Pontifícia Universidade Católica
do Rio de Janeiro

Felipe Calliari

High-Resolution OTDR with Embedded Precise Fault Analysis

Tese de Doutorado

Thesis presented to the Programa de Pós–Graduação em Engenharia Elétrica of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia Elétrica.

Advisor : Prof. Gustavo Castro do Amaral
Co-advisor: Prof. Michael Lunglmayr

Rio de Janeiro
September 2021

Pontifícia Universidade Católica
do Rio de Janeiro

Felipe Calliari

High-Resolution OTDR with Embedded Precise Fault Analysis

Thesis presented to the Programa de Pós–Graduação em Engenharia Elétrica of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia Elétrica. Approved by the Examination Committee:

Prof. Gustavo Castro do Amaral
Advisor
Centro de Estudos em Telecomunicações – PUC-Rio

Prof. Michael Lunglmayr
JKU

Prof. Guilherme Penello Temporão
Centro de Estudos em Telecomunicações – PUC-Rio

Dr. Marcelo Roberto Baptista Pereira Luis Jimenez
Centro de Estudos em Telecomunicações – PUC-Rio

Prof. Alan Conci Kubrusly
Centro de Estudos em Telecomunicações – PUC-Rio

Dr. Fernando Alves Rodrigues
INMETRO

Dr. Rogerio Passy
MLS Wireless

Rio de Janeiro, September 1st, 2021

Felipe Calliari

Felipe Calliari graduated from the Pontifícia Universidade Católica do Rio de Janeiro (Rio de Janeiro, Brasil) in Electrical Engineering with both Computer Electronics and Telecommunications emphases and later received a M.S. degree in Electrical Engineering in 2017 from the same university. He is a member of the Optoelectronics Laboratory of Center for Telecommunication Studies in PUC-Rio.

Bibliographic data

Calliari, F.

High-Resolution OTDR with Embedded Precise Fault Analysis / Felipe Calliari; advisor: Gustavo Castro do Amaral; co-advisor: Michael Lunglmayr. – 2021.

123 f: il. color. ; 30 cm

Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2021.

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Engenharia de Telecomunicações – Teses. 3. FPGA. 4. Linearized Bregman Iterations. 5. Comunicações Ópticas. 6. Monitoramento de Fibras Ópticas. 7. Reflectômetro Óptico no Domínio do Tempo. 8. Optoeletrônica.. I. Amaral, G. C.. II. Lunglmayr, M.. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 621.3

This thesis is dedicated to the memory of my grandfather,
Ilar Mário Calliari

Acknowledgments

I would like to thank my wife, Beatriz Mynssen, for the unconditional support, companion and love, without which this thesis could not have been written.

To my advisers and friends, Gustavo Amaral and Michael Lunglmayr, my sincere and deep gratitude for guidance, incentive and opportunity.

To all my family and friends, who directly or indirectly contributed to this work.

To my colleagues and friends of PUC–Rio for the support and friendship.

I would also like to thank to the brazilian agency CNPq for the financial support, without which this work would not have been realized.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Calliari, F.; Amaral, G. C. (Advisor); Lunglmayr, M. (Co-Advisor). **High-Resolution OTDR with Embedded Precise Fault Analysis**. Rio de Janeiro, 2021. 123p. Tese de Doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Optical fibers are susceptible to mechanical stress and may be damaged or broken, thus physical layer supervision is essential to identify these failures and remediate them as quickly as possible. In order to hasten and simplify the scheduling process of the in-field repairing units, an automated fiber measurement system based on a digital signal processing (DSP) unit capable of autonomously identifying fault positions was developed. By combining this DSP unit with a Tunable Photon-Counting OTDR, it is possible to create such a device. This work presents the development of the building blocks for such device.

Keywords

FPGA; Linearized Bregman Iterations; Optical Communications; Optical Fiber Monitoring; Optical Time Domain Reflectometry; Optoelectronics.

Resumo

Calliari, F.; Amaral, G. C.; Lunglmayr, M.. **OTDR de alta-resolução com análise precisa de falhas integrada**. Rio de Janeiro, 2021. 123p. Tese de Doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

As fibras ópticas são suscetíveis ao estresse mecânico e podem ser danificadas ou quebradas, portanto, a supervisão da camada física é essencial para identificar essas falhas e remediá-las o mais rápido possível. Com o objetivo de agilizar e simplificar o processo de agendamento de unidades de reparo em campo, foi desenvolvido um sistema automatizado de medição de fibras baseado em uma unidade de processamento digital de sinal (DSP) capaz de identificar as posições das falhas de forma autônoma. Combinando esta unidade de DSP com um OTDR de contagem de fótons, é possível criar tal dispositivo. Este trabalho apresenta o desenvolvimento dos blocos de construção para tal dispositivo.

Palavras-chave

FPGA; Linearized Bregman Iterations; Comunicações Ópticas; Monitoramento de Fibras Ópticas; Reflectômetro Óptico no Domínio do Tempo; Optoeletrônica.

Table of contents

List of figures

List of tables

List of Abreviations

APD	Avalanche Photodiode
AWG	Arrayed Waveguide Grating
BRAM	Block RAM
CCLs	CMOS Configuration Latches
CLB	Configurable Logic Blocks
CMOS	Complementary Metal-Oxide Semiconductor
CORDIC	COordinate Rotation DIgital Computer
DFB	Distributed Feedback Laser
DS	Dispersion-Shifted Fiber
DSP	Digital Signal Processing
DUT	Device Under Test
DWDM	Dense Wavelength-Division Multiplex
EDFA	Erbium-Doped Fiber Amplifier
F8-PMLL	Figure-Eight Passive Mode-Locked Laser
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
FSR	Free-Spectral Range
FUT	Fiber Under Test
HD	High-Dynamic
HDL	Hardware Description Language
HR	High-Resolution
I/O	Input and Output
IP	Intellectual Property
IOB	Input and Output Blocks
LASER	Light Amplification by Stimulated Emission of Radiation
LASSO	Least Absolute Shrinkage and Selection Operator
LBI	Linearized Bregman Iterations
LUT	Look-Up Table
MCC	Matthews Correlation Coefficient
MMF	Multi-Mode Optical Fiber
MUX	Multiplexer
NALM	Nonlinear Amplifying Loop Mirror
NEP	Noise-Equivalent Power
NZDSF	Non-Zero Dispersion-Shifted Fiber
OFDR	Optical Frequency-Domain Reflectometer
OLCR	Optical Low-Coherence Reflectometer
OLS	Ordinary Least Square
OMP	Orthogonal Matching Pursuit
OSI	Open Systems Interconnection
OTDR	Optical Time-Domain Reflectometer

PAR	Place And Route
PAT	Parallel Adder Tree
PC	Polarization Controller
PC-OTDR	Photon-Counting OTDR
PLD	Programmable Logic Device
PLM	Physical Layer Monitoring Unit
PMLL	Passive Mode-Locked Laser
PMT	Parallel Multiplexer Tree
PON	Passive Optical Network
RAM	Random-Access Memory
RTL	Register Transfer Level
SFIXED	Signed Fixed-Point
SMF	Single-Mode Optical Fiber
SNR	Signal-to-Noise Ratio
SOA	Semiconductor Optical Amplifier
SPAD	Single Photon Avalanche Detector
TBD	Trend Break Detection
TDC	Time-to-Digital Converter
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
VSLI	Very-Large-Scale Integration
WDM	Wavelength-Division Multiplex
ν-OTDR	Photon-Counting OTDR

Intelligence is the ability to avoid doing work,
yet getting the work done.

Linus Torvalds.

1
Introduction

Human beings are social beings by nature and, therefore, have the need to communicate. Communication can be defined as the process by which we exchange information, and it is an integral part of our lives [1]. There are at least four forms of communication: verbal, non-verbal, written and visual. Among these, the main form of communication is writing, which is used from ancient times to modern times as the main form of record of human history and knowledge [1]. Long-distance communications were limited to mail, carrier pigeon, among others. In ancient times, the forms of long-distance communication were limited to the post, carrier pigeon, among others. In the 19th century, however, we saw the beginning of the era of electrical communications, with the rise of the electric telegraph and, later, the invention of the telephone. These systems transmitted the information directly through the communication channel as baseband analog signals. As communication systems evolved, we started to use larger bands of the electromagnetic spectrum and, later, modulated signals as a way to increase the capacity to transmit information. At that time, coaxial cable and microwave systems operated at bit rates of approximately 100 Mbit/s. Then, around 1950, optical fibers were suggested as the best solution for data transmission due to its almost unlimited bandwidth [2,3]. Nevertheless, it was only after the 1980s that optical communications systems became commercially viable, due to the reduction of fiber losses and the development of lasers, optical amplifiers and photon detectors [4–6].

In the early 1990s, with the advent of the internet and its popularization, telecom companies saw the demand for bandwidth went through the roof. The main reasons behind this growth are the growing interest of the population in data exchange and the emergence of new types of services, such as video streaming. For example, in a report [7] dated from 2020, Netflix and Youtube, two of the biggest video streaming services, reported they would reduce streaming quality during one month in Europe to prevent the internet collapsing. Thus, in the last years, we have seen the massive deployment of optical fibers promoted by the ever-increasing need for higher data transmission capacity. Despite great success, optical fibers are susceptible to mechanical stress and, under certain conditions, can be damaged or broken. For this reason, physical layer supervision is essential to identify these failures and remediate them as quickly as possible to ensure the robustness of all layers of the network, as a failure can cause the interruption of essential services, such as bank, telephone or internet services links. An example of an “essential service” outage occurred in 2013, when the Amazon.com’s homepage was down for approximately 45 minutes [8]. Although the cause of this event is unknown, it is estimated that this interruption cost the company $5 million.

Among optical reflectometry techniques, the Optical Time-Domain Reflectometry (OTDR) is the most widely used technique for fiber monitoring due to its long-haul, high-resolution, and tunability monitoring capabilities. Through the use of an OTDR it is possible to extract information about the optical link by accessing only one end of the fiber. Thereby a network operator located at a central transmission station can monitor all the fibers connected to it and schedule an in-field repairing unit as needed. However, a network operator might have several thousand optical fibers under his responsibility [9, 10]. Therefore, an automated fiber measurement system based on a digital signal processing unit capable of identifying fault positions in a fiber profile could alleviate their workload and, eventually, hasten and simplify the scheduling process of the in-field repairing units [11]. As OTDR systems frequently make use of FPGAs [11–13], an automatic fault detection system could be easily implemented in an FPGA. Thus, it is of special interest the implementation of the Linearized Bregman Iterations for Trend Break Detection presented in [14, 15], since it was specially tailored for implementation in an FPGA.

1.1 Outline of the Thesis

This thesis is organized as follows:

- Chapter 1 introduces the problem.
- In Chapter 2, we will introduce important concepts that may be needed in the later chapters. A brief presentation is made about the history and some important concepts of optical fibers and optical networks, then some of the most known fiber monitoring techniques and its characteristcs. Finally, we will show a brief summary of the Linearized Bregman Iterations algorithm, and a introduction about FPGA.
- In Chapter 3 we will present the results of a paper about a new acquisition mode for the High Resolution ν-OTDR. One of the main improvements is the utilization of sliding detection windows to acquire the fiber profile,

which improved the measurement time required to obtain a certain dynamic range.

- Chapter 4 briefly presents the implementation of the Linearized Bregman Iterations (LBI) tailored for Trend Break Detection (TBD) in an FPGA. Some considerations are also made about the implementation of the hardware and control structures, as well as, the resulting maximum clock frequency and the processing time for some levels of parallelism — denoted by the number of BRAMs. Finally, we present briefly performance comparison between the Orthogonal Matching Pursuit (OMP) and the LBI.
- In Chapter 5 we will present the results of a new methodology, called profile-splitting, which allows for high-dimensional datasets to be processed within reasonable times without the loss of performance. As well as an analysis of the modulation in λ according to the square root of a measurement's estimated SNR which yielded high-quality results.
- Finally, in Chapter 6 we will conclude this thesis with a summary and a perspective of possible future research.
- Appendix A lists the published papers during the doctorate period.
- Appendix B, Appendix C, and Appendix D present copies of the papers related to this work.

2
Theoretical Review

2.1
Fiber Optics

Brief History of Fiber Optics

An optical fiber is a dielectric monofilament structure that carries electromagnetic energy — in the form of electromagnetic radiation with frequencies on the order of $\sim 10^{14}$ Hz — through its core. Optical fibers, as we know them today, are made of ultra-pure silica (ultra-pure silicon dioxide SiO_2), a kind of transparent, and flexible glass with a straight section of diameter in the order of μm. Light is guided inside the fiber through the phenomenon of total internal reflection which is a consequence of the refractive index difference between the core and the cladding. This phenomenon was first documented in 1854 by John Tyndall. However, the first coated optical fibers were only conceived around 1950 by Brian O'Brien [16], when he discovered that the key to efficient light transmission was having a low difference in refractive indexes.

The next big breakthrough toward popularizing fiber optics came in 1961 when Theodore Maiman created the first LASER (Light Amplification by Stimulated Emission of Radiation). Theodore Maiman's LASER produced a narrow beam of monochromatic light with wavelength of approximately 694.3 nm [17, 18]. By operating in baseband of the order of $\sim 10^{14}$ Hz, laser systems can transmit large amounts of information when compared to other communication systems (electrical, radiowave and microwave).

However, at that time, optical fibers had power losses above 1000 dB/km. In 1970, fiber losses were be reduced to below 20 dB/km, to $\lambda \sim 1$ μm, as a result of work carried out at that time with the aim of reducing impurities in the glass. Near 1970, continuous operation of GaAs-based lasers at room temperature was demonstrated [4, 19]. It is important to note that Theodore Maiman's Laser worked in the pulsed regime.

In 1972, the attenuation of Multi-Mode Optical Fibers (MMF) at an operating wavelength of about 850 nm was $\sim$4 dB/km. The wavelength range between 700 nm – 900 nm is known as "the first optical windows", this window was very attractive because of the low cost optical sources and detectors in this band and the minimum attenuation occurred at 820 μm. Currently, fiber losses in this window are ∼3 dB/km which is near the theoretical limit [20–22].

Still in the 1970s, interest in the region of wavelengths of the order of 1300 nm, known as the "second optical window" or minimum dispersion window, arose because of its 0.5 dB/km attenuation, but it only came into use in the mid 1980s with the invention of the first GaInAs photodiodes in 1977 by Pearsall [23–25]. The GaInAs photodiodes operate in the wavelength range 1100 nm $< \lambda <$ 1700 nm. The third optical window, also known as the minimum attenuation window, presents the theoretical minimum optical loss, of about 0.2 dB/km, for silica-based fibers, see Figure 2.1, and its utilization was delayed because of the large dispersion around 1550 nm and the difficulty of operating in this band since (optical sources and detectors are quite expensive) [19]. Later, Non-Zero Dispersion-Shifted Fiber (NZDSF) were created, which guaranteed connections with data traffic above 40 Gb/s [6].

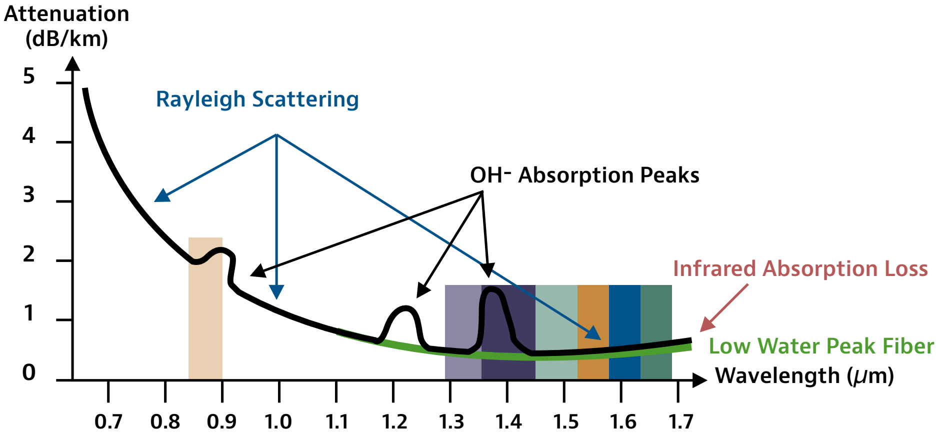


Figure 2.1: Fiber attenuation as a function of wavelength [20].

Optical Fiber

An optical fiber is a crystalline structure monofilament element composed of ultra-pure SiO_2 with dielectric properties capable of carrying electromagnetic energy in the range of optical and infrared frequencies. Optical fibers are composed of a core, a cladding, and some kind of protective coating, as seen in Figure 2.2. The cladding has the function of providing a lower refractive index (n_2) than the core (n_1), that is $n_2 < n_1$, to reduce the dispersion caused by changes in the refractive index at the interface between the two media, while the plastic coating's purpose is to protect the fiber.

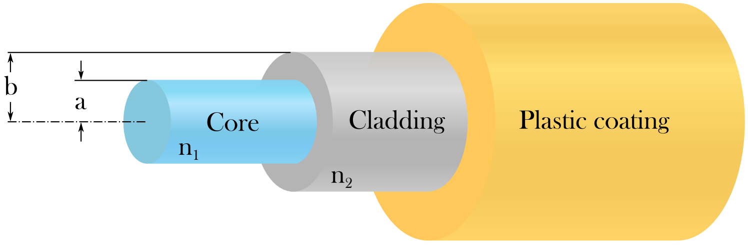
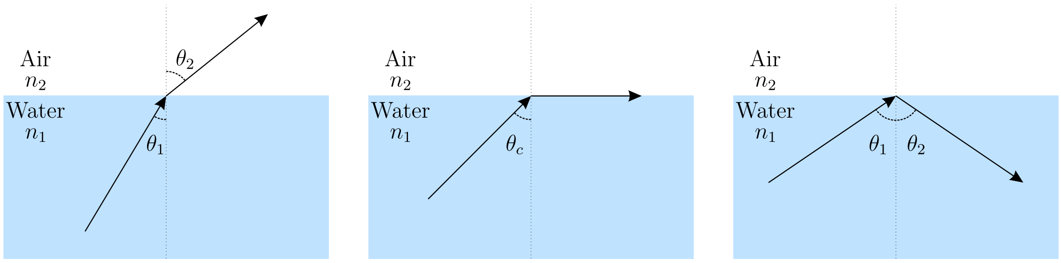


Figure 2.2: An optical fiber with an inner core, an cladding and an outer plastic coating.

The light propagates inside the fiber core through the phenomenon of total reflection of light that occurs when the angle of incidence is greater than the critical angle. Through Snell-Descartes' law, Equation 2-1, it is possible to see that when $\theta_2 > 90°$ the light is trapped inside the optical fiber which acts as a waveguide, see Figure 2.3.

$$n_1 \sin\theta_1 = n_2 \sin\theta_2 \tag{2-1}$$



(a) Refraction (b) Critical angle (c) Total internal reflection

Figure 2.3: Refraction of light at the border between two media [26].

Optical fibers can be classified as Single-Mode Optical Fiber (SMF) or Multi-Mode Optical Fiber (MMF), depending on the number of modes they can carry. For Multi-Mode Fiber, it is possible to *estimate* the number of modes supported by it through $M \approx V^2/2$, where V is defined by

$$V = \frac{2\pi a}{\lambda} \cdot \sqrt{n_1^2 - n_2^2} \tag{2-2}$$

where a is the radius of the fiber core, λ is the light wavelength, n_1 and n_2 are the refractive indices of the core and the cladding, respectively. A fiber will support only one propagation mode, i.e., it will be an SMF, if the parameter V, often called normalized frequency, is lower than 2.405, which is the first root of the Bessel function J_0. In this condition, $V < 2.405$, only the HE_{11} mode is able to propagate inside the fiber and this mode only ceases to exist when the radius of the fiber core is zero [6].

In general, SMF has a core size that range from 8 to 10 μm, has higher spectral efficiency, i.e., it has higher-bandwidth and can support even more traffic by using several wavelengths when used in combination with Wavelength Division Multiplexers (WDM), are suitable for long distance transmissions, and have lower attenuation than MMF, equipment used for SMF is more expensive than for MMF, coupling SMF is more difficult, in fact, the first SMF connectors with losses of less than 3 dB ($\sim$ 0.2 dB) only appeared around 1983 [27,28], and the main source of attenuation is the chromatic dispersion — each wavelength travels at a different speed, arriving at the destination at different times, thus causing pulse broadening.

On the other hand, MMF has a core size that range from 50 to 62.5 μm, the modes exchange energy through a nonlinear and non-reciprocity mode-coupling process [25,29,30], are suitable for medium distance transmissions due to being less expensive, according to the core's radial refractive index variation profile, see Figure 2.4, they can be classified as step-index and graded-index MMF, and the main source of attenuation in MMF is the modal dispersion — in which different propagation modes take different paths, arriving at the destination at different times, thus causing pulse broadening. Graded-index MMF is designed to reduce the modal dispersion inherent in step-index MMF, therefore, they have higher-bandwidth in exchange for being more expensive and more difficult to produce. In the typical MMF graduated index, the core's refractive index profile decays, usually, parabolically to the cladding's refractive index.

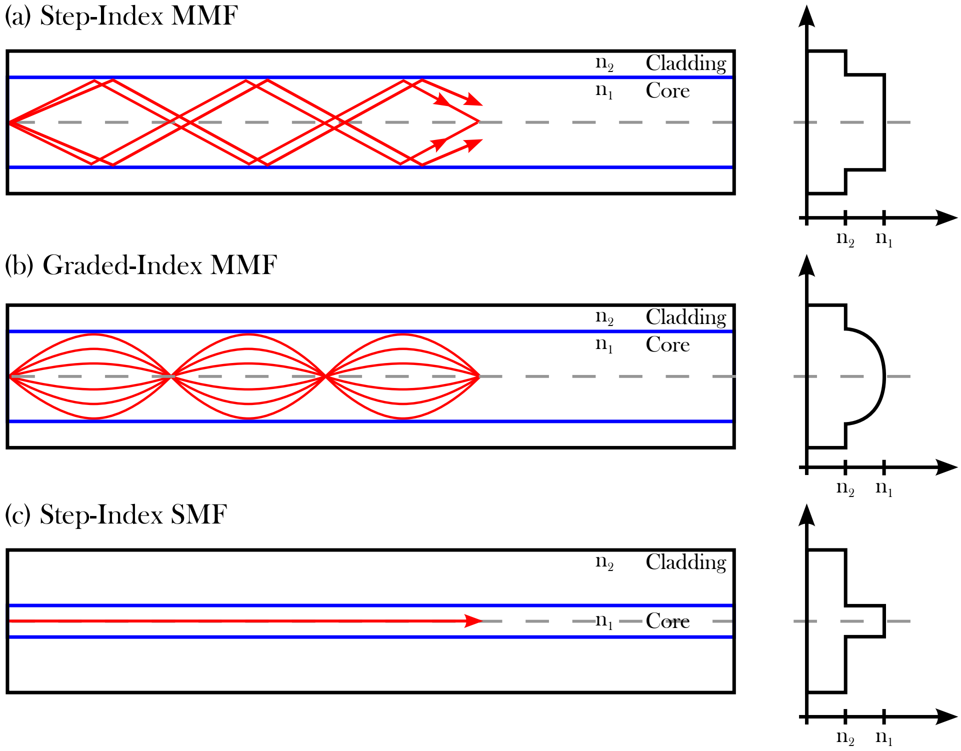


Figure 2.4: Optical fiber cross section showing ray propagation and radial refractive index profile.

2.2 Optical Networks

Ever since mankind began to live in society, man began to feel the need to communicate with each other [26]. Although there are several ways to communicate, we can classify them as verbal, nonverbal, written, and visual communication. Among these, the main form of communication is writing, which is used from ancient times to modern times as the main form of record of human history and knowledge [1]. Long-distance communications were limited to mail, carrier pigeon, among others. The first high-speed communication system was the Semaphore (Optical Telegraph), built by Charles Chappe in 1792, which was capable of transmitting messages over distances of ∼200 km in 15 minutes but was dependent on favorable weather conditions [31]. After Morse invented the electric telegraph in 1837, being capable of transmitting up to 120 bits/s through the use of qualified operators [6], Chappe's system became obsolete.

The electric telegraph is considered to be the beginning of the age of electrical communications. However, it fell into disuse after the invention of the telephone by Alexander Graham Bell. Bell's telephone transmitted the human voice in the form of a baseband analog signal, i.e., the signal was transmitted directly over the communication channel. As the amount of information grew, we started to use larger bands of the electromagnetic spectrum and, later, modulated signals as a way to increase the capacity to transmit information.

In the late 1950s, scientists started looking to optics as a possible solution for enhancing the capacity of telecommunications systems [2]. However, the first optical communication systems only became commercially viable after the 1980s, due to the reduction in fiber optic losses and the development of key elements such as semiconductor lasers, optical amplifiers, and photodetectors [4–6]. In the beginning, optical networks were mainly used as trunk lines by telephone companies, cable TV, among others, to provide network access for several clients while sharing a set of equipment. A trunk line consists of a high-capacity link composed of a single cable or several cables, or optical fibers, to transmit multiplexed signals to multiple users or broadcast signals to a large group of users [5].

Over 27 years, the advances in fiber-optic communication systems have been so great and fast that the period spanning 1975 to 2002 can be divided into five generations [4]. Figure 2.5 shows the evolution and maturation of fiber-optic technology as the BL product begins to saturate. The BL product is a commonly used figure of merit for fiber-optic communication systems that shows the maximum bit rate (B) given a certain spacing (L) between the repeaters.

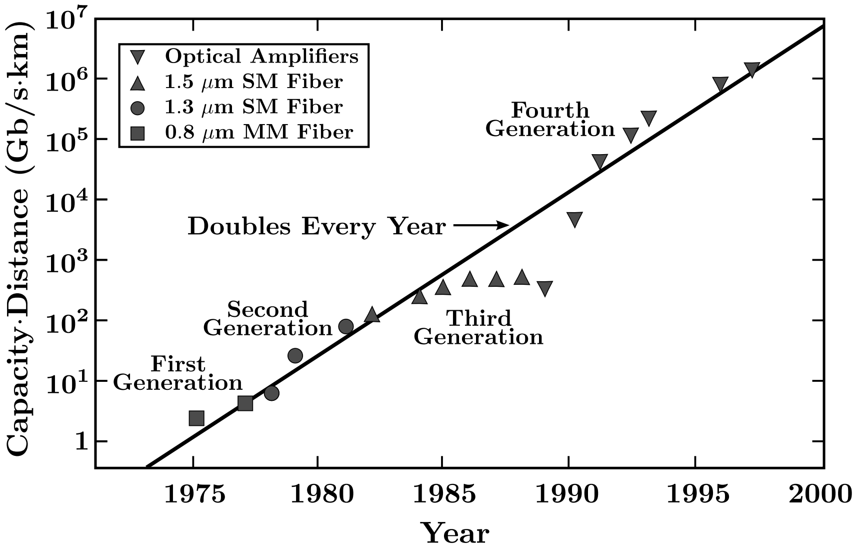


Figure 2.5: Increase in the BL product over the period 1975 to 2000 though four generations of fiber-optic communication systems [4].

The first generation of fiber-optic communication systems only became available commercially in 1980, employing GaAs semiconductor lasers operating near 850 nm. This generation did not last long as fiber losses were 3 dB/km, and these systems operated at bit rates of 45 Mb/s through the use of up to 10 km-repeater spacings. The second generation operated at the second optical window, which was known for its minimum dispersion and fiber losses below 1 dB/km, operating at a wavelength near 1.3 μm. The maximum bit rate of this generation was limited to 100 Mb/s in MMF or 1.7 Gb/s in SMF with repeaters spaced about 50 km apart. The third generation was delayed due to the high dispersion near 1.55 μm even though it presented fiber losses of the order of 0.2 dB/km, which is the limit fundamental imposed by Rayleigh scattering, and only came into operation in 1990 through the use of Dispersion-Shifted (DS) SMF and the development of Distributed Feedback Lasers (DFBs) that had a single longitudinal mode dominant. This generation could transmit information at bit rates of up to 10 Gb/s over a distance of up to 100 km.

With the advent of the internet and its popularization in the early 1990s, telecom companies saw the demand for bandwidth went through the roof. By 1996, the fourth generation became available, deploying the Wavelength-Division Multiplex (WDM) technique for enhancing the communication systems' capacity, in combination with recently developed optical amplifiers that allowed for increased repeater spacing [32]. The WDM technique makes use of multiple lasers operating at different wavelengths to stream multiple data over a single optical fiber. Its adoption required the development of new optical devices, such as optical lasers operating at slightly different wavelengths, optical multiplexers, and demultiplexers [4,6]. In [33], a WDM system operating at 1 Tbit/s over a 55 km link was shown. This system was composed of 25 lasers using polarization multiplexing totalling 50 channels operating at 20 Gb/s. According to [2], "in most WDM systems, fiber losses are compensated periodically using erbium-doped fiber amplifiers spaced 60–80 km apart". It is important to note that a repeater is an optoelectronic device that recovers the original electrical data from an optical signal and then retransmit this as an optical signal, and that, with the advent of optical amplifiers, the use of this type of repeaters is no longer required.

The major changes towards the fifth generation were the use of Raman amplification [34,35], the utilization of Dense Wavelength-Division Multiplex (DWDM) and the focus on making the WDM systems more efficient spectrally by using more advanced modulation schemes in which the information is encoded using amplitude, frequency, phase and polarization individually or altogether [2, 4, 6, 36]. Table 2.2 summarizes the information shown above. The main concern of the next generation, the sixth generation of fiber optic communication systems, is related to the maximum capacity of a single-mode fiber, and the main solutions include the use of multi-core fibers, the Spatial Division Multiplexing (SDM) technique [2,37,38] and All Optical Networks structures [39–41].

	Year implemented	Bit rate	Repeater spacing	Operating wavelength	Fiber losses
First Generation	1980	45 Mb/s	10 km	0.8 μm	3 dB/km
Second Generation	1985	100 Mb/s (MMF) 1.7 Gb/s (SMF)	50 km	1.3 μm	1 dB/km
Third Generation	1990	10 Gb/s (SMF)	100 km	1.55 μm	0.2 dB/km
Fourth Generation	1996	40 Gb/s (SMF) 1 Tb/s (WDM)	> 10,000 km	1.45 to 1.62 μm	0.2 dB/km
Fifth Generation	2002	> 40 Gb/s (SMF) 11 Tb/s (DWDM)	> 24,000 km	1.53 to 1.57 μm	0.2 dB/km

Table 2.2: Evolution of optical fiber communication systems and their main characteristics.

Optical fibers exhibit several advantages over other transmission media (twisted pairs, coaxial cables, microwaves, etc.) including their immunity to electromagnetic interference, their low power loss over distance, their smaller cross-sectional area, and lower weight in relation to metallic cables, their low manufacturing cost and almost unlimited bandwidth [2,37]. Furthermore, during their ∼35 year lifetime, it needs no maintenance, and, after their lifespan, they can still be used for seismic event detections [42,43], among other applications.

In contrast, fibers also have disadvantages, the main ones being: the high initial cost of implantation, and the limitation of optical power due to non-linear effects and due to the risk of “fiber fuse”. Despite their high reliability, optical fibers can sometimes be damaged by a variety of causes [44, 45]. On land, fibers can be damaged due to works, storms, or accidents, and, in the ocean, fibers can be disrupted by ship anchors, trawls used by fishing boats, marine animals, especially sharks which can chew the protection of the optical fibers, and natural disasters. In general, mechanical stress is highly prejudicial and should be avoided.

Due to its reliability and almost unlimited bandwidth, optical fibers networks are often used to interconnect networks across the world, as depicted in Figure 2.6. The internet’s core network or backbone, composed of the fibers depicted in Figure 2.6, is responsible for virtually all the data on the internet. This core is made up of individual high-speed fiber optic networks that form pairs to create the Internet backbone, and it is divided into smaller parts, smaller backbones, through internet exchange points, to keep traffic and data transmission from slowing down. As the backbone connects all smaller networks, it is possible to access any network through it. Clearly, as the deployment cost is relatively high, in order to keep this system working, it is necessary to monitor this entire network of cables spread across the planet, in addition to a fiber redundancy system, so that if something happens to the main fiber, customer service is not interrupted.

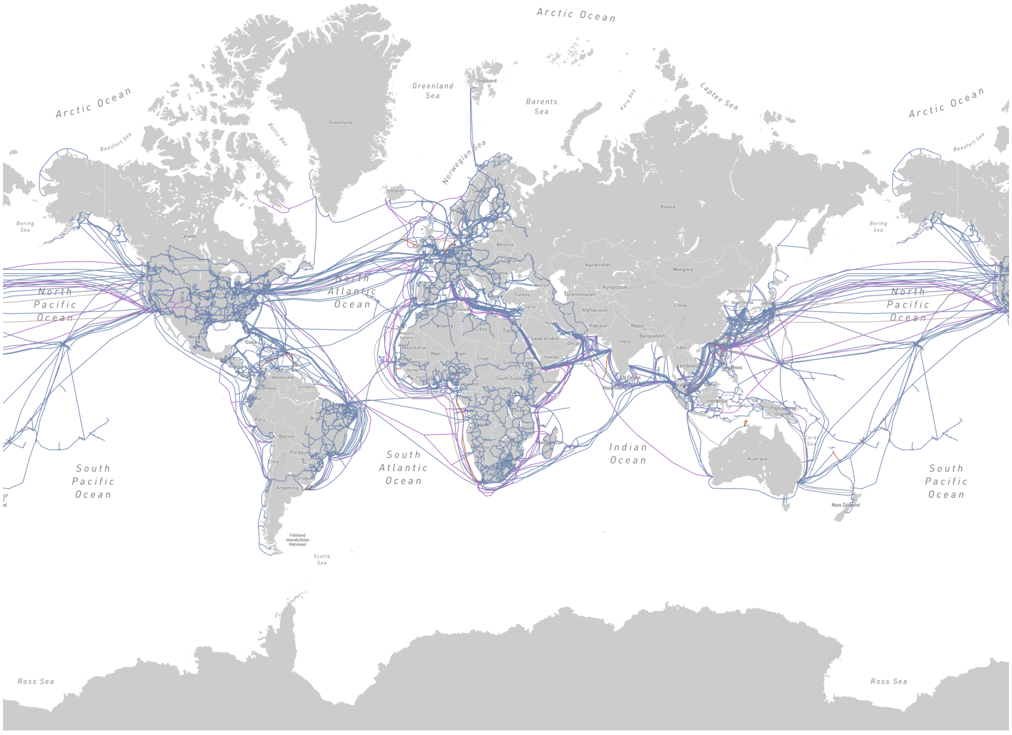


Figure 2.6: Global Internet Backbone Map [46].

2.3 Optical Monitoring Systems

Fiber monitoring is essential to enable long-distance high-speed optical telecommunications links since fibers can be jeopardized by the mechanical hazards presented in Section 2.2. Additionally, the supervision of optical networks' physical layer is fundamental since a fault can cause the suspension of essential services, such as a bank, telephone, or internet service links [11].

Optical reflectometry is the most used technique for the characterization of optical links since it only requires access to one end of the fiber [47]. Among the known types of optical reflectometry, stand out Optical Low-Coherence Reflectometry (OLCR), Optical Frequency-Domain Reflectometry (OFDR), and Optical Time-Domain Reflectometry (OTDR). Another remarkable feature is the detection scheme which can be a coherent or a direct detection scheme. The techniques that use a direct detection scheme are generally limited in resolution and reflection sensitivity when compared to the coherent techniques.

Direct and Coherent Detection Schemes

The direct detection technique is relatively simple to implement since only the reflected optical signal is incident on the detector. This optical signal is directly converted into an electrical signal — a photocurrent, see Equation 2-3 — proportional to the optical power which can be written as:

$$I_d = \mathcal{R}\, P_S \tag{2-3}$$

where $\mathcal{R}$ is the responsivity of the detector and P_S is the received optical signal incident on the optical detector. This detection scheme is relatively insensitive to the polarization of light or the phase of the optical signal.

Although coherent detection is generally more complex to implement, it offers certain advantages, see Figure 2.7, such as better dynamic range and greater sensitivity [4, 47]. The idea behind coherent detection is to mix the input signal with a reference signal, commonly referred to as a local oscillator (LO), and then detect the result of that interference with a photodiode. The beat between these optical fields generates a photocurrent that has information about the amplitude, frequency, or phase of the optical signal. The resulting photocurrent can be written as:

$$I_d = \mathcal{R}\left[P_S + P_{LO} + 2\sqrt{P_S P_{LO}}\cos\left(\omega_{IF} + \Delta\phi\right)\right] \tag{2-4}$$

where $\omega_{IF} = \omega_S - \omega_{LO}$ is the intermediate frequency and $\Delta\phi$ is the optical phase difference between the received optical signal, P_S, and the local oscillator,

P_{LO}. There are two coherent detection techniques: homodyne ($\omega_{IF} = 0$) and heterodyne ($\omega_{IF} \neq 0$).

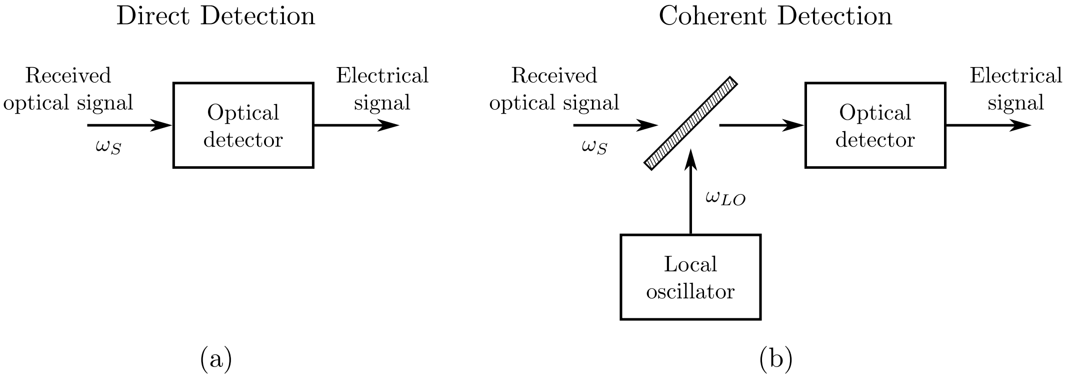


Figure 2.7: Schematic representation showing the differences between direct and coherent detection. Based on [47].

Coherent detection requires knowledge of the carrier phase, as the signal of interest is usually the term containing the square root as it describes the interference between the signal and local oscillator. Furthermore, the power of the local oscillator acts as an amplifying factor that multiplies the signal power resulting in an increased signal sensitivity [47].

Optical Low-Coherence Reflectometry

The OLCR technique is based on a coherent detection scheme and makes use of a low coherence source. A continuous-wave low-coherence source signal is launched into a Michelson interferometer, then it is split and goes to the test arm and the reference arm. The reference arm has a movable reference mirror whose position can be varied resulting in an optical delay for the returning light. Interference fringes will only be observed when the optical distance, which can be translated to optical delay, between the two arms is within the source's coherence length. If the optical distance between the two arms is greater than the coherence length of the source, the detector will only see a constant power, as no interference will occur. The measurement is then made by placing the device to be measured in one arm and varying the fiber length of the other arm. Each reflection in the test arm generates a coherence peak with a width proportional to the source's coherence length.

As stated in [47]: In practice, the mirror in the reference arm is often moved at a constant velocity, v_m, which causes the interference fringes to occur at a Doppler frequency, f_d, given by:

$$f_d = \frac{2\,v_m}{\lambda} \tag{2-5}$$

where λ is the average wavelength of the low-coherence source. The resulting signal then passes through an envelope detector and a low-pass filter. So by saving the magnitudes and position of the reference mirror it is possible to create a reflectometry trace. The biggest advantages of the OLCR technique are its high spatial resolution and reflection sensitivity. However, the measurement range is quite limited and determined by the distance over which a movable mirror can be scanned, its often limited to only tens of centimeters up to a few meters [48].

Optical Frequency-Domain Reflectometry

The Optical Frequency-Domain Reflectometer (OFDR) uses a continuous-wave light source with varying frequency to obtain the frequency-domain transfer function of the fiber by measuring the amplitude and phase of the reflected signal at each frequency [49]. The time-domain reflectometry trace is then obtained using the inverse Fourier transform on this data [50]. OFDR systems fall into two main classes: coherent (C-) and incoherent OFDR (I-OFDR).

Coherent OFDR systems use a tunable laser source in which the optical carrier frequency is linearly swept, this signal is then split into two paths: a reference arm with a fixed mirror and a test arm that probes a device under test (DUT). The reference signal and a time-delayed signal returning from a reflection point of the DUT coherently interfere at the fiber optic splitter. Further spectral analysis allows for the location of the faults in the DUT that generated these reflections. In I-OFDR, a continuous-wave laser has its optical carrier modulated by a radio frequency signal whose frequency is swept over a frequency span either by using a step-frequency or a sweep frequency method. The interference beat between reference and backscattered signal is performed in the electrical domain instead of in the optical domain as for C-OFDR systems [51]. C-OFDR systems generally offer greater spatial resolution and dynamic range than I-OFDR systems [49, 52, 53]; however, their measurement range is usually limited by the coherence properties of the laser [54–57]. The spatial resolution of an OFDR system can be determined by

$$\Delta z = \frac{c}{2\eta}\frac{1}{\Delta f_s} \tag{2-6}$$

where c is the speed of light, η is the refractive index of the fiber, and Δf_s is the frequency span of the measurement.

Optical Time-Domain Reflectometry

Among the known types of optical reflectometry, the most used is the Optical Time-Domain Reflectometry (OTDR). An OTDR is an optoelectronic device used to measure the impulse response of an optical fiber, i.e., to measure the impulse responde of the fiber the condition of one pulse inside a fiber at a time must be satisfied [58]. From this, it is possible to extract characteristics and information about the integrity of the optical link. Some of the features obtained from this reflectometry trace are an estimation of the fiber length, fiber attenuation, fault location, loss due to connectors or splices, and the possibility of gradual or sudden degradation of the fiber as a function of time through comparisons with previous fiber tests [59, 60].

