresponding to each EDFA device in each gain range;

2. build the training data, meaning the computation of the new dataset columns, as explained in section 3.3.5;
3. execute the GLS algorithm to compute the polynomial coefficients;
4. store the coefficients in *.parquet* files.

The results of the running time of the program for the six different models are reported in table 3.2.

Device	Gain	Time [s]	Total row count	Time per row [ms]
EDFA-17	LOW	2.70	21318	0.13
EDFA-35	HIGH	15.04	113696	0.13
EDFA-35	LOW	35.08	195712	0.18
EDFA-ILA-L	LOW	16.58	102624	0.16
JUNIPER	HIGH	12.05	74928	0.16
JUNIPER	LOW	17.83	116928	0.15

Table 3.2: Training time of the noise figure polynomial models.

The computation time continues to depend on the size of the training datasets. The program takes up to around 0.18 ms per dataset row to compute the model coefficients. Adding this to the previous step, it takes approximately 7.8 s to build a model from a single *.mat* dataset containing 10^4 combinations of total input power, target gain and tilt, and active channel central frequency. This time increases to 1 min 18 s for 10^5 combinations, and 13 min for 10^6 sets.

3.5.6 Training dataset size vs. accuracy

The measurement of the power spectral densities required to train this model can be very time consuming. This is due to the number of required OSA measurements and is dependent on the number of active channels. Taking as an example the EDFA-17 dataset in low gain range, it was characterized with the following input parameters:

1. Total input power: from -10 dBm to 6 dBm with a step of 2 dBm (9 values);
2. Gain: from 14 dB to 20 dB with a step of 1 dB (7 values);
3. Tilt: from -5 dB to 5 dB, with a step of 1 dB (11 values).

In this case, 9 input power spectra and 693 output power spectra are required. If the channel density is high and the spacing between them is low, a lower resolution bandwidth is necessary to accurately capture the noise level between adjacent channels. Since the measurement time is inversely proportional to the square of the resolution bandwidth, this can significantly increase the required time per dataset.

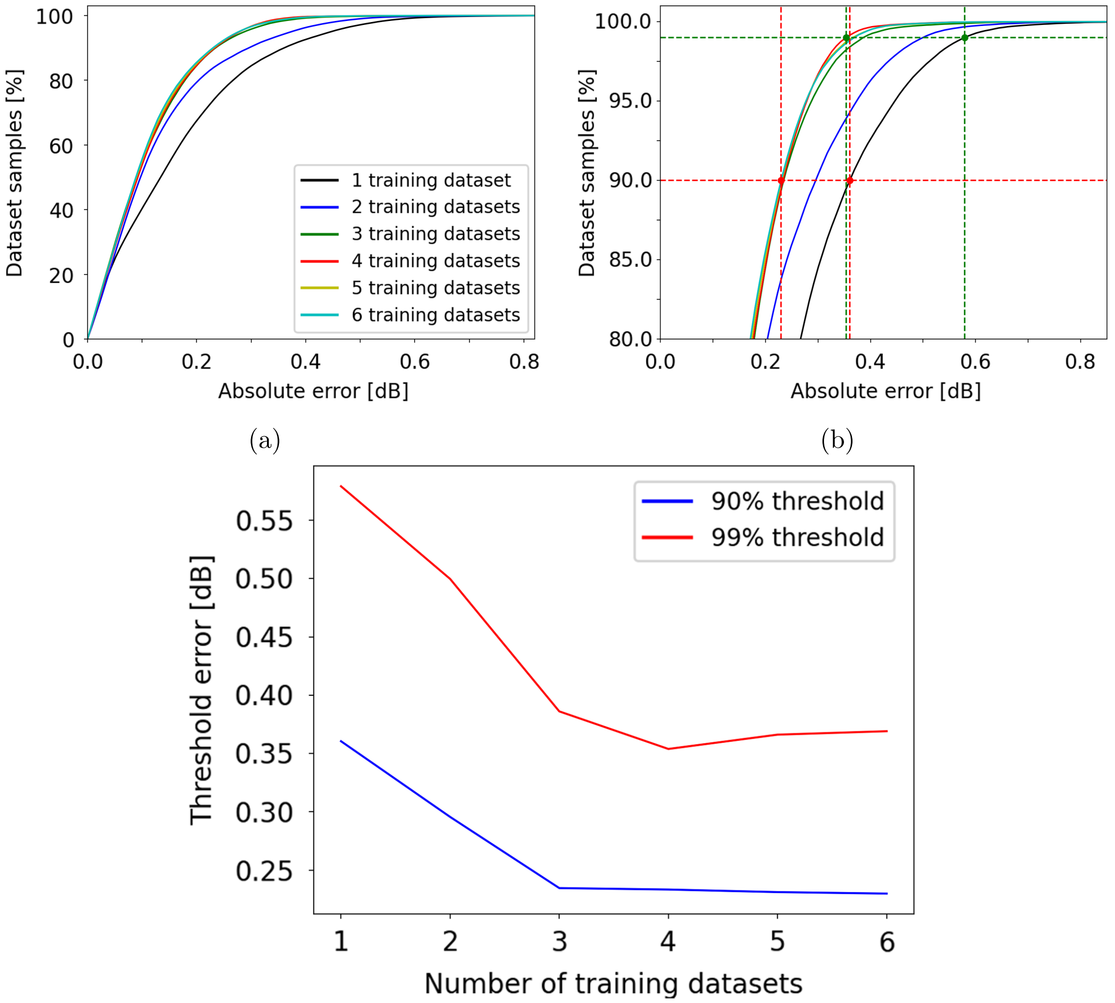
This is why considerations are made regarding the model's accuracy concerning the number and size of training datasets. As mentioned earlier, the model aims to be flexible, allowing a balance between accuracy and the cost of acquiring training data.

Two types of analyses are conducted. Firstly, the model will be trained on the full dataset of one or more devices, and its accuracy will be tested on the remaining datasets. This represents the 'worst case' scenario, as there is less correlation between different devices. Secondly, all datasets will be combined, and a random train/test split will be performed on the data, with accuracy evaluated as a function of the split ratio. For these tests, the Cisco L-band EDFA datasets will be used, as six of them are available.

Accuracy vs. number of datasets

The first analysis examines the accuracy of the model as a function of the number of measurement datasets used for training. The test results are evaluated using the concatenation of all device datasets. The results, presented in terms of cumulative distribution of absolute error, are shown in Figure 3.21.