To do so, an optical probe pulse is launched towards the fiber under test (FUT), through an optical circulator, and then the Rayleigh backscattered power is measured by a photodetector connected to port 3 of this circulator, see Figure 2.8. The measured optical power is recorded as a function of time as a reflectometry trace which will now be referred to as OTDR trace, see Fig. 2.9. The nature of such light falls into two classes: scattered light and reflected light. Light can be reflected due to discontinuity in refractive indexes along the fiber or due to such a reflective device such as a Bragg grating or a Fiber Optic Reflective-Coated Patch Cable. The scattered light is usually associated with the Rayleigh backscattering. As the other reflectometry techniques, the OTDR only needs to access one end of the fiber, i.e., a central transmission station can monitor all the fibers connected to it *in loco* without the need to install an apparatus for monitoring in each of the multiple nodes of the optical network.

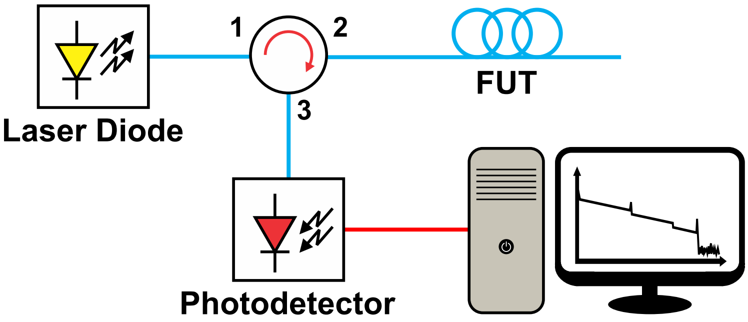


Figure 2.8: Block diagram of an OTDR.

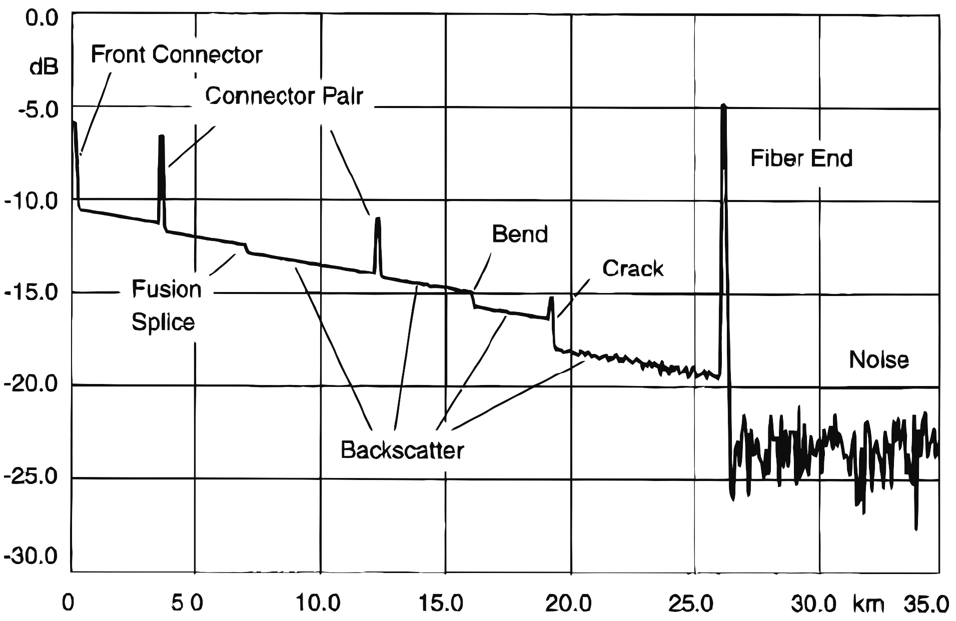


Figure 2.9: A typical OTDR trace with some of the most common fiber fault cases [47].

Basic concepts of OTDR

Rayleigh Scattering — The Rayleigh scattering phenomenon is an elastic scattering of electromagnetic radiation that occurs due to particles with a radius smaller than the electromagnetic radiation wavelength [61, 62], typically less than $\frac{\lambda}{15}$ [63]. Even though the scattering is omnidirectional, a small portion of the scattered propagates in the opposite direction of the incoming light [47,64]. The Rayleigh coefficient in optical fibers can be calculated from Equation 2-7 [65],

$$\alpha_s = \frac{8\pi^3}{3\lambda^4} n^8 p^2 k T_f \beta \tag{2-7}$$

where: α_s is the Rayleigh scattering coefficient (unit-less), λ is the wavelength of light (m), n is the refraction index in the fiber core, p is the photo-elastic coefficient of the glass, k is the Boltzmann constant (1.3806 $\mathrm{m^2kgs^{-2}K^{-1}}$), β is the isothermal compressibility (fractional change in the volume of the fiber as pressure changes at a constant temperature), T_f is a fictive temperature, representing the temperature at which the density fluctuations are “frozen” in the material (K) [66]. Early estimates of α_s values for SMF operating at 1550 nm were of about $\alpha_s \cong 3.9 \cdot 10^{-5}$ $\mathrm{m^{-1}}$, which in decibel units is 0.17 dB/km [47]. More recent studies estimate the limit of the Rayleigh scattering loss as $\alpha_s = 0.116$ dB/km at 1550 nm [67–69].

Dynamic Range — Another important parameter is the dynamic range, expressed in dB, which refers to the maximum fiber loss that can be measured, in other words the maximum length of an optical link. The dynamic range is defined by the combination of photodetector sensitivity, the optical pulse width and the peak power. The dynamic range is determined by the difference between the initial backscatter level and the noise level for a signal-to-noise ratio equal to 1 [47, 59, 61].

Spatial Resolution — The spatial resolution, often called "two-point" spatial resolution, refers to the mininum distance in which two reflections can be resolved. In time-domain reflectometry, the spatial resolution depends mainly on the optical pulse width launched into the fiber. As the pulse travels along in the fiber, it will widen due to dispersion effects. The spatial resolution can be determined by

$$\Delta z = \frac{c}{2\eta}\Delta t_s \tag{2-8}$$

where c is the speed of light, η is the refractive index of the fiber, and Δt_s is the system response time [47]. The system response time parameter depends on the photodetector's response time and optical pulse width and is given by

$$\Delta t_s = \begin{cases} \tau_p & \text{when } \tau_s \ll \tau_p \\ \sqrt{\tau_p^2 + \tau_s^2} & \text{otherwise} \end{cases} \tag{2-9}$$

where τ_p is the optical pulse width and τ_s is the response time of the photodetector. It is also important to note that there is a trade-off between dynamic range and spatial resolution, i.e., the narrower the optical pulse, the less energy it will have, which results in high spatial resolution and low dynamic range.

Dead Zone — Dead zones are related to strong reflections which saturate the optical receiver. After such an event, the receiver takes some time to recover its sensitivity, resulting in information loss. The attenuation dead zone is defined as the distance between the rising edge of the reflection event to the point in which the reflection is within a 0.5 dB margin over the backscattering level, and will be at least the width of the pulse plus the recovery time of the photodetector [47, 61]. The recovery time is the time the photodetector takes to readjust to its maximum sensitivity [58].

Modifications to the conventional OTDR

OTDR techniques have been widely adopted in the field for fiber monitoring, and several modifications have been proposed with the objective of increasing the dynamic range while also enhancing the spatial resolution. Increasing the dynamic range while still improving the spatial resolution is a challenging task, since both features are dictated by the pulse width and peak of the optical probe pulse. Some of the most outstanding OTDR techniques are Tunable-OTDR, Coded OTDR, Chaos OTDR, and Photon-Counting OTDR.

The Tunable-OTDR makes use of highly coherent optical sources and optical amplifiers to generate spectrally narrow pulses adapted to WDM networks [70]. Coded OTDR, instead of a single pulse at a time, uses pseudo-random code sequences to modulate the probing pulse, and correlation techniques are employed to recover the fiber trace [71, 72]. Coded OTDRs are able to increase the dynamic range while maintaining the same spatial resolution as a conventional OTDR that employs only one pulse at a time. Chaos OTDR was developed with the aim of overcoming the limitations in pseudo-random code sequences generation through the use of chaotic laser light which generates a real random signal with a bandwidth of several tens of GHz which, in turn, leads to higher spatial resolution [73–75]. However, the Chaos OTDR cannot be used to measure the fiber attenuation, it can only detect reflection events by cross-correlation method [75].

Classical OTDR techniques make use of p-i-n or Avalanche Photodiodes to measure the backscattered optical power in a continuous fashion. Thus, the photodetector's bandwidth is very relevant since higher bandwidth photodetectors may help to increase the maximum spatial resolution. However, increasing the bandwidth also increases the Noise Equivalent Power (NEP), which translates to a reduction in the dynamic range [47, 76–78].

A further development, the Photon Counting OTDR, was proposed with the advent of the Geiger-mode Single-Photon Detector. Such devices are based on InGaAs/InP Avalanche Photodiodes (APDs), hence the name Single Photon Avalanche Detector (SPAD). When an APD operates in Geiger-mode it becomes very sensitive to the point it can respond to the arrival of a single photon. For this, it is applied an above the breakdown voltage across the device's terminals, enhancing its sensitivity, so that the energy of a single photon is capable of starting the avalanche process [76, 77]. In this mode of operation, the APDs are only active during a very small time window, which is usually called a "gate". In gated mode, such APDs can attain high photon detection efficiencies and extremely low dark count rates [77]. If a photon (or more than one photon) impinges on its active region during the gate window it achieves saturation and a quenching circuit (which can be passive or active) is necessary to remove it from this state and send it to a relaxed state. The required time to return the APD to a relaxed state after a detection is called Dead Time.

The principal advantage of the Photon-Counting OTDR (PC-OTDR) over a conventional OTDR is that, in this case, due to the single-photon counting operation, the noise floor is independent of the receiver bandwidth which alleviates the compromise relationship between spatial resolution and dynamic range. The spatial resolution is now is limited by the performance of high-speed electronics, and the noise floor is now set by the quantum efficiency and the shot noise from the background dark counts of the SPAD [47]. However, a drawback of the PC-OTDR is the slower data acquisition due to the dead time after a detection during which no data is acquired [79]. Thus, the gated operation requires an intelligent management system to reduce monitoring periods and enhance the acquisition of statistically relevant data. One way to improve the data acquisition rate of PC-OTDRs was demonstrated in [12], where a train of gates was used to trigger the SPAD multiple times; the results show spatial resolution of 6 meters and descent achievable dynamic range within minute monitoring periods. A similar scheme was used in [80] to improve the data acquisition rate of a parallel work introduced in [13, 81]; the results show spatial resolution of 4.6 cm and 15 dB dynamic range.

2.4 Tunability and Pulse Generation

The tunability in PC-OTDR is another very important aspect due to the use of Dense Wavelength-Division Multiplex (DWDM) in recent optical networks to stream multiple data over a single optical fiber. The current DWDM networks operate with either 40 channels with spectral width of 0.8 nm and 80 channels with spectral width of 0.4 nm, although 0.2 nm and 0.1 nm standards have also been proposed [11, 82, 83]. In order to achieve tunability, a Tunable PC-OTDR makes use of highly coherent optical sources to generate optical pulses within the desired bandwidth and center wavelength. In reality, there are several ways to generate short optical pulses: gain switching, Q switching, mode locking, among others [25, 84–88]. However, what stands out the most is the mode locking offering the best bandwidth-to-pulse width ratio [25]. Indeed, mode-locked lasers can achieve near transform-limited pulse generation by employing both semiconductor and fiber gain media [80, 89–91].

Mode-locking is a technique for generating short high-power optical pulses through the interference of a several monochromatic waves equally spaced in frequency and with equal phases [25,89], see Figure 2.10. This interference can be written mathematically by

$$p(t) = \sum_{n=-N/2}^{N/2} c_n e^{j2\pi(f_0+n\cdot f_{FSR})t}, \tag{2-10}$$

where c_n is the weight of each component, f_0 is laser's center frequency, f_{FSR} is the optical cavity's free-spectral range (FSR), and $\mathrm{M} = N+1 \approx \frac{\Delta f}{f_{FSR}}$ is the number of monochromatic waves permitted inside the gain bandwidth of the medium (Δf) [25,80], see Fig. 2.10. Also, tunability can be enforced by the medium's gain bandwidth (Δf) or by the use of a tunable bandpass filter. This equation resembles the exponential form of the Fourier series, thus implying the optical pulses are emitted periodically during the mode-locked operation. Also, after the sum of the M modes, the pulse duration can be written as $\tau_P = \frac{T_P}{\mathrm{M}}$ where $T_P = \frac{1}{f_{FSR}}$ is the train of pulses' period, and knowing the mean optical power $\bar{\mathrm{I}}$ it is possible to estimate the peak power as $\mathrm{M}\bar{\mathrm{I}}$ [25].

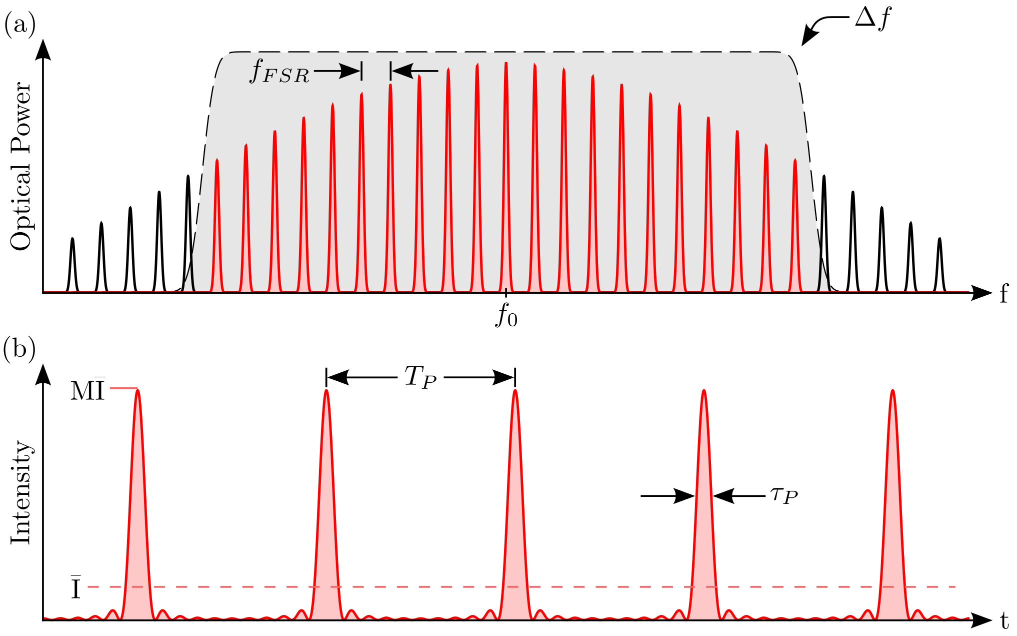


Figure 2.10: (a) Optical spectrum of a mode-locked laser. (b) Period pulse train of a mode-locked laser.

In practice, mode-locking is used to lock the monochromatic waves in phase, and can be achieved either actively (through active modulation devices) or passively. The fiber based passive mode-locked lasers (PMLL) attracts special attention due to the absence of active modulation devices, the fiber's low propagation loss, possibility of creating high quality factor fiber cavities and the fact that erbium-doped fiber amplifiers (EDFAs) and semiconductor optical amplifiers (SOAs) can be used as gain media [80, 90–92]. Also, the PMLLs generally offer rapid response and recovery times which leads to shorter pulses when compared with active mode-locked lasers [25]. In order to lock the modes in phase, several mechanisms can be used, some of them are: dispersion management [91,93,94], non-linear polarization rotation [95–97], self-phase modulation [25,30,90,98,99], a non-linear phase shift induced by the Kerr effect that causes the fiber's refraction index to change with the optical intensity $n(I) = n+n_2I$, where the n_2 term is a material-dependent incremental refractive-index change, and the constructive/destructive interference of counter-propagating longitudinal modes in a non-linear amplifying loop mirror (NALM) [11,90,98].

2.5
Automatic Fiber Monitoring System

In recent decades, especially with the advent of the Internet, the amount of information we exchange has grown exponentially. The main reasons behind this growth are the growing interest of the population in data exchange and the emergence of new types of services, such as video sharing and the emergence of services such as Youtube and Netflix. However, this ended up creating two big problems: the transmission of this information over long distances and the increase in the amount of information, which results in increased bandwidth requirements. The first fiber-optic communication systems emerged to fill this gap. In a nutshell, optical fibers offer low attenuation and "almost" unlimited bandwidths. Thus, in the last years, we have seen the massive deployment of optical fibers promoted by the ever-increasing need for higher data transmission capacity.

Despite their great success, optical fibers are susceptible to mechanical stress and, under certain conditions, may break. For this reason, physical layer supervision is essential to identify such faults and remedied them as rapidly as possible to ensure the robustness of all network layers. OTDR techniques have been widely adopted in the field for fiber monitoring due to their long-haul, high-resolution, and tunability monitoring capabilities. However, it the role of the network operator to identify fiber breaks, extreme bends, and defective connectors to schedule an in-field repairing unit as fast as possible. Also, a network operator might have several thousand optical fibers under his responsibility [9,10]. Therefore, an automated fiber measurement system based on a digital signal processing unit capable of identifying fault positions in a fiber profile could alleviate their workload and, eventually, hasten and simplify the scheduling process of the in-field repairing units [11]. As OTDR systems frequently make use of FPGAs [11–13], an automatic fault detection system could be easily implemented in an FPGA. In [100] and [14], the authors provided a thorough comparison between several signal processing techniques with the aim of finding faults in an OTDR trace. Among these signal processing techniques, the Linearized Bregman Iterations stand out for being a class of implementation-efficient and low-complexity algorithms for solving combined ℓ_1/ℓ_2 minimization problem [14, 15].

2.6 Linearized Bregman Iteration

Finding the solution of a linear equation $\mathbf{A}\boldsymbol{\beta} = \mathbf{y}$ arises from many applications. Some such applications may be simple to solve while others may be very tough, depending on the characteristics of the problem, for example: if $\mathbf{A}$ is a non-singular matrix, i.e., $\mathbf{A}$ is an invertible matrix, the presence of noise in $\mathbf{y}$, among others. Examples of direct applications are solving a system of linear equations for an electronic circuit [101], modern control systems [102], machine learning [103, 104], among others. However, there are applications where $\mathbf{A}$ can be a dense matrix and the solution we seek is sparse, as, for example, in compressed sensing [105–107] or the detection of breaks in a fiber profile [12, 14].

The so-called basis pursuit problem can be expressed as a constrained ℓ_1 minimization problem as follows,

$$\min_{\boldsymbol{\beta}} ||\boldsymbol{\beta}||_1 \;\; \text{s.t.} \;\; \mathbf{A}\boldsymbol{\beta} = \mathbf{y}, \tag{2-11}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{\beta} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$. Due to the ℓ_1 norm, this problem is convex [106, 108, 109] and, under certain circumstances, gives the sparsest solution [110]. Although the result is a solution of Eq. 2-11, it may not be unique [110, 111] even more so in the presence of noise. However, by using a soft thresholding operator as follows,

$$\text{shrink}\left(\mathbf{v}_j, \lambda\right) = \text{sign}\left(\mathbf{v}_j\right) \cdot \max\left(|\mathbf{v}_j| - \lambda, 0\right) = \begin{cases} \mathbf{v}_j - \lambda & \text{if } \mathbf{v}_j > \lambda \\ 0 & \text{if } \mathbf{v}_j \in [-\lambda, \lambda] \\ \mathbf{v}_j + \lambda & \text{if } \mathbf{v}_j < \lambda \end{cases} \tag{2-12}$$

it is possible to obtain a solution that minimizes the ℓ_1 norm in some sense [105]. The addition of this operator produces a sparse vector and removes the noise [103, 105, 112]. The Linearized Bregman Iterations (LBI) is a class of implementation-efficient and low-complexity algorithm [14, 15] for solving combined ℓ_1/ℓ_2 minimization problem, in the presence of noise [14, 105, 110, 113], of the form

$$\min_{\boldsymbol{\beta}} \lambda||\boldsymbol{\beta}||_1 + \frac{1}{2}||\boldsymbol{\beta}||_2^2 \;\; \text{s.t.} \;\; \mathbf{A}\boldsymbol{\beta} = \mathbf{y}, \tag{2-13}$$

where $\mathbf{A}$ is a dictionary matrix, in which each column is a candidate, $\boldsymbol{\beta}$ is the vector containing the weight of each candidate, and $\mathbf{y}$ is the original data set, which in this context of trend break detection in reflectometry traces is the fiber profile.

In [14], the Linearized Bregman Iterations (LBI) algorithm was adapted for trend break detection in OTDR traces, see Fig. 2.9. An arbitrary OTDR fiber profile can be interpreted as a weighted sum of step functions in the following form

$$y(z) = \alpha r\left(z\right) + \sum_{i \in F} a_i u\left(z - z_i\right) \tag{2-14}$$

where $y(z)$ is a noise-free representation of an OTDR trace, z is the distance inside the fiber, α is the fiber attenuation coefficient, $r(z)$ is the ramp function, $u(z)$ is the step function, F is the set of fault indices, a_i and z_i are the weight and position in which each fault occurs, respectively [14]. As the number of faults in a fiber profile is usually much smaller than its length, the identification of fault positions can be assumed to be sparse. It is also interesting to note that the addition of the $||\boldsymbol{\beta}||_2^2$ term causes the objective function to become strongly convex, therefore the solution of this minimization problem is unique [105, 110]. However, if $\lambda \to \infty$, the solution tends to a solution of Eq. 2-11 [105]. The Linearized Bregman Iterations can be summarized in the following algorithm [106].

Algorithm 1 Linearized Bregman Iterations (LBI)

```
Input: Measurement vector $\mathbf{y}$, $\mathbf{A}$, $\lambda$, $N$
$\boldsymbol{\beta}^{(0)} \leftarrow \mathbf{0}$
$\mathbf{v}^{(0)} \leftarrow \mathbf{0}$
for $k = 1..N$ do
    for $j = 1..n$ do
        $\boldsymbol{\beta}_j^{(k+1)} \leftarrow \text{shrink}\left(\mathbf{v}_j^{(k)}, \lambda\right)$
    end for
    $\mathbf{v}^{(k+1)} \leftarrow \mathbf{v}^{(k)} + \mu \mathbf{A}^T \left(\mathbf{y} - \mathbf{A}\boldsymbol{\beta}^{(k)}\right)$
end for
Output: Estimated $\boldsymbol{\beta}$
```

In the specific case of fiber profiles, each column of the dictionary matrix, $\mathbf{A}$, is a delayed step function. However, it is also necessary to take into account the slope, which represents the fiber attenuation. Thus, the final form of $\mathbf{A}$ is

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 2 & 1 & 1 & 0 & \cdots & 0 & 0 \\ 3 & 1 & 1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ m-1 & 1 & 1 & 1 & \cdots & 1 & 0 \\ m & 1 & 1 & 1 & \cdots & 1 & 1 \end{bmatrix}, \tag{2-15}$$

with m rows and $n = m+1$ columns. As shown in [14], an important feature of this dictionary matrix is that, due to its special structure, it allows for a further simplication that can be incorporated into the sparse Kaczmarz algorithm [114], preventing the need to store and load this matrix. The Kaczmarz's algorithm is an iterative form for solving $\mathbf{Ax} = \mathbf{y}$, as shown below

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \frac{\mathbf{y}_i - \left\langle \mathbf{a}_i, \mathbf{x}^{(k)} \right\rangle}{||\mathbf{a}_i||_2^2} \overline{\mathbf{a}_i} \tag{2-16}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{a}_i$ is the i$^{\text{th}}$ row of $\mathbf{A}$, $i = k \mod m$ and $\overline{\mathbf{a}_i}$ is the complex conjugate of $\mathbf{a}_i$. Thus, line 7 of Algorithm 1 can be rewritten as

$$\mathbf{v}^{(k+1)} \leftarrow \mathbf{v}^{(k)} + \frac{1}{||\mathbf{a}_i||_2^2} \mathbf{a}_i \left(\mathbf{y}_i - \mathbf{a}_i^T \boldsymbol{\beta}^{(k)} \right), \tag{2-17}$$

where the lefthand side of this equation is the step width in the LBI algorithm, contributing to the algorithm's convergence speed. By analyzing the dictionary matrix, $\mathbf{A}$, the squared norm $||\mathbf{a}_i||_2^2$ can be calculated as

$$||\mathbf{a}_i||_2^2 = i^2 + i, \tag{2-18}$$

with the squared term i^2 being the limiting factor for the convergence speed of the algorithm [14]. For this reason, a scaling factor σ for the first column of $\mathbf{A}$ was used, resulting in the algorithm shown below. It is important to note that this algorithm, Algorithm 2, was specially tailored for implementation in an FPGA.

Algorithm 2 Linearized Bregman Iterations for Trend Break Detection

```
Input: Measurement vector $\mathbf{y}$, $\lambda$, $N$
$\boldsymbol{\beta}^{(0)} \leftarrow \mathbf{0}$
$\mathbf{v}^{(0)} \leftarrow \mathbf{0}$
for $k = 1..N$ do
    $i \leftarrow \text{mod}\left((k-1), m\right) + 1$        ▷ cyclic re-use of rows of $\mathbf{A}$
    for $j = 1..(i+1)$ do
        $\boldsymbol{\beta}_j^{(k+1)} \leftarrow \text{shrink}\left(\mathbf{v}_j^{(k)}, \lambda\right)$
    end for
    $||\mathbf{a}_i||_2^2 \leftarrow \sigma^2 i^2 + i$
    $e \leftarrow \left(\mathbf{y}_i - \sigma i \boldsymbol{\beta}_1^{(k)} - \sum_{s=2}^{i+1} \boldsymbol{\beta}_s^{(k)}\right)$
    $d \leftarrow \frac{1}{||\mathbf{a}_i||_2^2} e$
    $\mathbf{v}_1^{(k+1)} = \mathbf{v}_1^{(k)} + \sigma i d$
    for $j = 2..(i+1)$ do
        $\mathbf{v}_j^{(k+1)} = \mathbf{v}_j^{(k)} + d$
    end for
end for
Output: Estimated $\boldsymbol{\beta}$
```

2.7 FPGA

A Field Programmable Gate Array (FPGA) is a programmable electronic device that is used to implement arbitrary digital circuits. Early FPGAs evolved from small arrays of programmable logic and interconnect networks to massive amounts of arrays of programmable logic, memory blocks, high-speed I/O and interconnect networks, highly specialized blocks, such as blocks of digital signal processing (DSP), clock management, hard memory controllers and transceivers. The interconnect networks, also called switching matrices, are responsible for the connection between the aforementioned blocks. By programming the switch matrices that connects these blocks, it is possible to create virtually any type of digital circuit.

In order to implement some logic structure in an FPGA, eletronic design automation (EDA) tools are used to generate a bitstream file that contains a binary representation of the configuration of how each resource is connected internally in an FPGA, including wire connections, contents of memory blocks and logic elements. The EDA tool follows three steps to generate the bitstream file. First, a precise and formal description of the eletronic device, written in a Hardware Description Language (HDL), such as VHDL or Verilog, is analyzed and translated into Register Transfer Level (RTL) design throught a process called “Synthesis”. The RTL-level design is used to create a netlist containing primitive logical elements. Then, a process called “Place and Route” (PAR) is used to interpret this netlist using the available resources on the target device. This step also takes into account timming constraints. Finally, the resulting design is captured as a bitstream file.

FPGA Architecture

An FPGA is a highly flexible and reprogrammable device that can be reprogrammed, on-demand, an unlimited number of times. Whenever we configure the FPGA, configuration bitstream is stored in highly robust CMOS configuration latches (CCLs) designed primarily for data integrity [115]. The CCLs, as well as Static Random-Access Memories (RAM), do not retain their data when power is removed. Therefore, the FPGA must be reconfigured every time it is turned off. The FPGA configuration process can be performed by an external non-volatile memory device or an external smart device, such as a microprocessor.

Although knowing the FPGA architecture is not a prerequisite for using an FPGA, since Hardware Description Languages (HDLs) provide abstraction layers, through direct instantiation of primitive components it is possible to achieve higher performances. It is interesting to note that EDA tools perform numerous optimizations during synthesis and PAR processes. Optimizations can be performed to achieve the highest clock frequency, reduce power consumption, or decrease area usage, i.e., use the minimum FPGA resources, useful for large projects. Figure 2.11 provides an overview of the architecture of a modern FPGA [115], and of its fundamental programmable functional elements.

- Configurable Logic Blocks (CLBs) are used to implement logical functions, sequential and combinational, and also to store data.
- Block RAMs (BRAMs) are 18-Kbit memories that can be used to store data. Each BRAM can also be used as two independent 9-Kbits memories.
- DSP Blocks consists of highly specialized blocks of 18 × 18 bit two's complement multiplier and a 48-bit accumulator.
- Input/Output Blocks (IOBs) manages the data that enters and leaves the device.
- Digital Clock Manager (DCM) are fully digital blocks to delay, multiply, divide or apply a phase to a clock signal.

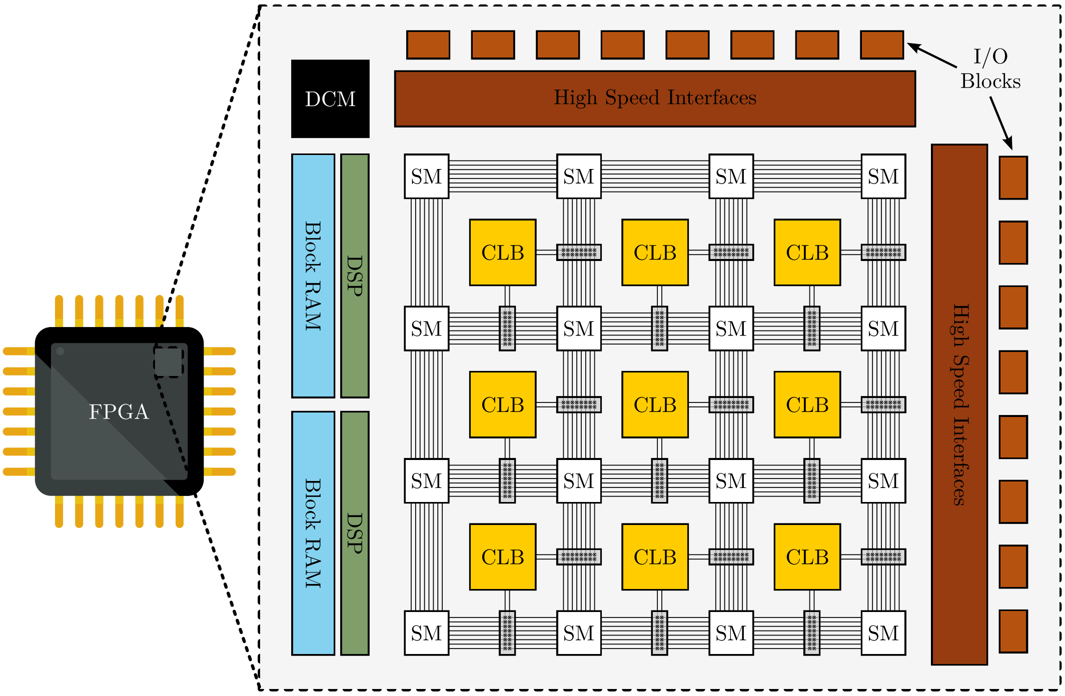


Figure 2.11: Block diagram of the architecture of an FPGA. Based on the Spartan-6 Family Architecture [116].

The Configurable Logic Blocks (CLBs) are the main logic resources of an FPGA and can be used to implement synchronous or asynchronous combinational circuits. The internal structure of a CLB may change depending on the FPGA Architecture, on the Spartan-6 Family Architecture, these blocks are composed of two slices, which are divided into three types (SLICEX, SLICEL, and SLICEM). The SLICEX is the most basic and consists of four Look-Up Tables (LUTs), eight flip-flops and can be used to implement combinational logic, sequential logic, and memory functions. Table 2.4 shows the functions that each type of slice is able to provide. It is important to note that although FPGAs are commonly used to implement digital circuits, they do not have logic gates. The n-bit Look-Up Tables (LUTs) are memories that can be used to implement any n-inputs logical function, where the inputs are connected to the address signal and the output is the truth table of this logical function.

	Slice type		
Supported functions	SLICEX	SLICEL	SLICEM
Four 6-input LUT function generators	✓	✓	✓
Eight storage elements (D flip-flop)	✓	✓	✓
Two 8-to-1 multiplexers	✗	✓	✓
Carry and arithmetic logic	✗	✓	✓
Four 64x1 distributed RAM blocks	✗	✗	✓
Four 32-bit shift registers	✗	✗	✓

Table 2.4: Comparative table of supported functions by each type of slice.

Early hardware designers had to deal with just a few hundred transistors. However, with the advent of integrated circuits (ICs), many more transistors could be placed in a single chip. In the 1970s, the world's first commercially available microprocessor, the Intel 4004 [117, 118], was released with about 2300 transistors. In the early 1990s, driven by technology and business, microprocessor designs passed a million transistors [119]. At this time, the Very Large-Scale Integration (VSLI) technology had already been developed and engineers were creating integrated circuits through schematic and they performed simulations with the help of hardware description languages such as Verilog and VHDL. The HDL languages gained, even more, popularity after the introduction of the logic synthesis step, by transforming the HDL code into Register Transfer Level (RTL), in order to refine the RTL code at the netlist level and to perform logical error checking steps, to then transform that RTL code into a design implementation in logical ports.

The first Programmable Logic Devices (PLDs), composed of small arrays of programmable logic and switching matrices, were rapidly superseded by the FPGAs. As FPGAs continued to grow in size they started to incorporate more and more custom resources such as DSP blocks, BRAMs, and full adders inside the CLBs. These custom resources, often called “hardened” resources, are not programmable devices as they are designed specifically to perform a task [120–122]. In addition, FPGA vendors typically provide Intellectual Property (IP) Cores, reusable components that intelligently use the capabilities of the FPGA’s Architecture to accomplish specific tasks. IP Cores offer the advantage of being proven designs, with interface and behavior verified by its supplier, that can be used as building blocks in large projects. Some of the areas that FPGAs stand out the most are their ease of prototyping, real-time applications, and their pipelining and parallelization capabilities. By combining its pipelining and parallelization capabilities, it is possible to implement specific hardware that can offer higher performances than the most powerful CPUs or GPUs. Even though the maximum operating frequency of an FPGA is not very high, for example, a high-end FPGA operates at the most at $\sim$1 GHz [123, 124].

3
Fast Acquisition Tunable High-Resolution ν-OTDR

Optical fibers are, inarguably, one of the most important elements of modern telecommunication networks, having allowed for long-haul high-bandwidth links to be implemented around the world [125]. They offer, thus, the basis for the Open Systems Interconnection (OSI) and its reliability is fundamental for the robust operation of all the higher-level network layers. Mechanical damage, the protagonist of optical fiber transmission impairments, can cause from minor to debilitating losses and, therefore, must be evaluated timely and precisely [126].

3.1
Motivation and Contributions

One way of discovering the position and magnitude of the mechanical damage imposed into a fiber is to monitor the backscattered light originated from an interrogation signal [127]. When the intensity of the back-propagating light is measured over time, and the position of the mechanical damage is determined based on the speed of light inside the fiber, the method is known as Optical Time Domain Reflectometry (OTDR) [60]. In OTDR applications, spatial resolution and dynamic range come as a trade-off since strengthening the pulse for enhanced reach usually affects the pulse width and diminishes the 2-point resolution [4]. To alleviate such trade-off, a PC-OTDR can be assembled, where a single-photon detector is employed and a detection management system must be integrated for consistent results [128].

In this chapter, a 15 dB dynamic range and 4.6 cm spatial resolution tunable PC-OTDR is presented along with a joint FPGA and TDC-based detection management system that allows several regions of the fiber to be interrogated by the same optical pulse. The main overall contributions of this work are discriminated below:

– An optical pulse generation unit, implemented by a figure-8 passively mode-locked laser, containing a tunable optical filter that ensures that the monitoring pulses are generated within the desired bandwidth and center wavelength for WDM applications.

- A detection procedure that allows for increased data acquisition rates by a factor that depends on the operator's choice of detection window length and detection gate train period, but is shown to reach at least 20-fold.
- Centimeter-range monitoring of km-range fibers in under 20 minutes with 15 dB dynamic range in a wavelength division multiplexing (WDM) scenario.

3.2 PC-OTDR System

Combining a pulse generation unit based on a figure-8 PMLL and a data acquisition unit based on an FPGA and a TDC culminates in an optical reflectometry-based measurement system that yields a spatial resolution in the centimeter range with both achievable dynamic range and data acquisition rates higher than related state-of-the-art systems; furthermore, the system is tunable around the telecommunication C-band and exhibits narrow bandwidth, making it ideal for supervision of currently deployed optical networks relying on wavelength division multiplexing (WDM), such as WDM Passive Optical Networks (WDM-PON). In Fig. 3.1, the assembled monitoring system is depicted in detail. Data acquisition is performed in synchronization with an enabling pulse from a high-speed optical switch that guarantees that a single probing pulse is traversing the fiber at a time. This triggering pulse, in turn, is synchronized to the incoming pulse from the PML by detecting a small portion of its output using a 90/10 beam splitter.

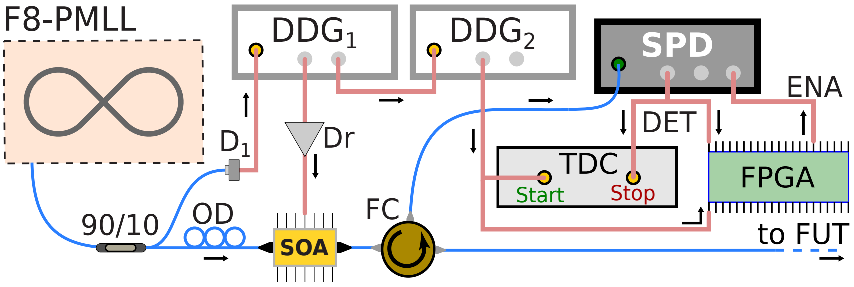


Figure 3.1: Block diagram of the PC-OTDR system including the synchronization with the F8-PMLL; optical fibers are blue lines and pink lines are electrical signals. D$_1$, *p-i-n* photodetector; DDG, Digital Delay Generator; Dr, electronic driver; OD, optical delay; FC, fiber circulator; FUT, fiber under test. The black arrows indicate the direction of the electric and optical signals for ease of visualization.

In Fig. 3.2a, the structure of the PMLL is depicted: on the right-hand side, the optical cavity contains an unidirectional EDFA that determines the direction of propagation, and a variable optical bandpass filter that determines the center wavelength of emission; on the left-hand side, the nonlinear amplifying loop mirror (NALM) contains a bi-directional semiconductor optical amplifier (SOA). The PMLL's two main output characteristics are also demonstrated: the generation of ps-long optical pulses, Fig. 3.2b; and the tunability over the telecommunication C-band, Fig. 3.2c. Under fundamental mode operation, the F8-PMLL produces 303 ps-wide pulses (which limit the spatial resolution of the PC-OTDR to 3.07 cm) spaced by 224 ns (a repetition rate of 4.46 MHz), the former determined by a total 1.39 GHz emission bandwidth (confirmed with the measurement of a high-resolution – 16 MHz – optical spectrum analyzer to be 11 pm at all center wavelengths) and, the latter, by the overall length of the fiber cavity of 44.8 meters. To achieve such conditions, other than correctly adjusting the PCs, the injection current of the SOA should be approximately 150 mA, which translates into optical pulses with 4 pJ energy level and 13 mW peak power. Under these conditions, the wavelength tunability of the F8-PMLL is rather simple in terms of PC alignment; on the other hand, the pulse peak power limits the achievable dynamic range. In order to maintain the ease of tunability of the system while also increasing the dynamic range, the output pulses were sent to a second EDFA (depicted in Fig. 3.1), which yields optical pulses with 25 pJ energy level and 83 mW peak power. Due to the broad bandwidth of the EDFA, the width of the optical pulses, after amplification, was not compromised, as presented in Fig. 3.2b.

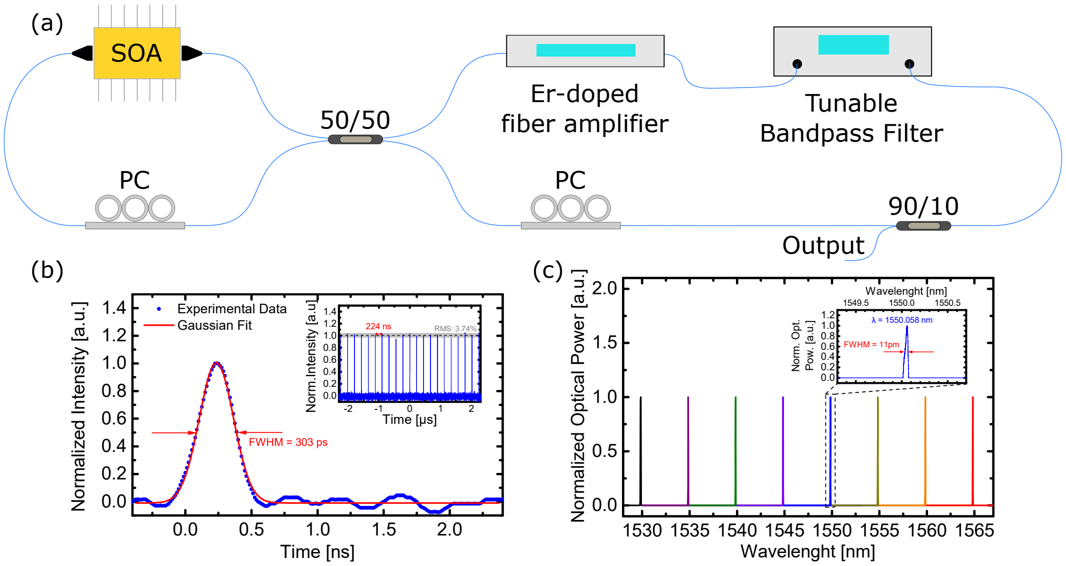


Figure 3.2: (a) Schematic of the passively mode-locked figure-8 fiber laser. PC, Polarization Controller. (b) Temporal shape and repetition rate of the generated optical signal in the F8-PMLL. (c) Center wavelength tunability and bandwidth of the employed F8-PMLL.