Figures 3.21a and 3.21b illustrate that models created using one or two datasets exhibit visibly poorer performance compared to the remaining cases, which show similar curves across the entire error range. The variations are significant, with the 90% threshold ranging between 0.22 dB and 0.36 dB and the 99% threshold between 0.35 dB and 0.58 dB. Figure 3.21c depicts these error threshold values as a function of the number of datasets. It shows that accuracy steadily improves until three datasets are used, after which it plateaus with only slight improvements as the number increases up to six. From these findings, it is concluded that including data from multiple devices in the training process is crucial to account for error contributions arising from manufacturing processes. However, the number of different devices can be kept reasonably low.



(a)

(b)

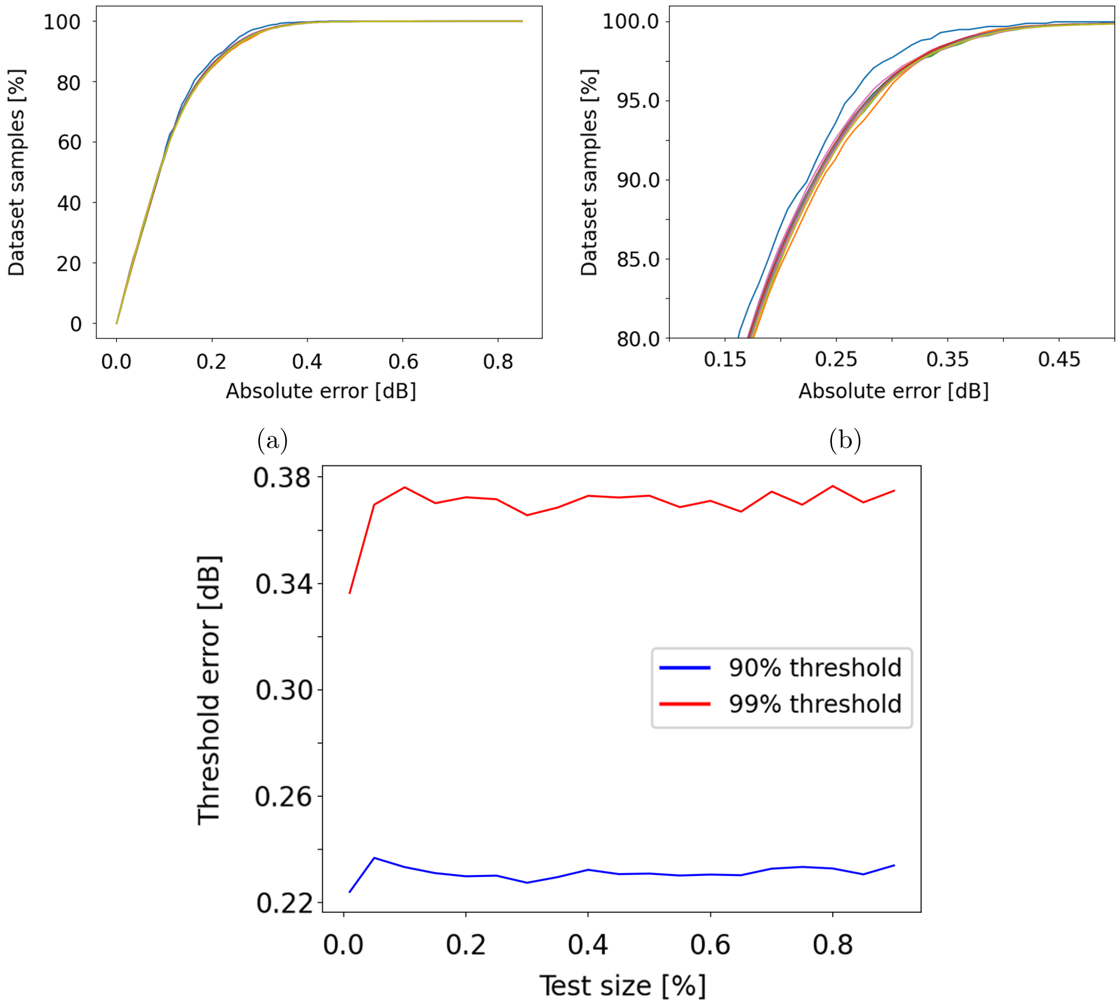
(c)

Figure 3.21: Cumulative distribution of absolute error of the L-band EDFA device, with respect to the number of training datasets. (a) overview of absolute errors; (b) zoom in the y-axis region between 80% and 100%; (c) 90% and 99% thresholds vs. number of datasets.

Accuracy vs. train/test split ratio

This second analysis aims at evaluating the impact of train and test split ratio on model accuracy. Instead of using datasets from different devices, the models are created by merging the six measurement datasets and varying the percentage of the random split between training and test sets. The results, presented in terms of cumulative distribution of absolute error, are shown in Figure 3.22.

Figures 3.22a and 3.22b demonstrate that the model accuracy does not exhibit



(a)

(b)

(c)

Figure 3.22: Cumulative distribution of absolute error of the L-band EDFA device, with respect to the train/test splitting ratio. Legend is omitted due to the superposition of results. (a) overview of absolute errors; (b) zoom in the y-axis region between 80% and 100%; (c) 90% and 99% thresholds vs. test size.

sensible changes with train and test splitting ratio. The curves remain consistently close together across the entire range, except for the curve representing a 1% test size (in blue), which performs slightly better in the zoomed region. Figure 3.22c, which shows the 90% and 99% thresholds as function of the test size, further confirms the stability of the absolute error with respect to this parameter. Based on these considerations and those from the previous section, we can conclude that the optimal strategy for achieving good model performance while minimizing the time spent on measuring EDFA power density spectra is to use measurements from multiple different devices, even with reduced datasets. For instance, increasing the step of the total input power, gain, and tilt would be a beneficial approach to reduce the number of measured power spectra. However, reducing the number of tested devices would not be advisable.

3.5.7 Model computational speed and memory footprint

This section focuses on the technical performance analysis of the polynomial model. Two metrics will be examined: computational speed, measured by the time required for the noise figure evaluation functions to execute; and memory footprint, defined as both the storage size of the model files and Python scripts in local memory, as well as the amount of computer memory (RAM) required for the library to perform computations. These metrics were evaluated for the Python implementation of the model evaluation class, but may vary for different solutions.

Computational speed

One of the primary objectives of the model was to ensure high time efficiency. As previously discussed, this model is designed not for standalone use, but for integration into larger and more comprehensive EDFA models, and a fast model is less likely to become a performance bottleneck within a larger system. The time analysis is conducted using the *timeit* module, part of the standard Python 3 library.

First, we will test the instantiation of the class and the initialization of the model.

```
e = EDFA_NF()
e.init_model("EDFA17_LOW_38")
```

The first line creates a class instance and calls the corresponding constructor, which initializes the internal attributes to their default value. The `init_model` function performs the following operations:

1. check if the *models* folder exists in the working directory and, if it is not present, raise an exception;

2. try to retrieve the model file with the name passed as parameter from the *models* folder, raising an exception in case the file does not exist;
3. with the aid of the *Pandas* library, open the model *.parquet* file and save the model name and polynomial coefficients inside two internal attributes.

The second function to be tested is the single-value noise figure calculation method.

```
nf = e.estimate_NF(p, g, t, f)
```

where `p, g, t, f` are four floating-point variables corresponding to the input parameters. This function performs the following operations:

1. check if the `init_model` function was called, and raise an exception if not;
2. call the `__warnings` private method, which checks if the input parameters are within the operating boundaries of the amplifier, and clamps their values if the corresponding clamping flag is set, otherwise just raises a warning to the user;
3. run the polynomial evaluation function.

Note that this calculation includes some overhead due to the input parameter controls, which may affect execution speed.

Lastly, the performance of the function that calculates the noise figure for multiple values simultaneously is evaluated.

```
nf = e.estimate_NF_array(p, g, t, f)
```

where `p, g, t, f` are of type `ndarray`, defined in the *NumPy* library. They are all the same length, and the calculation is performed element-wise. This function performs the same operations as the previous one, extended to the entire length of the arrays, and also checks that the four input arrays all have the same length.

The results for the execution time in ms, µs and ns units are reported in Table 3.3. For the array evaluation case, the values are divided by the array length, for a fair comparison with respect to the single evaluation case.

It is immediately apparent that the `init_model` method is orders of magnitude slower than the evaluation methods due to its file reading operations. Therefore,

Class method(s)	Exec. time [ms]	Exec. time [µs]	Exec. time [ns]
Initialization	40.76	40759.31	40759306.60
NF evaluation (single)	0.05	47.60	47600.11
NF evaluation (array)	0.00	0.61	613.81

Table 3.3: Execution time of the model evaluation Python class, divided by function.

it is more efficient to work with a single EDFA model for all computations and switch only when necessary to estimate the noise figure for different devices. If multiple models need to be accessed simultaneously, modifications to the library could be considered, such as storing multiple models within the class instance. This approach would speed up operations but increase the memory footprint of the class instance. The `init_model` method takes approximately the same time of 856 calls to the `estimate_NF` function, or a single call to the `estimate_NF_array` function with an array of about $6.6 \cdot 10^4$ elements. It is also possible to appreciate that the `estimate_NF_array` solution is about 80 times faster than the `estimate_NF` method, when evaluating per-element computation time with a large array length (in this test, the length was about $2 \cdot 10^4$). The `estimate_NF` method, on the other hand, is faster for single-element computation. Table 3.4 aims at finding the breakthrough, the number of elements after which `estimate_NF_array` becomes faster than repeated calls to `estimate_NF`.

Array length	Exec. time per sample [µs]
1	1479.72
2	746.99
5	298.97
10	150.46
20	76.76
33	47.39

Table 3.4: Execution time of the array estimation function for the noise figure, as a function of array length.

From these results, we can conclude that the `estimate_NF` method is faster up to an array length of around 33.

Memory footprint

In this section, the performance of the model in terms of disk size and RAM footprint will be evaluated. The combined size of the two Python scripts is about 36 kB. Each model file occupies about 10 kB. These files contain 720 64-bit floating-point coefficients, each labeled by a four-character string key. Assuming that the size of each key and coefficient is 4 and 8 bytes, respectively, the size of the information stored in each model file is $(4+8)\cdot 720 = 8640$ bytes, or approximately 8.43 kB. The difference between this calculated value and the actual file size is attributed to overhead from the *.parquet* file structure. The model can therefore be considered very lightweight for ordinary computers. To assess the RAM usage of the EDFA class, the *sys.getsizeof* method from the standard Python 3 library is utilized, combining the results for the function applied to the class instance and each internal attribute. The result is 26 334 B, or 25.72 kB. For a more detailed analysis, a memory profiler could be employed, which could provide insights into memory usage during noise figure computations. However, such analysis would also encompass other contributors to process memory, such as linked libraries, making it challenging to isolate the exact size of the `EDFA_NF` class instance.

3.5.8 Model usage in Deep Learning EDFA model

Lastly, the performance of the developed model is tested by employing it in a more comprehensive EDFA model. This testing phase is crucial because, as discussed in previous sections, this model was not designed to be employed by itself, but rather as part of a more complex solution. This deep learning software takes the signal power spectral density and the EDFA configuration settings as inputs, and returns the output PSD of the amplified signal. The model's success hinges on accurately estimating the amplifier noise figure. Previously, this estimation was a heuristic, fixed-value approach, independent of the input parameters and configuration. Determining this value involved manual iterations of the training and testing loop to find an optimal setting. The tests were performed on seven datasets obtained from devices by Juniper Networks. It is important to note that these datasets were selected because they represented he worst-case performance scenarios for the model, thus providing a stringest test of its capabilities. The performance results of the model with the fixed noise figure value and the estimation model are illustrated in tables 3.5, 3.6, 3.7, and 3.8.

From Tables 3.5 and 3.6 it is evident that the maximum and mean absolute noise figure errors are generally reduced with the use of the estimation model. This improvement is consistent across most datasets, although a couple of datasets exhibited only slightly worse results. The R^2 score, an internal model parameter which roughly estimates the proportion of the variation in the noise figure that is predictable from the input parameters, is reported in Table 3.7. This table indicates very little difference between the fixed-value approach and the estimation model, suggesting that both methods predict the noise figure variation to a similar degree. Finally, Table 3.8 shows the results in terms of RMSE. These results allow to conclude that errors are generally lower when the noise figure estimation model is employed, further demonstrating the model's effectiveness in reducing prediction errors and improving overall performance. As this test is a worst-case performance scenario, the model is expected to perform better for different devices.

Max. absolute error [dB] (lower is better)

Case	EDFA 1	EDFA 2	EDFA 3	EDFA 4	EDFA 5	EDFA 6	EDFA 7
Fixed NF	1.3914	1.1150	0.3941	0.4304	0.4176	0.5063	0.5245
Model NF	1.3099	1.1103	0.3730	0.4461	0.4067	0.4881	0.5351
	−5.86%	−0.42%	−5.35%	+3.65%	−2.61%	−3.59%	+2.02%

Table 3.5: Performance of deep learning EDFA model, maximum absolute error.

Mean absolute error [dB] (lower is better)

Case	EDFA 1	EDFA 2	EDFA 3	EDFA 4	EDFA 5	EDFA 6	EDFA 7
Fixed NF	0.1269	0.0849	0.0583	0.0590	0.0566	0.0587	0.0771
Model NF	0.1093	0.0816	0.0543	0.0596	0.0555	0.0625	0.0772
	−13.87%	−3.89%	−6.86%	+1.02%	−1.94%	+6.47%	+0.13%

Table 3.6: Performance of deep learning EDFA model, mean absolute error.