In order to showcase the capability of the system to perform tunable and fast fiber profile measurements with centimeter resolution, an arrayed waveguide grating (AWG) was employed. A 4-km feeder fiber (between the measurement system and the AWG) is followed by four different fibers ($L_1 = 4$ km, $L_2 = 8$ km, $L_3 = 0.9$ km, $L_4 = 3$ km) each connected to a different channel of the AWG, as follows: Ch 03, $\lambda = 1534.09$ nm, L_1; Ch 09, $\lambda = 1539.8$ nm, L_2; Ch 15, $\lambda = 1544.0$ nm, L_3; Ch 22, $\lambda = 1550.0$ nm, L_4. For these results, the measurement time was ~16.7 minutes. Interrogation of each individual channel is possible by tuning the F8-PMLL center wavelength accordingly, and the results are presented in Fig. 3.3; total measurement time is ~16.7 minutes for each of the channels.

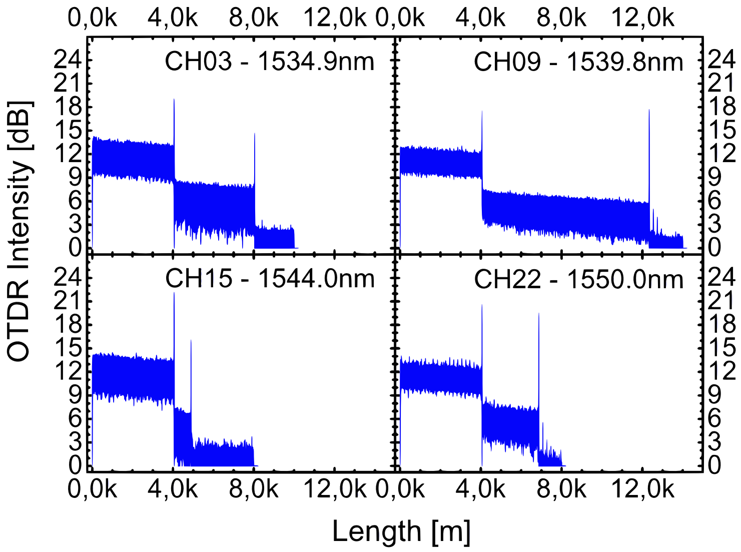


Figure 3.3: Full fiber measurements of four distinct AWG channels, corresponding to different wavelengths.

3.3 Conclusion

By combining the time resolution of a TDC with the management capabilities of an FPGA, a detection system could be developed that allows for high resolution measurements with high data acquisition rates. This structure, when employed together with an optical pulse generation system that allows for wavelength tunability and high-peak-power narrow pulses culminates into the fast acquisition tunable high-resolution photon-counting OTDR, with an achievable spatial resolution of 4.6 cm and 15 dB dynamic range. When compared to other monitoring solutions, which do not rely on photon-counting for fiber monitoring, the Fast Acquisition Tunable PC-OTDR exhibits an unique combination of characteristics that make it competitive with respect to the state-of-the-art.

4
FPGA-Embedded Linearized Bregman Iterations Algorithm for Trend Break Detection

Trend break detection, or level-shift detection, is a problem that permeates several science fields, and an efficient, accurate, and highly reliable processing unit to solve it is desirable. In this chapter, it is shown that by combining the flexible hardware design tools of Field Programmable Gate Arrays and the efficient Linearized Bregman Iterations algorithm allows for the development of such a unit. The manipulation of the data storage structure as well as the algorithm flow and control in hardware yields increased processing efficiency while maintaining all the observed qualities of the Linearized Bregman Iterations algorithm, such as low estimation error and high level-shift detection precision [15]. Due to its flexible memory structure, the proposed hardware architecture can be implemented in different-sized-FPGAs.

4.1
Motivation and Contributions

Recently, Linearized Bregman Iterations (LBI), a class of implementation-efficient and low-complexity algorithms, has been presented as an extremely attractive solution for trend break detection [14]. There, it was shown that the LBI algorithm outperforms the classical LASSO solution in the specific problem of trend break detection for fiber fault analysis. In this case, the better performance can be attributed to two factors. First it is well known that combined ℓ_1/ℓ_2 regularization terms can handle problems where estimation vectors with correlated elements are expected better than the LASSO solution [129] and the problem at hand is expected to have such correlations. Second, the strict convexity when using a combined ℓ_1/ℓ_2 term in the cost function has been reported to improve the convergence behavior of sparse estimation algorithms [110]. Furthermore, both the structure of the trend break detection problem and of the LBI algorithm's allow for simple hardware units, relying mainly on adders and efficient memory management, to conduct the core procedure, thereby avoiding hardware-complex multiplication and division operations [130].

In [14], the focus is on detection of trend breaks associated to fiber faults: to ensure robustness of network operation, detection of such faults must be remedied as fast as possible; this, in turn, creates a demand for highly reliable and fast trend break detection results, since a network operator might have several thousands of optical fibers under his responsibility [10,131] and, thus, the faster the data associated with one link can be processed, the faster information about all links will be available. The possibility of increasing the time efficiency of the algorithm due to its hardware implementation is of great interest in this context, as pictorially presented in Fig. 4.2a, where a centralized optical fiber network manager must operate several links simultaneously; expediting each measurement is extremely beneficial. Furthermore, as the measurement is often done by stand-alone Optical Time-Domain Reflectometry (OTDR) devices, the eventual goal is to implement the processing directly in such a device, where FPGAs are usually employed. Achieving an FPGA implementation of the detection algorithm is, then, preferable, since it would allow data acquisition and processing to be performed in the same embedded device, as pictorially presented in Fig. 4.2b, where the combination of a data acquisition hardware for the Physical Layer Monitoring (PLM) unit (adapted from [132]) along with the data processing hardware implementation of the LBI algorithm is depicted. The optical pulse generation (PG) unit launches a probing pulse into the fiber, while the data acquisition system, together with an optical circulator and an optical pulse detection (PD), acquires the reflected optical power and produce a fiber profile.

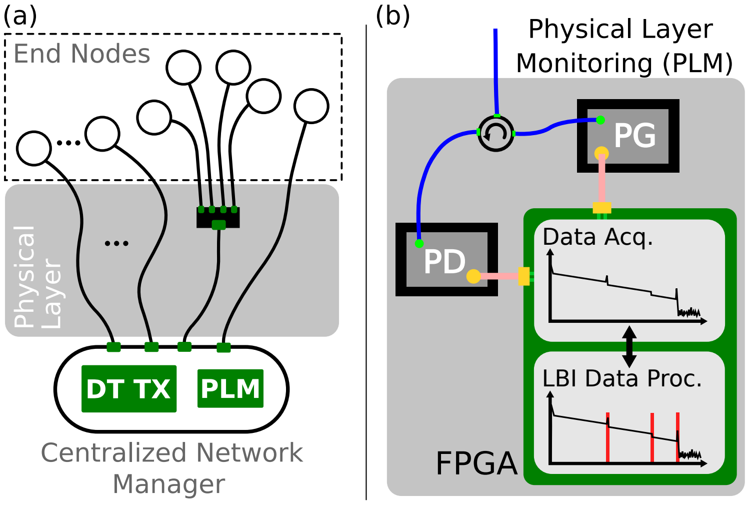


Figure 4.1: Motivation and applications of the proposed hardware-implementation of the Linearized Bregman Iteration algorithm. (a) (DT TX): data transmission unit; (PLM): physical layer monitoring unit. (b) (PG): pulse generation; (PD): pulse detection.

A novel hardware architecture for the LBI algorithm is presented and its main processing units are discussed in this chapter. A reduction factor on the elapsed algorithm time of approximately 100 could be achieved, which represents a substantial upgrade and warrants usage of dedicated hardware for trend break detection. The main overall contributions of this work are discriminated below:

- Speed-ups of the order of 100 times in the hardware implementation with respect to the software implementation of the LBI algorithm for trend break detection. The speed-ups are solely attributed to the hardware architecture and the usage of parallel-pipelined arithmetic units (such as the so-called pipelined adder tree) and to efficient memory organization and control, since no loss of performance is identified between hardware and software implementations.
- A comprehensive analysis of the hardware architecture, including the main processing and control units that allow for the algorithm structure to be implemented in hardware. This analysis provides a straightforward means to determining the total number of clock cycles necessary for the algorithm to elapse.
- A software implementation of the algorithm, that has been validated bitwise with respect to the hardware implementation, thereby allowing for the results of the proposed implementation to be verified in a simulated environment.
- Comparison, with respect to both performance and hardware characteristics, of the proposed hardware implementation with state-of-the-art hardware implementations of the Orthogonal Matching Pursuit algorithm, which is used in the context of sparse estimation and finds widespread interest in the current literature.

4.2 Architecture

It is a widespread notion that certain problems, despite their complexity, may be accelerated depending on the implementation; parallel programming, in which several parts of the same procedure are processed independently and simultaneously, is one of the most celebrated examples [133, 134]. Even though the LBI algorithm cannot be paralellized, there are computationally complex operations within the algorithm that can be paralellized to increase the processing efficiency. This is the case of the calculation of the instantaneous error after each iteration, which can be accelerated using a so called parallel adder tree (PAT). In order to harness the parallel speedup of the PAT, the entries of vector $\boldsymbol{\beta}$ must also be accessed in parallel, which can be accomplished through the instantiation of parallel Block RAMs (BRAMs), as depecited in

Fig. 4.3. The Pipelined Multiplexer Tree (PMT) is synchronized to the PAT such that, after a summation, the correct value of $\mathbf{y}$ is ready for subtraction. The value of μ_k is calculated in a pipelined CORDIC structure.

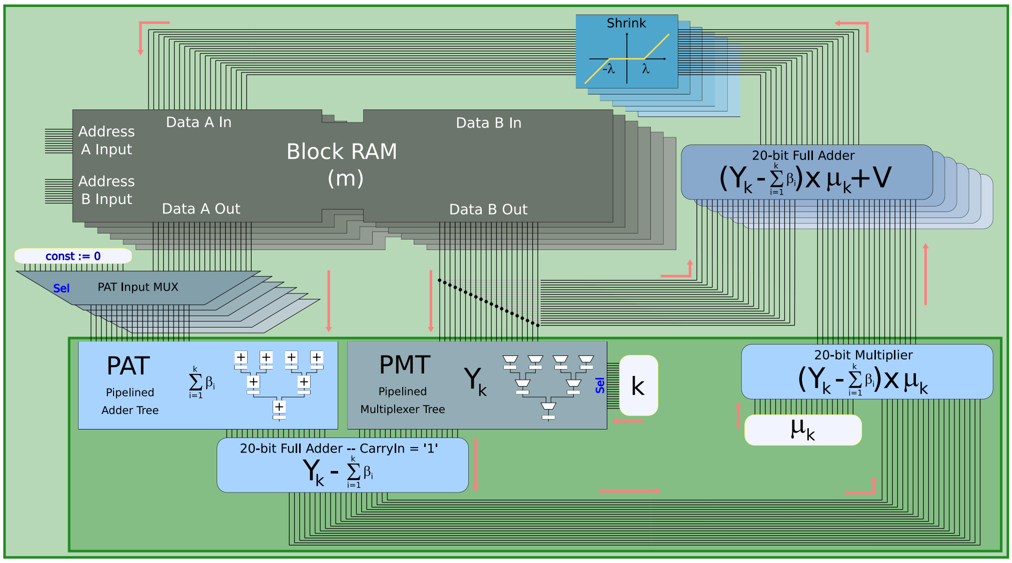


Figure 4.3: Hardware implementation of a single BRAM slice in the LBI core structure. The parallel structures are pictorially depicted in three-dimensional depth.

4.2.1 Master FSM

In order to control all the blocks depicted in Fig. 4.3, a so-called master Finite State Machine (FSM) has been implemented. The states, transitions and strobe signals depicted in Fig. 4.4 ensure the correct evolution of the algorithm. The FSM starts at an idle state then evolves to the loading state, where the control of the BRAMs is passed to the master FSM. Once the loading process has finished, the FSM waits for a "cordic_done" strobe to start the iterations. Each iteration consists of the following steps: the sum of betas, subtraction, multiplication, the update of $\mathbf{v}_j$, and the storage of $\mathbf{v}$ and $\boldsymbol{\beta}$. After L iterations, the maximum number of iterations, the unloading is performed, where the control of the BRAMs is passed to the master FSM, and the FSM returns to the idle state.

In order to validate the proposed hardware structure, the hardware has been synthesized and the maximum achievable clock frequency has been extracted. As said previously, PAT's performance depends on the number of entries of the vector $\boldsymbol{\beta}$ that can be accessed in parallel. However, as more BRAMs are used, the lower the maximum operating frequency of the FPGA will be. The results of Table 4.1 are in excellent agreement with the expectations

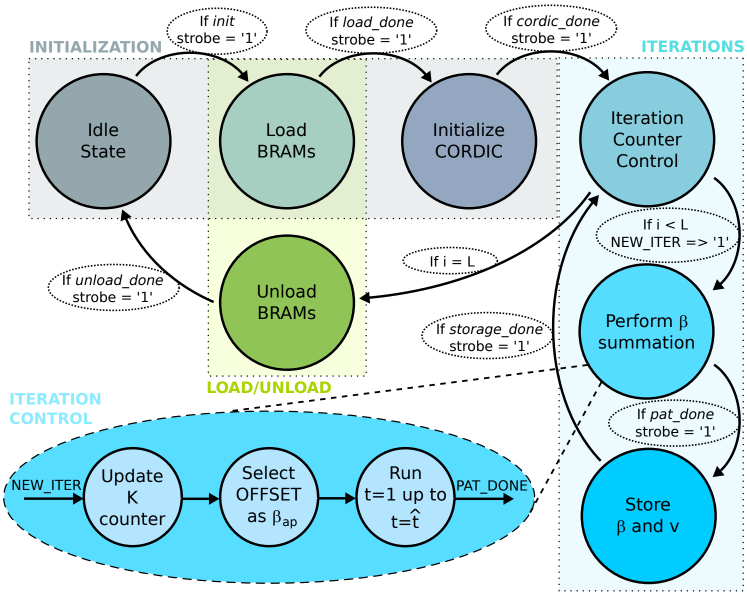


Figure 4.4: Block diagram of the master Finite State Machine.

in which it shows the influence of the resource usage, in terms of percentages of Adaptive Logic Modules (ALMs), on the maximum frequency of operation.

	Stratix V: 5SGSMD5K2F40C2				
BRAMs	ALMs	Registers	Memory [Bits]	Max. Clk.	Proc.
	out of 172,600		out of 41,246,720	Freq. [MHz]	Time [s]
1024	135,571(79%)	88,938	20,971,520(51%)	109.9	1.91
512	68,135(39%)	45,595	10,485,760(25%)	166.97	1.78
256	36,098(21%)	25,147	5,242,880(13%)	173.28	2.78
128	19,196(11%)	13,839	2,621,440(6%)	182.12	4.70
16	4,982(3%)	4,012	327,680(< 1%)	182.35	33.83
4	3,435(2%)	2,925	81,920(< 1%)	187.86	130.08

Table 4.1: FPGA Synthesis and Timing Results. Processing times calculated for $N = 10000$ and with $L = 6.5 \times 10^6$.

In [14], a software implementation of the same LBI was demonstrated, running on an Intel Xeon CPU E5-2690 v4 at 2.6 GHz and 512 GB RAM, with an algorithm elapsed time of 150 s for a 10000 point data series and considering 650 iterations per sample, i.e., the same conditions of the results of Table 4.1. The synthesized hardware achieved an almost two orders of magnitude speedup factor: a Stratix V FPGA with 1024 BRAMs running at a clock frequency of 109 MHz yielded a 1.91 second processing time, which translates to a speedup factor of 84 times.

4.3 Analysis

4.3.1 Real-World Dataset

The proposed hardware was subjected to test with a fiber profile acquired with a so-called Optical Time Domain Reflectometry (OTDR) device based on an FPGA [132]. This choice goes along with the motivation of the Introduction Section, since the acquisition system and the proposed hardware implementation of the analysis algorithm could, potentially, coexist in the same FPGA. The chosen fiber profile exhibits several interesting features related to fault detection: the presence of both high magnitude and low magnitude faults; and faults separated by few points. The results of both hardware and software implementations of the LBI algorithm are compared in Fig. 4.5. Here, one of the objectives is to analyze whether the codification of data in the hardware have an impact on the precision of the fault detection. This is because the 20-bit *SFIXED* format has been chosen to encode the data in the FPGA, as opposed to the 64-bit floating point in the software. The striking feature showcased in Fig. 4.5 is the fact that the *SFIXED* version of the algorithm, the blue trace, exhibits equivalent results to that of the red trace. The traces are vertically offset in order to highlight the correct trend break detection but the original baseline is the same. The inset displays a zoomed-in segment of the profile, where faults close to one another have been correctly identified by both versions of the algorithm.

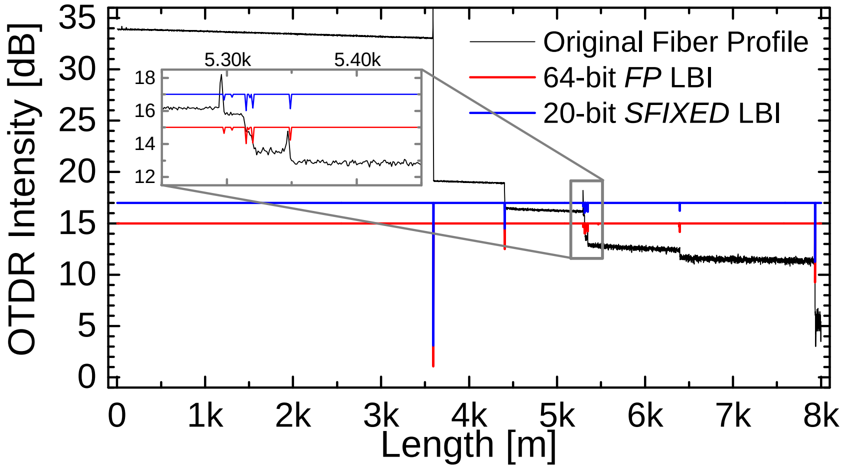


Figure 4.5: Results of trend break detection on a testbench real-world fiber profile acquired in laboratory conditions.

4.3.2 State-of-the-Art Comparison

One final question about the proposed hardware structure that implements the LBI algorithm is whether there are other methodologies that are better indicated for this context. An extensive literature review on this subject reveals that, when it comes to hardware implementations of sparse estimation methods, the majority of the works focus on the orthogonal matching pursuit (OMP) algorithm [135], which has, thus, been the focus of this comparison. The performance comparison entails the evaluation of figures of merit such as the *precision* and the MCC [136]. For the results of Table 4.2, both algorithms were fed multiple datasets, where the maximum number of trend breaks included in the datasets was five and the length was varied from 5000 to 15000 so that different dataset sizes could be considered. The performance of the LBI for trend break detection is much more pronounced, as can be seen from the results. At the same time, even though slower for small instance sizes, the complexity of the LBI structure allows for manipulation of much longer datasets, which is a necessity for trend break detection in the context of optical fiber analysis.

Algorithm	Precision	MCC
	$\left(\frac{TP}{TP+FP}\right)$	$\left(\frac{TP\cdot TN-FP\cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}\right)$
LBI	67.3%	0.81
OMP	8.2%	0.28

Table 4.2: Performance Comparison, where: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (TN).

4.4 Conclusions

The FPGA implementation of the Linearized Bregman Iterations algorithm adapted for trend break detection is a powerful signal processing tool as the results show that such an implementation provides impacting gains in processing time without loss of performance. From a digital hardware design perspective, the hardware structure of the algorithm combined with an efficient parallel memory access structure produces a pipelined design that can be easily controlled by a single master FSM. Furthermore, its comparison with state-of-the-art hardware implementations of sparse estimation algorithms (in particular, the Orthogonal Matching Pursuit algorithm), indicates clear advantages of the proposed architecture in terms of performance and hardware flexibility.

These results bring the goal of creating an embedded unit that combines data acquisition and processing a step closer to reality.

5
Profile-Splitting Linearized Bregman Iterations for Trend Break Detection Applications

Dealing with large datasets can be a tremendous challenge, even for low-complexity algorithms with efficient hardware implementations, as is the case of the Linearized Bregman Iterations (LBI) for Trend Break Detection (TBD) applications. The complexity of such algorithm can be shown to be quadratic with respect to the dataset length N. In this chapter, a new methodology is set forth called the profile-splitting, which allows for high-dimensional datasets to be processed within reasonable times without the loss of performance.

5.1
Motivation and Contributions

The TBD problem has a broad presence across different scientific fields with special recent special focus on fault detection in optical fiber profiles [14], where trend breaks are associated with faults. The datasets produced by optical fiber monitoring devices can contain several thousands of points [14] and, in such an application, fast trend break detection results are desired so that repair units can be quickly routed to fix optical network failures thus minimizing the downtime of the network. The profile splitting method is a way to reduce the required processing time by of a large dataset by evaluating multiple M-dimensional vectors that, together, compose the original data. Towards this goal, the main overall contributions of this work are:

- The unmatched trend break detection prowess of the LBI algorithm [14, 15] was upheld while dramatically increasing the processing speed by up to 5-fold factors with datasets as long as 50.000 points.
- The analysis of the impact of the signal-to-noise ratio (SNR) of a given sub-profile on the parameter λ of the LBI algorihtm and a simplified model that assumes the noise to be poissonian, allowing virtually any profile to be easily processed.
- Identification of a regime for the Mx1-dimensional sub-profiles where the performance is maintained while the processing times can still be reduced.

- Analysis of the potential advantages of the split-profile method when considering its hardware implementation and, furthermore, the resources in the currently available platforms.

5.2 Split-Profile Characterization

An interesting feature of the TBD problem in the context of fiber fault detection is the fact that the dataset contains sparse trend breaks but, also, a linear slope component associated with the attenuation experienced by light as it propagates in the fiber [60]. This, in turn, causes the SNR to decrease progressively as the distance between a certain point and the monitoring system increases. For this reason, as the original dataset is split into sub-profiles, a difference in SNR can be observed: sub-profiles associated with positions closer (in distance) to the measurement station (lower indices $n \in N$) exhibit higher SNR; those associated with distant positions (higher indices $n \in N$) exhibit lower SNRs. The SNR is a crucial aspect of signal processing using ℓ_1/ℓ_2 minimization, which is the core of the LBI algorithm.

$$\arg\min_{\boldsymbol{\beta}} \max_{\boldsymbol{\gamma}} \lambda\|\boldsymbol{\beta}\|_1 + \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 - \boldsymbol{\gamma}^T(\boldsymbol{A}\boldsymbol{\beta} - \boldsymbol{y}). \tag{5-1}$$

In the ideal noiseless case, the ℓ_2 portion of Eq. 5-1 would clearly be zero, and the algorithm would have no problem adding candidates and increasing the ℓ_1 portion of Eq. 5-1. As the SNR decreases, the algorithm must correctly balance the λ factor to achieve optimal signal denoising without flooding the selections with faults that are not really present in the dataset. In order to successfully process a low-SNR measurement, the λ factor can be increased, making the algorithm more tolerant to the squared error norm and more stringent in the selection of new candidates. The observation that the major noise source in the system can be modeled as a Poisson random variable [14,100] is a crucial point in consistently adjusting the λ factor for different sub-profiles. Furthermore, the ℓ_2 norm used to calculate the squared error between the estimated and original signals in the ℓ_1/ℓ_2 minimization essentially determines the norm of the error, which is defined as the square root of its total energy. Finally, by modulating λ according to the square root of a measurement's estimated SNR, the algorithm can adapt to different SNR regimes for each sub-profile generated out of an original OTDR profile, as shown below:

$$\lambda_{\text{sub-prof}} = \lambda_{\text{orig}}\sqrt{\text{SNR}_{\text{est}}}. \tag{5-2}$$

Figure 5.1 presents the rates of true and false positives, as well as the so-called Matthew's Correlation Coefficient (MCC) value output by two versions of the split-profile method (with and without the λ scaling and selection) and also by the original (non-split) LBI algorithm. The characterization introduced by the MCC makes it clear that there is a deep performance drop if λ is not correctly chosen. For these results, the split-profile length was fixed as 4500 points, and the original profile length was set to 15,000 points. A dataset of 1000 simulated profiles was used to increase the statistical relevance of the results.

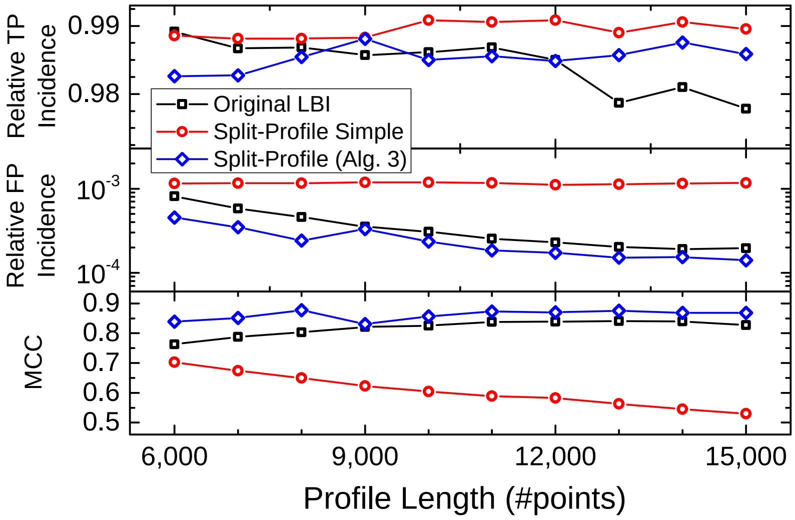


Figure 5.1: Performance comparison between three different versions of the analysis algorithm.

The value of $M = 4500$ has been determined based on an extensive characterization of the algorithm's performance versus the length of the sub-profiles. These results appear in Figure 5.2, where the original profiles, from which the sub-profiles were derived, contained 15,000 points. The baselines—in black dashes—correspond to the results of the original (non-split) LBI.

Finally, the timing comparison of the original and split-profile LBI's is showcased in Figure 5.3 for increasing dataset lengths. For shorter dataset lengths, the effect of the split-profile methodology is rather negligible since the original length is comparable to the individual split-profile length; however, as the dataset length increases above 20,000 points, a very clear distinction can be observed between the total elapsed times of the variants of the analysis algorithm. At 50.000 points, a 5-fold processing time advantage can be observed.

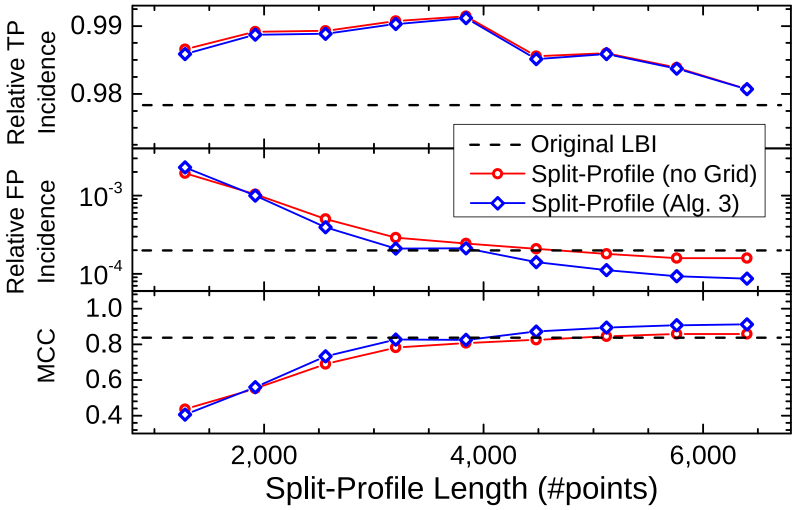


Figure 5.2: Performance of the split-profile methodology for varying split-profile lengths, in number of points.

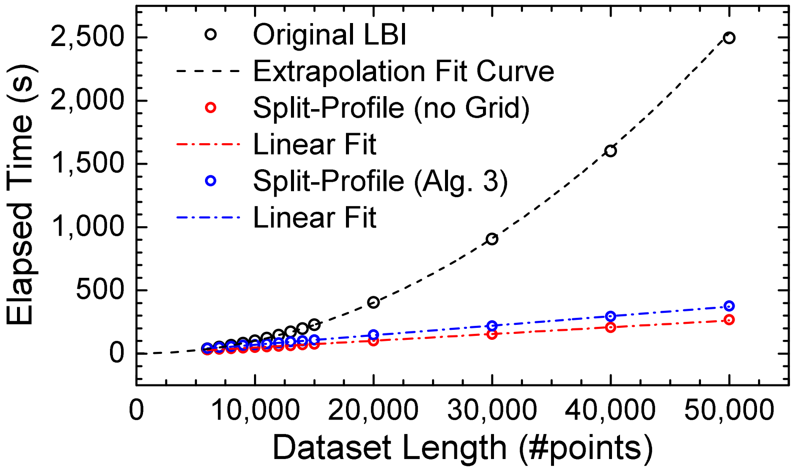


Figure 5.3: Final timing comparison between the original LBI and the corresponding split-profile LBI; two versions of the latter, with and without the λ-grid, are depicted.

5.2.1 Conclusions

The profile-split methodology applied to the LBI algorithm ensured comparable algorithm performance while overcoming the quadratic scaling of the elapsed time, which becomes a severe hindrance for long instance sizes. As it stands, no information from other sections of the original profile is necessary so that a given sub-profile can be processed, and thus, simultaneous processing of each of the sub-profiles is a possibility. Processing times in the order of hundreds of seconds for datasets as long as 50,000 points associated with an MCC value of 0.84 place the split-profile linearized Bregman iterations algorithm as the candidate of choice for trend break detection problems.

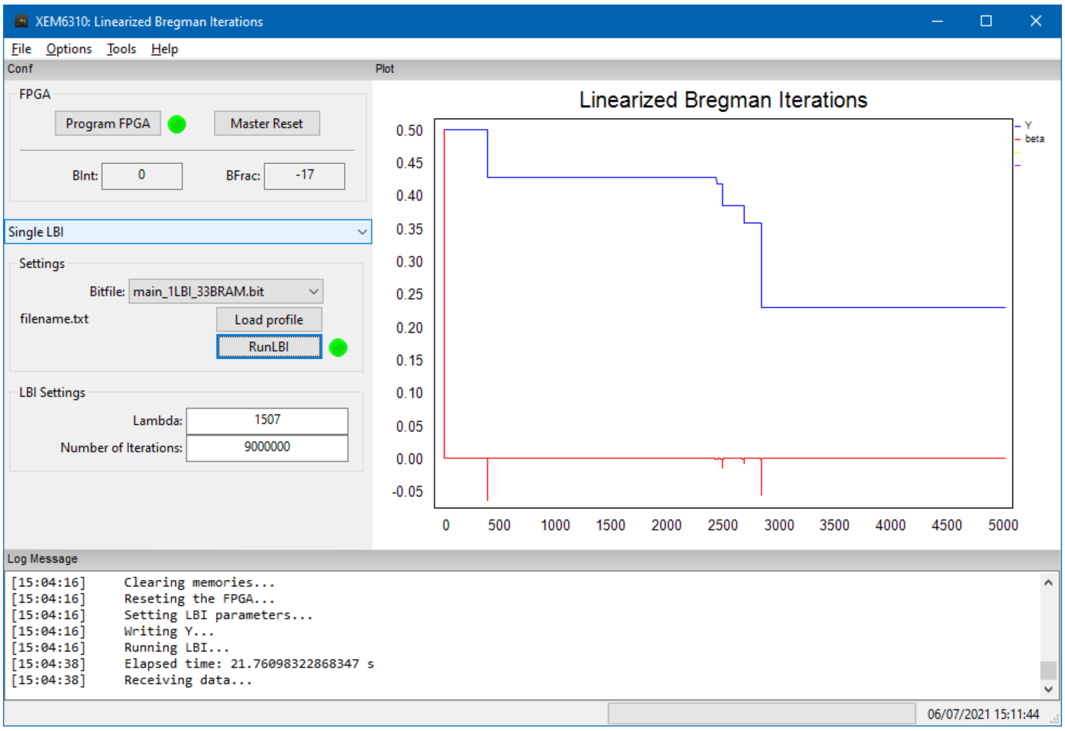
6 Conclusions

The fiber-optic communication systems emerged from the ever-increasing need for higher data transmission capacity. It is well known that fiber optics presents several advantages over other transmission media due to its immunity to electromagnetic interference, low power loss over distance, low manufacturing cost, and, mainly, due to its almost unlimited bandwidth. Nevertheless, optical fibers are made of glass, thus they are susceptible to mechanical stress and, under certain conditions, can be damaged or broken. Fiber optic companies often deploy redundant paths for high availability to avoid disruption of essential services during an in-field repairing. However, internet service providers usually use all available links, which increases their revenue. In these cases, a break may result in the end user's link being interrupted.

In order to identify such faults, a network operator routinely makes use of optical reflectometry devices. The role of a network operator is to identify fiber breaks and schedule an in-field repairing unit as fast as possible to ensure the robustness of all network layers. However, a network operator might have several thousand optical fibers under his responsibility. Thus, an automated fiber monitoring system is highly desired as it could alleviate the network operator's workload and, eventually, hasten and simplify the scheduling process of the in-field repairing units.

The main contributions of this thesis rely on developing the necessary building blocks for the creation of an automatic fiber monitoring system. Firstly, a new FPGA-based detection management system was developed to dramatically reduce the time needed to acquire an entire fiber profile in a High-Resolution PC-OTDR. The tunability and high-resolution features depend on generating narrow optical pulses spectrally adapted for DWDM systems. Thus, an optical pulse generation unit based on a figure-8 passively mode-locked laser was employed to produce 303 ps-wide optical pulses. Combining these units resulted in a fast acquisition tunable PC-OTDR with 4.6 cm spatial resolution and 15 dB dynamic range.

Subsequently, the Linearized Bregman Iteration algorithm adapted to Trend Break Detection was implemented in FPGA. This algorithm stands out for being a class of implementation-efficient and low-complexity algorithms for solving combined ℓ_1/ℓ_2 minimization problems. The implementation followed the steps presented in the following works [106], where the use of the Kaczmarz algorithm allowed the LBI algorithm to converge to an approximate least-squares solution (ℓ_2 norm) at a lower cost than other iterative methods, and [14], where optimizations towards the solution of Trend Break Detection problems, in special for OTDR applications, leads to a simpler matrix ($\mathbf{A}$) — composed mainly by ones and zeros, refer to Eq. 2-15, thus avoiding several complex hardware-multiplication operations. The resulting hardware implementation relied mainly on adders and efficient memory management controlled by a simple Finite State Machine (FSM), and presented speed-ups of the order of 100 times with respect to the software implementation. Furthermore, considering that the algorithm complexity can be shown to be quadratic with respect to the dataset length, further optimizations were implemented. These optimizations included a profile-splitting methodology and modulation of lambda according to the SNR difference between the generated sub-profiles and the original profile. It is important to emphasize that the actual implementation, see Fig. 6.1, does not count with slope detection, whose implementation is in progress and will be evaluated in future work.



BInt: 0 BFrac: -17

Single LBI

Bitfile: main_1LBI_33BRAM.bit

filename.txt

Lambda: 1507

Number of Iterations: 9000000

```
[15:04:16]     Clearing memories...
[15:04:16]     Reseting the FPGA...
[15:04:16]     Setting LBI parameters...
[15:04:16]     Writing Y...
[15:04:16]     Running LBI...
[15:04:38]     Elapsed time: 21.76098322868347 s
[15:04:38]     Receiving data...
```

06/07/2021 15:11:44

Figure 6.1: LBI fault detection results extracted from the FPGA.

6.1 Future works

Several works can be suggested as continuation of this study, the most relevant are summarized below.

- Implementation of the slope block in order to confirm the simulated results in [137]. Special attentions should be drawn to the effect of the inclusion of the slope component and to the value of σ, as shown in Algorithm 2.
- Analysis of the appearance of fault clusters due to a reduced number of iterations or a short distance between faults should be taken, as the appearance of fault clusters may diminish the measurement's spatial resolution. A method based on approximate deconvolution for increasing the spatial resolution and further reducing the number of iterations without affecting the performance of the algorithm is being studied.
- In this work, the building blocks for the creation of an automated fiber monitoring system were presented. However, it is extremely important to serialize the data obtained by the FPGA-based PC-OTDR so that these data can later be used by the Linearized Bregman Iterations adapted for Trend Break Detection. It is important to note that the OTDR stores data sequentially, i.e., $\mathbf{y} = [\mathbf{y(1)}, \mathbf{y(2)}, \mathbf{y(3)}, \mathbf{y(4)}, \ldots, \mathbf{y(N)}]$, while LBI algorithm's data storage is distributed over a number of parallel BRAMs structured as follows:

$$\begin{array}{cc} & \begin{array}{cccc} \text{BRAM}(1) & \text{BRAM}(2) & & \text{BRAM}(M) \end{array} \\ t\Big\uparrow & \begin{bmatrix} \mathbf{y}[TM+1] & \mathbf{y}[TM+2] & \cdots & \mathbf{y}[TM+M] \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{y}[2M+1] & \mathbf{y}[2M+2] & \cdots & \mathbf{y}[2M+M] \\ \mathbf{y}[M+1] & \mathbf{y}[M+2] & \cdots & \mathbf{y}[M+M] \\ \mathbf{y}[1] & \mathbf{y}[2] & \cdots & \mathbf{y}[M] \end{bmatrix} \\ & \xrightarrow[m]{\qquad} \end{array}, \tag{6-1}$$

where M is the number of parallel BRAMs and $T = \left\lceil \frac{N}{M} \right\rceil$, where the ceiling operator is denoted by $\lceil \cdot \rceil$.
- Furthermore, in order to have a fully embedded system, all components must be controllable and be placed side-by-side. Special attention must be taken to the figure-8 passively mode-locked laser, since a feedback loop control must be implemented to ensure the mode-locking operation [138].

Bibliography

[1] A. Gokhale, *Introduction to telecommunications.* Cengage Learning, 2004.

[2] G. P. Agrawal, “Optical Communication: Its History and Recent Progress,” in *Optics in Our Time*, M. D. Al-Amri, M. El-Gomati, and M. S. Zubairy, Eds. Springer International Publishing, 2016, pp. 177–198.

[3] K. C. Kao and G. A. Hockham, “Dielectric-fibre surface waveguides for optical frequencies,” in *Proceedings of the Institution of Electrical Engineers*, vol. 113, no. 7. IET, 1966, pp. 1151–1158.

[4] G. P. Agrawal, *Fiber-Optic Communication Systems*, 4th ed. John Wiley & Sons, Inc, 2010.

[5] T. Li, “Advances in optical fiber communications: An historical perspective,” *IEEE Journal on Selected Areas in Communications*, vol. 1, no. 3, pp. 356–372, 1983.

[6] G. Keiser, *Optical Fiber Communications*, 4th ed. McGraw-Hill, 2011.

[7] H. Gold, “Netflix and YouTube are slowing down in Europe to keep the internet from breaking,” [Online]. Available: https://edition.cnn.com/2020/03/19/tech/netflix-internet-overload-eu/index.html, March 2020.

[8] J. Pepitone, “Amazon back after rare site crash,” [Online]. Available: https://money.cnn.com/2013/01/31/technology/amazon-down/index.html, February 2013.

[9] F. Sato, K. Tsuchiya, Y. Nagao, T. Hirama, R. Oka, and K. Takahashi, “Ultra-high-fiber-count optical cable for data center applications,” *SEI Technical Review*, vol. 86, no. 45, pp. 45–50, 2018.

[10] A. Hornsteiner, “Fiber optic technology trends in data transmission: Digitalization of data advance the need for constant upgrading of data networks,” *Optik & Photonik*, vol. 12, no. 4, pp. 20–24, 2017.

[11] F. Calliari, L. Herrera, J. P. von der Weid, and G. Amaral, “Centimeter-Resolution Long-Distance Optical Fiber Monitoring,” in *Optics, Photonics and Laser Technology 2018.* Springer, Cham, 2019, pp. 33–56.

[12] G. C. Amaral, J. D. Garcia, L. E. Herrera, G. P. Temporao, P. J. Urban, and J. P. von der Weid, "Automatic fault detection in wdm-pon with tunable photon counting otdr," *Journal of Lightwave Technology*, vol. 33, no. 24, pp. 5025–5031, 2015.

[13] L. Herrera, G. Amaral, and J. P. von der Weid, "Ultra-high-resolution tunable PC-OTDR for PON monitoring in avionics," in *Optical Fiber Communications Conference and Exhibition (OFC), 2015.* IEEE, 2015, pp. 1–3.

[14] M. Lunglmayr and G. C. Amaral, "Linearized bregman iterations for automatic optical fiber fault analysis," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 10, pp. 3699–3711, 2018.

[15] F. Calliari, G. C. do Amaral, and M. Lunglmayr, "Fpga-embedded linearized bregman iteration algorithm for trend break detection," *Eurasip journal on wireless communications and networking*, vol. 2020, no. 1, pp. 1–26, 2020.

[16] J. Hecht, *Understanding Fiber Optics*, 5th ed. Pearson/Prentice Hall, 2005.