Adjusted R^2 score [%] (higher is better)

Case	EDFA 1	EDFA 2	EDFA 3	EDFA 4	EDFA 5	EDFA 6	EDFA 7
Fixed NF	92.2129	95.3613	95.9068	97.7005	91.6659	94.8887	93.1631
Model NF	89.3877	95.2287	96.0442	97.5909	91.9185	94.2216	93.5089
	−2.82%	+0.09%	+0.14%	−0.11%	+0.25%	−0.67%	+0.35%

Table 3.7: Performance of deep learning EDFA model, adjusted R^2 score.

RMSE [dB] (lower is better)

Case	EDFA 1	EDFA 2	EDFA 3	EDFA 4	EDFA 5	EDFA 6	EDFA 7
Fixed NF	0.1908	0.1413	0.0810	0.0813	0.0782	0.0828	0.1093
Model NF	0.1715	0.1390	0.0773	0.0828	0.0758	0.0885	0.1081
	−10.11%	−1.63%	−4.57%	+1.84%	−3.07%	+6.88%	−1.10%

Table 3.8: Performance of deep learning EDFA model, RMSE.

Chapter 4

Differential Group Delay modeling

In recent times, advancements in optical components and techniques have significantly improved fiber-optic transmission systems, meeting the growing bandwidth demands from diverse applications. Next-generation goals focus on creating fully transparent, high-capacity coherent transmission systems with polarization diversity for long-haul distances. To effectively design such systems, a thorough understanding of fiber-channel transmission impairments is paramount, as these impairments introduce complex signal distortions [38]. High-bit-rate optical transmission systems are vulnerable to optical-fiber-based issues such as chromatic dispersion (CD), polarization-mode dispersion (PMD), and fiber nonlinearities. Systems are typically designed to minimize or compensate for these effects using specialized fibers and devices [39]. In this analysis, we focus on Differential Group Delay (DGD), a parameter that is directly related to PMD, as the latter is intended as the average value of the former. The goal is to study the beheavior of this parameter in devices such as the optical multiplexers and demultiplexers inside a Wavelength Cross Connect (WXC) card. The objective is to see if DGD can be adjusted by varying the device attenuation, and if DGD depends linearly on the number of cascaded multiplexers and demultiplexers, as if it would with fiber spans, or if the parameter exhibits a different behavior.

4.1 Theoretical background

Single-mode optical fibers (SMF) support two orthogonally polarized modes (LP_{01x} and LP_{01y}) with identical transverse distribution. Ideally, both modes would propagate through the fiber with the same propagation constant, ensuring they travel at the same speed. However, imperfections arising during manufacturing or external factors like vibrations, bending, and temperature fluctuations can introduce birefringence into the fiber, altering their propagation constants. This birefringence varies along the fiber, causing random variations in the transmitted modes and phase differences (polarization mode coupling effect), leading to polarization mode dispersion (PMD). At higher optical power levels, the fiber's nonlinear nature becomes significant due to susceptibility, causing an intensity-dependent refractive index and propagation constant, which affects the orthogonal components of the transmitted optical pulse [38]. The effect of this birefringence results in a mismatch of group velocities between the two orthogonal components. The DGD of an optical fiber can be evaluated as:

$$DGD \approx \frac{d\Delta\beta}{\delta\omega}L = \frac{L}{c}B = \frac{L}{c}\frac{\lambda}{L_P} \tag{4.1}$$

where $\Delta\beta$ is the variation between the propagation constants of the two modes, L is the fiber length, c is the speed of light, λ is the light pulse wavelength, B is the birefringence and L_P is the beat length. These two parameters are defined as:

$$B = n_{slow} - n_{fast},\ L_P = \frac{\lambda}{B} \tag{4.2}$$

where n_{slow}, n_{fast} are the refractive indices of the two birefringent axes of the optical fiber. Note that DGD depends linearly on fiber length, with the other parameters being due to fiber material and geometrical structure. Therefore, in case of multiple spans of optical fibers connected together, DGD can be considered an additive metric. The objective of this analysis is to see if this assumption also holds true for more complex devices such as optical multiplexers and demultiplexers, and if the internal attenuation setting of those devices influences DGD in any way.

4.2 DGD and WXC attenuation

In this section, the effect of the target attenuation set on the optical multiplexers and demultiplexers on the resulting DGD is evaluated. The devices under test are WXC cards from Cisco, which can be used to provide colorless multiplexing and demultiplexing to ROADM nodes. The card allows to set a device attenuation between 0 and 20 dB. Higher attenuation values may require to employ the polarization analyzer in its highest sensitivity setting, in order to effectively capture the low output power level, and this has to be taken into consideration because, as will be evident in the later sections, instrument sensitivity impacts on the DGD reading.

4.2.1 Test setup

In order to measure DGD, a test setup is required, such as the one illustrated in Figure 4.1.

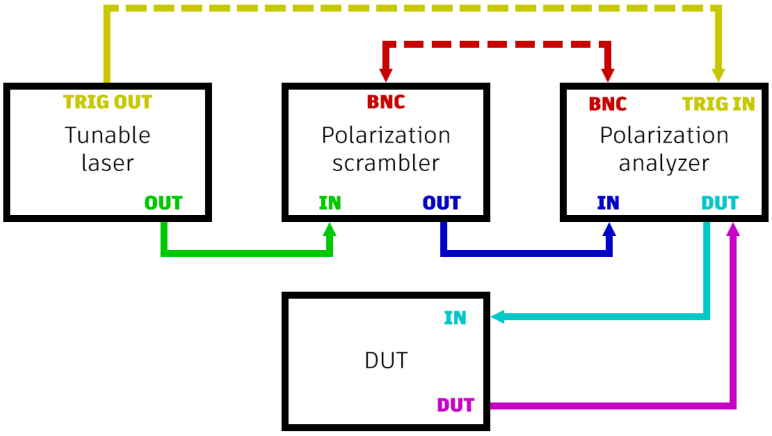


Figure 4.1: DGD measurement setup, showing each functional block and their interconnections. Solid lines represent optical connections, while dashed lines refer to electrical connections.