[17] T. H. Maiman, "Stimulated Optical Emission in Fluorescent Solids. I. Theoretical Considerations," *Phys. Rev.*, vol. 123, pp. 1145–1150, Aug 1961. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.123.1145

[18] T. H. Maiman, R. H. Hoskins, I. J. D'Haenens, C. K. Asawa, and V. Evtuhov, "Stimulated Optical Emission in Fluorescent Solids. II. Spectroscopy and Stimulated Emission in Ruby," *Phys. Rev.*, vol. 123, pp. 1151–1157, Aug 1961. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.123.1151

[19] H. J. R. Dutton, *Understanding Optical Communications*, ser. ITSO networking series. Prentice Hall PTR, 1998. [Online]. Available: https://books.google.com.br/books?id=iHCFQgAACAAJ

[20] J. Laferrière, G. Lietaert, R. Taws, and S. Wolszczak, "Reference Guide to Fiber Optic Testing – Volume 1," VIAVI Solutions, Tech. Rep., 2015. [Online]. [Online]. Available: https://comms.viavisolutions.com/Fiber-Optics-Test-Guide-Volume-1-vi96906

[21] R.-J. Essiambre, G. Kramer, P. J. Winzer, G. J. Foschini, and B. Goebel, “Capacity limits of optical fiber networks,” *Journal of Lightwave Technology*, vol. 28, no. 4, pp. 662–701, 2010.

[22] G. Ballou, *Handbook for sound engineers.* Taylor & Francis, 2013.

[23] T. Pearsall and M. Papuchon, “The $Ga_{0.47}In_{0.53}As$ homojunction photodiode — A new avalanche photodetector in the near infrared between 1.0 and 1.6 μm,” *Applied Physics Letters*, vol. 33, no. 7, pp. 640–642, 1978.

[24] T. Pearsall, “$Ga_{0.47}In_{0.53}As$: A ternary semiconductor for photodetector applications,” *IEEE Journal of Quantum Electronics*, vol. 16, no. 7, pp. 709–720, 1980.

[25] B. E. Saleh and M. C. Teich, *Fundamentals of Photonics*, 3rd ed. John Wiley & Sons, Inc, 2020.

[26] F. Calliari, “Automatic High-Dynamic and High-Resolution Photon Counting OTDR for Optical Fiber Network Monitoring,” M.Sc. dissertation, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil, 2017.

[27] J. Stern, D. Payne, D. McCartney, P. Healey, P. Lindsey, D. Russ, and J. Stewart, “Field installation of a 31.5 km monomode optical fibre system operated at 140 Mbit/s and 650 Mbit/s,” *Electronics Letters*, vol. 18, no. 14, pp. 631–632, 1982.

[28] G.-D. Khoe and H. Lydtin, “European optical fibers and passive components: status and trends,” *IEEE journal on selected areas in communications*, vol. 4, no. 4, pp. 457–471, 1986.

[29] J. M. Kahn, K.-P. Ho, and M. B. Shemirani, “Mode coupling effects in multi-mode fibers,” in *Optical Fiber Communication Conference.* Optical Society of America, 2012, pp. OW3D–3.

[30] G. P. Agrawal, *Nonlinear Fiber Optics*, 5th ed. Elsevier Science & Technology Books, 2013.

[31] W. F. Giozza, E. Conforti, and H. Waldman, *Fibras ópticas: tecnologia e projeto de sistemas.* Makron Books, 1991.

[32] T. Otani, K. Goto, H. Abe, M. Tanaka, H. Yamamoto, and H. Wakabayashi, “5.3 gbit/s 11300 km data transmission using actual submarine cables and repeaters,” *Electronics Letters*, vol. 31, no. 5, pp. 380–381, 1995.

[33] A. Gnauck, A. Chraplyvy, R. Tkach, J. Zyskind, J. Sulhoff, A. Lucero, Y. Sun, R. M. Jopson, F. Forghieri, R. Derosier *et al.*, "One Terabit/s Transmission Experiment," in *Optical Fiber Communication Conference*. Optical Society of America, 1996, p. PD20.

[34] S. Singh, A. Singh, and R. Kaler, "Performance evaluation of EDFA, RAMAN and SOA optical amplifier for WDM systems," *Optik*, vol. 124, no. 2, pp. 95–101, 2013.

[35] M. S. Bhamrah, A. Atieh *et al.*, "SOA/EDFA/RAMAN optical amplifier for DWDM systems at the edge of L & U wavelength bands," *Optical Fiber Technology*, vol. 52, p. 101971, 2019.

[36] K. Fukuchi, T. Kasamatsu, M. Morie, R. Ohhira, T. Ito, K. Sekiya, D. Ogasahara, and T. Ono, "10.92-Tb/s (273× 40-Gb/s) triple-band/ultra-dense WDM optical-repeatered transmission experiment," in *Optical Fiber Communication Conference*. Optical Society of America, 2001, p. PD24.

[37] Y. Miyamoto and H. Takenouchi, "Dense space-division-multiplexing optical communications technology for petabit-per-second class transmission," *NTT Technical Review*, vol. 12, no. 12, pp. 1–7, 2014.

[38] R.-J. Essiambre and A. Mecozzi, "Capacity limits in single-mode fiber and scaling for spatial multiplexing," in *OFC/NFOEC*, 2012, pp. 1–3.

[39] A. A. Saleh and J. M. Simmons, "All-optical networking—evolution, benefits, challenges, and future vision," *Proceedings of the IEEE*, vol. 100, no. 5, pp. 1105–1117, 2012.

[40] L. Li, P. G. Patki, Y. B. Kwon, V. Stelmakh, B. D. Campbell, M. Annamalai, T. I. Lakoba, and M. Vasilyev, "All-optical regenerator of multichannel signals," *Nature communications*, vol. 8, no. 1, pp. 1–11, 2017.

[41] M. Esmail, A. Ragheb, H. Fathallah, and S. Alshebeili, "Demonstration of photonics-based switching of 5g signal over hybrid all-optical network," *IEEE Photonics Technology Letters*, vol. 30, no. 13, pp. 1250–1253, 2018.

[42] E. F. Williams, M. R. Fernández-Ruiz, R. Magalhaes, R. Vanthillo, Z. Zhan, M. González-Herráez, and H. F. Martins, "Distributed sensing of microseisms and teleseisms with submarine dark fibers," *Nature communications*, vol. 10, no. 1, pp. 1–11, 2019.

[43] J. B. Ajo-Franklin, S. Dou, N. J. Lindsey, I. Monga, C. Tracy, M. Robertson, V. R. Tribaldos, C. Ulrich, B. Freifeld, T. Daley *et al.*, "Distributed acoustic sensing using dark fiber for near-surface characterization and broadband seismic event detection," *Scientific reports*, vol. 9, no. 1, pp. 1–14, 2019.

[44] J. P. Kilmer, A. DeVito, H. H. Yuce, C. J. Wieczorek, J. P. Varachi, and W. T. Anderson, "Optical cable reliability: lessons learned from post-mortem analyses," in *Fiber Optics Reliability: Benign and Adverse Environments IV*, vol. 1366. International Society for Optics and Photonics, 1991, pp. 85–92.

[45] S. Zemon, A. Budman, T. Wei, E. Eichen, and K. Ma, "Decay of transmitted light during fiber breaks-implications for break location," *Journal of Lightwave Technology*, vol. 12, no. 9, pp. 1532–1535, 1994.

[46] Infrapedia, Inc., "Terrestrial and Submarine Cable Map," [Online]. Available: https://www.infrapedia.com/app, May 2021.

[47] D. Derickson, *Fiber Optic Test and Measurement*, 1st ed. Prentice Hall, 1997.

[48] D. Gifford, M. Froggatt, M. Wolfe, S. Kreger, and B. Soller, "Millimeter resolution reflectometry over two kilometers," in *33rd European Conference and Exhibition of Optical Communication.* VDE, 2007, pp. 1–3.

[49] W. Eickhoff and R. Ulrich, "Optical frequency domain reflectometry in single-mode fiber," *Applied Physics Letters*, vol. 39, no. 9, pp. 693–695, 1981.

[50] H. Ghafoori-Shiraz and T. Okoshi, "Fault location in optical fibers using optical frequency domain reflectometry," *Journal of lightwave technology*, vol. 4, no. 3, pp. 316–322, 1986.

[51] G. C. Amaral, A. Baldivieso, J. D. Garcia, D. C. Villafani, R. G. Leibel, L. E. Y. Herrera, P. J. Urban, and J. P. von der Weid, "A Low-Frequency Tone Sweep Method for In-Service Fault Location in Subcarrier Multiplexed Optical Fiber Networks," *Journal of Lightwave Technology*, vol. 35, no. 10, pp. 2017–2025, May 2017.

[52] H. Barfuss and E. Brinkmeyer, "Modified optical frequency domain reflectometry with high spatial resolution for components of integrated

optic systems," *Journal of Lightwave Technology*, vol. 7, no. 1, pp. 3–10, 1989.

[53] G. Mussi, N. Gisin, R. Passy, and J. Von der Weid, "-152.5 db sensitivity high dynamic-range optical frequency-domain reflectometry," *Electronics Letters*, vol. 32, no. 10, pp. 926–927, 1996.

[54] R. MacDonald, "Frequency domain optical reflectometer," *Applied Optics*, vol. 20, no. 10, pp. 1840–1844, 1981.

[55] S. Venkatesh and D. W. Dolfi, "Incoherent frequency modulated cw optical reflectometry with centimeter resolution," *Applied optics*, vol. 29, no. 9, pp. 1323–1326, 1990.

[56] Z. Ding, X. S. Yao, T. Liu, Y. Du, K. Liu, Q. Han, Z. Meng, J. Jiang, and H. Chen, "Long measurement range ofdr beyond laser coherence length," *IEEE Photonics Technology Letters*, vol. 25, no. 2, pp. 202–205, 2012.

[57] J. Geng, C. Spiegelberg, and S. Jiang, "Narrow linewidth fiber laser for 100-km optical frequency domain reflectometry," *IEEE Photonics Technology Letters*, vol. 17, no. 9, pp. 1827–1829, 2005.

[58] Anritsu, "Understanding OTDRs," [Online]. Available: https://dl.cdn-anritsu.com/en-au/test-measurement/files/Technical-Notes/White-Paper/Anritsu_understanding_otdrs.pdf, 11 2011.

[59] B. L. Danielson, "Optical time-domain reflectometer specifications and performance testing," *Applied optics*, vol. 24, no. 15, pp. 2313–2322, 1985.

[60] M. Barnoski, M. Rourke, S. Jensen, and R. Melville, "Optical time domain reflectometer," *Applied optics*, vol. 16, no. 9, pp. 2375–2379, 1977.

[61] L. Thévenaz, *Advanced Fiber Optics: Concepts and Technology.* EPFL press, 2011.

[62] J. W. Strutt, "Xxxiv. on the transmission of light through an atmosphere containing small particles in suspension, and on the origin of the blue of the sky," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 47, no. 287, pp. 375–384, 1899.

[63] E. Hecht, *Optics*, 4th ed. Addison Wesley, 1998.

[64] R. Ellis, "Explanation of reflection features in optical fiber as sometimes observed in otdr measurement traces," *Corning White Paper*, 2007.

[65] J. E. Odhner, "Covert ground and port surveillance using hyperbox®: Rayleigh backscattering from fiber optics," in *2016 IEEE International Carnahan Conference on Security Technology (ICCST)*, Oct 2016, pp. 1–5.

[66] K. Tajima, "Low-loss optical fibers realized by reduction of rayleigh scattering loss," in *OFC'98. Optical Fiber Communication Conference and Exhibit. Technical Digest. Conference Edition. 1998 OSA Technical Digest Series Vol. 2 (IEEE Cat. No. 98CH36177).* IEEE, 1998, pp. 305–306.

[67] K. Saito, M. Yamaguchi, H. Kakiuchida, A. Ikushima, K. Ohsono, and Y. Kurosawa, "Limit of the rayleigh scattering loss in silica fiber," *Applied physics letters*, vol. 83, no. 25, pp. 5175–5177, 2003.

[68] Y. Tamura, H. Sakuma, K. Morita, M. Suzuki, Y. Yamamoto, K. Shimada, Y. Honma, K. Sohma, T. Fujii, and T. Hasegawa, "The first 0.14-db/km loss optical fiber and its impact on submarine transmission," *Journal of Lightwave Technology*, vol. 36, no. 1, pp. 44–49, 2018.

[69] W. Zhi, R. Guobin, L. Shuqin, and J. Shuisheng, "Loss properties due to rayleigh scattering in different types of fiber," *Opt. Express*, vol. 11, no. 1, pp. 39–47, Jan 2003. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-11-1-39

[70] D. Villafani Caballero, J. P. von der Weid, and P. J. Urban, "Tuneable OTDR measurements for WDM-PON monitoring," in *2013 SBMO/IEEE MTT-S International Microwave Optoelectronics Conference (IMOC)*, Aug 2013, pp. 1–5.

[71] N. Park, J. Lee, J. Park, J. G. Shim, H. Yoon, J. H. Kim, K. Kim, J.-O. Byun, G. Bolognini, D. Lee *et al.*, "Coded optical time domain reflectometry: principle and applications," in *Asia-Pacific Optical Communications*. International Society for Optics and Photonics, 2007, pp. 678 129–678 129.

[72] R. Liao, M. Tang, C. Zhao, H. Wu, S. Fu, D. Liu, and P. P. Shum, "Harnessing oversampling in correlation-coded OTDR," *Optics express*, vol. 27, no. 2, pp. 1693–1705, 2019.

[73] A. Wang and Y. Wang, "Chaos correlation optical time domain reflectometry," *Science China Information Sciences*, vol. 53, no. 2, pp. 398–404, 2010.

[74] Z. Xie, L. Xia, Y. Wang, C. Yang, C. Cheng, and D. Liu, "Fiber Fault Detection With High Accuracy Using Chaotic Signal From an SOA Ring Reflectometry," *IEEE Photonics Technology Letters*, vol. 25, no. 8, pp. 709–712, April 2013.

[75] X. Dong, A. Wang, J. Zhang, H. Han, T. Zhao, X. Liu, and Y. Wang, "Combined attenuation and high-resolution fault measurements using chaos-OTDR," *IEEE Photonics Journal*, vol. 7, no. 6, pp. 1–6, 2015.

[76] F. Scholder, J.-D. Gautier, M. Wegmüller, and N. Gisin, "Long-distance otdr using photon counting and large detection gates at telecom wavelength," *Optics communications*, vol. 213, no. 1-3, pp. 57–61, 2002.

[77] P. Eraerds, M. Legré, J. Zhang, H. Zbinden, and N. Gisin, "Photon counting otdr: advantages and limitations," *Journal of Lightwave Technology*, vol. 28, no. 6, pp. 952–964, 2010.

[78] J. Hu, Q. Zhao, X. Zhang, L. Zhang, X. Zhao, L. Kang, and P. Wu, "Photon-counting optical time-domain reflectometry using a superconducting nanowire single-photon detector," *Journal of lightwave technology*, vol. 30, no. 16, pp. 2583–2588, 2012.

[79] S. Cova, M. Ghioni, A. Lotito, I. Rech, and F. Zappa, "Evolution and Prospects for Single-Photon Avalanche Diodes and Quenching Circuits," *Journal of modern optics*, vol. 51, no. 9-10, pp. 1267–1288, 2004.

[80] F. Calliari, M. M. Correia, G. P. Temporão, G. C. Amaral, and J. P. Von Der Weid, "Fast acquisition tunable high-resolution photon-counting otdr," *Journal of Lightwave Technology*, vol. 38, no. 16, pp. 4572–4579, 2020.

[81] L. Herrera, F. Calliari, J. Garcia, G. C. Amaral, and J. P. von der Weid, "High resolution automatic fault detection in a fiber optic link via photon counting otdr," in *Optical Fiber Communication Conference.* Optical Society of America, 2016, pp. M3F–4.

[82] C. Xiong, M. Tang, C. Ke, Z. Feng, Q. Wu, L. Xu, S. Fu, W. Tong, P. P. Shum, and D. Liu, "Experimental demonstration of ultra-dense wdm-pon with seven-core mcf-enabled self-homodyne coherent detection," *IEEE Photonics Journal*, vol. 9, no. 2, pp. 1–7, 2017.

[83] X. Guan, W. Shi, and L. Rusch, "Ultra-dense wavelength-division multiplexing with microring modulator," *Journal of Lightwave Technology*, 2021.

[84] M. Delgado-Pinar, D. Zalvidea, A. Diez, P. Pérez-Millán, and M. Andrés, "Q-switching of an all-fiber laser by acousto-optic modulation of a fiber bragg grating," *Optics express*, vol. 14, no. 3, pp. 1106–1112, 2006.

[85] J. Huang, S. Huang, H. Chang, K.-W. Su, Y.-F. Chen, and K.-F. Huang, "Passive q switching of er-yb fiber laser with semiconductor saturable absorber," *Optics express*, vol. 16, no. 5, pp. 3002–3007, 2008.

[86] S. Fu, Q. Sheng, X. Zhu, W. Shi, J. Yao, G. Shi, R. A. Norwood, and N. Peyghambarian, "Passive q-switching of an all-fiber laser induced by the kerr effect of multimode interference," *Optics express*, vol. 23, no. 13, pp. 17 255–17 262, 2015.

[87] B. Ryvkin, E. A. Avrutin, and J. T. Kostamovaara, "Asymmetric-waveguide laser diode for high-power optical pulse generation by gain switching," *Journal of lightwave technology*, vol. 27, no. 12, pp. 2125–2131, 2009.

[88] A. Rosado, A. Pérez-Serrano, J. M. G. Tijero, Á. Valle, L. Pesquera, and I. Esquivias, "Enhanced optical frequency comb generation by pulsed gain-switching of optically injected semiconductor lasers," *Optics express*, vol. 27, no. 6, pp. 9155–9163, 2019.

[89] M. Bass *et al.*, *Fiber Optics Handbook: Fiber, Devices, and Systems for Optical Communications.* McGraw-Hill, 2021.

[90] M. A. Chernysheva, A. A. Krylov, P. G. Kryukov, and E. M. Dianov, "Nonlinear amplifying loop-mirror-based mode-locked thulium-doped fiber laser," *IEEE Photonics Technology Letters*, vol. 24, no. 14, pp. 1254–1256, 2012.

[91] M. M. Correia, P. Tovar, F. Calliari, G. C. Amaral, and J. P. von der Weid, "Picosecond pulse-width ultra-low repetition rate passive mode-locked fibre ring laser," *Electronics Letters*, vol. 56, no. 18, pp. 947–948, 2020.

[92] H.-G. Weber and M. Nakazawa, *Ultrahigh-speed optical transmission technology.* Springer Science & Business Media, 2007, vol. 3.

[93] Y. Chen, F. Kärtner, U. Morgner, S. Cho, H. Haus, E. Ippen, and J. Fujimoto, "Dispersion-managed mode locking," *JOSA B*, vol. 16, no. 11, 1999.

[94] M. J. Ablowitz, T. P. Horikis, and B. Ilan, "Solitons in dispersion-managed mode-locked lasers," *Physical Review A*, vol. 77, no. 3, p. 033814, 2008.

[95] X. Yang, Z. Li, E. Tangdiongga, D. Lenstra, G. Khoe, and H. Dorren, “Sub-picosecond pulse generation employing an soa-based nonlinear polarization switch in a ring cavity,” *Optics express*, vol. 12, no. 11, pp. 2448–2453, 2004.

[96] T. Liu, D. Jia, J. Yang, J. Chen, Z. Wang, and T. Yang, “An ultra-long cavity passively mode-locked fiber laser based on nonlinear polarization rotation in a semiconductor optical amplifier,” *Laser Physics*, vol. 23, no. 9, p. 095005, 2013.

[97] A. Komarov, H. Leblond, and F. Sanchez, “Passive harmonic mode-locking in a fiber laser with nonlinear polarization rotation,” *Optics communications*, vol. 267, no. 1, pp. 162–169, 2006.

[98] M. E. Fermann, F. Haberl, M. Hofer, and H. Hochreiter, “Nonlinear amplifying loop mirror,” *Optics Letters*, vol. 15, no. 13, pp. 752–754, 1990.

[99] P. E. Powers, *Field guide to nonlinear optics.* Society of Photo-Optical Instrumentation Engineers (SPIE), 2013.

[100] J. P. von der Weid, M. H. Souto, J. D. Garcia, and G. C. Amaral, “Adaptive filter for automatic identification of multiple faults in a noisy otdr profile,” *Journal of Lightwave Technology*, vol. 34, no. 14, pp. 3418–3424, 2016.

[101] A. S. Sedra, D. E. A. S. Sedra, K. C. Smith *et al.*, *Microelectronic circuits.* New York: Oxford University Press, 1998.

[102] R. C. Dorf and R. H. Bishop, *Modern Control Systems.* Pearson, 2011.

[103] K. Isogawa, T. Ida, T. Shiodera, and T. Takeguchi, “Deep shrinkage convolutional neural network for adaptive noise reduction,” *IEEE Signal Processing Letters*, vol. 25, no. 2, pp. 224–228, 2017.

[104] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for machine learning.* Cambridge University Press, 2020.

[105] J.-F. Cai, S. Osher, and Z. Shen, “Linearized bregman iterations for compressed sensing,” *Mathematics of computation*, vol. 78, no. 267, pp. 1515–1536, 2009.

[106] M. Lunglmayr and M. Huemer, “Efficient linearized bregman iteration for sparse adaptive filters and kaczmarz solvers,” in *Sensor Array and*

Multichannel Signal Processing Workshop (SAM), 2016 IEEE. IEEE, 2016, pp. 1–5.

[107] Z. Han, H. Li, and W. Yin, *Compressive sensing for wireless networks*. Cambridge University Press, 2013.

[108] S. Mallat, *A Wavelet Tour of Signal Processing*, 3rd ed. Boston: Academic Press, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123743701000136

[109] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM journal on computing*, vol. 24, no. 2, pp. 227–234, 1995.

[110] S. Osher, Y. Mao, B. Dong, and W. Yin, "Fast linearized bregman iteration for compressive sensing and sparse denoising," *Commun. Math. Sci.*, pp. 93–111, 2010.

[111] D. L. Donoho and M. Elad, "On the stability of the basis pursuit in the presence of noise," *Signal Processing*, vol. 86, no. 3, pp. 511–532, 2006.

[112] L. Hoeltgen, "Bregman iteration for optical flow," M.Sc. dissertation, Saarland University, 2010.

[113] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier analysis and applications*, vol. 14, no. 5, pp. 877–905, 2008.

[114] S. Karczmarz, "Angenaherte auflosung von systemen linearer gleichungen," *Bull. Int. Acad. Pol. Sic. Let., Cl. Sci. Math. Nat.*, pp. 355–357, 1937.

[115] Xilinx, Inc., "Spartan-6 FPGA Configuration: User Guide," [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug380.pdf, 2019.

[116] ——, "Spartan-6 FPGA Configurable Logic Block," [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug384.pdf, 2010.

[117] F. Faggin, "The making of the first microprocessor," *IEEE Solid-State Circuits Magazine*, vol. 1, no. 1, pp. 8–21, 2009.

[118] W. Aspray, "The Intel 4004 microprocessor: What constituted invention?" *IEEE Annals of the History of Computing*, vol. 19, no. 3, pp. 4–15, 1997.

[119] N. Tredennick, “Technology and business: forces driving microprocessor evolution,” *Proceedings of the IEEE*, vol. 83, no. 12, pp. 1641–1652, 1995.

[120] K. E. Murray, J. Luu, M. J. Walker, C. McCullough, S. Wang, S. Huda, B. Yan, C. Chiasson, K. B. Kent, J. Anderson *et al.*, “Optimizing fpga logic block architectures for arithmetic,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 6, pp. 1378–1391, 2020.

[121] D. L. N. Hettiarachchi, V. S. P. Davuluru, and E. J. Balster, “Integer vs. floating-point processing on modern fpga technology,” in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2020, pp. 0606–0612.

[122] B. Grady and J. H. Anderson, “Synthesizable heterogeneous fpga fabrics,” in *2018 International Conference on Field-Programmable Technology (FPT)*. IEEE, 2018, pp. 222–229.

[123] Xilinx, Inc., “Virtex UltraScale+ FPGA Data Sheet: DC and AC Switching Characteristics,” [Online]. Available: https://www.xilinx.com/support/documentation/data_sheets/ds923-virtex-ultrascale-plus.pdf, 2021.

[124] Altera Corporation, “Backgrounder. Stratix 10: The Most Powerful, Most Efficient FPGA for Signal Processing,” [Online]. Available: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/backgrounder/stratix10-floating-point-backgrounder.pdf, 2020.

[125] S. Kumar and M. J. Deen, *Fiber optic communications: fundamentals and applications*. John Wiley & Sons, 2014.

[126] P. J. Urban, G. Vall-Llosera, E. Medeiros, and S. Dahlfort, “Fiber plant manager: An otdr-and otm-based pon monitoring system,” *IEEE Communications Magazine*, vol. 51, no. 2, pp. S9–S15, 2013.

[127] K. O. Hill and G. Meltz, “Fiber bragg grating technology fundamentals and overview,” *Journal of lightwave technology*, vol. 15, no. 8, pp. 1263–1276, 1997.

[128] G. C. Amaral, L. E. Herrera, D. Vitoreti, G. P. Temporão, P. J. Urban, and J. P. der von Weid, “WDM-PON monitoring with tunable photon counting OTDR,” *IEEE Photonics Technology Letters*, vol. 26, no. 13, pp. 1279–1282, 2014.

[129] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical learning with sparsity: the lasso and generalizations*, 1st ed. Monographs on Statistics and Applied Probability 143: CRC press, 2015.

[130] U. Meyer-Baese, "Digital signal processing with field programmable gate arrays," *Springer*, 2004.

[131] F. Sato, K. Tsuchiya, Y. Nagao, T. Hirama, R. Oka, and K. Takahashi, "Ultra-high-fiber-count optical cable for data center applications," *SEI Technical Review*, no. 86, pp. 45–50, April 2018.

[132] F. Calliari, L. E. Y. Herrera, J. P. von der Weid, and G. C. Amaral, "High-dynamic and high-resolution automatic photon counting otdr for optical fiber network monitoring," *Proceedings of the 6th International Conference on Photonics, Optics and Laser Technology - Volume 1: PHOTOPTICS*, pp. 82–90, 2018.

[133] R. Kastner, J. Matai, and S. Neuendorffer, "Parallel programming for fpgas," *ArXiv e-prints arXiv:1805.03648*, May 2018.

[134] H. M. Waidyasooriya, M. Hariyama, and K. Uchiyama, "FPGA-Oriented Parallel Programming," in *Design of FPGA-Based Computing Systems with OpenCL*. Springer, 2018, pp. 45–74.

[135] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, March 2008.

[136] B. W. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," *Biochimica et Biophysica Acta (BBA)-Protein Structure*, vol. 405, no. 2, pp. 442–451, 1975.

[137] G. Castro do Amaral, F. Calliari, and M. Lunglmayr, "Profile-splitting linearized bregman iterations for trend break detection applications," *Electronics*, vol. 9, no. 3, p. 423, 2020.

[138] R. Iegorov, T. Teamir, G. Makey, and F. Ilday, "Direct control of mode-locking states of a fiber laser," *Optica*, vol. 3, no. 12, pp. 1312–1315, 2016.

A
Publications

In this appendix all the papers developed during the doctorate are listed.

Related to this work

The three papers related to this work were published in 2020. Following "chronologically", by chronologically I mean in the order in which they were developed and not in the order of publication, we have as first article: "*Fast Acquisition Tunable High-Resolution Photon-Counting OTDR*", which was published in Electronics Letters.

The other two articles talk about the implementation of the LBI algorithm in an FPGA. These are: "*FPGA-Embedded Linearized Bregman Iterations Algorithm for Trend Break Detection*", which was accepted for publication in EURASIP Journal on Wireless Communications and Networking in 2020, and "*Profile-Splitting Linearized Bregman Iterations for Trend Break Detection Applications*", which was published in the MDPI Electronics journal also in 2020.

Not related to this work

The articles and publications presented below are not or are not directly related to this work.

Papers

- "*Impact of a remote lab on teaching practices and student learning*" published in Computers & Education in 2018.
- "*Embedded time-multiplexed AMCC and OTDR signals for analog radio over fiber links*" published in Optics Communications in 2019.
- "*Picosecond pulse-width ultra-low repetition rate passive mode-locked fibre ring laser*" published in Electronics Letters in 2020.
- "*Alignment-free characterization of polarizing beamsplitters*" published in Applied Optics in 2019.

Conferences

- "*High-Dynamic and High-resolution Automatic Photon Counting OTDR for Optical Fiber Network Monitoring*" presented at the 6th International Conference on Photonics, Optics and Laser Technology, 2018, Funchal, Madeira, Portugal.
- "*Transmitter-Embedded AMCC, LTE-A and OTDR signal for Direct Modulation Analog Radio over Fiber Systems*" presented at the Optical Fiber Communication Conference, 2018, San Diego, USA.
- "*Remote Over-saturation Amplification of Optical fiber Monitoring Pulses*" presented at the 7th International Conference on Photonics , Optics and Laser Technology, 2019, Prague, Czech Republic.
- "*Concept Maps and Self-Regulated Learning: an Exploratory Study*" presented at the 2020 IEEE Global Engineering Education Conference (EDUCON), 2020, Porto, Portugal.

Chapters of Books

- "*Integration of an LMS, an IR and a Remote Lab*" as part of the Lecture Notes in Networks and Systems book series from Springer International Publishing in 2018.
- "*Centimeter-Resolution Long-Distance Optical Fiber Monitoring*" as part of the Springer Series in Optical Sciences book series from Springer International Publishing in 2019.

B
Fast Acquisition Tunable High-Resolution ν-OTDR

The paper "*Fast Acquisition Tunable High-Resolution Photon-Counting OTDR*" was published in the Journal of Lightwave Technology in 2020.

Fast Acquisition Tunable High-Resolution Photon-Counting OTDR

Felipe Calliari, Marlon M. Correia, Guilherme Penello Temporão, Gustavo C. Amaral, *Member, IEEE*, and Jean Pierre von der Weid, *Senior Member, IEEE*

***Abstract*—A 15 dB dynamic range and 4.6 cm spatial resolution tunable photon-counting optical time-domain reflectometer (PC-OTDR) is presented along with a Field Programmable Gate Array (FPGA)-based detection management system that allows several regions of the fiber to be interrogated by the same optical pulse, increasing the data acquisition rate when compared to previous solutions. The optical pulse generation is implemented by a tunable figure-8 passive mode-locked laser providing pulses with the desired bandwidth and center wavelength for WDM applications in the C-band. The acquisition rate is limited by the afterpulse effect and dead time of the employed gated avalanche single-photon detectors. The devised acquisition system not only allows for centimeter-resolution monitoring of fiber links as long as 12 km in under 20 minutes but is also readily adapted to any other photon-counting strategy for increased acquisition rate. The system provides a 20-fold decrease in acquisition times when ompared with state-of-the-art solutions, allowing affordable times centimeter-resolution long-distance fiber measurements.**

***Index Terms*—Optical fiber monitoring, optical time domain lectometry, single-photon detection.**

I. Introduction

OPTICAL fibers are, inarguably, one of the most important elements of modern telecommunication networks, having owed for long-haul high-bandwidth links to be implemented ound the world [1]. They offer, thus, the basis for the Open stems Interconnection (OSI) and its reliability is fundamental the robust operation of all the higher-level network layers. echanical damage, the protagonist of optical fiber transmission pairments, can cause from minor to debilitating losses and, erefore, must be evaluated timely and precisely [2].

One way of discovering the position and magnitude of the mechanical damage imposed into a fiber is to monitor the back-propagating light originated from an interrogation signal; the nature of such light falls into two classes: scattered light, usually associated with the Rayleigh backscattering of light from the atomic nuclei of the fiber material; and reflected light, usually associated with the discontinuity of index of refraction along the fiber or to a reflective structure placed along the fiber, such as a Bragg grating; the last of which figures as the basis of several of the proposed and implemented distributed fiber sensors [3].

When the intensity of the back-propagating light is measured over time, and the position of the mechanical damage is determined based on the speed of light inside the fiber, the method is known as Optical Time Domain Reflectometry (OTDR) [4]. In OTDR applications, spatial resolution and dynamic range come as a trade-off since strengthening the pulse for enhanced reach usually affects the pulse width and diminishes the 2-point resolution [5]. To alleviate such trade-off, a photon-counting OTDR (PC-OTDR) can be assembled, where a single-photon detector is employed and a detection management system must be integrated for consistent results [6]. In recent works [7], [8], PC-OTDR systems have been developed where a train of detection gates allows for several positions of the fiber to be interrogated with a single optical pulse. Employing the same mechanism in systems with centimeter-range resolutions, which can benefit both optical fiber monitoring and sensor applications, has been elusive. In [9], for instance, a system making use of two different PC-OTDR variants was demonstrated in order to achieve centimeter-range resolution in km-range fibers without the drawback of low data acquisition rate imposed by the high-resolution acquisition procedure. There, extra signal processing steps had to be taken into account, which reduced the total monitoring speed.

In this work, a 15 dB dynamic range and 4.6 cm spatial resolution tunable photon-counting optical time-domain reflectometer (PC-OTDR) is presented along with a joint Field Programmable Gate Array (FPGA) and Time-to-Digital Converter (TDC)-based detection management system that allows several regions of the fiber to be interrogated by the same optical pulse. The optical pulse generation is implemented by a figure-8 passively mode-locked laser containing a tunable optical filter that ensures that the probing pulses are generated within the desired bandwidth and center wavelength for WDM applications. The developed detection procedure allows increasing the data acquisition rate by a factor that depends on the operator's

Manuscript received October 28, 2019; revised March 28, 2020; accepted April 17, 2020. Date of publication April 27, 2020; date of current version July 28, 2020. This work was supported by brazilian agencies Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), and Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ). *(Corresponding author: Felipe Calliari.)*

Felipe Calliari, Marlon M. Correia, Guilherme Penello Temporão, and Jean Pierre von der Weid are with the Center for Telecommunications Studies, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro - RJ 22451-900, Brazil (e-mail: felipe.calliari@opto.cetuc.puc-rio.br; mcorreia@opto.cetuc.puc-rio.br; temporao@opto.cetuc.puc-rio.br; vdweid@opto.cetuc.puc-rio.br).

Gustavo C. Amaral is with the Center for Telecommunications Studies, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro - RJ 22451-900, Brazil, and also with the QC2DLab, Kavli Foundation, Technical University of Delft 2628 CD, Delft, The Netherlands (e-mail: gustavo@opto.cetuc.puc-rio.br).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JLT.2020.2990872

choice of detection window length and detection gate train period. The results indicate that portions of the fiber separated by 200 meters can be interrogated by the same optical pulse without debilitating effects coming from the afterpulse effect. Centimeter-range monitoring of km-range fibers in under 20 minutes in a wavelength division multiplexing (WDM) scenario are demonstrated in this paper.

The paper is divided as follows. The architecture and its sub-systems are presented in Section II, where the arrangement of the devices, individual characteristics, and data management are presented and discussed. Section III approaches the characteristics of the full system with respect to its achievable spatial resolution and dynamic range, tunability, acquisition rate, and limitations on such. Section IV provides discussion on possible applications and optimizations of the system and concludes with the major contributions and future works.

II. Architecture

The basic operation of a PC-OTDR begins with the optical pulse generation sub-system, which creates light pulses at a rate such that only one pulse traverses the fiber at a time. Under this condition, back-propagating light created due to either scattering or reflection of the interrogation pulse can be collected by means n optical circulator and sent to a single-photon detector. It is task of the data acquisition sub-system to correctly associate ections to time windows such that, with an estimate of the r's refractive index and the speed of light, one can determine positions where the back-propagating signals originate from. nathematical form, $d_i = \frac{c}{2\,n}\,t_i$, where c is the speed of light in uum, n is the refractive index of the fiber and t_i is the round-travel time of the i-th detected pulse, taken into account by factor 2 in the denominator.

n order to simplify the analysis of the OTDR signal, one can ore the non-linear effects associated to the transmission of ical signals, which allow the employment of linear system tractions such as the transfer function and the impulse response. In this context, and since the optical fiber is a two-port work, the OTDR procures to measure the impulse response) of the fiber's S_{11} parameter, or $h_{11}(t)$; in fact, optical frequency-domain reflectometry (OFDR) techniques [10] such as the Incoherent- and Coherent-OFDR attempt to assess the transfer function $H_{11}(f)$ of the fiber, which is intimately related to $h_{11}(t)$ via a Fourier transform. Due to the intrinsic attenuation of light inside an optical fiber, $h_{11}(t)$, which, when translated into distance is usually referred to as the fiber profile, exhibits an exponential decay; it is, thus, usual to display the fiber profile in logarithmic scale with a linear negative slope, where eventual breaks can be associated to power losses [11], [12].

It becomes clear that the spatial resolution achievable in the measurement of the fiber profile is limited by the temporal width of the interrogating pulse, $p(t)$, since

$$h_{11}^{\text{eff}}(t) = h_{11}(t) * p(t), \tag{1}$$

where $h_{11}^{\text{eff}}(t)$ stands for the effectively measured fiber profile. At the same time, the overall energy contained in $p(t)$ will determine the amount of attenuation that it can withold while still allowing for a higher-than-one signal-to-noise ratio (SNR > 1) measurement in the detector. Assuming saturation of the optical pulse generation structure, in order to increase the optical energy contained in the probing pulse, its width must be increased, thereby constituting the so-called trade-off between spatial resolution and dynamic range. Attempting to extract the maximum of both ends of this trade-off, one can employ sources of high-power narrow optical pulses and low noise-equivalent power (NEP) detectors. Moreover, management and synchronization of pulse generation and data acquisition is imperative for efficient long-reach high-resolution measurement. In the following sub-sections, these two sub-systems (optical pulse generation and data acquisition) are presented in detail.

A. Optical Pulse Generation

The optical pulse generation sub-system is comprised of a figure-eight passively mode-locked laser (F8-PMLL). Such laser configuration has been shown to produce sub-picosecond pulses at different wavelengths [13], rendering its application for a tunable, long-reach, high-resolution time-domain reflectometer. Mode-locked lasers rely on saturable absorbers to create a fixed phase relation between the spectral modes allowed within its optical resonator cavity. Under the mode-locked condition, the temporal shape of the output signal is given by a coherent weighted sum of such allowed spectral modes or, mathematically:

$$p(t) = \sum_{n=-N/2}^{N/2} c_n e^{-i2\pi(\omega_0 + n\cdot f_0)t}, \tag{2}$$

where c_n are the individual weigths of each spectral modes (or Fourier coefficients), ω_0 is the laser's center frequency, f_0 is the optical cavity's free-spectral range (FSR), and N is the total number of allowed spectral modes within the laser bandwidth $\Delta\omega$, where $N = \Delta\omega / f_0$. Equation 2 states that the cavity modes constitute a Fourier series of the output signal, which will correspond to periodically emitted pulses since the spectrum, during mode-locked operation, takes the form of a frequency comb.

Mode-locking can be achieved either actively (through intensity and/or phase modulation) or passively in either free-space or fiber cavities [14]. Among others, the main advantages of fiber-based mode-locked lasers include low propagation loss, the possibility of creating high Q value kilometer-long fiber cavities, and the fact that erbium-doped fiber amplifiers (EDFAs) and fiber-pigtailed semiconductor optical amplifiers (SOAs) can be used as the gain media [14]. Figuring as physical mechanisms that guarantee mode-locking in PML fiber lasers are nonlinear polarization rotation [15], dispersion-managed stretched pulse propagation [16], and the constructive/destructive interference of counter-propagating longitudinal modes in a so-called nonlinear amplifying loop mirror (NALM) [17]. The latter allows for the optical switching of the input optical pulses according to the following mathematical relations:

$$\begin{aligned} \mathcal{T} &= \tfrac{I_{\text{out}}}{I_{\text{in}}} = \tfrac{\mathrm{G}(1-\cos[\Delta\phi_{\text{NL}}(\mathrm{G}-1)])}{2} \\ \mathcal{R} &= 1 - \mathcal{T}, \end{aligned} \tag{3}$$

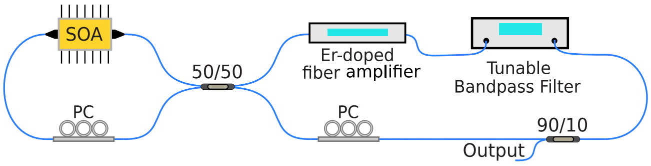


Fig. 1. The overview of the passively mode-locked figure-8 fiber laser. On the left-hand side, the nonlinear amplifying loop mirror (NALM) is depicted. On the right-hand side, the optical cavity with an unidirectional EDFA that determines the direction of propagation, and a variable optical bandpass filter that determines the center wavelength of emission are depicted. PC, Polarization Controller.

where $I_{\text{in/out}}$ are the input/output optical intensities, G is the amplifier gain, $\Delta\phi_{\text{NL}}$ is the difference in non-linear phase shift between the counter-propagating fields in the NALM, and $\mathcal{T}$ and $\mathcal{R}$ are the transmissivity and reflectivity of the NALM, respectively.

When the NALM is combined with a closed fiber loop, it constitutes a figure-8 laser (F8L), an extremely versatile fiber structure that finds applications in, for instance, soliton generation, and has been first reported by Duling [18]. Recently, an F8L realized with an SOA in the NALM and an EDFA in the optical cavity has been reported [19], whose design is reproduced in this work and depicted in Fig. 1. The NALM (left-hand side in Fig. 1) is composed by an SOA, a Polarization Controller (PC), and single-mode fibers totalling a length of $L_1 \approx 8.42$ meters. The optical cavity (right-hand side in Fig. 1) is formed by a PC, a 90/10 coupler (that allows one to couple a portion of the optical signal generated in the F8L to the output), a tunable band pass filter (BPF),an unidirectional EDFA (incorporating an optical isolator), and single-mode fibers totaling a length of $L_2 \approx 36.38$ meters.

In the NALM, a periodic transmission and reflection of the optical pulses occurs as a function of the instantaneous input power and the non-linear phase shift between the longitudinal counter-propagating modes supported by the NALM cavity. Therefore, if an optical pulse is formed in the F8-PMLL and its peak power satisfies the transmission conditions of the NALM, it will propagate in the F8-PMLL according to the direction dictated by the optical isolator; otherwise, it will be reflected by the NALM and eliminated by the EDFA's isolator [20]. The polarization controller of the left-hand side loop (1) can be adjusted to control the birefringence in the fiber and, in turn, the non-linear phase shift factor in Eq. 3. This way, the interference of the counter-propagating longitudinal modes on the 50/50 coupler can be adjusted such that only input pulses of a certain intensity are transmitted, while lower pulses are reflected and extinguished in the EDFA's isolator. The center wavelength of emission of the F8L is dictated by the BPF, which is tunable in the range 1530–1565 nm covering the telecommunication C-band as depicted in Fig. 2.

Under fundamental mode operation, the F8-PMLL presented in Fig. 1 produces 303 ps-wide pulses spaced by 224 ns (a repetition rate of 4.46 MHz), the former determined by a total 1.39 GHz emission bandwidth and, the latter, by the overall length of the fiber cavity of 44.8 meters. To achieve such conditions, other than correctly adjusting the PCs, the injection current of the SOA should be approximately 300 mA, which translates into optical pulses with 4 pJ energy level and 13 mW peak power. Under these conditions, the wavelength tunability of the F8-PMLL is rather simple in terms of PC alignment; on the other hand, the pulse peak power limits the achievable dynamic range. In order to maintain the ease of tunability of the system while also increasing the dynamic range, the output pulses were sent to a second EDFA, which yields optical pulses with 25 pJ energy level and 83 mW peak power. Due to the broad bandwidth of the EDFA, the width of the optical pulses, after amplification, was not compromised, as presented in Fig. 3; the output spectrum, measured in a electrical spectrum analyzer (ESA), corroborates the timing characteristics of the optical signal, and is presented in Fig. 4.

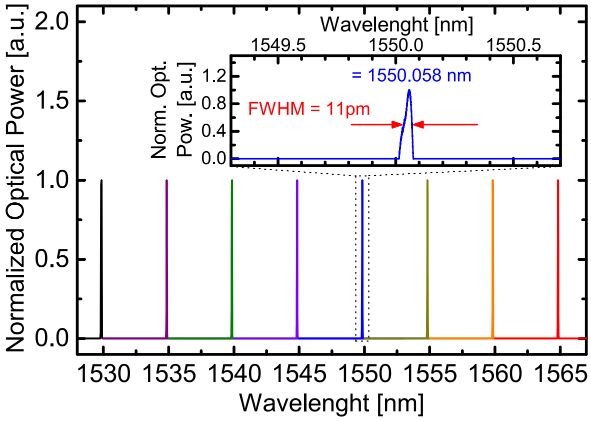


Fig. 2. Center wavelength tunability of the employed F8-PMLL. The bandwidth, measured with a high-resolution (16 MHz) optical spectrum analyzer, is 11 pm at all center wavelengths.

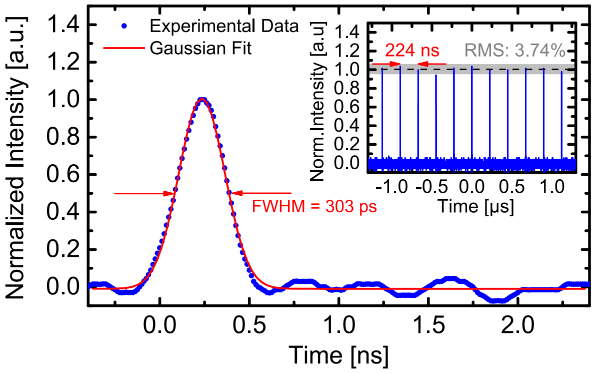


Fig. 3. Temporal shape of the generated optical signal in the passive mode-locked figure-8 laser. The pulse separation of 224 ns matches the resonator length of 44.8 meters and the pulse temporal width is 303 ps, which limits the spatial resolution of the PC-OTDR to 3.07 cm.

B. Data Acquisition

Data acquisition is performed in synchronization with an enabling pulse from a high-speed optical switch that guarantees that a single probing pulse is traversing the fiber at a time. This is necessary since the rate of pulse emission from the PML described in the previous subsection (which is entirely

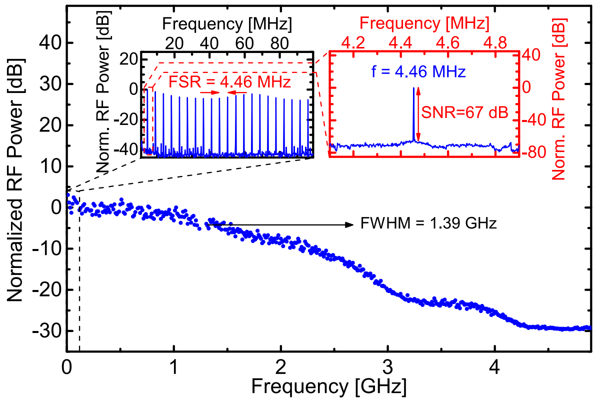


Fig. 4. Spectral characterization of the generated optical signal with respect to the central wavelength of 1550.058 nm. The spectral comb-shape exhibits a finesse $\mathcal{F} = 6371.43$, with a bandwidth (δf) of 700 Hz and a free-spectral range (FSR) of 4.46 MHz, which matches the resonator length of 44.8 meters.

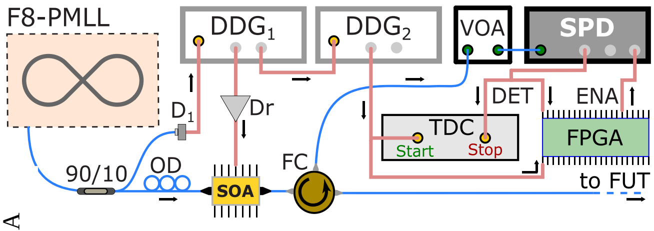


Fig. 5. Block diagram of the data acquisition system including the synchronization with the F8-MPLL; optical fibers are blue lines and pink lines are electrical signals. D$_1$, *p-i-n* photodetector; Dr, electronic driver; OD, optical delay; FC, fiber circulator; VOA, variable optical attenuator; FUT, fiber under test. The black arrows indicate the direction of the electric and optical signals for ease of visualization.

determined by the cavity length of the optical resonator) might match this condition. The pulse selection is accomplished through an SOA triggered by a short (2 ns, 300 mA) pulse, which, under these conditions, imparts a 1 dB insertion loss on the optical pulse and creates an ASE noise level of −9.6 dBm, setting the probing pulse's extinction ratio to 27 dB. This triggering pulse, in turn, is synchronized to the incoming pulse from the PMLL by detecting a small portion of its output using a 90/10 beam splitter.

The block diagram of the data acquisition sub-system is depicted in Fig. 5. Synchronization is crucial not only between the PML and the SOA but, also, between these and a Field Programmable Gate Array (FPGA), a Time-to-Digital Converter (TDC), and the single-photon detector that manage the detection of the backscattered portion of the probe pulse. A Digital Delay Generator (DDG$_1$), triggered by detections from the F8-PMLL pulses, is responsible for reducing the rate of pulses launched into the fiber by means of an internal prescaler. It is also responsible for generating a delayed trigger (τ_p) for a second DDG (DDG$_2$) that works in burst mode; the importance of τ_p will be clarified further on. DDG$_2$ then generates a sequence of evenly-spaced (τ_b) electrical pulses that act on the TDC as a *start* pulse and are also routed to the FPGA. The FPGA is programmed such that the pulse received by DDG$_2$ goes through an AND gate and is enabled according to a pre-defined signal e_r, whose function will also be clarified further on; presently, e_r is considered always high, for simplicity.

The output of the FPGA's AND gate is directed to the enable input of the SPD, opening a detection window (τ_d) whose width can be tuned between 5 and 35 ns. If a detection occurs within this detection window, the detection pulse of the SPD is directed to the *stop* input of the TDC where the time difference between *start* and *stop* pulses is determined. The detection pulse is also sent to the FPGA, that records the occurrence of a detection. In case no detection occurs, the subsequent pulse from DDG$_2$ will reinitialize the process and the FPGA will record the absence of a detection in the previous detection window. The FPGA can, then, produce a simple binary indexed list to backtrack the detection events: 1's correspond to the occurrence of a detection; and 0's to no detection.

The fact that the system is able to reinitialize the data acquisition at every new cycle of the burst of DDG$_2$ allows a single optical pulse to probe regions of the fiber spaced by $d = \frac{c}{2n}\tau_b$. In order to reconstruct the fiber profile, one is required to measure the time between the pulse being launched into the fiber and any of the *stop* pulses, which can be accomplished by combining the information from the FPGA and the TDC. This is due to the TDC's data storage procedure, which piles up the time between the *start* and *stop* pulses only when a corresponding *stop* occurs; the FPGA provides, thus, a means of associating the times in the TDC to the correct detection windows which produced them. Furthermore, since τ_p represents an offset in time between the probing pulse being launched into the fiber and the opening of the detection windows, it must also be taken into account in order to write any detection time as:

$$\tau_{\text{det}} = \tau_p + i\tau_b + \tau^i_{\text{TDC}}, \tag{4}$$

where i corresponds to the index of the respective enabling pulse of DDG$_2$ in which a detection was recorded and τ_{TDC} is the time stored in the TDC after a detection.

Clearly, in order to probe all the points in the fiber, τ_p must be swept so that it covers the distance between two consecutive detection windows. The total number of steps for τ_p can then be calculated according to $\frac{\tau_b}{\tau_d}$ and has a direct impact on the total measurement time, as will be discussed in the next section. In order to clarify the whole procedure, Fig. 6 depicts, in (a), the time panels of all the relevant optical and electrical signals associated to the data acquisition subsystem. In (b), the event lists contained within the FPGA and the TDC are pictorially depicted as well as the procedure for combining the two lists and creating the fiber profile.

III. The Fast Acquisition Tunable High-Resolution PC-OTDR

Combining the previously described sub-systems culminates in an optical reflectometry-based measurement system that yields a spatial resolution in the centimeter range with both achievable dynamic range and data acquisition rates higher than related state-of-the-art systems; furthermore, the system is tunable around the telecommunication C-band and exhibits narrow bandwidth, making it ideal for supervision of currently

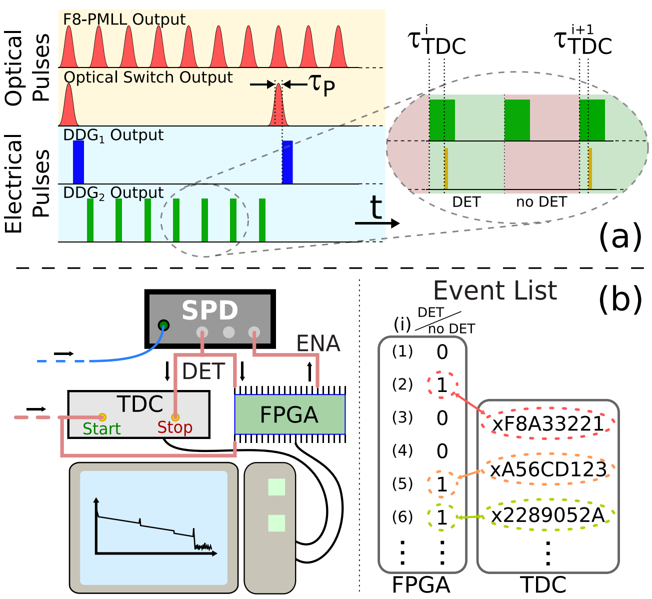


Fig. 6. Data acquisition subsystem procedure. (a) Time panel of relevant electrical and optical pulses. The zoomed region represents two different situations, i.e., detection windows with and without an associated detection pulse. Furthermore, the time between the enabling of the detector and the arrival of the detection pulse (different for the two depicted events) is recorded by the TDC as τ_{TDC}. (b) The detection system for reconstruction of the fiber profile based on the event lists from the FPGA and the TDC. On the left hand side, the connections between the devices is depicted as a block diagram including a personal computer; optical fibers are blue lines, pink lines are electrical signals, and black lines are data buses. The right hand side depicts the combination of the results from the event lists that allow for the reconstruction of $\tau_{\det}$.

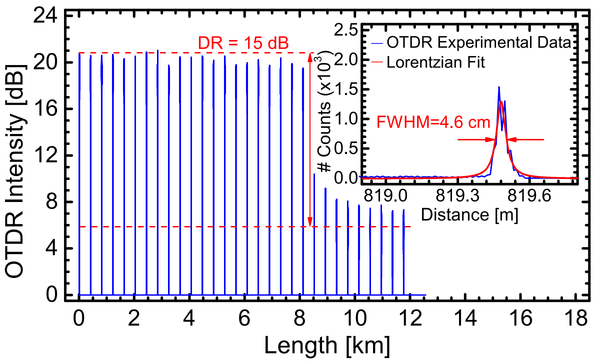


Fig. 7. Dynamic range (15 dB) and spatial resolution (4.6 cm) achieved with the proposed OTDR system. The wavelength of operation of the F8-PMLL for these results was 1550.058 nm, and the total measurement time was 5 minutes.

deployed optical networks relying on wavelength division multiplexing (WDM), such as WDM Passive Optical Networks (WDM-PON).

In order to characterize the system's operation parameters, OTDR profiles of different fibers have been measured; depending on the figure of merit of interest, the condition of the measurement also changed. This is because, even with the higher data acquisition rates propitiated by the proposed system, measurements of km-long fibers that achieve the full dynamic range of the system still require long measuring times. In Fig. 7, for instance, the dynamic range and spatial resolution are showcased using an 8-km fiber; only the measurements of detection windows associated to a fixed τ_p were acquired, however, but during an extended 5-min measurement time (t_{meas}).

For this measurement, a detection window $\tau_d = 20$ ns and a burst period $\tau_b = 2$ μs were used, yielding a 15 dB dynamic range, calculated by taking the difference, in dB, between the initial point of the profile and the 1.5-dB point below the noise level (which coincides with the average value of the noise) [21], i.e., detections associated to positions "outside" the fiber. The value of the dynamic range agrees well with the prediction based on the mean number of photons per detection window at the initial position of the fiber (0.45), the rate of emission of probing pulses into the fiber (9 KHz), and the dark count rate of the detector (5 per second). The VOA at the input of the SPD guarantees that the mean number of photons impinging on the SPD is such that multi-photon detection is reduced so that the OTDR trace exhibits no saturation effects [8]. The VOA attenuation is set to ~15 dB, calculated based on the pulse peak power at the input of the fiber, the photon's energy, and the Rayleigh scattering coefficient. Under these conditions, the rate difference between detections (4100 per second) and dark counts equates to 14.5 dB, a good estimate of the achieved dynamic range in Fig. 7. The SPD operates at an efficiency of 5%, and the counting rate is 180,000 per second.

The total amount of time necessary to reproduce the results with the same dynamic range, but for all points in the fiber, would be $\frac{\tau_b}{\tau_d} t_{\mathrm{meas}} \approx 8$ hours, which is the reason why these were not measured. The spatial resolution was determined by the full width at half maximum of a lorentzian fit into a reflection peak at the fiber's end to be 4.6 cm as shown in the inset of Fig. 7. This value, which differs from the limit induced by the temporal width of the probing pulse, is attributed, mainly, to the jitter of the SPD, although the jitter associated to the F8-PMLL and to both the TDC and DDG also contribute. These values have been experimentally determined to be 350 ps, 20 ps, 40 ps, and 45 ps, respectively, which, combined with the pulse width, amount to a spatial resolution of 4.9 cm, which agrees well with the experimentally determined values.

In order to showcase the capability of the system to perform tunable and fast fiber profile measurements with centimeter resolution, an arrayed waveguide grating (AWG) was employed. A 4-km feeder fiber (between the measurement system and the AWG) is followed by four different fibers ($L_1 = 4$ km, $L_2 = 8$ km, $L_3 = 0.9$ km, $L_4 = 3$ km) each connected to a different channel of the AWG, as follows: Ch 03, $\lambda = 1534.09$ nm, L_1; Ch 09, $\lambda = 1539.8$ nm, L_2; Ch 15, $\lambda = 1544.0$ nm, L_3; Ch 22, $\lambda = 1550.0$ nm, L_4. For these results, the measurement time for each of the values of τ_p was set to ten seconds, and the number of total steps necessary in order to cover all the points in the fiber was $\frac{\tau_b}{\tau_d} = \frac{2\mu s}{20\,\mathrm{ns}} = 100$, totalizing a measurement time of ~16.7 minutes. Interrogation of each individual channel is possible by tuning the F8-PMLL center wavelength accordingly, and the results are presented in Fig. 8.

Under these conditions, the amount of detection events are still not enough to reach the full dynamic range of the system, as previously commented, and as clearly depicted in the traces of Fig. 8. Even under such short measurement time conditions,

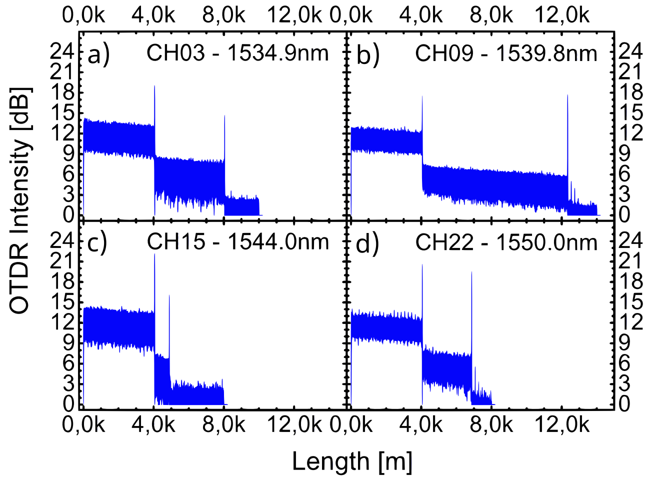


Fig. 8. Full fiber measurements of four distinct AWG channels, corresponding to different wavelengths. The tunability of the F8-PMLL allows for each channel to be probed individually. Total measurement time is ∼16.7 minutes for each of the channels. Detection peaks observed at the end of the fiber profiles correspond to afterpulsing due to the high reflection peak at the fiber end.

however, the system was able to achieve a quite expressive ∼12 dB dynamic range. Furthermore, the system showcases able centimeter-resolution measurements of km-range fibers ıinute-range times, an outstanding result in comparison with e-of-the-art systems. Due to its performance, the system is ıbed the Fast Acquisition High-Resolution Photon Counting DR.

Acquisition Rate and Afterpulse Probability

t is noteworthy that the burst period τ_b determines the ed-up in data acquisition of the proposed system, since a ʒle optical probing pulse sent into the fiber can be used to ɪrrogate multiple regions. Enabling more detection windows optical pulse, which corresponds to decreasing τ_b, even ugh positive in the sense of higher data acquisition rate, an associated effect related to the afterpulse probability of employed gated avalanche single-photon detectors. As the effective detector *deadtime* between two detection windows is reduced, the probability of a delayed release of a trapped carrier inside the semiconductor junction increases exponentialy [22].

For regions of the fiber where the counts associated to Rayleigh backscattered photons is above the noise (coming either from the intrinsic dark count rate of the detector of from the afterpulsing probability), the effect is not apparent; however, when one analyzes the detections outside of the fiber, where only the detector's intrinsic noise (dark count rate) should play a role, the effect becomes clear, as in positions $L \geq 8.2$ km of Fig. 7. In fact, if the separation between detection windows is too small, one can extrapolate the impact of the afterpulsing effect and expect that it completely degrades the measured OTDR profile, erasing all relevant information about the fiber.

Analyzing the impact of the afterpulse probability on the detections permits finding a set of parameters compatible with the sought after higher data acquisition rates while still minimizing as much as possible its contribution. Experimental results are presented in Fig. 9, where the difference, in dB, between the intrinsic noise level of the system, associated to the detector's dark count rate, and the noise level when the afterpulse is present, has been used as a figure of merit when the detection window length τ_d and the burst period τ_b are varied. In order to provide more insight into the results, the horizontal axis is presented in both μs and in meters, i.e., the associated distance separation, in the fiber, between two consecutive detection windows.

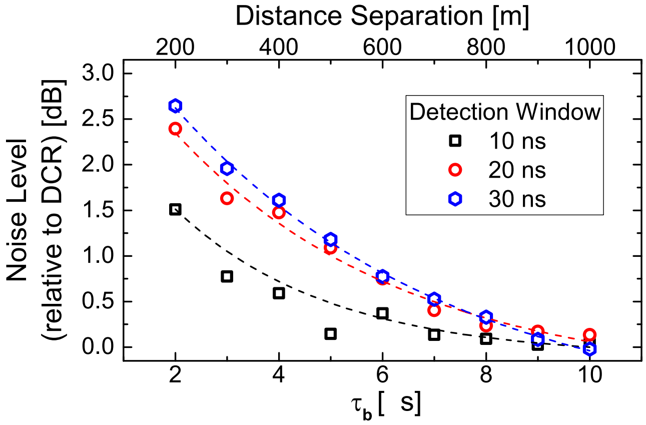


Fig. 9. Noise level, in dB, relative to the SPD's intrinsic dark count rate as the separation between detection windows and the width of such windows is varied. The exponential decay is expected from the behavior of the afterpulse probability as a function of *deadtime* [23].

As can be clearly seen from Fig. 9, the afterpulse probability dominates the noise baseline level unless the separation between consecutive detection windows is longer than 1 km. Simultaneously, the number of necessary windows $\frac{\tau_b}{\tau_d}$ for complete fiber measurements increases by a factor of 5, under these conditions, with respect to the measurements of Fig. 8. Therefore, and since the impact of the afterpulse is not as critical on the noise level as the impact of increasing τ_b is on the timing, the parameters used in the measurement could be loosely selected as $\tau_b = 2$ μs and $\tau_d = 20$ ns.

B. Discussion

It is clear that one of the major benefits of the proposed system is the capability of monitoring wavelength multiplexed networks with telecom-compatible wavelengths, which is available due to the tunability of the F8-PMLL. However, short-distance high-capacity optical networks operating at the near-infrared region of the spectrum are attracting a lot of attention [24], and high-resolution monitoring is of great interest in this context. In fact, a photon-counting OTDR system operating at this wavelength and making use of a gain-switched laser was recently demonstrated [25]. There, the acquisition rate is still limited since, for each probing pulse launched into the fiber, only a single detection window is enabled. Fortunately, the herewith proposed acquisition system can be readily adapted to work in that and any other photon-counting OTDR system with equal or even higher gains, since the silicon-based single-photon detectors compatible with that system exhibit lower dark count rates, higher detection efficiency, and smaller afterpulse probability.

Systems such as the one in [26], where a superconducting nanowire single-photon detector compatible with the telecom wavelength was employed in a PC-OTDR, or the one in [27], where a 30-dB dynamic range was achieved but with a total monitoring time of 6 hours, would likewise benefit from the proposed acquisition system.

It is noteworthy that the solution found in [9] to achieve cm-resolution fault detection at long-distances would also benefit from the high-rate acquisition system. There, a *coarse* and long-reach PC-OTDR system first acquires information about the fiber profile, which is followed by a signal processing routine that identifies possible candidates that are, then, individually interrogated by a *fine* PC-OTDR system. The limitation on the acquisition system, however, causes the total monitoring time to scale with the number of identified candidates, which is overcome with the current acquisition system, since multiple fault candidates can be interrogated simultaneously.

On one hand, thus, the acquisition system proposed and demonstrated in this work enables the high-resolution monitoring of long-distance fibers by greatly increasing the rate of data acquisition. On the other, the much higher event detection rate on the SPAD uncovers an application bottleneck associated to the maximum data capacity of the employed TDC. The fact ıt the current TDC is limited to 180 thousand samples per ɔond caused the detection efficiency to be reduced to 5% order to avoid loss of data; for higher values of detection iciency the detection rate overloads the TDC. Employing TDC with higher data capacity will allow for even higher ta acquisition gains and could usher the replacement of the rrent avalanche single-photon detectors by superconducting nowire single-photon detectors (SNSPD) that offer not only gher detection efficiency but also free-running operation.

The gain in speed over other centimeter-resolution PC-OTDR lutions is determined by the number of extra detection winws that the proposed system is capable of offering. Since this mber will change if the fiber length increases and can also leveraged against an increased contribution of the afterpulse ect (as shown in Fig. 9), this number is hardly determined. For measurement conditions showcased in Fig. 7, however, 20 detection windows (within the fiber span) are available, which translates into a 20-fold factor in acquisition time in comparison with state-of-the-art solutions where only a single detection window is opened per optical pulse sent into the fiber [9], [25]. For the results of Figure 8, because the length of the fiber changes, the timing gain for the full-fiber measurement would vary between 10 and 20 using the current (limiting) devices and a reduced efficiency of the SPD. State-of-the-art (Superconducting Nanowire) Single-Photon Detectors offer absence of afterpulsing, higher than 90% detection efficiency, and as high as 100 ns deadtimes with 100 ps jitter. Allied to high-speed TDC modules, the system could perform the measurements showcased in Figs. 7 and 8 in under a minute with higher resolution (due to the reduced jitter) and higher dynamic range (due to the absence of afterpulsing).

When compared to other monitoring solutions, which do not rely on photon-counting for fiber monitoring, the Fast Acquisition Tunable PC-OTDR exhibits an unique combination of characteristics that make it competitive with respect to the state-of-the-art. For instance, a chaos-OTDR able to achieve similar (centimeter) resolution in long distance measurements has been demonstrated [28], with monitoring times as low as 2 ms. However, the technique is limited to identifying reflective events in a fiber. It is important to highlight that, even though used in a different context in [28], filtered broadband sources are not indicated for the measurements presented in Fig. 7 and 8. The fact that the demonstrated spatial resolution can be maintained for long-distance measurements is only possible due to the tunable F8-PMLL, which creates short-pulses with a relatively narrow spectral bandwidth. This way, chromatic dispersion does not severely limit the spatial resolution as more distant positions of the fiber are probed. In e.g. [29], a filtered broad band source was used to perform centimeter resolution measurements and a study of the impact of the chromatic dispersion (as the spectral bandwidth of the filter was varied) in the achievable spatial resolution was performed.

IV. Conclusion

By combining the time resolution of a TDC with the management capabilities of an FPGA, a detection system could be developed that allows for high resolution measurements with high data acquisition rates. This structure, when employed together with an optical pulse generation system that allows for wavelength tunability and high-peak-power narrow pulses culminates into the fast acquisition tunable high-resolution photon-counting OTDR, with an achievable spatial resolution of 4.6 cm and 15 dB dynamic range. The current bottleneck of the system is the throughput of the TDC, that handles a limited amount of detection events. Improving the TDC's data handling capacity, narrowing the pulse width of the F8-PMLL, and migrating the detection system in order to make use of superconducting nanowire single-photon detectors are the main future points of investigation, which have the potential to lead to sub-centimeter spatial resolution with an even higher achievable dynamic range. The realization of a system where a single optical pulse can be used to monitor distinct positions of a fiber in a high-resolution photon-counting OTDR enables dramatic gains in acquisition time and, in turn, full long-distance fiber measurements with centimeter resolution.

References

[1] S. Kumar and M. J. Deen, *Fiber Optic Communications: Fundamentals and Applications*. Hoboken, NJ, USA: Wiley, 2014.

[2] P. J. Urban, G. Vall-Llosera, E. Medeiros, and S. Dahlfort, "Fiber plant manager: An OTDR-and OTM-based PON monitoring system," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. S9–S15, Feb. 2013.

[3] K. O. Hill and G. Meltz, "Fiber bragg grating technology fundamentals and overview," *J. Lightw. Technol.*, vol. 15, no. 8, pp. 1263–1276, 1997.

[4] M. Barnoski, M. Rourke, S. Jensen, and R. Melville, "Optical time domain reflectometer," *Appl. Opt.*, vol. 16, no. 9, pp. 2375–2379, 1977.

[5] G. P. Agrawal, *Fiber-optic communication systems*, vol. 222, Hoboken, NJ, USA: Wiley, 2012.

[6] G. C. Amaral, L. E. Herrera, D. Vitoreti, G. P. Temporão, P. J. Urban, and J. P. von der Weid, "Wdm-PON monitoring with tunable photon counting OTDR," *IEEE Photon. Technol. Lett.*, vol. 26, no. 13, pp. 1279–1282, Jul. 2014.

[7] L. Herrera, F. Calliari, J. Garcia, G. Amaral, and J. von der Weid, "High resolution automatic fault detection in a fiber optic link via photon counting otdr," in *Proc. Opt. Fiber Commun. Conf.*, 2016, Paper M3F–4.

[8] G. C. Amaral, J. D. Garcia, L. E. Herrera, G. P. Temporao, P. J. Urban, and J. P. von der Weid, “Automatic fault detection in wdm-pon with tunable photon counting OTDR,” *J. Lightw. Technol.*, vol. 33, no. 24, pp. 5025–5031, 2015.

[9] F. Calliari, L. E. Herrera, J. P. von der Weid, and G. C. Amaral, “High-dynamic and high-resolution automatic photon counting OTDR for optical fiber network monitoring,” in *Proc. 6th Int. Conf. Photon., Opt. Laser Technol.*, 2018, vol. 1, pp. 82–90.

[10] M. Wegmuller, J. P. Von Der Weid, P. Oberson, and N. Gisin, “High resolution fiber distributed measurements with coherent ofdr,” in *Proc. ECOC 2000*, Munich, Germany, vol. 11, no. 4, 2000, p. 109.

[11] J. P. von der Weid, M. H. Souto, J. D. Garcia, and G. C. Amaral, “Adaptive filter for automatic identification of multiple faults in a noisy otdr profile,” *J. Lightw. Technol.*, vol. 34, no. 14, pp. 3418–3424, 2016.

[12] M. Lunglmayr and G. C. Amaral, “Linearized bregman iterations for automatic optical fiber fault analysis,” *IEEE Trans. Instrum. Meas.*, vol. 68, no. 10, pp. 3699–3711, Oct. 2018.

[13] M. Salhi, F. Amrani, H. Leblond, and F. Sanchez, “Analytical investigation of a figure-eight single-pulse all-fiber laser based on a nonlinear amplifying loop mirror,” *Physical Rev. A*, vol. 82, no. 4, 2010, Art. no. 043834.

[14] H.-G. Weber and M. Nakazawa, *Ultrahigh-Speed Optical Transmission Technology*, vol. 3, Berlin, Germany: Springer Science & Business Media, 2007.

[15] A. Komarov, H. Leblond, and F. Sanchez, “Passive harmonic mode-locking in a fiber laser with nonlinear polarization rotation,” *Opt. Commun.*, vol. 267, no. 1, pp. 162–169, 2006.

[16] Y. Chen *et al.*, “Dispersion-managed mode locking,” *JOSA B*, vol. 16, no. 11, pp. 1999–2004, Nov. 1999.

[17] M. E. Fermann, F. Haberl, M. Hofer, and H. Hochreiter, “Nonlinear amplifying loop mirror,” *Opt. Lett.*, vol. 15, no. 13, pp. 752–754, 1990.

[18] I. N. Duling, “All-fiber ring soliton laser mode locked with a nonlinear mirror,” *Opt. lett.*, vol. 16, no. 8, pp. 539–541, 1991.

[19] M. Kues *et al.*, “Passively mode-locked laser with an ultra-narrow spectral width,” *Nature Photon.*, vol. 11, no. 3, p. 159, 2017.

[20] H.-R. Chen *et al.*, “12 ghz passive harmonic mode-locking in a 1.06 μ m semiconductor optical amplifier-based fiber laser with figure-eight cavity configuration,” *Opt. Lett.*, vol. 38, no. 6, pp. 845–847, 2013.

[21] D. Derickson, C. Hentschel, and J. Vobis, *Fiber optic test and measurement*, vol. 8. Englewood Cliffs, NJ, USA: Prentice Hall, 1998.

[22] S. Cova, M. Ghioni, A. Lotito, I. Rech, and F. Zappa, “Evolution and prospects for single-photon avalanche diodes and quenching circuits,” *J Modern Opt.*, vol. 51, no. 9-10, pp. 1267–1288, 2004.

[23] S. Cova, A. Lacaita, and G. Ripamonti, “Trapping phenomena in avalanche photodiodes on nanosecond scale,” *IEEE Electron Device Lett.*, vol. 12, no. 12, pp. 685–687, Dec. 1991.

[24] D. A. Miller, “Optical interconnects to silicon,” *IEEE J. Sel. Topics Quantum Electron.*, vol. 6, no. 6, pp. 1312–1317, Nov./Dec. 2000.

[25] B. Li *et al.*, “850nm gain-switched pulse laser and its application in photon counting OTDR,” in *Proc. Opt. Metrology Inspection Ind. Appl. V*, 2018, vol. 10819, Art. no. 1081915.

[26] J. Hu *et al.*, “Photon-counting optical time-domain reflectometry using a superconducting nanowire single-photon detector,” *J. Lightw. Technol.*, vol. 30, no. 16, pp. 2583–2588, 2012.

[27] P. Eraerds, M. Legré, J. Zhang, H. Zbinden, and N. Gisin, “Photon counting OTDR: Advantages and limitations,” *J. Lightw. Technol.*, vol. 28, no. 6, pp. 952–964, 2010.

[28] Z. Wang *et al.*, “Long-range and high-precision correlation optical time-domain reflectometry utilizing an all-fiber chaotic source,” *Opt. Express*, vol. 23, no. 12, pp. 15 514–15 520, 2015.

[29] L. E. Ynoquio Herrera, G. C. Amaral, and J. P. von der Weid, “Remote fiber bragg grating-based sensor characterization with ultra-high-resolution tunable photon counting OTDR,” in *Proc. SBMO/IEEE MTT-S Int. Microw. Optoelectronics Conf.*, 2017, pp. 1–4.

C
FPGA-Embedded Linearized Bregman Iterations Algorithm for Trend Break Detection

The paper "*FPGA-Embedded Linearized Bregman Iterations Algorithm for Trend Break Detection*" was accepted for publication in EURASIP Journal on Wireless Communications and Networking in 2020.

RESEARCH

FPGA-Embedded Linearized Bregman Iterations Algorithm for Trend Break Detection

Felipe Calliari[1*], Gustavo C. Amaral[1,2] and Michael Lunglmayr[3]

Abstract

Detection of level shifts in a noisy signal, or trend break detection, is a problem that appears in several research fields, from biophysics to optics and economics. Although many algorithms have been developed to deal with such a problem, accurate and low-complexity trend break detection is still an active topic of research. The Linearized Bregman Iterations have been recently presented as a low-complexity and computationally-efficient algorithm to tackle this problem, with a formidable structure that could benefit immensely from hardware implementation. In this work, a hardware architecture of the Linearized Bregman Iterations algorithm is presented and tested on a Field Programmable Gate Array (FPGA). The hardware is synthesized in different sized FPGAs and the percentage of used hardware, as well as the maximum frequency enabled by the design, indicate that an approximately 100 gain factor in processing time, concerning the software implementation, can be achieved. This represents a tremendous advantage in using a dedicated unit for trend break detection applications. The proposed architecture is compared with a state-of-the-art hardware structure for sparse estimation and the results indicate that its performance concerning trend break detection is much more pronounced while, at the same time, being the indicated solution for long datasets.

Keywords: Linearized Bregman Iterations; Trend Break Detection; FPGA

1 Introduction

Trend break detection in the presence of noise is a broad problem that can be found across different research fields [1–4]. For that reason, several different methodologies have been proposed in the literature [5–7], with the ones that make use of ℓ_1 regularization to counter the problem's inherent high-dimensionality arguably figuring as the most successful ones [8, 9]. Such an approach is required for highly reliable estimation results [7]. Even though such regularization allows the problem to be solved in a computationally efficient manner (usually associated to a complexity which is proportional to a polynomial function of the number of inputs), the fact that a computer can solve the problem does not necessarily mean that the result is achieved quickly, practically speaking. In certain contexts, achieving elapsed algorithm times in the order of seconds as opposed to minutes may yield a substantial impact on the application [10].

It is a widespread notion that certain problems, despite their complexity, may be accelerated depending on the implementation; parallel programming, in which several parts of the same procedure are processed independently and simultaneously, is one of the most celebrated examples [11]. Field Programmable Gate Arrays (FPGAs) are extremely versatile hardware structures that offer [12–14]: great flexibility to design high speed high-density digital hardware; easiness of programability and reconfiguration; energy efficiency; high resource utilization; low cost; and the possibility to combine parallel processing structures with serial control units. FPGAs have been used as a versatile computing platform accelerating algorithms through dedicated and carefully designed architectures in a wide range of fields [15] such as cryptography [16], image processing [17], and machine learning [18].

1.1 Motivation and Contributions

Recently, Linearized Bregman Iterations (LBI), a class of implementation-efficient and low-complexity algorithms, has been presented as an extremely attractive solution for trend break detection [1]. There, it was shown that the LBI algorithm outperforms the classical LASSO solution in the specific problem of trend break detection for fiber fault analysis. In this case, the better performance can be attributed to two factors.

[*] Correspondence: felipe.calliari@opto.cetuc.puc-rio.br
[1] Center for Telecommunications Studies, Pontifical Catholic University of Rio de Janeiro, Marquês de São Vicente, 22451-900, Rio de Janeiro, BR
Full list of author information is available at the end of the article

First it is well known that combined L1/L2 regularization terms can handle problems where estimation vectors with correlated elements are expected better than the LASSO solution [19] and the problem at hand is expected to have such correlations. Second, the strict convexity when using a combined L1/L2 term in the cost function has been reported to improve the convergence behavior of sparse estimation algorithms [20]. Furthermore, both the structure of the trend break detection problem and of the LBI algorithm's allow for simple hardware units, relying mainly on adders and efficient memory management, to conduct the core procedure, thereby avoiding hardware-complex multiplication and division operations [21]. In [1], the focus is on detection of trend breaks associated to fiber faults: to ensure robustness of network operation, detection of such faults must be remedied as fast as possible; this, in turn, creates a demand for highly reliable and fast trend break detection results, since a network operator might have several thousand optical fibers under his responsibility [22,23] and, thus, the faster data associated with one link can be processed, the faster information about all links will be available.

The possibility of increasing the time efficiency of the algorithm due to its hardware implementation is of great interest in this context, as pictorially presented in Fig. 1-a. Furthermore, as the measurement is often done by stand-alone Optical Time-Domain Reflectometry (OTDR) devices, the eventual goal is to implement the processing directly in such a device, where FPGAs are usually employed. Achieving an FPGA implementation of the detection algorithm is, then, preferable, since it would allow data acquisition and processing to be performed in the same embedded device, as pictorially presented in Fig. 1-b.

In this work, the hardware implementation of the LBI algorithm is studied in depth and is simulated and synthesized for different FPGAs. A novel hardware architecture is presented and its main processing units are discussed. VHDL simulation environments enable a step-by-step comparison and validation of the processing stages referenced by the computer algorithm implementation [1]. Hardware synthesis results allow determination of both device usage with different FPGA sizes and maximum clock frequency; the latter, combined with the average number of clock cycles per iteration loop, make total processing time calculation possible for different problem instance sizes. A reduction factor on the elapsed algorithm time of approximately 100 is achieved, which represents a substantial upgrade and warrants usage of dedicated hardware for trend break detection. The main overall contributions of this work are discriminated below:

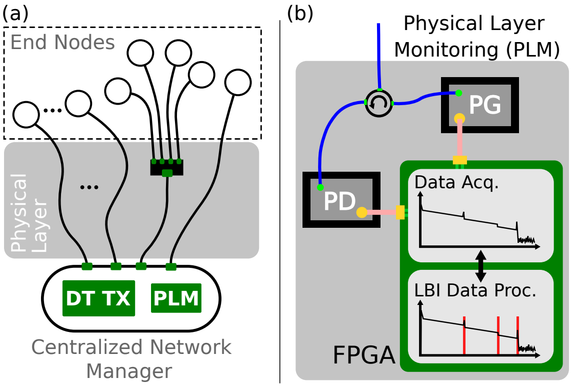


Figure 1 Motivation and applications of the proposed hardware-implementation of the Linearized Bregman Iteration algorithm. Depicted in (a) is the scenario where a centralized optical fiber network manager must deal with several links simultaneously and, being able to expedite the measurement is beneficial; (DT TX): data transmission unit; (PLM): physical layer monitoring unit. Depicted in (b) is the combination of a data acquisition hardware for the PLM (adapted from [10]) along with the data processing hardware implementation of the LBI algorithm. The optical pulse generation (PG) unit launches a probing pulse into the fiber, while the data acquisition system, together with an optical circulator and an optical pulse detection (PD), acquires the reflected optical power and produce a fiber profile.

- Speed-ups of the order of 100 times in the hardware implementation with respect to the software implementation of the LBI algorithm for trend break detection. The speed-ups are solely attributed to the hardware architecture and the usage of parallel-pipelined arithmetic units (such as the so-called pipelined adder tree) and to efficient memory organization and control, since no loss of performance is identified between hardware and software implementations.
- A comprehensive analysis of the hardware architecture, including the main processing and control units that allow for the algorithm structure to be implemented in hardware. This analysis provides a straightforward means to determining the total number of clock cycles necessary for the algorithm to elapse.
- A software implementation of the algorithm, that has been validated bit-wise with respect to the hardware implementation, thereby allowing for the results of the proposed implementation to be verified in a simulated environment.
- Comparison, with respect to both performance and hardware characteristics, of the proposed hardware implementation with state-of-the-art hardware implementations of the Orthogonal Matching Pursuit algorithm, which is used in the context of sparse

estimation and finds widespread interest in the current literature.

The paper is divided as follows. In Section 3, a brief review of the LBI algorithm for trend break detection is performed, including the structure of the candidate matrix and the pseudocode based on which the hardware architecture is developed. Section 4 presents the digital hardware architectural concept as well as focused descriptions of its main units; the estimated number of clock cycles until the algorithm elapses is derived based on this architecture. In Section 5, comparative results between the simulated hardware implementation and its Julia-based software counterpart are discussed. Synthesis parameters for two target FPGAs (Altera Cyclone V and Altera Stratix V) are also reported. Case studies (both for a real-world fiber profile and simulated data series) are discussed in Section 6.1. Section 6.2 is dedicated to the comparative analysis between the proposed architecture and a state-of-the-art algorithm in the context of trend break detection. Finally, Section 6.2 concludes the paper.