A tunable laser is fed into a polarization scrambler, which rapidly randomizes the SOP of the input signal, and then into a polarization analyzer. The role of the polarization scrambler is to remove any effects due to polarization, by providing the possibility to evaluate the mean of results for multiple random polarization states. The polarization analyzer is connected both to the scrambler output and to the DUT. It splits the signal into two paths, one of which is sent into the DUT, and then compares the two paths, to measure the impact of the DUT on the signal. The scrambler and analyzer are synchronized via an electrical BNS connection. Moreover, as the tunable laser performs a sweep, varying the laser central frequency over a set range, a connection is needed between the laser and the polarization analyzer, so that the laser source can trigger the measurement whenever the laser frequency is changed. The DUT is shown in Figure 4.2. In this case, it consists in one multiplexer from a WXC under test, whose DGD is studied as a function of the set attenuation.

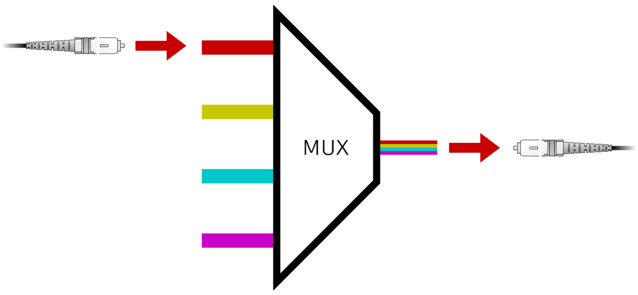


Figure 4.2: DGD attenuation DUT. Only a multiplexer is employed.

4.2.2 DGD results

The results for the DGD measurements as a function of the laser central frequency for different values of device attenuation are reported in Figure 4.3.

It is immediately possible to see that all results exhibit a similar mean value, possibly suggesting that attenuation does not have a significant impact on DGD. The frequency curve appears to be more noisy for the 20 dB attenuation case. This can be due to the fact that the polarization analyzer had to operate in its high sensitivity mode for this last case, as otherwise the input power would be below a detectable value. Figure 4.4 shows the mean value of the DGD across all frequencies of the span as a function of the set attenuation.

From the experimental results, it is possible to conclude that the attenuation that can be set on the WXC devices does not cause sensible changes in the DGD.

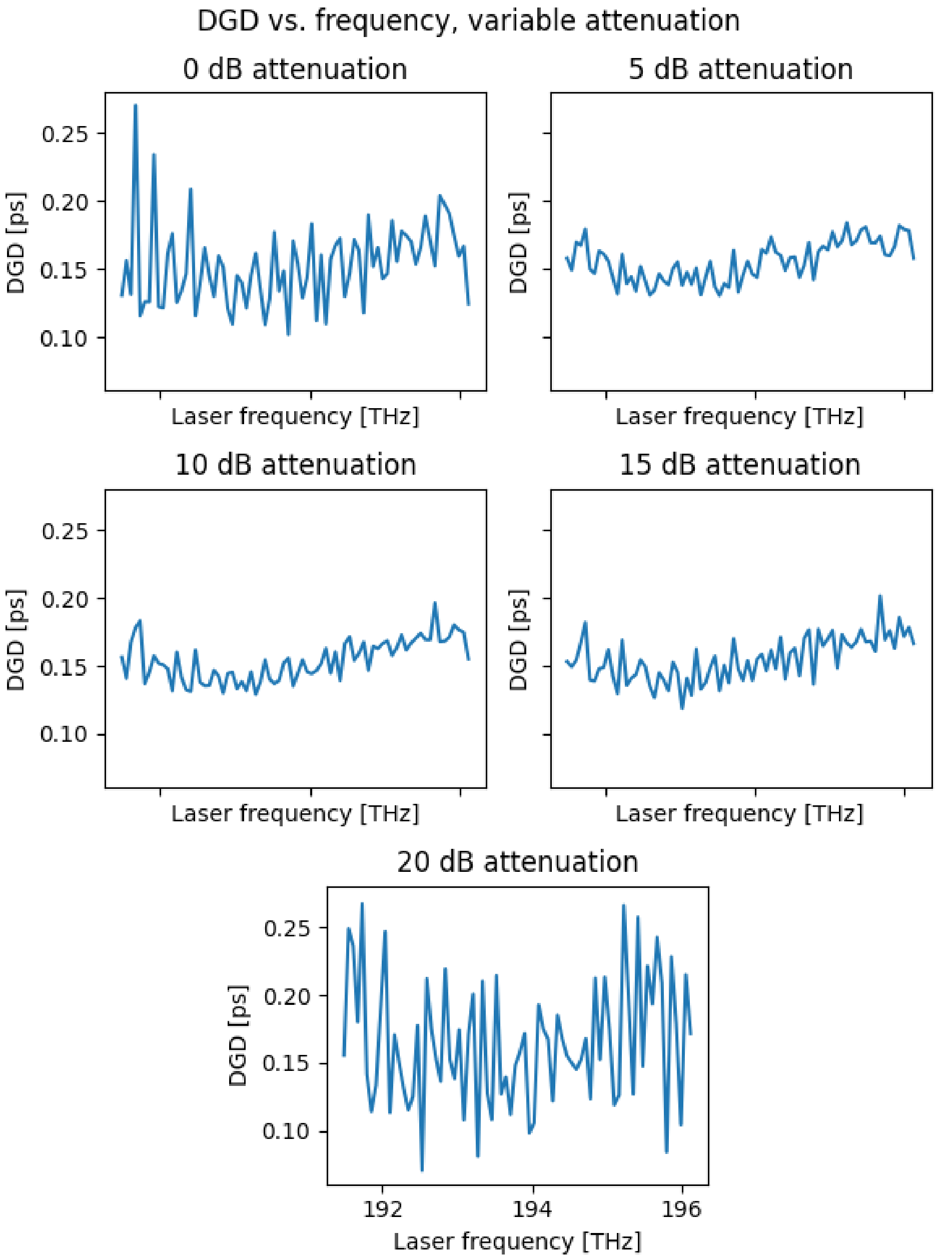


Figure 4.3: DGD spectrum vs. attenuation.

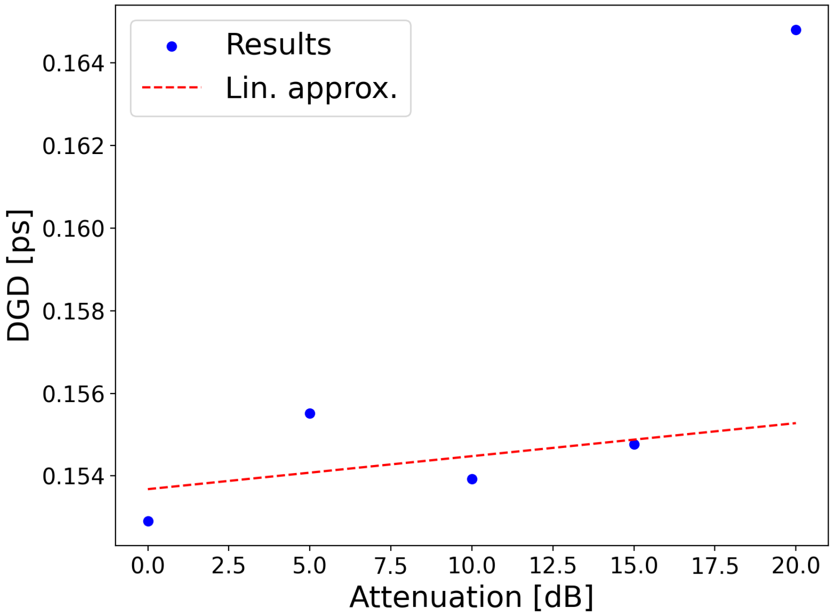


Figure 4.4: DGD spectrum vs. attenuation. The linear interpolation is shown, which does not account for the single high sensitivity data point (upper right corner).