2 Methods

The aim of this work was to develop and evaluate a hardware architecture to accelerate sparse estimation for reliably solving the trend break detection problem. To evaluate the capabilities of the proposed architecture the following methods have been used:

- A commercially available state-of-the-art synthesis software (INTEL QUARTUS PRIME) was used to evaluate the maximum clock frequency and the device occupation of the design for different sizes (i.e. the number of used block RAMs in parallel).
- An analytical description was developed and used (validated by hardware simulation) to calculate the number of clock cycles necessary to perform the estimation tasks.
- A bit-true simulation environment was developed and, after bit-true validation against a commercially available hardware simulation tool (Mentor Graphics Modelsim), it was used to evaluate the performance of the proposed architecture in large-scale simulation studies.
- The bit-true simulation results of the hardware implementation are compared to double precision floating-point results in terms of their averaged squared error norms.
- A real-world data set measured in the lab by an Optical Time-Domain Reflectometry measurement device has been used to evaluate the algorithm, both in double precision floating point as well as in its bit-true quantized version used in hardware.
- The LBI algorithm discussed in this work is compared to the OMP algorithm based on simulation studies and the thereby calculated figures of merit: the precision and the Matthews correlation coefficient.
- The LBI algorithm and the OMP algorithm are compared based on their computational complexity analysis.

3 The Linearized Bregman Iterations Algorithm for Trend Break Detection

Under the assumption that the trend break detection problem is a sparse one, i.e., the number of candidate vectors that describe the signal of interest is much smaller than the number of observations, it can be cast into the combined ℓ_1/ℓ_2 problem of the form [1]:

$$\min_{\boldsymbol{\beta}} \lambda||\boldsymbol{\beta}||_1 + \frac{1}{2}||\boldsymbol{\beta}||_2^2 \;\; s.t. \;\; \mathbf{A}\boldsymbol{\beta} = \mathbf{y}, \tag{1}$$

where $\mathbf{A}$ is the dictionary, with each candidate vector stored in a column, $\boldsymbol{\beta}$ is the vector containing the coefficients of the weighted linear combination of dictionary vectors that will approximate the signal of interest represented by the data series $\mathbf{y}$, and λ is a parameter that adjusts the weight of the ℓ_1 versus the ℓ_2 norms in the cost function. Adaptation of the Linearized Bregman Iterations algorithm to trend break detection has been presented in [1] in a context where a linear trend is also expected in the signal of interest. In order to simplify and generalize the implementation, this linear trend is not considered in the current implementation. Incorporating the linear trend in the proposed architecture is, however, straightforward.

Throughout the manuscript, the length, in data points, of the signal of interest $\mathbf{y}$ will be defined as N, i.e., $\mathbf{y}$ and $\boldsymbol{\beta}$ are N-dimensional vectors and $\mathbf{A}$ is an $N \times N$ matrix. The Linearized Bregman Iterations algorithm has a cyclic structure, involving, in a single iteration, an approximate gradient descent (AGD) followed by a non-linear shrink function of the form: $\text{shrink}(\mathbf{v}_j, \lambda) = \max(|\mathbf{v}_j| - \lambda, 0) \cdot \text{sign}(\mathbf{v}_j)$ [24]. Due to the special structure of the candidate dictionary matrix $\mathbf{A}$ for the trend break detection problem, namely:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 1 & 0 & \cdots & 0 & 0 \\ 1 & 1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & 1 & \cdots & 1 & 1 \end{bmatrix}, \tag{2}$$

its storage is not necessary for the AGD calculation, as the latter can be rewritten as

$$\begin{aligned}\mathbf{v}^{(i+1)} &= \mathbf{v}^{(i)} + \frac{\mathbf{a}_k}{||\mathbf{a}_k||_2^2}\left(\mathbf{y}_k - \mathbf{a}_k^{\mathrm{T}}\boldsymbol{\beta}^{(i)}\right) \\ &= \mathbf{v}^{(i)} + \frac{\mathbf{a}_k}{||\mathbf{a}_k||_2^2}\left(\mathbf{y}_k - \sum_{s=1}^{k+1}\beta_s\right)\end{aligned} \tag{3}$$

where the $\mathbf{a}_k$ represent rows of the candidate matrix, the superscripted i represents the iteration index, and the index $k \in [1:N]$ controls the cyclic reuse of rows of $\mathbf{A}$ as the iteration index evolves, i.e., $k = \mathrm{mod}\left((i-1), N\right) + 1$.

The $\mathbf{a}_k$, in turn, have an interesting structure that allows the AGD to be further optimized and the calculation to be performed only for those indices where $\mathbf{a}_{k,j} \neq 0$. In other words (and also considering the fact that $||\mathbf{a}_i||_2^2 = k$),

$$\mathbf{v}_j^{(i+1)} = \begin{cases} \mathbf{v}_j^{(i)} + \frac{1}{k}\left(\mathbf{y}_k - \sum_{s=1}^{k+1}\boldsymbol{\beta}_s\right), & a_{k,j} = 1 \\ \mathbf{v}_j^{(i)}, & a_{k,j} = 0 \end{cases}, \tag{4}$$

which, considering computational implementation, translates into accessing and manipulating only those values of vector $\mathbf{v}^{(k)}$ up to index j. A final observation of the structure of matrix $\mathbf{A}$ (namely, the fact that it is a square matrix) reveals that a single index k is sufficient to control an iteration of the algorithm. The resulting procedure, presented as pseudocode in Algorithm 1, efficiently solves the trend break detection problem with low memory usage.

Algorithm 1 Linearized Bregman Iteration for Trend Break Detection

```
Require: Measurement vector $\mathbf{y}$, $\lambda$, $\boldsymbol{\beta}_{\text{start}}$, $\mathbf{v}_{\text{start}}$, $L$
Ensure: Estimated $\hat{\boldsymbol{\beta}}$
$\boldsymbol{\beta}^{(0)} \leftarrow \boldsymbol{\beta}_{\text{start}}$
$\mathbf{v}^{(0)} \leftarrow \mathbf{v}_{\text{start}}$
$i \leftarrow 1$
while $i < L$ do
    $k \leftarrow \mathrm{mod}\ ((i-1), N) + 1$   ▷ cyclic re-use of rows of $\mathbf{A}$
    $\mu_k \leftarrow \frac{1}{k}$
    $e \leftarrow \left(\mathbf{y}_k - \sum_{s=1}^{k+1}\beta_s^{(i)}\right)$
                                   ▷ instantaneous error with inner product
    $d \leftarrow \mu_k e$
    for $j = 1..k$ do
        $\mathbf{v}_j^{(i+1)} \leftarrow \mathbf{v}_j^{(i)} + d$
        $\boldsymbol{\beta}_j^{(i+1)} \leftarrow \mathrm{shrink}\left(\mathbf{v}_j^{(i+1)}, \lambda\right)$
    end for
    $i \leftarrow i + 1$
end while
```

For the presented algorithm, on average $(N-1)/2$ additions per iteration are required. This can be seen from Algorithm 1: the first iteration requires 1 addition; the second 2; and the last N; culminating in $N \cdot (N+1)/2$ additions for N iterations. It is interesting to note that a single multiplication operation is necessary after the result of the summation is performed, which greatly simplifies the procedure and avoids an overload of complex arithmetic structures. As it has been shown in [1], Linearized Bregman Iterations lead to less complex algorithms than alternative approaches such as the Adaptive LASSO [7]. Using an FPGA, one could speed up the additions thereby speeding up a single iteration, resulting in a speedup of the whole algorithm. Indeed, as is shown here, this solution allows the core algorithm to be processed within a fraction of the time it would take on a high-end server processor.

3.1 Hardware Implementation Considerations of the LBI Algorithm

3.1.1 The Ordinary Least Squares Final Step

It is important to note that Algorithm 1 is an adaptation of the pseudo-code presented in [1], where only the computation-heavy part of the procedure is depicted. Its purpose is to identify the relevant non-zero values of the $\hat{\boldsymbol{\beta}}$ vector that compose the output or, in other words, reduce the dimension of the detection space focusing on the subspace spanned by the relevant candidate vectors. After this procedure, it is usual to perform an Ordinary Least Square (OLS) in this reduced subspace in order to remove any biasing introduced by the algorithm; operating on the reduced subspace found by the LBI drastically reduces the complexity of the OLS. This step, which involves matrices transposing and inverting, can be efficiently conducted in a standard personal computer and, even though this could also be implemented in the same hardware structure that contains the core algorithm [1], the goal of this work is to present the latter and the OLS step is left as a post-processing step to be performed in a different processing unit.

With respect to the OLS post-processing, interesting functionalities of so-called Systems On Chip (SOC), which combine FPGAs with embedded CPUs as, e.g., in the used Cyclone V, can be harnesed for this goal. In fact, they enable the delegation of the reduced subspace OLS problem to the embedded CPU. Measurements have been performed on the SOC contained in the Cyclone V running Linux (Ubuntu 16.04) on its dual-core Arm processor. For typical OLS instance sizes of 15000x20 (a reasonable upper-bound according to [1] for a 15000 samples fiber profile), a run-time of about 0.15 seconds in the Julia language has been obtained. It is noteworthy that, in contrast, running the LBI core algorithm (typically requiring 100-200 seconds on a much more powerful XEON Server processor) is practically not feasible on such an embedded CPU. However, as the measurement results of the OLS problem show, and due to its negligible time overhead, delegating the OLS post-processing step to an embedded CPU is indeed feasible. Once again, even though the ordinary Kaczmarz algorithm approximately solves the OLS problem and can re-use blocks from the Sparse Kaczmarz, on which the LBI algorithm is based, thereby allowing the implementation of both in the same FPGA chip, this adaptation would require several design add-ons, which are not the focus of this research.

In summary, the reason to presently delegate the OLS step to a CPU is threefold: first, a CPU can easily deal with the problem in the reduced subspace identified by the core procedure of the LBI; second, it allows this work to focus on the core structure of the LBI algorithm; and, finally, since an embedded CPU might be already available (as is the case of the CYCLONE V studied here) performing the OLS on the CPU of a SOC might be the more practical approach.

3.1.2 Scaling and Arithmetic Dynamic Range

Also left as a pre-processing step is the scaling of the data vector $\mathbf{y}$, which is necessary to ensure the correct behaviour of step 7 in Algorithm 1 when using the 20-bit fixed-point format – this seemly arbitrary value will be clarified in Section 6.1 utilizing limited arithmetic dynamic range. In other words, one must make sure that no overflows of the arithmetic dynamic range is observed when performing the summation of $\boldsymbol{\beta}$ values. The scaling is intimately connected to the available arithmetical dynamic range, which, in turn, is connected to the memory resources of the FPGA board, thereby constituting a design-related compromise relationship. In case of over-scaling, the arithmetic dynamic range will be hindered; to overcome this, a higher number of bits can be assigned to the data points, which then increases the resources necessary in the FPGA board. On the other hand, in case of under-scaling, the results may overflow, creating errors that can jeopardize the algorithm's convergence. A scaling factor consistent with the algorithm's convergence can be determined according to the following considerations.

The major source for overflows is the sum calculation of $\boldsymbol{\beta}$ values in line 7 of Algorithm 1. Empirical tests conducted based on the testbench developed in [1] indicated that a scaling based on dividing the data vector $\mathbf{y}$ by its maximum value allowed to obtain the results shown in Sect. 5 without harming overflow effects. This is due to the firmly non-expansive property [25] of the shrink function as well as the negative feedback of the error between $\sum_{s=1}^{k+1} \boldsymbol{\beta}_s^{(i)}$ and y_k (line 7 of Algorithm 1). To clarify the negative feedback effect, one could multiply both sides of line 7 by -1, i.e., $-e = \mathbf{y}_k - \sum_{s=1}^{k+1} \boldsymbol{\beta}_s^{(i)}$. This procedure would require μ_k, in line 10, to also be multiplied by -1.

The algorithm can, then, be interpreted as a stabilizing loop on the values of $\mathbf{v}_j^{(i)}$ with the mentioned negative feedback on the deviation between the sum of $\boldsymbol{\beta}$ values (functions of $\mathbf{v}_j^{(i)}$) and the corresponding $\mathbf{y}_k$, which causes overshoots of the sum of $\boldsymbol{\beta}$ values to be immediately corrected in the next iterations. This leads to the fact that, even in worst case scenarios (multiple up and down trend breaks in the measurements), the sum of $\boldsymbol{\beta}$ values scarcely goes above unit, considering the above-mentioned normalization procedure. Moreover, even though its ratio of occurrence is negligible, in the case an overshoot occurs, the excess value would be small thereby not compromising the convergence of the algorithm, as the performance of the quantized version in Sect. 5 demonstrates. The closeness of these results to the ones obtained using double precision floating point shows that, practically, harming effects due to overflow can be neglected.

4 FPGA Architecture

An inherently iterative algorithm and, thus, sequential in nature, the LBI cannot have its iterations parallelized. The operations within an iteration, on the other hand, could greatly benefit from parallelization. The challenge is, thus, to design an architecture that allows high parallelism but that still keeps the effort for control logic, as well as the datapath, manageable. Parallelism in digital hardware not only benefits from parallel calculation units but also, and sometimes more crucial, from an architecture that efficiently feeds the required data to the calculation units in a parallel fashion. Although today's FPGAs typically provide a considerable number of block RAMs (BRAMs), these are implemented in such a way that the number of BRAMs is reduced in exchange for extensive individual memory depth as it is beneficial for many applications.

For the current architecture, this represents a tradeoff: while the values of $\boldsymbol{\beta}$ and $\mathbf{v}$ benefit from parallel access, the values of y are preferably accessed in sequential order and, at the same time, using dedicated BRAMs for $\mathbf{y}$ would limit the scalability and flexibility of the algorithm. For this reason, a combined parallel adder tree and parallel multiplexer tree architecture have been implemented for convenient parallel access to the estimation variables ($\boldsymbol{\beta}$ and $\mathbf{v}$) involved in the core calculations of the algorithm as well as efficient data routing of the values of $\mathbf{y}$.

As already mentioned, even though the iterative nature of the Linearized Bregman Iterations algorithm does not allow for parallelization over the iterations, two core operations that permit parallel pipelining can be identified within a single iteration, as presented in

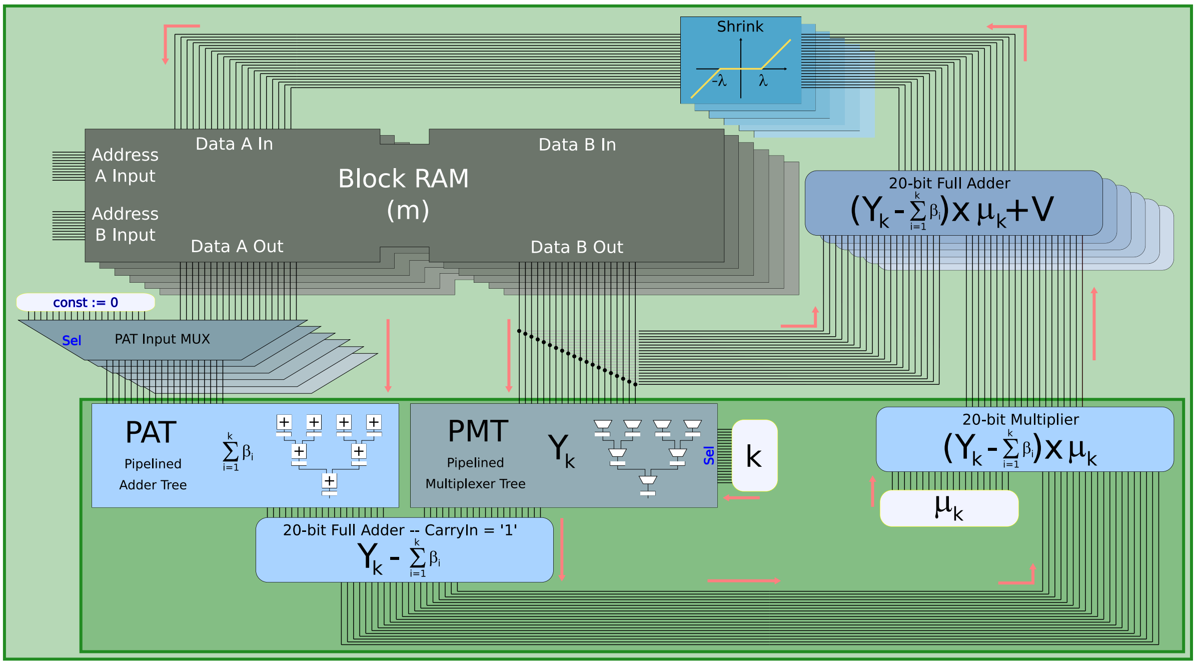


Figure 2 Hardware implementation of a single BRAM slice in the LBI core structure. The parallel structures are pictorially depicted in three-dimensional depth. The Pipelined Multiplexer Tree (PMT) is synchronized to the PAT such that, after a summation, the correct value of $\mathbf{y}$ is ready for subtraction. The value of μ_k – refer to Algorithm 1 – is calculated in a pipelined CORDIC structure.

Algorithm 1: the summation of k entries of the vector $\boldsymbol{\beta}$; and the processing (including update, shrinkage, and storage) of vectors $\mathbf{v}$ and $\boldsymbol{\beta}$. By instantiating parallel memory structures, both operations, that represent computational bottlenecks of the algorithm's iterations, can be optimized. On one hand, the summation can be efficiently performed in a so-called *parallel adder tree* (PAT) (logarithmic number of time steps) given that the data can be accessed in parallel. On the other, parallel processing of the data in vectors $\mathbf{v}$ and $\boldsymbol{\beta}$ can also be accelerated if load/storage can be performed in parallel. Since the algorithm relies on the computation of several iterations to converge, optimizing these two procedures allows for substantial gains in processing time.

4.1 Memory Structure

In order to harness the parallel speedup of the PAT, the entries of vector $\boldsymbol{\beta}$ must also be accessed in parallel, which can be accomplished through the instantiation of parallel Block RAMs (BRAMs). The data storage is structured as follows:

$$\begin{array}{cc} & \begin{array}{cccc} \text{BRAM}(1) & \text{BRAM}(2) & & \text{BRAM}(M) \end{array} \\ t\Big\uparrow & \begin{bmatrix} \boldsymbol{\beta}[TM+1] & \boldsymbol{\beta}[TM+2] & \cdots & \boldsymbol{\beta}[TM+M] \\ \vdots & \vdots & \cdots & \vdots \\ \boldsymbol{\beta}[2M+1] & \boldsymbol{\beta}[2M+2] & \cdots & \boldsymbol{\beta}[2M+M] \\ \boldsymbol{\beta}[M+1] & \boldsymbol{\beta}[M+2] & \cdots & \boldsymbol{\beta}[M+M] \\ \boldsymbol{\beta}[1] & \boldsymbol{\beta}[2] & \cdots & \boldsymbol{\beta}[M] \end{bmatrix}, \\ & \xrightarrow[m]{} \end{array} \qquad (5)$$

where M is the number of parallel BRAMs available in the FPGA. In such a structure, a single arbitrary BRAM, say m, will contain the entries:

$$[m + tM] \forall t \in [0; T], m \in [1; M] : T = \left\lceil \frac{N}{M} \right\rceil,$$

where the ceiling operator is denoted by $\lceil \cdot \rceil$.

Vector $\boldsymbol{\beta}$, however, is not the only vector stored throughout processing: vectors $\mathbf{y}$ and $\mathbf{v}$ are also necessary. Since all these contain the same number N of entries, the data is sectioned such that the address depth of each BRAM is divided into three slices with address pointers (ap) associated with $\boldsymbol{\beta}$ ($\boldsymbol{\beta}_{\text{ap}}$), $\mathbf{y}$ ($\mathbf{y}_{\text{ap}}$), and $\mathbf{v}$ ($\mathbf{v}_{\text{ap}}$); $\boldsymbol{\beta}_{\text{ap}}$ is arbitrarily set to zero. Under this rationale, entries of vectors $\mathbf{v}$ and $\mathbf{y}$ would appear at addresses $t+\mathbf{v}_{\mathrm{ap}}$ and $t+\mathbf{y}_{\mathrm{ap}}$, respectively, even though, for simplicity, only entries of vector $\boldsymbol{\beta}$ are shown in Eq. 5. Using this data storage structure, all positions $[tM+1 : tM+M]$ of either vectors can be accessed from parallel BRAMs within a clock cycle; such data segment will henceforth be referred to as a *parallel row*, with t the *parallel row* pointer following its definition in Eq. 5. A block diagram of the digital hardware architecture depicting a single BRAM and including the major structures of the LBI algorithm hardware implementation is presented in detail in Fig. 2.

Apart from the PAT, a Pipelined Multiplexer Tree (PMT) is used to select the specific value of the data series $\mathbf{y}$, namely $\mathbf{y}(k)$, from which the result of the partial summation of $\boldsymbol{\beta}$ is subtracted from (refer to line 7 of Algorithm 1). The architecture of the PMT is such that the number of stages meets that of the PAT, so synchronization between the two outputs is naturally ensured. Furthermore, the selection key that acts on each stage of the PMT is derived from the cyclic iteration index, k.

The value of $\mu_k = \frac{1}{k}$, which involves a computation-heavy division, has been delegated to a pipelined CORDIC structure instead of a Look-Up Table (LUT) since the goal is to delegate the BRAMs for storage of vectors $\boldsymbol{\beta}$, $\mathbf{v}$, and $\mathbf{y}$. The stages of the pipeline are pre-filled before the iterations are started, requiring a number of clock cycles equal to its length for complete fill-up: this number has been chosen to be $\mathrm{F} = 20$, providing an accurate estimation of the results of $1/k$. Moreover, stage propagation is enabled at each new iteration, ensuring that the correct value is always available without limiting the number of clock cycles per iteration. The reason behind choosing the pipelined CORDIC structure instead of a LUT for the $1/k$ calculation is, then, threefold: (i) preventing memory to be reserved for data other than the vectors $\boldsymbol{\beta}$, $\mathbf{v}$, and $\mathbf{y}$; (ii) no negative implications on the maximum clock frequency of the design, as evidenced by the Place and Route results of the full structure – refer to Section 5; (iii) no negative influence on the number of clock cycles for each iteration since the initial F clock cycles are not a part of the iteration but, rather, of the initialization step.

Based on this memory structure, the amount of clock cycles necessary to complete the calculation of d (lines 5 to 8 in Algorithm 1) depends both on the number of data points and on the depth of the PAT, which, in turn, depends on the number of instantiated (or available) parallel BRAMs in the hardware structure. For an arbitrary iteration cycle, with cyclic index k, the equation that relates these values to the total number of clock cycles is $C'_{\mathrm{r}} = \left\lceil \frac{k}{M} \right\rceil + \lceil \log_2 M \rceil$, where the subscript refers only to the reading and processing of $\boldsymbol{\beta}$ values up to the output of the PAT. Taking into account also the subsequent subtraction and multiplication steps – refer to Fig. 2 –, each taking one clock cycle, the total number of clock cycles amounts to $C_{\mathrm{r}} = \hat{t} + \lceil \log_2 M \rceil + 2$, where $\hat{t} = \left\lceil \frac{k}{M} \right\rceil$ denotes the maximum value of t during an iteration.

4.2 PAT Input Control

Even though a *parallel row* is accessible at each clock cycle due to the parallel instantiation of the BRAMs, clearly not all values in the row will be used during a given iteration with index k. For that reason, a multiplexer (*PAT input MUX* in Fig. 2) is connected immediately after the BRAM output with its remaining input connected to a null value. Due to the additive identity property of zero, the output of the multiplexer can be directed to the PAT without the corruption of the result while accommodating the parallel storage structure.

The selection signal that controls the *PAT input MUX* is derived based on the fact that replacing BRAM outputs by zero is only necessary during the last *parallel row* access, i.e., when $t = \left\lceil \frac{k}{M} \right\rceil = \hat{t}$. Selection is, thus, based on an auxiliary counter that records the aforementioned value and on a so-called unary code (or thermometer code), which encodes the last column index that contributes to the sum. Fig. 3 depicts the control unit responsible for handling the BRAM input address and selection of PAT inputs.

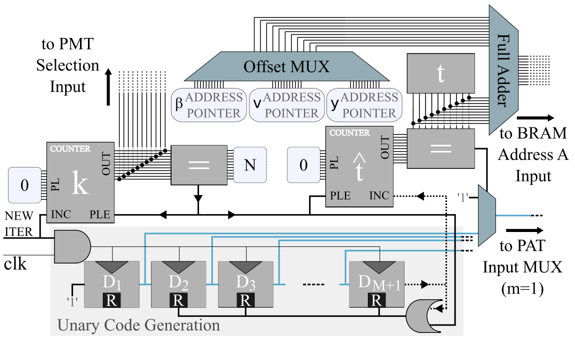


Figure 3 Iteration control architecture. Only the selection for $m = 1$ is depicted for clarity. The Unary Code is used since, at a new iteration k, all the outputs from 1 to k should be made available, i.e., the difference between the iteration k and the iteration $k+1$ is the habilitation of output $k+1$; the Unary Code produces the required output with a minimal footprint.

A NEW ITER strobe generated by the control unit and the clock signal are the necessary inputs. The cyclic iteration index counter k is implemented through a simple counter with parallel load dependent on the comparison with the signal length N. The unary code propagates at each new iteration and, when the $(M+1)^{\text{th}}$ stage is reached, it auto-resets while also incrementing the $\hat{t}$ counter. The unary code acts on *PAT input MUX* when $t = \hat{t}$ and the different address pointers are combined with the counter t to produce the correct BRAM address.

4.3 $\boldsymbol{\beta}$ and $\mathbf{v}$ Storage

An indispensable step of the algorithm is the correct storage of the vectors $\boldsymbol{\beta}$ and $\mathbf{v}$ after processing. According to Algorithm 1, all the elements of vector $\boldsymbol{\beta}$ are processed by the shrink function right after processing of the vector $\mathbf{v}$. As previously pointed out, acceleration of the storage procedure tackles one of the algorithm's bottlenecks. Both the facts that the BRAMs allow for writing and reading from two independently addressed ports and that if λ is set to zero in the shrink function it implements the identity transformation have been harnessed to perform data storage optimization, as it is detailed as follows.

One of the BRAM's ports (taken as B without loss of generality in Fig. 2) is responsible for reading the values of $\mathbf{v}$ from the memory while the other port (A) is responsible for storing the values of $\boldsymbol{\beta}$ and $\mathbf{v}$. The addresses are controlled such that, on the first clock cycle, values of $\mathbf{v}$ in a *parallel row* are read (through port B), processed in the 20-bit full adder, and sent to the shrink function with $\lambda = 0$. Therefore, at the following clock cycle, the stable value of $\mathbf{v}$ can be stored (through port A) at the same time as the value of λ is changed in the shrink function and processes the values of $\mathbf{v}$ being read (through port B). In the third clock cycle, a stable value of $\boldsymbol{\beta}$ is stored (through port A) while the values of $\mathbf{v}$ from the following *parallel row* are accessed (through port B), initiating a new storage cycle for a subsequent *parallel row.*

The net number of clock cycles per *parallel row* storage is, thus, two if one does not compute the very first and last accesses; therefore, the number of clock cycles necessary at an arbitrary iteration with cyclic index k is $C_{\text{s}} = 2\left\lceil \frac{k}{M} \right\rceil + 2 = 2\hat{t} + 2$. Two extra clock cycles are also necessary for the hand-shaking protocol between the iteration control unit (presented in Fig. 3) – whose control over the BRAMs address is releaved – and the writing unit that takes over control and stores vectors β and v, i.e., $C'_{\text{s}} = 2\hat{t} + 4$.

The loading to and from the BRAM (either before starting the algorithm or after the analysis is performed) is performed via a Shift-Register that controls the Memory Enable of each of the BRAMs and routes the addresses and data ports to I/O ports of the entity. This way, data can be streamed and connected directly to all the input ports of the BRAMs while the addresses and Memory Enables ensure the correct loading of the values. Two extra clock cycles are necessary, one at the beginning of the streaming and one in the end, so the Shift-Registers are correctly initialized. Furthermore, to store and later access all three vectors, this streaming procedure must be repeated three times, sequentially, thereby creating the necessity of extra $2(2+3N)$ clock cycles for loading to and from the BRAMs. When compared to the number of clock cycles required by the iterations, however, it becomes clear that this number is negligible and has not been included in the clock cycle analysis at the end of the section. The data transfer hardware described above, which only corresponds to a small device utilization overhead, is included in the synthesis results of the next section.

4.4 Master FSM

In order to control all the aforementioned steps of the design, a so-called master Finite State Machine (FSM) is implemented. The states, transitions, and strobes depicted in Fig. 4 ensure the correct evolution of the algorithm. The FSM starts at an idle state and, based on an *init* strobe, evolves to the loading state, where control of the BRAMs is granted to the streaming structure. Once loading is done, the state evolves to the CORDIC initialization (for $1/k$ calculation) and, after that, to the iteration control state. This state initializes a counter, which saturates at the total value of iterations L; in case the counter value is below L, the state evolves first to the $\boldsymbol{\beta}$ summation (subsequently triggering a NEW ITER strobe—refer to Fig. 2) and, then, to the storage of $\boldsymbol{\beta}$ and $\mathbf{v}$; if, however, the counter value is equal to L, the iterations are done, and unloading can start. Finally, after unloading is performed, the FSM returns to the idle state, where it waits for an upcoming *init* strobe.

4.5 Total Clock Cycle Estimation

After analysis of the PAT processing and the data storage structure, the total number of clock cycles for an iteration can be determined. According to the previous analysis, combining C_s, the number of clock cycles necessary for storage with the determined C_{r}, the number of clock cycles necessary for the partial summation of $\boldsymbol{\beta}$ in the PAT as well as the necessary operations to determine $\mathbf{y}$, the total number of clock cycles spent in an arbitrary iteration with index k is $C_{\text{T}} = 3(\hat{t}+2) + \lceil \log_2 M \rceil$.

The total number of clock cycles taken by the algorithm to elapse can be easily derived from this equation by summation over L, the total number of iterations:

$$C = F + \sum_{i=1}^{L} \left[3\left(\left\lceil \frac{((i-1)\,\%N)+1}{M} \right\rceil + 2 \right) + \lceil \log_2 M \rceil \right].$$

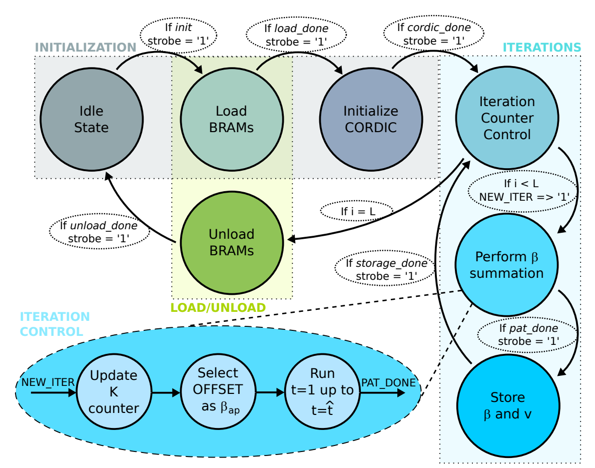


Figure 4 Block diagram of the master Finite State Machine that controls the steps necessary for correct data processing with the core algorithm structure depicted in Fig. 2. A simplified block diagram of the Iteration Control unit of Fig. 3 is also presented.

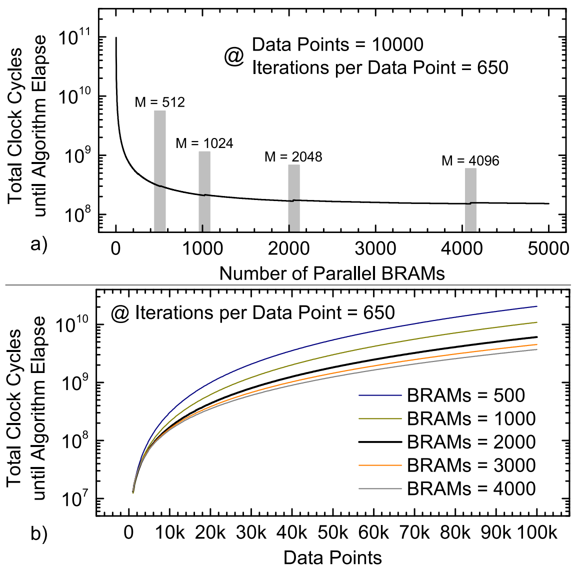


Figure 5 Estimate of total number of clock cycles necessary for the algorithm to elapse considering the presented architecture. (a) Dependence with respect to the number of available parallel BRAMs for a fixed number of 10000 data points and 650 iterations per data point. The gray-shaded areas highlight the transition between powers of 2, which manifests as sharp increases in the calculated value of C. (b) Dependence with respect to the number of data points for a fixed number of available BRAMs and 650 iterations per data point. The highlighted curve corresponds to 2000 BRAMs, which is the maximum available for the largest target FPGA studied here.

(6)

The factor F in Eq. 6 accounts for pre- and post-processing instructions performed by the control unit such as: master resets; granting control over the BRAMs; and, most importantly, preemptively filling up the pipelined CORDIC that calculates μ_k. However, as will be described in the next Section, the value of F is much smaller than the total number of clock cycles taken by the core procedure.

Fig. 5 presents the dependence of the total number of clock cycles until the algorithm elapses with both the number of available parallel BRAMs for a fixed number of data points and with the number of data points for a fixed number of available BRAMs. In both cases, the iterations per data point (defined as L/N) is fixed at 650, a realistic value that will be discussed in further sections. Considering a maximum clock frequency achievable in the target FPGA to be around 100 MHz, a 10000-point data series would be processed in less than two seconds, which represents an approximately 100 gain factor when compared to the Julia implementation reported in [1].

The results of Fig. 5-a provide an interesting analysis point: a stagnation of the contribution of M, as it increases, to the decrease of the number of clock cycles necessary for the algorithm to elapse. As Eq. 6 indicates, the impact of M in the total number of clock cycles is of the form $1/M$. Therefore, as M increases, its contribution to decreasing C becomes relatively marginal. Moreover, it is important to note that there is also a second part in the term inside the summation of Eq. 6, which scales up with M, i.e., $\lceil \log_2 M \rceil$. Therefore, as the $1/M$ term reaches a relative stagnation with higher M, the contribution of the $\lceil \log_2 M \rceil$ term grows, which balances out for high values of M creating the observed stagnation, which is visually pronounced in Fig. 5 due to the logarithmic scale.

5 Validation and Synthesis

Comparison between the software-defined hardware implementation of the Linearized Bregman Iterations algorithm using the architecture presented in the previous Section and its software implementation counterpart [1] permits validating the former. In order to provide a bit-true validation, the *SFIXED* standard used in the VHDL simmulation was implemented in Julia allowing one to accompany, step-by-step, the evolution of the algorithm on both platforms and identify any discrepancies. Due to the fact that the rounding procedure is the same for both, no such discrepancies were observed; the fixed point Julia simulation code outputs *exactly* the same values as of the hardware implementation. The validation of the hardware implementation and the demonstration of its equivalence to the Julia *SFIXED* implementation creates a versatile tool to estimate the performance of the FPGA results on a software environment.

For the simulation of the hardware implementation, the ModelSim VHDL simulation environment was employed. In such an environment, both the evolution of the algorithm as well as the number of clock cycles necessary to run each iteration can be extracted, so the results of Eq. 6 can also be ascertained. Even though an extremely reliable and versatile tool, VHDL simulation offers a drawback in terms of running time: simulating a high number of BRAMs or a large dataset can be extremely time-consuming. For this reason, a predetermined set of parameters (data points, iterations, and number of BRAMs) were chosen to showcase the validity of the hardware implementation.

Table 1 contains the information regarding the simulation of the hardware structure under the different parameter conditions, where B stands for the number of BRAMs, and L and N follow the previously defined notation. The estimated number of clock cycles based on Eq. 6 that appear in Table 1 take into account the required $F = 21$ extra clock cycles for initialization and control, but exclude the up/down loading steps of data into and out of the BRAMs. The asterisk in the last column indicates that 2048 BRAMs is actually above the 2000 maximum available number of BRAMs with 20 bit-wide data entries in the target Altera Stratix V FPGA, but could be implemented in a larger device.

Table 1 VHDL Simulation – Validation

	N = 10 B = 4 L = 10	N = 10 B = 4 L = 100	N = 100 B = 4 L = 100	N = 100 B = 4 L = 1000	N = 1000 B = 4 L = 1000
C (Eq. 6)	155	1361	4721	47021	384521
Clk. Cyc.	155	1361	4721	47021	384521
	N = 1000 B = 128 L = 1000	N = 5000 B = 1024 L = 5000	N = 10000 B = 1024 L = 10000	N = 15000 B = 1024 L = 15000	N = 15000 B = 2048* L = 15000
C (Eq. 6)	26269	124301	321781	592461	442989
Clk. Cyc.	26269	124301	321781	592461	442989

The results of Table 1 are in excellent agreement with the expectations, which translate into: validation of the hardware implementation as well as a demonstration of its equivalence to the Julia *SFIXED* software implementation; and verification of the validity of Eq. 6, which, in turn, is a validation of the results of Fig. 5. The concluding step of this Section is, then, to synthesize the hardware so that the maximum achievable clock frequency can be extracted. As previously commented, the clock frequency, combined with the total number of clock cycles necessary for the algorithm to elapse, can be used to estimate the amount of time the algorithm will take to execute. Furthermore, as a by-product of the synthesis results, it is possible to assess the percentage of FPGA resources occupied by the architecture, which, in turn, provides the means for selecting the target FPGA for hardware implementation. The results are summarized in Table 2, where the Intel Quartus Prime synthesis software was used. It is important to note that, due to the complexity of the Place and Route (PAR) problem, it is not reasonable to assume that the synthesis software will always find the global optimum. Therefore, the results of Table 2 should be interpreted as lower bounds of the optimal achievable clock frequency for each design instance and the discrepancies in these to be within the uncertainty of the PAR procedure.

Up to 1024 BRAMs could be instantiated in the Stratix V, with as high as 109 MHz maximum clock frequency yielding a 1.91 second processing time for 10000-long data series considering 650 iterations per sample (overall, 6.5 million iterations). This result represents a 100 speedup factor in processing time with respect to software implementations under the same data conditions but running on a Intel Xeon CPU E5-2690 v4 at 2.6 GHz and 512 GB RAM [1], a major achievement, which advocates for the dedicated hardware solution applied to the trend break detection problem. It is also interesting to note that, for a smaller FPGA, the Cyclone V, a ~ 16 gain factor with respect to the software implementation was achieved, which is interesting in the sense that smaller FPGAs exhibit, generally, significantly lower costs, but could still deliver processing times in the range of a few seconds.

The results from Table 2 also indicate that a compromise between instantiation of a higher number of BRAMs (which reduces the total number of clock cycles necessary for the algorithm to elapse as determined by Eq. 6) and the maximum achievable clock frequency exists. In fact, the processing time for 512 instantiated BRAMs was lower than that of 1024 because the gain in clock frequency superseded that of the reduction of clock cycles. Again, it should be mentioned that the PAR problem is an extremely complex one and the algorithms that solve it may not always reach the best possible solution, so the clock frequency values obtained should be interpreted as lower bounds. Finally, to put the results into an application proned perspective, fiber profiles as long as 50 km could be analyzed in search for breaks in under 10 seconds [1].

6 Results and Discussion

6.1 Case Study Results

Validation of the software Julia *SFIXED* implementation performed in Section 5 allows one to investigate

Table 2 Target FPGAs Synthesis and Timing Results

Processing times calculated for $N = 10000$ and with $L = 6.5 \times 10^6$

	Stratix V: 5SGSMD5K2F40C2					Cyclone V: 5CSXFC6D6F31C6				
BRAMs	ALMs out of 172,600	Registers	Memory [Bits] out of 41,246,720	Max. Clk. Freq. [MHz]	Proc. Time [s]	ALMs out of 41,910	Registers	Memory [Bits] out of 5,662,720	Max. Clk. Freq. [MHz]	Proc. Time [s]
1024	135,571(79%)	88,938	20,971,520(51%)	109.9	1.91	–[1]	–[1]	–[1]	–[1]	–[1]
512	68,135(39%)	45,595	10,485,760(25%)	166.97	1.78	–[1]	–[1]	–[1]	–[1]	–[1]
256	36,098(21%)	25,147	5,242,880(13%)	173.28	2.78	36,548(87%)	24,741	2,621,440(46%)	81.13	5.94
128	19,196(11%)	13,839	2,621,440(6%)	182.12	4.70	20,178(48%)	13,475	1,310,720(23%)	95.37	8.98
16	4,982(3%)	4,012	327,680(< 1%)	182.35	33.83	5,000(12%)	3,928	163,840(3%)	92.94	66.37
4	3,435(2%)	2,925	81,920(< 1%)	187.86	130.08	3,427(8%)	2,929	40,960(< 1%)	91.73	266.40

[1] : Design too large to fit into device.

aspects of the hardware implementation in a more suitable simulation environment. This is important due to the amount of simulation workload necessary to yield statistically relevant results. To provide a complete overview, the analysis is split into two steps. First, an experimental dataset extracted in a laboratory environment, with standard measurement devices, is processed by the bit-true validated Julia *SFIXED* implementation and by the original 64-bit floating version of the algorithm [1]. These results are evaluated in terms of their trend break detection capabilities or, in other words, the performance of the two versions of the algorithm.

After analysis with real-world data, which, unfortunately, is limited to the availability of resources in the laboratory, the second step is to submit the algorithm to simulated datasets that contain the same features as the real-world ones; in [1], the creation of a testbench of simulated datasets is discussed in detail, specifically in regards to noise addition, with very pronounced resemblance between the real-world and simulated results; this, in turn, allows for statistically relevant investigation of the performance of the algorithm. It should be noted that, for all the results presented in this Section, the bit-width of data points for the *SFIXED* format was fixed at 20, where the reason behind this will be clarified in Section 6.1.2.