4.3 DGD and WXC cascade

4.3.1 Test setup

The setup used to measure DGD as a function of the number of multiplexers and demultiplexers in series is the same as that arranged for the previous case. The only change is in the DUT, which is reported in Figure 4.5.

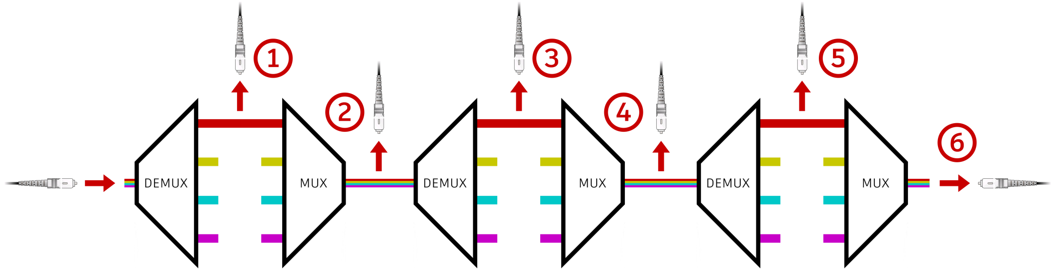


Figure 4.5: DGD cascade DUT. A series of multiplexers and demultiplexers is employed. The polarization analyzer is connected to various points in the chain (1 to 6) to perform the measurements.

The first device of the chain is a demultiplexer. The optical fiber which is connected back to the polarization analyzer is moved further along the device chain for each measurement.

4.3.2 DGD results

The results for the DGD measurements as a function of the laser source frequency for a cascade of a variable number of multiplexers and demultiplexers are reported in Figure 4.6.

We can see that the mean value of the DGD steadily increases with the number of devices in series, which is expected, as it would be assumed to be an additive quantity. As the cascade of multiple devices also causes a non-null signal attenuation due to insertion loss, even if the target attenuation is set to be 0 dB, when a number of about 5-6 devices is reached it is necessary to switch to the high sensitivity mode of the polarization analyzer. Also, the plot corresponding to the longest chain of devices, and therefore the highest attenuation, exhibits a significantly more noisy frequency curve, also due to the high sensitivity.

Figure 4.7 shows the mean value of the DGD across all frequencies of the span as a function of the set attenuation.

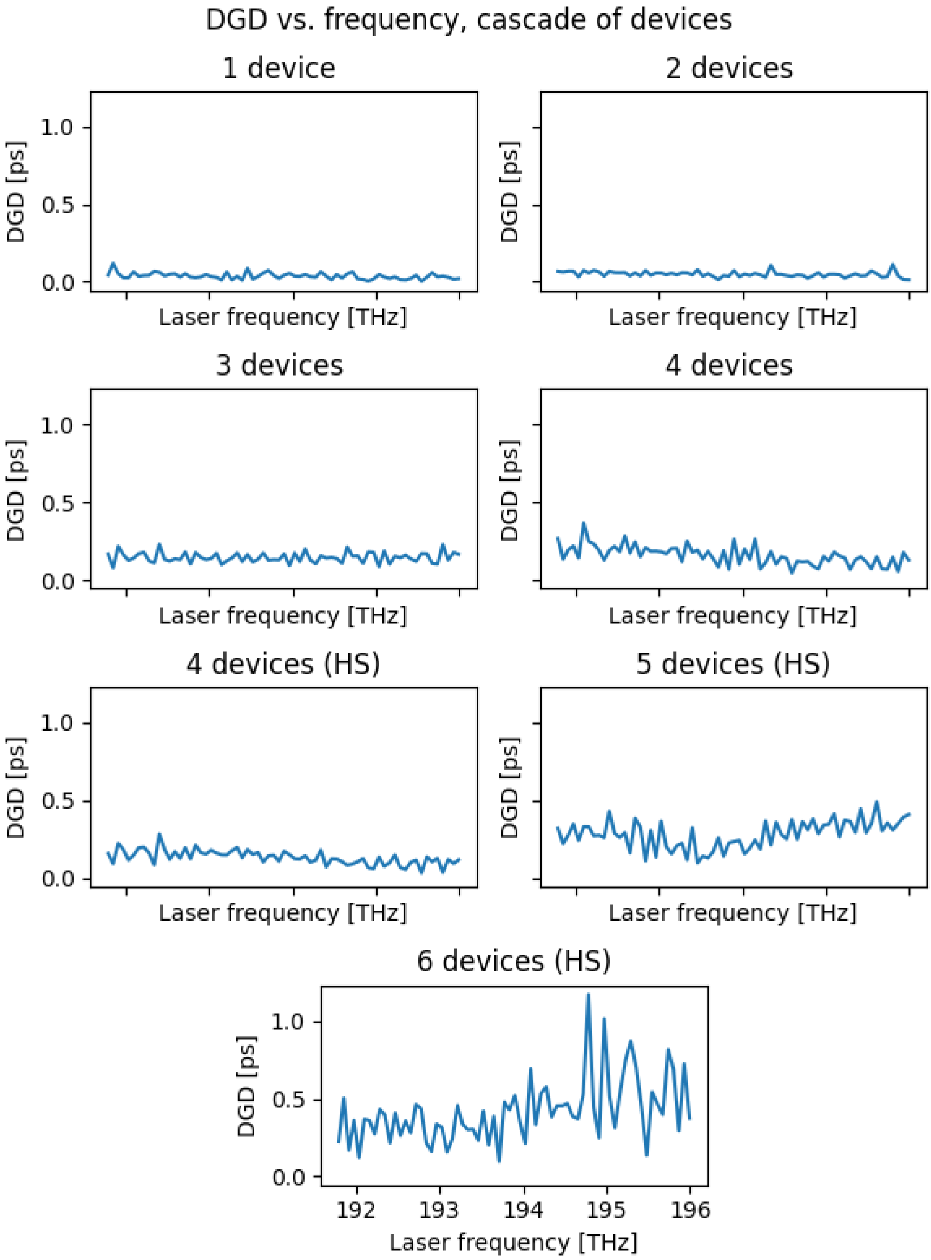


Figure 4.6: DGD spectrum vs. number of devices.

It is possible to see that, in high sensitivity, the addition of a device to the chain results in a higher difference in the DGD. The curve follows a linear trend, taking into account some deviation due to noise and instrument uncertainty, especially in high sensitivity.

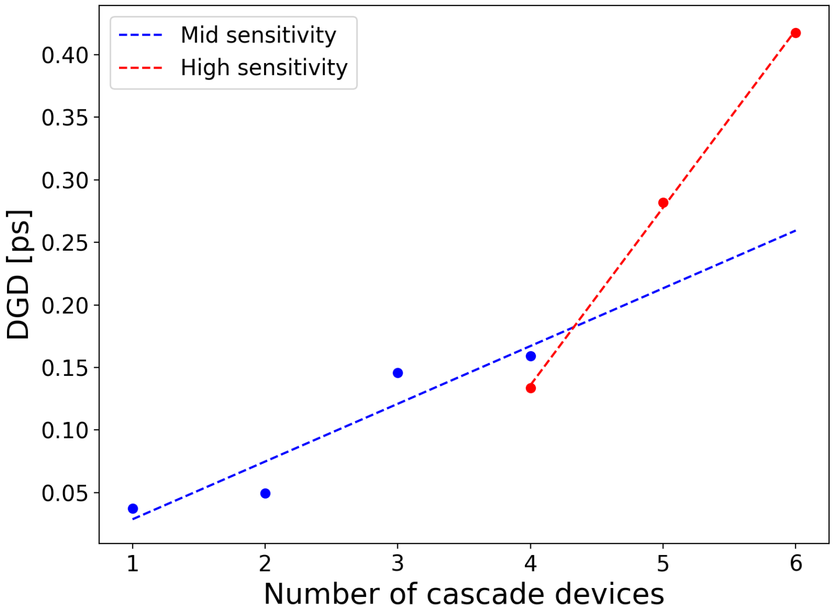


Figure 4.7: DGD spectrum vs. number of devices, highlighting the change in the linear interpolation slope for the different sensitivity settings.

4.4 Conclusions on DGD measurements

The results confirm a linear relationship between DGD and the number of connected devices. However, for accurate comparison across different setups, it's crucial to explicitly report the polarization analyzer sensitivity alongside DGD measurements, as it directly affects the final values.

Conclusions

The development of a polynomial model for the noise figure of Erbium Doped Fiber Amplifiers (EDFA) is presented. The model utilizes a machine learning algorithm to determine the polynomial coefficients, and aims to achieve high accuracy, as well as other qualities necessary for easy and efficient integration inside more comprehensive models, such as good computational speed, device independence, user-friendliness, and portability. The results indicate that the model demonstrates acceptable accuracy, comparable to the uncertainty in the dataset measurements. Its polynomial structure enables rapid nosie figure computation and makes it very light and portable, as the model files only contain the floating-point representation of the model coefficients.

The study also includes an analysis of the Differential Group Delay (DGD) affecting the internal devices of a Wavelength Cross Connect (WXC) card. This analysis aims at verifying the linear behavior of this parameter with respect to the number of traversed optical multiplexers and demultiplexers, and to determine any effect of the tunable device attenuation. The results show that variable attenuation does not impact DGD, and its behavior with the number of devices is , similar to that observed with consecutive optical fiber strains. The sensitivity of the optical instrumentation influences the slope of this linear behavior.

References

[1] Liu Yichen, Liu Xiaomin, Zhang Yihao, Cai Meng, Fu Mengfan, Zhong Xueying, Yi Lilin, Hu Weisheng, and Zhuge Qunbi. Building a digital twin of EDFA: a grey-box modeling approach. pages 1–2, 2023.

[2] Zhu S., Gutterman C., Montiel A. D., Yu J., Ruffini M., Zussman G., and Kilper D. Hybrid Machine Learning EDFA Model. *Optica Publishing Group*, 2020.

[3] Liu Yichen, Liu Xiaomin, Liu Lei, Zhang Yihao, Cai Meng, Yi Lilin, Hu Weisheng, and Zhuge Qunbi. Modeling EDFA Gain: Approaches and Challenges. *Photonics*, 8(10), 417, 2021.

[4] Vittorio Curri, Mattia Cantono, and Roberto Gaudino. Elastic All-Optical Networks: A New Paradigm Enabled by the Physical Layer. How to Optimize Network Performances? *J. Lightwave Technol.*, 35(6):1211–1221, Mar 2017.

[5] Vittorio Curri. Synergistic Use of AI and Physics Models in Planning and Controlling Multi-Band Optical Networks. pages 1–2, 2023.

[6] Rocco D'Ingillo, Andrea D'Amico, Renato Ambrosone, Emanuele Virgillito, Vittorio Gatto, Stefano Straullu, Francesco Aquilino, and Vittorio Curri. Deep Learning Gain and Tilt Adaptive Digital Twin Modeling of Optical Line Systems for Accurate OSNR Predictions. 28th International Conference on Optical Network Design and Modelling, 2024.

[7] Yuren You, Zhiping Jiang, and Christopher Janz. Machine Learning-Based EDFA Gain Model. pages 1–3, 09 2018.

[8] Shengxiang Zhu, Craig Gutterman, Alan Diaz Montiel, Jiakai Yu, Marco Ruffini, Gil Zussman, and Daniel Kilper. Hybrid Machine Learning EDFA Model. page T4B.4, 01 2020.

[9] Giacomo Borraccini, Vittorio Gatto, Andrea D'Amico, Stefano Straullu, Francesco Aquilino, Antonino Nespola, Stefano Piciaccia, A. Tanzi, Gabriele Galimbert, and Vittorio Curri. Gain Profile Characterization and Modeling for Dual-Stage EDFA Abstraction and Control. *IEEE Photonics Technology Letters*, PP:1–1, 01 2023.

[10] Andrea D'Amico, Stefano Straullu, Antonino Nespola, Ihtesham Khan, Elliot London, Emanuele Virgillito, Stefano Piciaccia, Aberto Tanzi, Gabriele Galimberti, and Vittorio Curri. Using machine learning in an open optical line system controller. *Journal of Optical Communications and Networking*, 12, 02 2020.

[11] Al-Amri Mohammad D., El-Gomati Mohamed M., and Zubairy M. Suhail. *Optics in Our Time*. Springer, 2016.

[12] John Bray. The Engineers of the Early Multichannel Telephony Coaxial Cable Systems. pages 135–151, 1995.