6.1.1 Analysis of a Real-World Dataset

The subject of the experimental analysis of the proposed hardware implementation was chosen to be the dataset corresponding to the measurement of an optical fiber profile. The data acquisition was performed with a so-called Optical Time Domain Reflectometry (OTDR) device based on an FPGA [10]. The OTDR provides the user with information of the fiber's integrity (the fiber profile) by measuring the optical power that is reflected back from the optical fiber when a probing pulse is launched into it [26]; the result is displayed in logarithmic scale (dB), as depicted by the black trace in Fig. 6. This choice goes along with the motivation of the Introduction Section, since the acquisition system and the proposed hardware implementation of the analysis algorithm could, potentially, coexist in the same FPGA. The chosen fiber profile exhibits several interesting features related to fault detection: the presence of both high magnitude and low magnitude faults; and faults separated by few points.

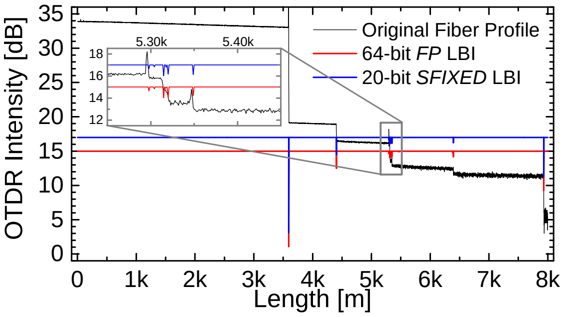


Figure 6 Results of trend break detection on a testbench real-world fiber profile acquired in laboratory conditions. Red and blue traces correspond, respectively, to the results from the 64-bit floating point and 20-bit *SFIXED* bit-true versions of the LBI algorithm. The traces are vertically offset in order to highlight the correct trend break detection but the original baseline is the same. The inset displays a zoomed-in segment of the profile, where faults close to one another have been correctly identified by both versions of the algorithm.

The fact that the LBI algorithm, in its original version, provides accurate trend break detection results had already been determined in [1] and is again verified by the red trace in Fig. 6. The striking feature showcased in Fig. 6, however, is the fact that the bit-true validated *SFIXED* version of the algorithm, the blue trace, exhibits equivalent results, indicating that the proposed hardware implementation of the LBI algorithm using the 20-bit *SFIXED* format upholds the performance of its 64-bit floating point counterpart. One important comment is regarding the linear slope component that is clearly part of the original signal in Fig. 6. In [1], the slope component has been included in the candidate matrix **A** for completeness; however, this slope can be pre-compensated since it is a standard value for most commerical optical fibers as specified in ITU-T G.652. In order to maintain the LBI algorithm, as presented in Algorithm 1, general with respect to any trend break detection problem but, at the same time, allow for data from optical fiber monitoring to be processed by this version of the algorithm, the slope has been pre-compensated.

6.1.2 Testbench of Simulated Datasets

The objective of using a set of simulated datasets, and, therefore, accumulating statistically relevant data (results of over 15000 different simulated datasets were analysed), is twofold: firstly, as it was mentioned in Section 3, to empirically determine the interval of number of iterations per sample for which the estimation quality reaches a reliable level; and, secondly, to compare the quality of estimation between the 64-bit floating point implementation, and the 20-bit *SFIXED* implementation for profiles with different number of data points and with different number of iterations per sample.

Evaluation of the bit-width parameter has been conducted based on the limited number of possible bit-width configurations of the BRAMs in the target FPGAs considered here; supported bit widths for both the STRATIX V and the CYCLONE V are: 10, 20, and 40 bits [27]. All three configurations have been tested and the results are as follows. The 10-bit *SFIXED* implementation proved to be too limited in terms of correctly expressing the input data, since large discrepancies with respect to the benchmark results ran in a personal computer with a 64-bit floating point precision have been observed. As the bit-width was increased to 20, these discrepancies were drastically diminished, indicating a more equivalent expression of the data series and, also, a sufficient arithmetical dynamic range. Further increasing the bit-width to 40, however, did not effect any significant change in the performance of the algorithm and, since this configuration requires twice the memory usage when compared to the 20-bit one, while also increasing the complexity of the arithmetical structures in the FPGA, the value of 20 bits has been set as an operational parameter for all the tests.

To clarify this analysis, an estimation error metric has been used, following the definition in [1]: noiseless sparse vectors with randomly sorted magnitudes ($\boldsymbol{\beta}_{\text{ideal}}$) are used to create datasets using the candidate matrix **A**, to which white Gaussian noise is added; the results of the estimation are $\boldsymbol{\beta}_{\text{est}}$, which are then compared to $\boldsymbol{\beta}_{\text{ideal}}$ using the squared error norm. The closer to zero error between the estimated and ideal vectors, the better is the estimation. The results of such estimation error metric are depicted in Fig. 7 for different values of L and for both the 20-bit *SFIXED* and the 64-bit floating point implementations; the results are extremely similar, indicating that no information is lost due to the 20-bit *SFIXED* implementation. Here, the intrinsic slope of optical fiber profiles has not been considered once again, as discussed in the previous sub-section.

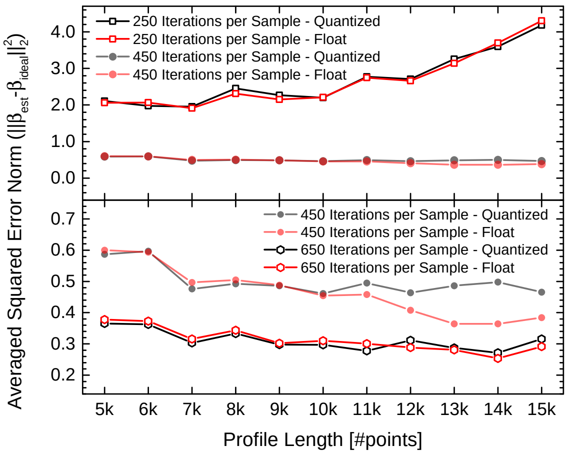


Figure 7 Estimation results for both SFIXED and floating point implementations for different profile lengths and different numbers of iterations per sample; L is simply this value multiplied by the number of points (N) in the profile. The results are divided into two panels for ease of visualization.

From the results of Fig. 7, it is possible to conclude that the differences in estimation between the 20-bit *SFIXED* and 64-bit double implementations are negligible, i.e., the hardware implementation will have no problems achieving comparable estimation accuracy as the software version, e.g. in [1]. A large variety of test cases (different lengths and fault scenarios) have also been tested, and the results point to the fact that a 20-bit fixed point resolution provides comparable results in terms of detection performance when compared to longer bit length words. The conclusion, therefore, is that significant deviations for other data sets are highly unlikely. Furthermore, it is important to note that, as discussed in depth in [1], it is impractical to expect the fault detection algorithm to be able to resolve any fault (with an arbitrary magnitude) for two main reasons: first, the amount of information gained from identifying a fault below a certain level (say, smaller than 0.1) is close to none, since the impact of this fault on the optical link transmission is minimum; second, in order to achieve a level of sensitivity that allows one to identify extremely small faults would require the processing time to be also extremely high, as also discussed in [1]. With that being said, it can be expected, from the results, that the gain in sensitivity from increasing the bit-length of the words beyond 20 falls within a practical sensitivity issue and, thus, becomes irrelevant.

Furthermore, it becomes clear that, after 450 iterations per sample, the accuracy stabilizes, with an averaged squared error norm value in the order of 0.5. This indicates that the results from Fig. 5 with 650 iterations per sample are indeed realistic. Finally, this result is also useful when interpreted along with those of Fig. 5: a compromise between the total number of clock cycles before the algorithm elapses and the quality of the estimation can be found and, in specific cases, one of these can be sacrificed (increasing the processing time or allowing for a worse estimate) to boost the other (faster results or extremely precise estimation). In real world applications, the expected accuracy (given a set of parameters including bit-length and total number of algorithm iterations) can be determined in advance and provided to the user, so that the constraints are known *a priori*. If necessary, the memory word length can be adjusted such that it is larger or equal to the bit-width that yields the desired calculation precision.

6.2 State-of-the-Art Hardware Implementations of Sparse Estimation Algorithms – Comparison with the Proposed Architecture

The LBI algorithm, in order to be suitable for tackling the trend break detection problem, was modified from its original format; the especial structure of the candidate matrix, for instance, allowed one to derive the expressions in Eqs. 3 and 4, which greatly simplify the procedure [1]. This fact prompts the natural question of whether there are other methodologies that, even though originally intended to be used in sparse estimation, could be steered towards the application at hand. Moreover, this would allow for a comparison between the herewith proposed hardware implementation of the LBI algorithm with other similar structures. An extensive literature review on this subject reveals that, when it comes to hardware implementations of sparse estimation methods, the majority of the works focus on the orthogonal matching pursuit (OMP) algorithm [28], which has, thus, been the focus of this comparison.

Although an interesting approach, where efficient architectures have been proposed such as [29,30], OMP typically assumes that the number of non-zero entries of the vector $\boldsymbol{\beta}$ – and, thus, the number of candidates that compose the signal of interest – is known. For applications where this assumption is valid, the method is highly efficient and, in addition, provides recovery guarantees, i.e., the certainty that, if present, the candidate will be found. For the cases that have been described in the introductory section of this document, however, this is a rather detrimental characteristic; for the specific case of trend break detection associated to fiber fault detection, for instance, this assumption would yield results that do not translate the true underlying trend of the original signal. Furthermore, the OMP algorithm relies, for optimal performance, on a small value of the so-called *mutual incoherence*

$$\mu = \max_{i \neq j} \frac{|\tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_j|}{\|\tilde{\mathbf{a}}_i\|_2 \|\tilde{\mathbf{a}}_j\|_2}, \tag{7}$$

with $\tilde{\mathbf{a}}_i$, $\tilde{\mathbf{a}}_j$ as the i^{th} and j^{th} columns of $\mathbf{A}$, respectively; OMP requires the value μ to be smaller than $\frac{1}{2k-1}$ for guaranteed recovery of the sparse positions [31]. As one can easily check from (2), in the case of the candidate matrix associated to trend break detection, the value μ is close to its maximum value of 1 when it comes to close neighbouring columns of $\mathbf{A}$, which could significantly decrease the OMP algorithm's performance in this scenario.

In order to propitiate an interesting comparison and further discussion about the disparities between the two algorithms and, moreover, between their respective hardware implementations, a framework for utilizing the OMP algorithm in a trend break detection scenario has been developed and is as follows. A sparsity factor ξ is defined and the maximum number of non-zero elements within the estimation problem is determined using ξ and the total number of points in the dataset of interest, N; this maximum non-zero number is herewith defined as $N_0 = \xi N$. It is important to note that, since the actual number of non-zero elements that are present in the signal of interest is not known *a priori*, this approach will always be suboptimal when compared to an algorithm, such as the LBI, that does not necessitate such information. The OMP algorithm is, then, ran normally with N_0 iteration loops while the intermediate estimation results are submitted to a consistent information criterion for determining the best sparse approximation; the Bayesian Information Criterion [32], which balances the squared error norm and the resulting sparsity for model identification, has been employed. The complete procedure, dubbed OMP for Trend Break Detection, is structured in Algorithm 2, and has been directly adapted from [33].

The OMP for Trend Break Detection algorithm involves three main procedures, defined in Algorithm 2, as follows. The optimization problem of line 7 allows one to find the candidate with the highest overlap with the residual of previous iterations. This candidate is removed from the measured signal and the new residual is determined (in line 11) with an intermediate estimate ($\hat{\boldsymbol{\beta}}_i$) being generated as a by-product

Algorithm 2 OMP for Trend Break Detection

```
**Require:** Measurement vector $\mathbf{y}$, Sparsity Level $\xi$,
Candidate Matrix $\mathbf{A}$
**Ensure:** Estimated $\hat{\boldsymbol{\beta}}_{\text{out}}$
1: $\mathbf{r}_0 = \mathbf{y}$
2: $\boldsymbol{\Phi}_0 = []$
3: $\Lambda_0 = \{\emptyset\}$
4: $N_0 = \xi N$
5: $\text{BIC}_{\text{res}} = \infty$
   → Innitiate OMP algorithm
6: for $i = 1 : N_0$ do
7:     $\lambda_i = \arg\max_{j=[1,N]/\Lambda_{i-1}} |\langle \mathbf{r}_{i-1}, \mathbf{A}_j \rangle|$
8:     $\Lambda_i = \Lambda_{i-1} \bigcup \{\lambda_i\}$
9:     $\boldsymbol{\Phi}_i = [\boldsymbol{\Phi}_{i-1}\ \mathbf{A}_j]$
10:    $\hat{\boldsymbol{\beta}}_i = \arg\min_x ||\boldsymbol{y} - \boldsymbol{\Phi}_i x||$
11:    $\mathbf{r}_i = \mathbf{y} - \boldsymbol{\Phi}_i \hat{\boldsymbol{\beta}}_i$
   → Introducing the BIC as a function
12:    $\text{BIC}_{\text{new}} = \text{BIC}\left(\hat{\boldsymbol{\beta}}_i, i\right)$
13:    if $\text{BIC}_{\text{new}} < \text{BIC}_{\text{res}}$ then
14:        $\hat{\boldsymbol{\beta}}_{\text{out}} = \hat{\boldsymbol{\beta}}_i$
15:        $\text{BIC}_{\text{res}} = \text{BIC}_{\text{new}}$
16:    end if
17: end for
```

(in line 10). Finally, the best intermediate estimate is found using the BIC and output as the final OMP estimate.

6.2.1 Performance Comparison Based on Software Implementation

Being a well-known algorithm in the sparse estimation community, several implementations of the OMP can be found as well as detailed explanations of the procedure; here, a *Cholesky*-decomposition-based OMP has been used as reference [33]. In order to maintain consistency with respect to the Julia implementation of the LBI, the code has been written in Julia.

The first comparison step, based on the software implementation, allows for the comparison of both algorithms to be analyzed. This entails, as set forth in [1], the evaluation of figures of merit such as the *precision* and the *Matthews correlation coefficient* (MCC) [34], where a so-called Contingency Table is used to derive the aforementioned figures of merit, and revolves around the determination of so-called True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) associated with the detection of breaks in the dataset. Assessing the performance with statistically relevant results is an important part of the performance comparison, so the framework of the testbench of simulated datasets, discussed in Section 6.1-B, has been once again employed. For the results herewith presented, the maximum number of trend breaks included in the simulated datasets was five and, since the comparison is not related to timing, ξ was set as 0.01, i.e., one percent of the total number of points in the dataset. Finally, N was varied from 5000 to 15000 so that different dataset sizes could be considered in the analysis.

Table 3 Performance Comparison

Algorithm	Precision	MCC
	$\left(\frac{TP}{TP+FP}\right)$	$\left(\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}\right)$
LBI	67.3%	0.81
OMP	8.2%	0.28

As the results of Table 3 demonstrate, the performance of the OMP for Trend Break Detection does not reach the performance level exhibited by the LBI, even when the sparsity level N_0 is made much larger than the actual number of breaks present in the dataset. On one hand, the precision indicates how confident one can be about a break identified by the algorithm, which represents a huge impact on the application. If one considers, for instance, an optical fiber network manager that must take actions regarding the repairment of a cable, the cost of deploying a mobile unit creates a demand for high precision in the analysis results. On the other hand, the MCC is an interesting figure of merit from a theoretical point of view, since it stands for a correlation coefficient between the observed and predicted binary classifications; in this case, the break/no-break classification: a value of +1 corresponds to perfect prediction; and a value of 0 corresponds to random classification results. It is possible to observe, with this respect, that the LBI algorithm is very much closer to correctly predicting the underlying trend of the dataset (with an MCC of 0.81) than the OMP algorithm, which exhibits a low MCC value of 0.28. The conclusion of this analysis indicates that, even in case the performance of the OMP is mantained when the structure is adapted to hardware, as has been demonstrated in Sections 5-A and -B for the LBI, it will remain as a suboptimal choice for trend break detection with respect to its performance.

6.2.2 Comparison Based on Hardware Implementation

The in-depth description and analysis of the LBI hardware structure presented in Section 4 allowed one to derive the total number of clock cycles taken, for a given set of input parameters, for the algorithm to elapse. Similar analysis for the OMP is, unfortunately, not available in the literature; since the focus of this document is not on the hardware implementation of the OMP, the comparison must, then, be performed using a different figure of merit. In [33], two interesting figures of merit are presented: the total processing time (0.34 ms) for fixed parameters (a degree of sparsity of

36 (p_1), a (p_2) 256-long measurement vector and 1024 (p_3) samples); and the total number of multiplication and addition operations as a function of $p_{1..3}$. There, the data precision, in bits, was 18, as opposed to the 20-bit *SFIXED* considered, here, for the LBI, which is within a comparable range.

For the closest comparable instance size for trend break detection (1024×1024), the hardware implementation of the LBI would elapse (considering the data in Table 2 in 64 ms, an $\sim$200 factor with respect to the OMP. This result is, however, ambiguous and further analysis is necessary, which has been subdivided into: complexity; and device usage. Analysis of the complexity has been carried out indirectly by evaluating the total number of multiplications necessary for both hardware architectures. The total number of multiplications necessary for the OMP increases cubically [33] in the degree of sparsity and quadratically in the instance size, while that for the LBI increases linearly (as discussed throughout Sections 3 and 4). Even though for small instance sizes, such as the previously mentioned 1024×1024 problem, the OMP exhibits a total number of necessary multiplications much lower than the LBI, the relation quickly inverts as the instance size is increased to, for instance, 15000×15000, a quite small dataset for optical fiber analysis, and higher. This result clearly evidenciates why, in [33], the digital signal processing (DSP) core of the target FPGA is almost at its usage limit for a relative small instance size and the fact that, as the instance size increases the OMP becomes a less attractive solution to trend break detection.

Even though the target FPGAs for the architecture presented here (the Altera Stratix V) and in [33] (the Xilinx Virtex-6) are not the same, they are in the same range of FPGA sizes and, thus, allow for device usage comparison. For the OMP, the device usage (where memory is included) is associated to the necessary matrix inversion for line 10 in Algorithm 2 and the storage of the candidate matrix elements. Therefore, the size of the hardware depends on the problem size and increases tremendously if the degree of sparsity increases; in [33], the modest $p_1 = 36$, $p_2 = 256$, and $p_3 = 1024$ problem occupied 76% of the DSP slices available, 69% of the available BRAMs, and a total of 16% slices. In the proposed LBI architecture, not only the number of occupied BRAMs is more flexible (the architecture of the $\boldsymbol{\beta}$ matrix in Eq. 5 can be made more vertical or more horizontal) but also the relative number of occupied slices is also smaller; a 128 BRAMs design, which enables up to 35000-long datasets, for instance, would only occupy 11% of the FPGA, and no DSP slices, which, in this case, could be reserved for other useful procedures, such as the pre-processing mentioned in Section 3. It is interesting to mention that, according to Eq. 6, there is a tradeoff between the number of used BRAMs and the total number of clock cycles necessary for the algorithm to elapse, which would play a role in case the number of BRAMs was chosen to be either smaller or greater. In both cases, however, it is clear that the overall device usage of the proposed LBI structure is much more feasible, flexible, and economic for datasets with $\geq$10000 points.

7 Conclusions

Trend break detection, or level-shift detection, is a problem that permeates several science fields, and an efficient, accurate, and highly reliable processing unit to solve it is desirable. Combining the flexible hardware design tools of Field Programmable Gate Arrays and the efficient Linearized Bregman Iterations algorithm allowed for the development of such a unit. The manipulation of the data storage structure as well as the algorithm flow and control in hardware yielded an up to 100 times gain in processing time when compared to a personal computer while maintaining all the observed qualities of the algorithm, such as low estimation error and high level-shift detection precision. The speedup factor greatly depends on the memory availability in the target hardware and, even though this specific speed-up factor has been calculated for the Stratix V FPGA, a chip with more than 1000 BRAMs (each containing 1024 20-bit-long words) can achieve the same speedup level.

Due to its flexible memory structure, the proposed hardware architecture can be implemented in different sized-FPGAs, with the main distinctions being the amount of available dual-block RAMs and maximum achievable clock frequency, characteristics that are hardware-dependent. On a middle-sized chip such as the Altera Cyclone V, the hardware supports up to 256 parallel BRAMs with a maximum clock frequency of 81 MHz and a total processing time of 6 seconds for a 10000-long dataset and 650 iterations per sample. Such processing prowess can be directed towards on-line data supervision such as optical fiber monitoring, which constitutes an exciting future point of investigation. Furthermore, incorporating advanced signal processing techniques into the hardware design in order to eliminate any pre-processing step while increasing the convergence speed is also a sought-after goal for future studies.

Evaluation of the architecture using both real-world and simulated datasets making use of its bit-true hardware-validated software showed that the performance is not deteriorated due to necessary adjustments for hardware implementation. In fact, the comparison of the floating point and quantized versions of the algorithm yielded negligible discrepancies when analysing the squared error norm of the estimation. Further comparative analysis of the performance and hardware architecture with respect to a state-of-the-art algorithm for sparse estimation showed that the performance of the LBI for trend break detection is much more pronounced; at the same time, even though slower for small instance sizes, the complexity of the LBI structure allows for manipulation of much longer datasets, which is a necessity for trend break detection in the context of optical fiber analysis.

In summary, the FPGA implementation of the Linearized Bregman Iterations algorithm adapted for trend break detection, reported in the present manuscript, has fomented the following contributions: the description of the hardware structure of the algorithm and an efficient parallel memory access structure; the results that such an implementation provides, with impacting gains in processing time without loss of performance; its comparison with state-of-the-art hardware implementations of sparse estimation algorithms (in particular, the Orthogonal Matching Pursuit algorithm), indicating clear advantages of the proposed architecture in terms of performance and hardware flexibility; and the future points of investigation that it enables in the field of digital signal processing in an FPGA, especially with respect to creating an embedded unit for data acquisition and processing with direct applications in optical fiber analysis.

Abbreviations

AGD: Approximate Gradient Descent; BIC: Bayesian Information Criterion; BRAM: Block RAM; CORDIC: COordinate Rotation DIgital Computer; CPU: Central Processsing Unit; DSP: Digital Signal Processing; FN: False Negatives; FP: False Positives; FPGA: Field Programmable Gate Array; FSM: Finite State Machine; LASSO: Least Absolute Shrinkage and Selection Operator; LBI: Linearized Bregman Iterations; LUT: Look-Up Table; MCC: Matthews Correlation Coefficient; MUX: Multiplexer; OLS: Ordinary Least Square; OMP: Orthogonal Matching Pursuit; OTDR: Optical Time Domain Reflectometry; PAR: Place And Route; PAT: Parallel Adder Tree; PMT: Parallel Multiplexer Tree; RAM: Random Access Memory; SFIXED: Signed Fixed Point; SOC: System On Chip; TN: True Negatives; TP: True Positives; VHDL: Very-high-speed Integrated Circuit Hardware Description Language.

Acknowledgements

Financial support from Brazilian agency CNPq is acknowledged by F. Calliari. This work has been supported by the COMET-K2 "Center for Symbiotic Mechatronics" of the Linz Center of Mechatronics (LCM) funded by the Austrian federal government and the federal state of Upper Austria.

Author's contributions

All authors contributed equally to this work, read and approved the final manuscript.

Availability of data and materials

Data and source code are available from the corresponding author upon request.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Center for Telecommunications Studies, Pontifical Catholic University of Rio de Janeiro, Marquês de São Vicente, 22451-900, Rio de Janeiro, BR. [2]QC2DLab, Kavli Foundation, Technical University of Delft, Mekelweg 5, 2628CC Delft, The Netherlands. [3]Institute of Signal Processing, Johannes Kepler University, Altenbergerstraße 69, 4040 Linz, Austria.

References

1. Lunglmayr, M., Amaral, G.C.: Linearized bregman iterations for automatic optical fiber fault analysis. IEEE Transactions on Instrumentation and Measurement **68**(10), 3699–3711 (2018)
2. Basseville, M., Benveniste, A.: Design and comparative study of some sequential jump detection algorithms for digital signals. Acoustics, Speech and Signal Processing, IEEE Transactions on **31**(3), 521–535 (1983)
3. Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE **77**(2), 257–286 (1989)
4. Lorenz, D.A., Wenger, S., Schöpfer, F., Magnor, M.: A sparse kaczmarz solver and a linearized bregman method for online compressed sensing. In: Image Processing (ICIP), 2014 IEEE International Conference On, pp. 1347–1351 (2014). IEEE
5. Rea, W.S., Reale, M., Cappelli, C., Brown, J.A.: Identification of changes in mean with regression trees: an application to market research. Econometric Reviews **29**(5-6), 754–777 (2010)
6. Storath, M., Weinmann, A., Demaret, L.: Jump-sparse and sparse recovery using potts functionals. IEEE Transactions on Signal Processing **62**(14), 3654–3666 (2014)
7. von der Weid, J.P., Souto, M.H., Garcia, J.D., Amaral, G.C.: Adaptive filter for automatic identification of multiple faults in a noisy otdr profile. Journal of Lightwave Technology **34**(14), 3418–3424 (2016)
8. Kim, S.-J., Koh, K., Boyd, S., Gorinevsky, D.: ℓ_1 Trend Filtering. SIAM review **51**(2), 339–360 (2009)
9. Lunglmayr, M., Huemer, M.: Efficient linearized bregman iteration for sparse adaptive filters and kaczmarz solvers. In: Sensor Array and Multichannel Signal Processing Workshop (SAM), 2016 IEEE, pp. 1–5 (2016). IEEE
10. Calliari, F., Herrera, L.E.Y., von der Weid, J.P., Amaral, G.C.: High-dynamic and high-resolution automatic photon counting otdr for optical fiber network monitoring. Proceedings of the 6th International Conference on Photonics, Optics and Laser Technology - Volume 1: PHOTOPTICS, 82–90 (2018). INSTICC
11. Rumelhart, D.E., McClelland, J.L.: Parallel distributed processing: explorations in the microstructure of cognition. volume 1. foundations (1986)
12. Xie, J., Meher, P.K., Sun, M., Li, Y., Zeng, B., Mao, Z.-H.: Efficient fpga implementation of low-complexity systolic karatsuba multiplier over gf (2m) based on nist polynomials. IEEE Transactions on Circuits and Systems I: Regular Papers **64**(7), 1815–1825 (2017)
13. Greisen, P., Runo, M., Guillet, P., Heinzle, S., Smolic, A., Kaeslin, H., Gross, M.: Evaluation and fpga implementation of sparse linear solvers for video processing applications. IEEE Transactions on Circuits and Systems for Video Technology **23**(8), 1402–1407 (2013)
14. Cong, L., Xiaofu, W.: Design and realization of an fpga-based generator for chaotic frequency hopping sequences. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications **48**(5), 521–532 (2001)
15. Vanderbauwhede, W., Benkrid, K. (eds.): High-Performance Computing Using FPGAs. Springer, New York, NY (2013). doi:10.1007/978-1-4614-1791-0
16. Hu, J., Guo, W., Wei, J., Cheung, R.C.C.: Fast and generic inversion architectures overGF(2^m)using modified itoh-tsujii algorithms. IEEE Transactions on Circuits and Systems II: Express Briefs **62**(4), 367–371 (2015). doi:10.1109/TCSII.2014.2387612
17. Cardells-Tormo, F., Molinet, P.-.: Area-efficient 2-d shift-variant convolvers for fpga-based digital image processing. IEEE Transactions on Circuits and Systems II: Express Briefs **53**(2), 105–109 (2006). doi:10.1109/TCSII.2005.857091
18. Zhang, C., Fang, Z., Zhou, P., Pan, P., Cong, J.: Caffeine: Towards uniformed representation and acceleration for deep convolutional neural networks. In: 2016 IEEE/ACM International Conference on

Computer-Aided Design (ICCAD), pp. 1–8 (2016). doi:10.1145/2966986.2967011

19. Hastie, T., Tibshirani, R., Wainwright, M.: Statistical Learning with Sparsity: the Lasso and Generalizations, 1st edn. CRC press, Monographs on Statistics and Applied Probability 143 (2015)
20. Osher, S., Mao, Y., Dong, B., Yin, W.: Fast linearized Bregman iteration for compressive sensing and sparse denoising. Communications in Mathematical Sciences **8**(1), 93–111 (2010)
21. Meyer-Baese, U.: Digital signal processing with field programmable gate arrays. Springer (2004)
22. Sato, F., Tsuchiya, K., Nagao, Y., Hirama, T., Oka, R., Takahashi, K.: Ultra-high-fiber-count optical cable for data center applications. SEI Technical Review (86), 45–50 (2018)
23. Hornsteiner, A.: Fiber optic technology trends in data transmission: Digitalization of data advance the need for constant upgrading of data networks. Optik & Photonik **12**(4), 20–24 (2017)
24. Lunglmayr, M., Hiptmair, B., Huemer, M.: Scaled linearized bregman iterations for fixed point implementation. In: 2017 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–4 (2017). doi:10.1109/ISCAS.2017.8050534
25. Bauschke, H.H., Burachik, R.S., Combettes, P.L., Elser, V., Luke, D.R., Wolkowicz, H.: Fixed-point algorithms for inverse problems in science and engineering. Springer Publishing Company, Incorporated - Springer Optimization and Its Applications (2013)
26. Barnoski, M., Rourke, M., Jensen, S., Melville, R.: Optical time domain reflectometer. Applied optics **16**(9), 2375–2379 (1977)
27. IntelFPGA: High-Performance Computing Using FPGAs. Intel Corporation (2015). `https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/stratix-v/stx5_51001.pdf`
28. Candes, E.J., Wakin, M.B.: An introduction to compressive sampling. IEEE Signal Processing Magazine **25**(2), 21–30 (2008). doi:10.1109/MSP.2007.914731
29. Ge, X., Yang, F., Zhu, H., Zeng, X., Zhou, D.: An efficient fpga implementation of orthogonal matching pursuit with square-root-free qr decomposition. IEEE Transactions on Very Large Scale Integration (VLSI) Systems **27**(3), 611–623 (2019). doi:10.1109/TVLSI.2018.2879884
30. Rabah, H., Amira, A., Mohanty, B.K., Almaadeed, S., Meher, P.K.: Fpga implementation of orthogonal matching pursuit for compressive sensing reconstruction. IEEE Transactions on Very Large Scale Integration (VLSI) Systems **23**(10), 2209–2220 (2015). doi:10.1109/TVLSI.2014.2358716
31. Cai, T.T., Wang, L.: Orthogonal matching pursuit for sparse signal recovery with noise. IEEE Transactions on Information Theory **57**(7), 4680–4688 (2011)
32. Schwarz, G., *et al.*: Estimating the dimension of a model. The annals of statistics **6**(2), 461–464 (1978)
33. Rabah, H., Amira, A., Mohanty, B.K., Almaadeed, S., Meher, P.K.: Fpga implementation of orthogonal matching pursuit for compressive sensing reconstruction. IEEE Transactions on very large scale integration (VLSI) Systems **23**(10), 2209–2220 (2014)
34. Matthews, B.W.: Comparison of the predicted and observed secondary structure of t4 phage lysozyme. Biochimica et Biophysica Acta (BBA)-Protein Structure **405**(2), 442–451 (1975)

D
Profile-Splitting Linearized Bregman Iterations for Trend Break Detection Applications

The paper "*Profile-Splitting Linearized Bregman Iterations for Trend Break Detection Applications*" was published in the MDPI Electronics journal in 2020.

electronics

MDPI

Article

Profile-Splitting Linearized Bregman Iterations for Trend Break Detection Applications

Gustavo Castro do Amaral [1,2,*,†], Felipe Calliari [1,†] and Michael Lunglmayr [3,†]

[1] Center for Telecommunications Studies, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro 22451-900, Brazil; felipe.calliari@opto.cetuc.puc-rio.br

[2] QC2DLab, Kavli Foundation, Technical University of Delft, 2628 CD Delft, The Netherlands

[3] Institute of Signal Processing, Johannes Kepler University, 4040 Linz, Austria; michael.lunglmayr@jku.at

* Correspondence: gustavo@opto.cetuc.puc-rio.br

† These authors contributed equally to this work.

Received: 30 January 2020; Accepted: 18 February 2020; Published: 3 March 2020

Abstract: Trend break detection is a fundamental problem that materializes in many areas of applied science, where being able to identify correctly, and in a timely manner, trend breaks in a noisy signal plays a central role in the success of the application. The linearized Bregman iterations algorithm is one of the methodologies that can solve such a problem in practical computation times with a high level of accuracy and precision. In applications such as fault detection in optical fibers, the length N of the dataset to be processed by the algorithm, however, may render the total processing time impracticable, since there is a quadratic increase on the latter with respect to N. To overcome this problem, the herewith proposed profile-splitting methodology enables blocks of data to be processed simultaneously, with significant gains in processing time and comparable performance. A thorough analysis of the efficiency of the proposed methodology stipulates optimized parameters for individual hardware units implementing the profile-splitting. These results pave the way for high performance linearized Bregman iteration algorithm hardware implementations capable of efficiently dealing with large datasets.

Keywords: trend break detection; linearized Bregman iteration; optical time domain reflectometry; FPGA

1. Introduction

Linearized Bregman iterations [1] are a class of computationally efficient algorithms for solving the combined ℓ_1/ℓ_2 minimization problem, for which vast applications can be found in the fields of compressed sensing [2], image analysis [3], and signal denoising [4]. Both the fact that each iteration is of low complexity [5] and that a good denoising quality is achieved with even a relatively low number of iterations [5] make the algorithm a good candidate for real-time estimation problems such as beam forming [6]. Furthermore, an adaptation of the original formulation, the sparse Kaczmarz algorithm [7], has been considered for hardware implementation due to its vectorized structure, consisting of an approximate gradient descent followed by a non-linear shrink function [8].

A prolific problem that can be cast as a combined ℓ_1/ℓ_2 minimization is Trend Break Detection (TBD), which also finds applications in several fields of research [9,10]. In fact, the results presented in [11,12] indicated outstanding performance achieved by the Linearized Bregman Iteration (LBI) when applied to the TBD problem. Moreover, an efficient hardware implementation of the algorithm, with gains in processing time of about two orders of magnitude [12], attests to the prowess of the LBI for trend break detection.

Dealing with large datasets, however, can be a great challenge even for low-complexity algorithms with efficient hardware implementations. In fact, the algorithm's complexity can be shown to be

Electronics **2020**, *9*, 423; doi:10.3390/electronics9030423 www.mdpi.com/journal/electronics

quadratic with respect to the dataset length N—which is demonstrated in Section 2—and the timing necessary for the algorithm to converge becomes eventually impracticable. Even though the TBD problem has a broad presence across different scientific fields, the application focus, in this document, will be given to fault detection in optical fiber profiles, in which trend breaks are associated with such faults [13,14]. In such an application, timely trend break detection results are sought so that mobile repair units can be quickly deployed and the downtime of the network can be kept as small as possible so as not to affect the network users greatly [15,16]. Simultaneously, datasets produced by optical fiber monitoring devices can contain several thousands of points [11].

In this work, a new methodology to deal with high-dimensional TBD problems within the LBI framework is proposed. This is the profile-splitting method, where, instead of analyzing the profile as a single N-dimensional vector, the algorithm evaluates multiple M-dimensional vectors that, together, compose the original data. In Section 2, the combined ℓ_1/ℓ_2 minimization cast as a TBD problem is presented, as well as the structure of the profile-splitting method. Even though the gain in timing arising from this approach can be easily demonstrated, two fundamental issues arise.

The first issue regards the performance of the algorithm, which must be maintained in order for the method to be valid; otherwise, the method would be equivalent to sacrificing estimation performance for faster results, which could be achieved with different algorithms for even faster processing times [14]. Upholding the unmatched trend break detection prowess of the LBI [11,12] is, therefore, a crucial goal of the profile-splitting method, and the discussion and results are presented in Section 3.

The second issue regards the actual implementation of units that process the split-profiles simultaneously and within the same hardware. It is necessary to determine whether the available platforms contain enough resources such that the simultaneity leads to gains in timing as expressive as expected. Here, the parallel implementation on and resource availability of Field Programmable Gate Arrays (FPGAs) are studied and presented in Section 4. Section 5 covers the Materials and Methods, and the paper is concluded in Section 6, where possible future research directions and applications are debated.

2. The LBI Algorithm Applied to the TBD Problem

In order to apply sparse signal denoising techniques to the TBD problem, it is mandatory for the amount of trend breaks within the signal of interest to be much smaller than the number of observations. In this case, the original ℓ_0 problem of counting the number of identified breaks—which turns out to be an NP-problem—can be relaxed in such a way that the ℓ_0 pseudo-norm is replaced by the ℓ_1 norm while the sparsity of the result is still enforced for many applications [17].

Linearized Bregman iterations add a further ℓ_2 term in the cost function leading to the following problem formulation:

$$\arg\min_{\boldsymbol{\beta}} \lambda \|\boldsymbol{\beta}\|_1 + \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 \text{ s.t. } \boldsymbol{A}\boldsymbol{\beta} = \boldsymbol{y}. \tag{1}$$

Although adding this ℓ_2 norm to the cost function in general leads to less sparse results, this effect can be compensated by choosing large enough λ values [11]. The advantage of using such an additional ℓ_2 term is that the optimization problem becomes strongly convex, leading to better algorithmic properties [18]. Moreover, the constraint $\boldsymbol{A\beta} = \boldsymbol{y}$ is considered by the algorithm in a Lagrangian manner [2], for which the algorithm implicitly finds the values γ, leading to the following formulation of the problem [18]:

$$\arg\min_{\boldsymbol{\beta}} \max_{\gamma} \lambda \|\boldsymbol{\beta}\|_1 + \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 - \gamma^T(\boldsymbol{A}\boldsymbol{\beta} - \boldsymbol{y}). \tag{2}$$

Even though algorithmic variations that output γ can be employed, the focus here is primarily on the estimation result $\boldsymbol{\beta}$. Therefore, γ is treated as a byproduct of the algorithm, leaving λ as the only free parameter to be selected [11].

It is clear that the balance between sparsity and a close approximation between the filtered and original signals is controlled by the value of λ; at the same time, λ also has a strong influence on the overall elapsed time of the solving algorithms [11], making its selection a delicate task for ℓ_1/ℓ_2 estimation. If one chooses too large a value, the estimation results will be too sparse (i.e., featuring a high number of false negatives, as set forth by [11]); conversely, if one chooses too small a value, the results will be exceedingly non-sparse (i.e., featuring a high number of false positives [11]).

Furthermore, it can be shown that even with an exceedingly high value of λ, the results will scarcely contain a single non-zero β entry where the break can be expected, but rather, a cluster of non-zero β entries around the position of the break. Since, in many applications [11,13,19], it is important that the procedure returns a single value for the positions of breaks, techniques such as cluster analysis [20] or extensive searches on the reduced detected cluster subspace [21,22] can be employed to narrow down the positions. In [11], a procedure involving the detection of the highest identified peaks was put forth with high quality results and minimum computational effort, so this methodology is also considered here.

In summary, the whole procedure involves the detection of the reduced subspace containing positions around the break points, which is accomplished by the core trend break detection algorithm, in this case the LBI, and afterwards, the positions are determined following a peak search algorithm. The LBI algorithm, in turn, is presented in its vectorized form, the sparse Kaczmarz, in Algorithm 1, where several optimizations to consider the candidate matrix associated with the TBD problem [12] were already taken into account. The impact of the special structure of the candidate matrix is twofold: first, it prevents the algorithm from storing and accessing the candidate matrix values; second, it allows for reduced complexity as only additions are required for the employed scalar product operations.

Algorithm 1 Linearized Bregman iteration for trend break detection

```
Require: Measurement vector $\boldsymbol{y}$, $\lambda$, $\boldsymbol{\beta}_{\text{start}}$, $\boldsymbol{v}_{\text{start}}$, $L$
Ensure: Estimated $\hat{\boldsymbol{\beta}}$
$\boldsymbol{\beta}^{(0)} \leftarrow \boldsymbol{\beta}_{\text{start}}$
$\mathbf{v}^{(0)} \leftarrow \boldsymbol{v}_{\text{start}}$
$i \leftarrow 1$
while $i < L$ do
    $k \leftarrow \operatorname{mod}((i-1), N) + 1$                         ▷ cyclic re-use of rows of A
    $\mu_k \leftarrow \frac{1}{k}$
    $e \leftarrow \left(y_k - \sum_{s=1}^{k+1} \beta_s^{(i)}\right)$       ▷ instantaneous error with inner product
    $d \leftarrow \mu_k e$
    for $j = 1..k$ do
        $v_j^{(i+1)} \leftarrow v_j^{(i)} + d$
        $\beta_j^{(i+1)} \leftarrow \text{shrink}\left(v_j^{(i+1)}, \lambda\right)$
    end for
    $i \leftarrow i + 1$
end while
```

2.1. Application to Fiber Fault Detection

Remote detection of faults in optical fibers is an essential tool for the robust operation of modern optical networks, which compose the physical layer upon which the current high-rate telecommunication services are built. Single-ended solutions, such as Optical Time Domain Reflectometry (OTDR) [23], are preferred since the optical network manager can simultaneously feed the fiber with data signals and also probe it for faults [24]. The fundamental working principle of the OTDR is to send a probing pulse into the optical fiber and acquire the backscattered light as a function of time; this is, then, translated into distance with the knowledge of the average speed of light in the fiber.

The final product of the OTDR is a dataset containing the measured optical power as a function of distance; a piecewise linear function in log scale, corrupted by measurement noise, where the presence of faults are represented by sharp decreases in the power, which are associated with the trend breaks potentially identified by the LBI algorithm. Apart from trend breaks, the LBI algorithm can be adapted to accommodate a candidate corresponding to a linear slope in the dataset; this modification is crucial in fiber fault detection applications since, due to the intrinsic attenuation experienced by light as it propagates in the fiber, the dataset presents a negative slope. The absence of such a candidate would induce the procedure to include breaks that do not compose the original signal in order to minimize the squared error norm of Equation (1). In the upper panel of Figure 1, a usual dataset generated by an OTDR measurement is depicted; in the lower panel, LBI-based fault analysis results are presented in the cases where the slope component is absent or present in the model.