[13] Takanori Okoshi and Kazuro Kikuchi. Frequency stabilisation of semiconductor lasers for heterodyne-type optical communication systems. *Electronics Letters*, 16:179–181, 1980.

[14] Michael G. Taylor. Phase Estimation Methods for Optical Coherent Detection Using Digital Signal Processing. *Journal of Lightwave Technology*, 27(7):901–914, 2009.

[15] M.G. Taylor. Coherent detection method using DSP for demodulation of signal and subsequent equalization of propagation impairments. *IEEE Photonics Technology Letters*, 16(2):674–676, 2004.

[16] R-J Essiambre, G Kramer, PJ Winzer, GJ Foschini, and B Goebel. Capacity limits of optical fiber networks. *J. Lightwave Technol. 28:662–701*, 2010.

[17] National Institute of Information, Communications Technology, Aston Institute of Photonic Technologies, and Nokia Bell Labs. World Record 402 Tb/s Transmission in a Standard Commercially Available Optical Fiber, Jun 2024.

[18] Rodney Tucker, Jayant Baliga, Robert Ayre, Kerry Hinton, and Wayne Sorin. Energy Consumption in Optical IP Networks. *Journal of Lightwave Technology*, 27, 10 2008.

[19] Infinera Corporation. Future-proof Open Line Systems for Cloud Scale Networks. *White paper*, 2019.

[20] Yongcheng Li, Li Gao, Gangxiang Shen, and Limei Peng. Impact of roadm colorless, directionless, and contentionless (cdc) features on optical network performance. *J. Opt. Commun. Netw.*, 4(11):B58–B67, Nov 2012.

[21] Lumentum Operations LLC. ROADM Graybox: ROADM-20, 2024.

[22] Inc. Cisco Systems. Enhanced C-Band 96-Channel EDFA Amplifiers for the Cisco ONS 15454 MSTP Data Sheet, Jan 2016.

[23] Inc. Fujitsu Network Communications. Open Optical Transformation. *White paper*, 2018.

[24] Infinera Corporation. Open Optical Networks: Doing More With Less. *White paper*, 2019.

[25] Redpath Ian and Omdia. Open Optical Networks: A pratical guide for open line adoption for CSPs. *White paper*, 2022.

[26] Paul Goransson, Chuck Black, and Timothy Culver. *Software Defined Networks: A Comprehensive Approach.* Elsevier, 2016.

[27] Diego Kreutz, Fernando M. V. Ramos, Paulo Esteves Veríssimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*, 103(1):14–76, 2015.

[28] Yonghan Wu, Min Zhang, Lifang Zhang, Jin Li, Xue Chen, and Danshi Wang. Dynamic Network Topology Portrait for Digital Twin Optical Network. *Journal of Lightwave Technology*, 41(10):2953–2968, 2023.

[29] R. Vilalta, R. Casellas, Ll. Gifre, R. Muñoz, R. Martínez, A. Pastor, D. López, and J.P. Fernández-Palacios. Architecture to Deploy and Operate a Digital Twin Optical Network. In *2022 Optical Fiber Communications Conference and Exhibition (OFC)*, pages 1–3, 2022.

[30] Génesis Villa, Christian Tipantuña, Danny S. Guamán, Germán V. Arévalo, and Berenice Arguero. Machine Learning Techniques in Optical Networks: A Systematic Mapping Study. *IEEE Access*, 11:98714–98750, 2023.

[31] Alexander William Setiawan Putra, Minoru Yamada, Hiroyuki Tsuda, and Sumiaty Ambran. Theoretical Analysis of Noise in Erbium Doped Fiber Amplifier. *IEEE Journal of Quantum Electronics*, 53(4):1–8, 2017.

[32] Douglas M. Baney, Philippe Gallion, and Rodney S. Tucker. Theory and Measurement Techniques for the Noise Figure of Optical Amplifiers. *Optical Fiber Technology*, 6(2):122–154, 2000.

[33] Carlton M. Caves. Quantum limits on noise in linear amplifiers. *Phys. Rev. D*, 26:1817–1839, Oct 1982.

[34] Gyeong-il Kweon. Noise figure of optical amplifiers. *Journal of the Korean Physical Society*, 41:617–628, 11 2002.

[35] Masaki Asobe, Takeshi Umeki, and Osamu Tadanaga. Phase sensitive amplification with noise figure below the 3 db quantum limit using CW pumped PPLN waveguide. *Optics Express*, 20:13164–72, 05 2012.

[36] Agilent Technologies. EDFA Testing with the Interpolation Technique, 2000.

[37] W. S. Dorn. A generalization of Horner's rule for polynomial evaluation. 1961.

[38] Jan Litvik, Daniel Benedikovič, Marc Wuilpart, Milan Dado, and Michal Kuba. Numerical model for DGD estimation in optical transmission system. *Proceedings of SPIE - The International Society for Optical Engineering*, 8772, 04 2013.

[39] Alan Eli Willner, SM Reza Motaghian Nezam, Zhongqi Pan, and Qian Yu. Performance degradations and monitoring in optical networks. 5282:261–274, 2004.

Acknowledgements

Ringrazio tutti i membri del gruppo Planet per l'opportunità di svolgere questa tesi e il supporto quotidiano lungo il percorso.
Un ringraziamento particolare va a Rocco D'Ingillo, del gruppo Planet, e a Stefano Straullu, della fondazione Links, con cui ho potuto condividere maggiormente i miei progressi, i dubbi e le difficoltà.
Ringrazio Andrea Rosso, Enrico Miotto, Riccardo Schips e Vittorio Gatto, con cui ho condiviso il periodo di scrittura della tesi, per aver sempre reso l'ufficio un luogo animato, sereno, fresco e piacevole.

Ringrazio la mia famiglia, su cui ho sempre sentito di potermi appoggiare, per il sostegno, la comprensione, la vicinanza e la volontà di non farmi mancare nulla, dai piccoli gesti quotidiani alle cose più importanti. Sono grato di aver vissuto in un ambiente che mi ha concesso di dedicarmi con serenità allo studio e alle passioni coltivate negli ultimi anni.

Ringrazio i miei amici, con cui ho potuto condividere difficoltà, gioia, pressione e soddisfazione del mio percorso e che mi hanno sempre supportato con le parole e la compagnia. Un ringraziamento speciale ai gruppi di giochi di ruolo e da tavolo, per avermi offerto momenti di risate, svago e spensieratezza in cui poter ricaricare le energie, che mi hanno permesso di affrontare questo cammino con più leggerezza.

Ringrazio Marco Gregio, che, trascinandomi in svariati progetti, mi ha fatto riscoprire la passione di creare qualcosa con le mie mani.