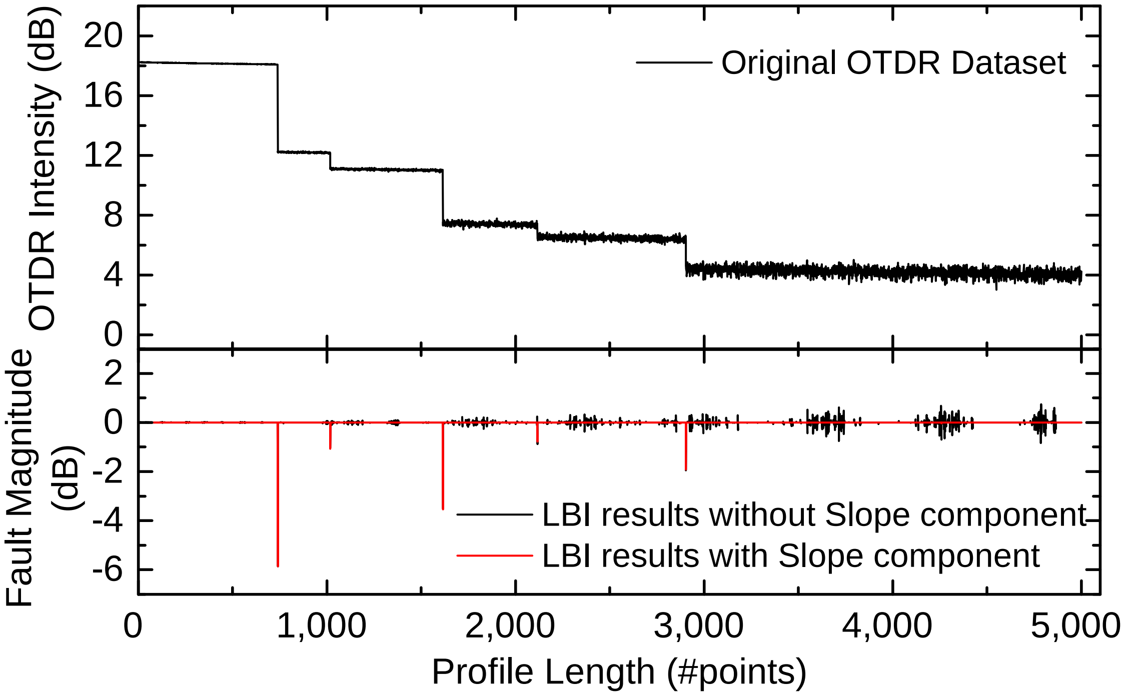


Figure 1. Effect of the inclusion of the slope component in the analysis of a fiber profile. In the upper panel, the original OTDR dataset containing the slope component is depicted. In the lower panel, two different LBI fault detection results are presented; when the slope is not included, a high incidence of false positives can be distinguished, which vanishes when the slope is considered.

The results of Figure 1 make it clear that the slope component is indispensable in the analysis of OTDR datasets. In fact, the extra components forced in the final results when the slope is not included in the model are so-called false positives. It is important to note that the true positives, or components added to the final algorithm result that indeed correspond to components in the original signal, do not vary from one result to the other; in other words, the algorithm is able to identify the true underlying components even when the slope is not included, the difference being whether the results are flooded by false positives or not.

Throughout this document, the case-study results are based on the analysis of fiber profiles such as the one depicted in Figure 1 by the LBI and split-profile LBI algorithms. In fact, the procedure described in Algorithm 2 has been set forth in [11] for the consistent generation (in a simulated environment) of datasets that capture the characteristics of real fiber datasets and is used here. The inputs are a set of fault positions and their respective magnitudes ($\boldsymbol{\beta}$), the length of the dataset (N), and the linear slope (α); the output is the dataset $\mathbf{y}$. Here, ones (N) constructs an all-ones matrix of size $N \times N$ and tril $(\cdot)$ extracts the lower triangular portion of a square matrix by setting the elements above the main diagonal to zero. The function ADDPOISSONNOISE reflects the detection of optical power by the photodetector and, consequently, the reduced signal-to-noise ratio (SNR) as the distance increases.

Algorithm 2 Fiber dataset simulation.

```
Require: Fault vector $\boldsymbol{\beta}_{\text{ideal}}$, $N$, $\alpha$
Ensure: Output dataset $\mathbf{y}$
$\mathbf{A} = [[1, .., N]^T, \text{tril}(\text{ones}(N))]$
$\boldsymbol{\beta}_{\text{ideal}} = [\alpha, \boldsymbol{\beta}_{\text{ideal}}]$
$\mathbf{y}_{\text{noiseless}} = \mathbf{A}\boldsymbol{\beta}_{\text{ideal}}$
$\mathbf{y} = \text{ADDPOISSONNOISE}(\mathbf{y}_{\text{noiseless}})$
```

2.2. Time Scaling with Data Length

Examination of the procedure described in Algorithm 1 allows one to extract the upper bound of the number of operations necessary for the LBI to process an instance of size N. In the explanation that follows, an iteration consists of the set of operations computed for a given value of i. According to Lines 6, 7, 8, and 10 of Algorithm 1, the first iteration ($k = 1$) requires, respectively, a division by an integer $\left(\mu_k = \frac{1}{k}\right)$, a single addition $\left(\beta_1^{(i)} + \beta_2^{(i)}\right)$, a multiplication $(d = \mu_k e)$, and another addition $\left(v_1^{(i+1)} = v_1^{(i)+d}\right)$. It becomes clear that the division and multiplication will appear only once per iteration, but the number of additions will depend on k and, therefore, will vary from one iteration to the next.

Following the steps of the second iteration ($k = 2$) clarifies this point, since the number of multiplications and divisions will remain at one, but the number of additions will be, now, four. From that, it is straightforward, using induction on k, to show that the total number of addition operations in the Nth iteration will be $2N$. On average, the number of addition operations per iteration ($\bar{\text{O}}$P) is, then:

$$\bar{\text{O}}\text{P} = \frac{2(N+1) \cdot \frac{N}{2}}{N} = N + 1. \tag{3}$$

It was demonstrated in [11] that, for consistent trend break detection results, the total number of iterations, L, should scale with the dataset length N, i.e., $L = \alpha \cdot N$. In this case, the total number of additions until the algorithm elapses is given by $\text{OP} = \alpha \left(N^2 + N\right)$, which results in an algorithm complexity of $O\left(N^2\right)$ in terms of additions. In Figure 2, the average elapsed time of the conventional LBI algorithm, for different instance sizes, is presented, where a second order polynomial in N has been used to fit the points with a resulting R^2 (R-squared) value of 0.999, validating this relation.

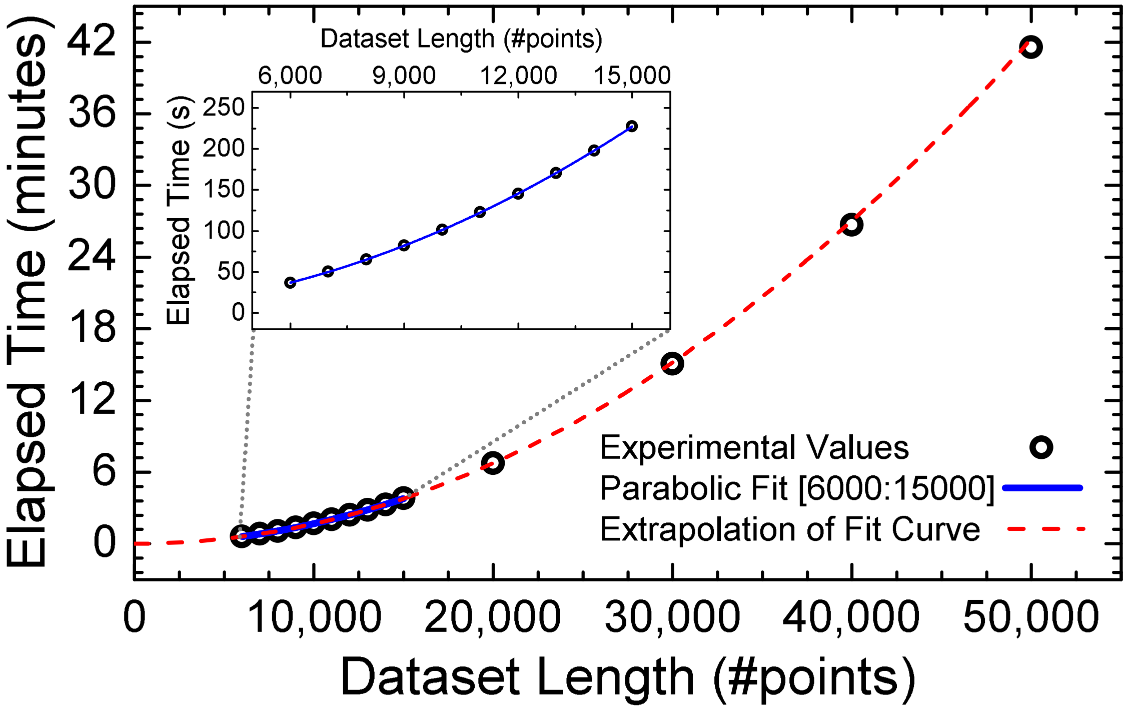


Figure 2. Elapsed times of the LBI algorithm for different dataset lengths. Following the complexity analysis of the algorithm, a parabolic curve is fitted to the first ten points of the curve (6000 to 15,000 points, marked in blue) and further extrapolated to the higher values. As can be seen, the curves fall exactly into the extrapolated curve, confirming the complexity analysis.

3. Results: The Profile-Split Methodology

The profile-split methodology attempted to solve the non-linear scaling of the elapsed algorithm time as a function of the length of the dataset. A necessary condition for the methodology to be successful was that the performance of the algorithm was not degraded as a result of the profile-split; in fact, overcoming the original performance results is a sought-after, but hardly achievable goal, since the established performance of the LBI is quite high [11].

Even though the idea behind it was quite simple—splitting the original profile into fixed-length sub-profiles and processing them individually—the constraint of upholding (or even further increasing) the algorithm's original performance created the necessity for a deeper analysis and understanding of the impact of the splitting methodology, which uncovered interesting effects related to the algorithm's procedure. Two such effects were identified as most prominent and, in the following subsections, were scrutinized. These were: the selection of the value of λ, which was intrinsically associated with the expected signal-to-noise ratio of a given sub-profile and changed dynamically from one sub-profile to another; and the impact of the chosen sub-profile length on both the algorithm's elapsed time and its performance.

3.1. λ Selection

An interesting feature of the TBD problem in the context of fiber fault detection is the fact that the dataset contained sparse trend breaks, but also, a linear slope component, which as previously mentioned, was associated with the attenuation experienced by light as it propagated in the fiber [23]. This, in turn, caused the signal-to-noise ratio to decrease progressively for distant positions (the reference for distance in this case was the measurement apparatus, i.e., the OTDR device).

For this reason, as the original profile was split into sub-profiles, the latter became very distinguished: sub-profiles associated with positions closer (in distance) to the measurement station (lower indices $n \in N$) exhibited higher SNR; while those associated with distant positions (higher indices $n \in N$) exhibited lower SNRs. The SNR, in turn, was an important aspect of the signal of interest from the point of view of the ℓ_1/ℓ_2 minimization, which was the core of the LBI procedure. In the ideal case, the absence of noise, the ℓ_2 portion of Equation (2) would clearly be zero, and the algorithm would have no problem adding candidates and increasing the ℓ_1 portion of Equation (2). As the noise was increased, the algorithm had to correctly balance the λ factor to achieve optimal signal denoising without flooding the selections with faults that were not really present in the dataset. In order to successfully process a low-SNR measurement, the λ factor could be increased, making the algorithm more tolerant to the squared error norm and more stringent in the selection of new candidates.

The observation that the major noise source in the system could be modeled as a Poisson random variable [11,14] was a crucial point in consistently adjusting the λ factor for different sub-profiles. This was because the SNR of a signal dominated by Poisson noise scaled with $\frac{1}{\sqrt{C}}$, where C represents the associated number of measurements, which, in turn, is related to the signal's energy. Furthermore, the ℓ_2 norm used to calculate the squared error between the estimated and original signals in the ℓ_1/ℓ_2 minimization essentially determined the norm of the error, which was defined as the square root of its total energy. Finally, then, by modulating λ according to the square root of a measurement's estimated SNR, the algorithm adapted to the difference in SNR between sub-profiles generated out of an original OTDR profile. In other words:

$$\lambda_{\text{sub-prof}} = \lambda_{\text{orig}} \sqrt{\text{SNR}_{\text{est}}}. \tag{4}$$

Through the analysis of this document, and for simplicity, SNR_{est} was calculated by taking the first position of a sub-profile and calculating its associated SNR (the function EstimatePoissonSNR in Algorithm 3). This prevented a more complex analysis of the SNR of a whole set of measurements and yielded high-quality results, as will be shown.

Estimation of a λ value that was adapted to a given SNR condition could, therefore, positively influence the estimation results and uphold the performance of denoising of the original profile. Unfortunately, there was no closed form for λ, as the characteristics of the datasets differed greatly; in [11], $\lambda = 0.5$ was used along with a hot-starting procedure and a so-called λ-grid, which allowed the algorithm to run iterations at different values of λ in a grid and select the one that best described the signal of interest according to an information criterion.

This procedure was also adopted here, where the Bayesian Information Criterion (BIC) [25] was employed. Even though the selection using a λ-grid required more iterations to be run and, therefore, more time until the algorithm converged, both the gain in timing with respect to the non-split version and the gain in performance due to the grid, as will be shown further on, advocated for its usage. The initial value of λ in the grid was then calculated based on the estimated SNR of the sub-profile. This procedure allowed both for each split-profile to be processed entirely independently (as no information from one was necessary for the other) and for the estimation performance to be equivalent to the case where the whole dataset was processed as one. The final procedure for λ selection in the split-profile LBI scenario is described in Algorithm 3.

It is important to note that, in case the dynamics of the system was different and the noise model was not Poissonian, the general principle of adjusting λ based on the SNR would still be valid. In other words, the fact that λ could be adapted according to the SNR of the signal of interest was a property of the ℓ_2/ℓ_1 minimization being considered (as λ weighed the ℓ_1 norm with respect to the ℓ_2 norm), and not of the trend break detection problem or even of the acquisition devices in OTDR systems. These considerations generalized this approach to other applications where splitting the original dataset was beneficial, provided an equally consistent noise model for estimating the SNR of a sub-profile was devised.

Algorithm 3 Split-profile with λ selection.

```
Require: $S_\lambda = \text{grid}\,(0.5 : \lambda_{\max})\,, \mathbf{y}_{\text{split}}$
Ensure: $\hat{\boldsymbol{\beta}}_{\text{best}}$
$\lambda^{\text{fact}} \leftarrow \textsc{EstimatePoissonSNR}(\max(\mathbf{y}_{\text{split}}))$
$S_\lambda \leftarrow \lambda^{\text{fact}} \cdot S_\lambda$
$L \leftarrow \alpha N$
$\hat{\boldsymbol{\beta}} \leftarrow \text{LBI}(\mathbf{y}, \lambda = S_\lambda(1), \boldsymbol{\beta}_{\text{start}} = \mathbf{0}, \mathbf{v}_{\text{start}} = \mathbf{0}, L)$
$b_{\text{best}} \leftarrow \text{BIC}(\hat{\boldsymbol{\beta}}, \mathbf{y})$
$\hat{\boldsymbol{\beta}}_{\text{best}} \leftarrow \hat{\boldsymbol{\beta}}_{\text{first}}$
for $\lambda \in S_\lambda$ do
    $\mathbf{v} \leftarrow \textsc{HotStart}(\hat{\boldsymbol{\beta}}, \lambda)$
    $\hat{\boldsymbol{\beta}} \leftarrow \text{LBI}(\mathbf{y}, \lambda, \boldsymbol{\beta}_{\text{start}} = \hat{\boldsymbol{\beta}}, \mathbf{v}_{\text{start}} = \mathbf{v}, 0.1L)$
    $\left[\hat{\boldsymbol{\beta}}_{\text{best}}, b_{\text{best}}\right] \leftarrow \text{BIC}(\hat{\boldsymbol{\beta}}, \mathbf{y}, \hat{\boldsymbol{\beta}}_{\text{best}}, b_{\text{best}})$
end for
```

In order to demonstrate the impact of the λ scaling derived from the estimated SNR in a given profile, as well as that of the λ selection using a grid, the relative incidences of true and false positives could be used; however, the so-called Matthews correlation coefficient, that can be calculated by:

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})\,(\text{TP} + \text{FN})\,(\text{TN} + \text{FP})\,(\text{TN} + \text{FN})}}, \tag{5}$$

allowed for a richer evaluation of the estimation quality. It qualifies the prediction of the break/no-break binary classification, where a $\text{MCC} = +1$ value stands for perfect prediction and $\text{MCC} = 0$ stands for random (or completely uncorrelated) predictions. Figure 3 presents the rates of true and false positives, as well as the MCC value output by two versions of the split-profile method (with and without the λ scaling and selection) and also by the original (non-split) LBI algorithm. For these results, the split-profile length was fixed as 4500 points, and the original profile length was set to 15,000 points; the dependence of the results on the choice of the split-profile length will be discussed in the next subsection. Furthermore, 1000 profiles were generated according to Algorithm 2 to increase the statistical relevance of the results.

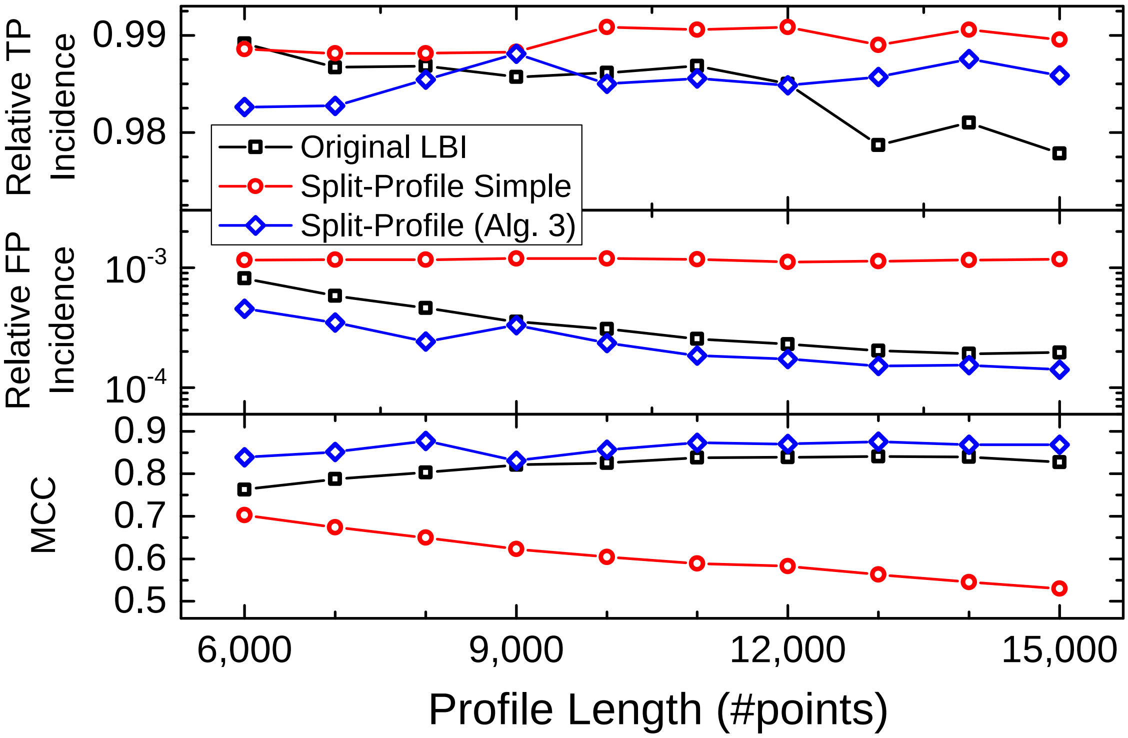


Figure 3. Performance comparison between three different versions of the analysis algorithm. Even though the differences of TPs and TFs might not seem extremely drastic for the split-profile simple, the broader characterization enabled by the MCC makes it clear that there is a deep performance drop if λ is not correctly chosen.

For all these and other results, the true and false positives were calculated by comparing the original fault positions in the $\hat{\beta}_{\text{ideal}}$ vector—used to create the datasets for analysis in Algorithm 2—with the ones estimated ($\hat{\beta}_{\text{best}}$) by three different versions of the analysis algorithm: the original (non-split) LBI, or Algorithm 1; the split-profile LBI without λ scaling and selection (split-profile simple); and the full version of the split-profile LBI, or Algorithm 3. The same procedure applied to true and false negatives.

The comparison between the three versions was fruitful because it allowed one to draw an important conclusion, which was the crux of the split-profile methodology: merely subdividing a dataset into a number of sub-profiles to be analyzed individually yielded a decrease in the performance; it was only through a consistent modification of the parameters (λ selection) that comparable—or even better—performances could be achieved.

3.2. Split-Profile Length Analysis

The choice of the split-profile length was, in contrast with the λ selection, a much less meaningful parameter of the split-profile methodology. However, it remained as a free parameter of this methodology and had to undergo scrutiny as well. As observed in Figure 4, the split-profile length imposed a minimum threshold that, if respected, maintained the quality of the results. There, the performance results for different split-profile lengths are depicted in terms of: the relative false positive and true positive incidence, as well as the MCC. For these results, 1000 datasets, each 15,000 points long, were sorted and constructed using Algorithm 2 and later analyzed with the split-profile LBI—Algorithm 3—using different split-profile lengths. At the same time, the same datasets were analyzed with the original (non-split) LBI algorithm, to serve as a control group; the results of this analysis represent the baselines—or the thresholds to be met—in the panels of Figure 4.

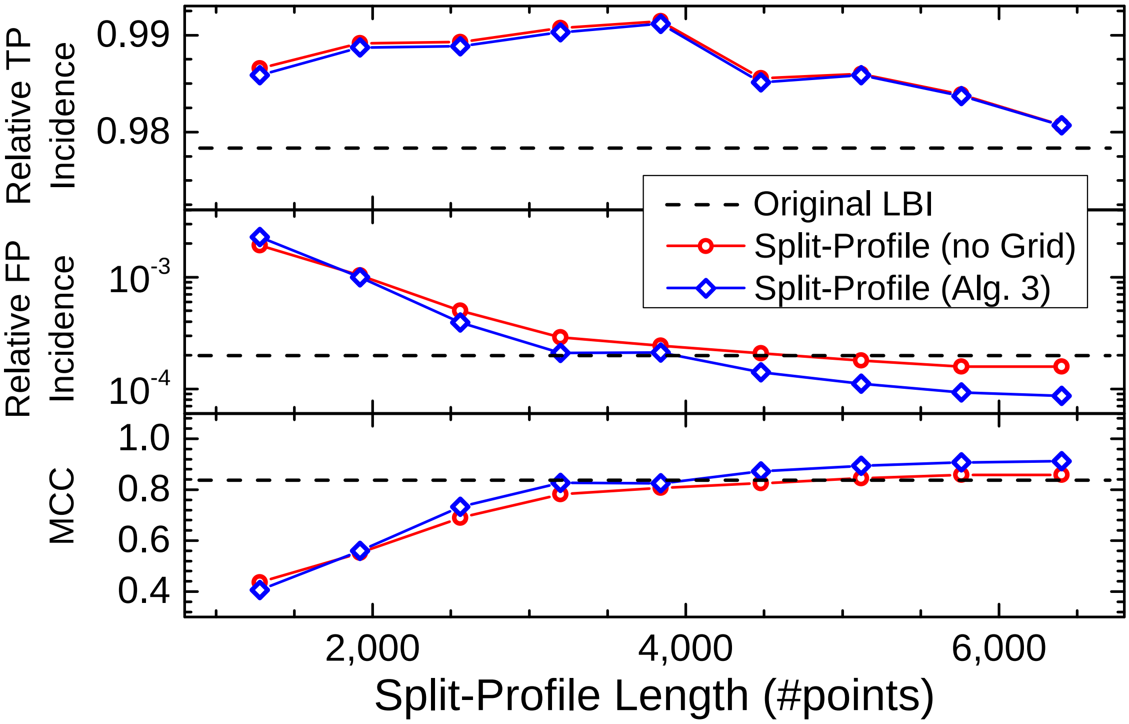


Figure 4. Performance of the split-profile methodology for varying split-profile lengths, in number of points. The original profiles, from which the sub-profiles were derived, contained 15,000 points. To increase the statistical validity of the analysis, 1000 datasets were generated according to Algorithm 2. The baselines—in black dashes—correspond to the results of the original (non-split) LBI. Above 4500 points, the performance recovered and slightly increased even when the λ-grid (Lines 7 through 10 in Algorithm 3) was not performed, as evidenced by a higher MCC value.

This effect had quite a discernible origin, since reducing the instance size analyzed by any sparse technique reduced its signal denoising capabilities. In summary, the results indicated that the minimum split-profile length that allowed the original performance to be met was $\sim$4500. Below this value, the split-profile LBI produced an overflow of false positives, and the MCC dropped significantly. As the length increased, however, the total time necessary for the algorithm to elapse also increased; therefore, the chosen operational value was set as the minimum such that the original performance was met. The choice of 4500 for the results of Figure 3 followed from this analysis.

3.3. Timing Results

The parameters and procedures that allowed the split-profile LBI to uphold the performance level set by the original LBI algorithm when applied to trend break detection in optical fiber profiles was determined with a thorough analysis in the previous section. These were the λ selection and scaling, the initial value of $\lambda = 0.5$ in the grid, and the split-profile length of $\sim$4500 points.

Finally, then, it was left to determine the impact of the split-profile methodology on the elapsed time of the algorithm, which was the driving motivation behind its development. For this analysis, the results of Figure 2 were used as a reference. Furthermore, two different timing curves are presented in Figure 5: the full processing time of Algorithm 3, including the extra necessary iterations to perform the λ-grid selection; and the processing time without the grid, i.e., in case the algorithm was to be halted before Line 7 of Algorithm 3. This result was important to showcase the minimum relative impact that the extra iterations required by the grid had on the total processing time in contrast with the impact that they had on the overall performance of the split-profile LBI, as shown in Figure 3.

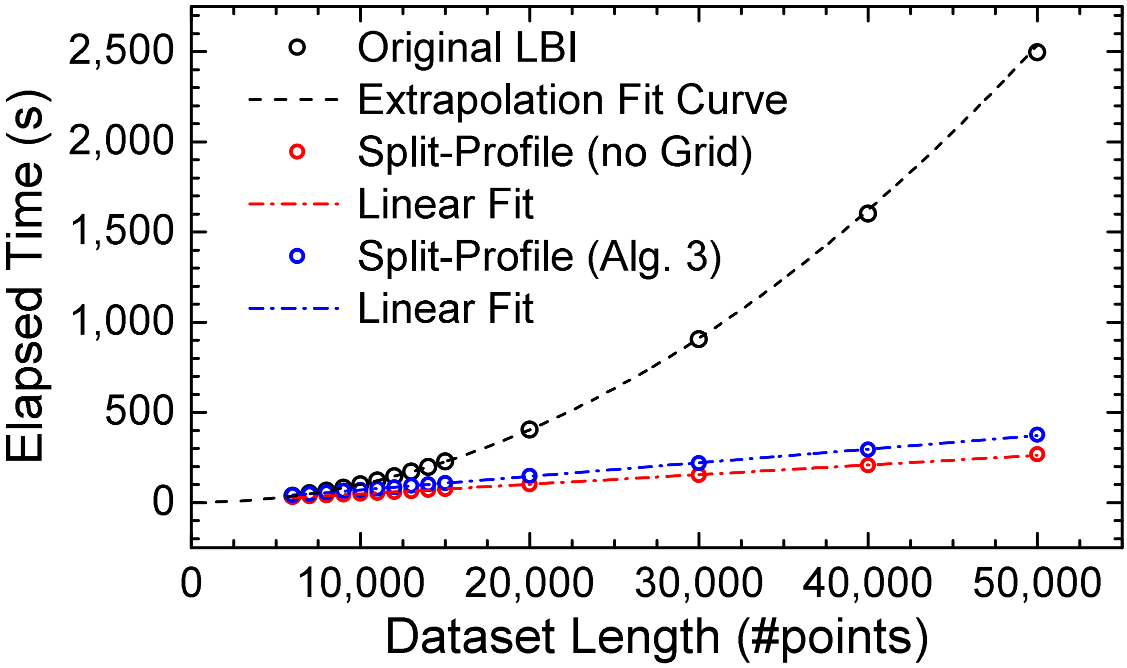


Figure 5. Final timing comparison between the original LBI and the corresponding split-profile LBI; two versions of the latter, with and without the λ-grid, are depicted. For shorter dataset lengths, the effect of the split-profile methodology is rather negligible since the original length is comparable to the individual split-profile length; however, as the dataset length increases above 20,000 points, a very clear distinction can be observed between the total elapsed times of the variants of the analysis algorithm.

By adjusting the length and the selection of λ, the performance of the split-profile LBI was upheld, and simultaneously, a dramatic decrease in the total processing time was observed. In fact, as the results of Figure 5 showed, once the length was fixed, the time scaled linearly with the length of the dataset being processed instead of quadratically: as the split length was fixed, the overall processing time was comprised of the time required for processing one split-profile multiplied by the number of total splits plus a processing overhead. The results of the original LBI algorithm are reproduced from Figure 2—in black along with its parabolic fit—for reference. The results of the complete split-profile LBI—Algorithm 3—are shown in blue, which represent a minor increase with respect to those in red, which correspond to not performing the λ-grid. For both these results, a linear fit was used, with an associated R^2 value of 0.999 and 0.998 when considering and not considering the grid, respectively. As in the analysis of the timing of the original LBI, the linear fit was performed, for the split-profile LBI, only within the first points, in the range $[6000 : 15,000]$, and then extrapolated to the other points, showing that the fit was indeed consistent as the length of the dataset increased.

4. Discussion: Parallelization of Profile-Split in Dedicated Hardware

The results of Figure 5, even though quite striking, were acquired after running the algorithm on a workstation (Intel Xeon CPU e5-2690 v4 at 2.6 GHz and 512 GB RAM). As set forth in [12], however, the core of the LBI algorithm could benefit immensely from hardware-dedicated implementation. There, individual Block RAMs (BRAMs) were structured such that data from each iteration could be fed in parallel to an arithmetical unit, greatly expediting the procedure. In fact, the total number of clock cycles necessary for the algorithm to elapse given the total number of iterations and the amount of available memory structures (BRAMs) in such a dedicated hardware was analytically determined and confirmed for commercial FPGAs [12] and was as follows:

$$C\left(N, M, \alpha, F\right) = F + \sum_{i=1}^{L} \left[3 \left(\left\lceil \frac{\left(\left(i-1\right)\%N\right)+1}{M} \right\rceil + 2 \right) + \lceil \log_2 M \rceil \right], \tag{6}$$

where the operator $\lceil\cdot\rceil$ denotes the ceiling operation, N is the instance size, L is the total number of iterations ($L = \alpha N$, M is the number of available BRAMS, and F is an (overall negligible) offset associated with initialization and control instructions ($F = 22$) [12]. By associating this value with the maximum achievable clock frequency in the target FPGA, the processing time could be determined, i.e.,

$$T_{\text{proc}} = \frac{C(N, M, \alpha, F)}{f_{\max}}. \tag{7}$$

In a profile-split scenario, pre-determined stretches of data of fixed size ($N{\sim}4500$ according to Section 3) were processed individually. The free-parameters of Equation (6) were, thus, reduced to α, M, and $f_{\max}$. The first, albeit a choice of the operator, was shown to yield highly accurate results at values $\alpha = 450$ or greater [12] and could also be fixed. The latter two were limited by the availability of resources on the FPGA chip and the optimality of the clock distribution inside the chip. In order, thus, to expand the analysis, two target FPGA chips, namely Altera's CYCLONE V and STRATIX V, were selected as potential platforms for the implementation of the parallel split-profile LBI. Making use of the hardware design established in [12] for an FPGA-embedded LBI unit, the resource usage and maximum estimated clock frequency for different unit sizes were determined and are presented in Figure 6. Here, the unit size was determined by the number of BRAMs included in the design.

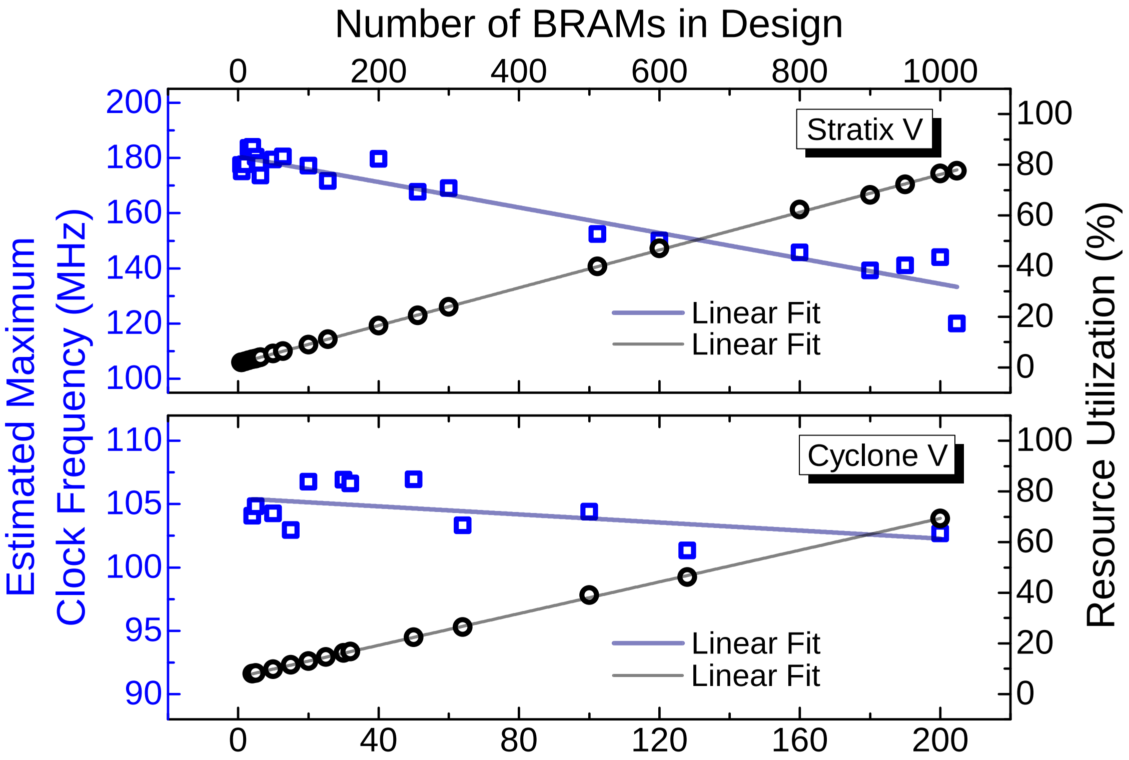


Figure 6. Resource and timing analysis for two FPGA chips of different sizes: the smaller CYCLONE V; and the mid-range STRATIX V. These results correspond to the optimal place-and-route procedure of commercially available hardware-design software, namely the QUARTUS PRIME software.

Simple linear regressions applied to each of the datasets—also depicted in Figure 6—enabled one to estimate, given the number of BRAMs in the design, both the resource usage and the maximum clock frequency for the target FPGAs. Based on these, operation conditions for different hardware options could be determined. Here, three such options were considered: the original LBI hardware, consisting of a single unit that utilized the chip's full resource capacity and processed the whole dataset without splitting; a sequential split-profile implementation, where a single unit making use of the chip's full resource capacity processed individual sub-profiles one after the other; and the simultaneous split-profile implementation, where multiple units, each making use of a portion of the chip's full resource capacity, processed individual sub-profiles at the same time.

The objective of the comparative analysis between the three options was to gauge: the speed-up level of the split-profile methodology as opposed to the original LBI in a hardware scenario; the minimum processing time achieved by the split-profile LBI; and finally, the tradeoff between simultaneous processing and full resource utilization. In order to perform the analysis correctly, whose results are depicted in Figure 7, the following practical observations are due:

- The analysis was restricted to the STRATIX V, since the CYCLONE V did not offer enough resources for a fruitful analysis for a large range of profile lengths.
- Due to the fact that the difference in processing times between the split-profile and the original LBI was insignificant for datasets containing fewer than 20,000 points—according to the analysis of Figure 5—this was set as the starting point of the analysis.
- The goal was to observe the trend of the total processing time curve for the three hardware options (original LBI, sequential split, simultaneous split) as the dataset length increased. Therefore, up to 10^7 points were considered for the analysis.
- Due to the data storage architecture and assuming a 20 bit long word BRAM with 1024 positions, each BRAM had a data capacity of 333 points [12].

- The BRAM storage capacity could be combined with the stipulated size of the profile-split ($N_{\text{split}} = 4480$) in order to determine the minimum number of BRAMs assigned to each unit (14).
- The curve that associated the resource usage in the FPGA to the number of instantiated BRAMs (Figure 6) could be used to determine the maximum number of individual split-profile units that could be simultaneously instantiated in the STRATIX V (38, with 98.08% resource usage) and, in turn, the maximum number of data points that could be stored ($38 \cdot 4480 = 170{,}240$, highlighted in Figure 7).
- The curve that associated the resource usage in the STRATIX V to the number of instantiated BRAMs (Figure 6) could be used to determine the maximum number of BRAMs that could be used for the original LBI implementation (1392, with 99.06% resource usage) and, in turn, the maximum number of data points that could be stored in the FPGA (333 × 1392 = 463,536, highlighted in Figure 7).
- If the dataset to be processed was longer than the maximum storage capacity of the hardware, full processing would not be possible in a single step, requiring the data points to be processed in batches.
- Processing data in batches required extra time necessary for loading/unloading of data to be taken into account. According to [12], this procedure took one clock cycle per data point and, although negligible, was included for completeness.
- Provided the sub-profile segmentation was respected through batch processing, no further care was needed in a profile-split scenario.
- In the original LBI implementation, batch processing required an additional λ selection step (as was necessary in the split-profile case), otherwise the performance would decrease drastically. The procedure described in Section 3 could be employed such that the original performance was maintained.

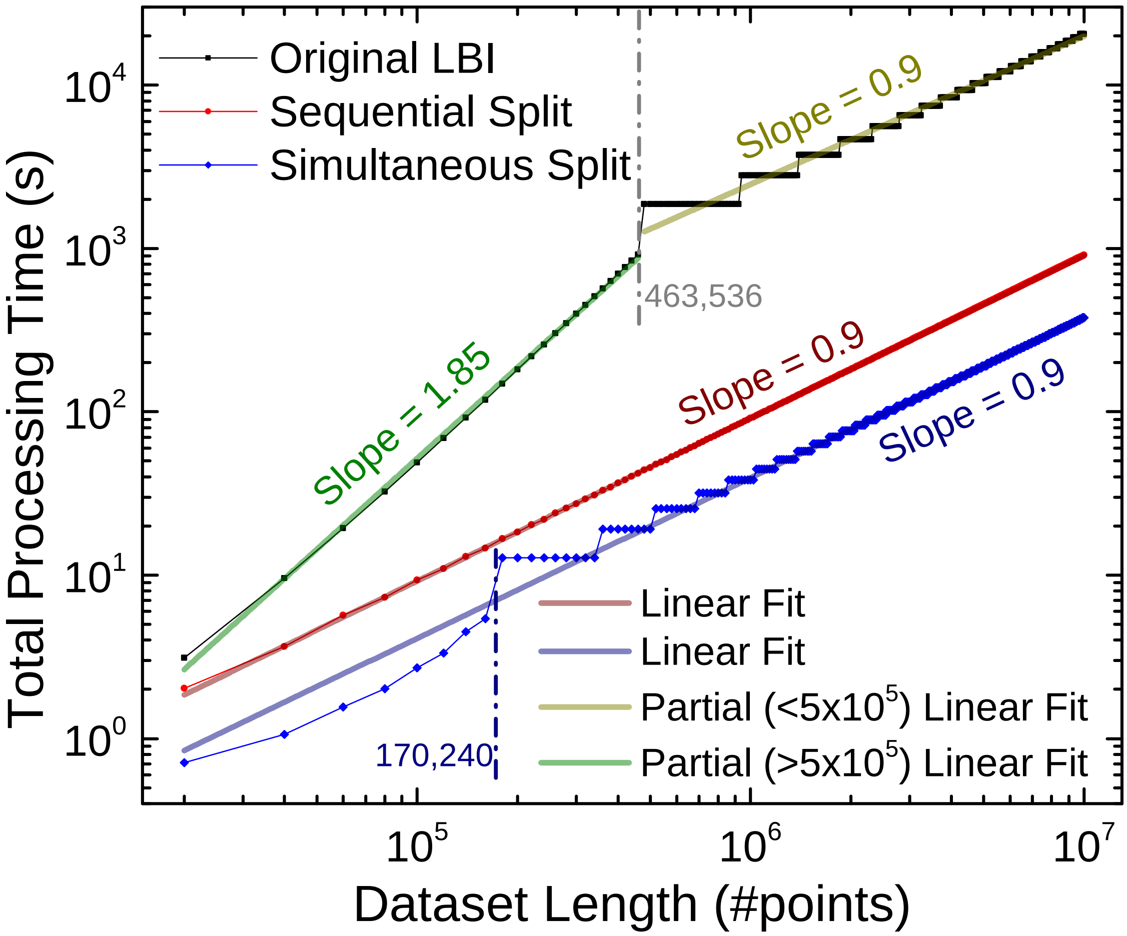


Figure 7. Total processing time as a function of the dataset length for three possible hardware implementations in the STRATIX V. The results are displayed in LOG-LOG scale for ease of visualization. Striking features, discussed in the text, are: the slope break of the original LBI curve, which corresponds to the transition to batch processing; the equivalence between the asymptotic slopes of all three curves; and the dominance of the simultaneous split-profile hardware in terms of processing time over the other options.

The most straightforward observation that could be made from Figure 7 is that the simultaneous split-profile, which made use of multiple units of the split-profile LBI instantiated in the same hardware, outperformed the other solutions. The sequential split-profile hardware, when implemented in the STRATIX V, exhibited a ∼3-fold asymptotic time increase with respect to the simultaneous split-profile hardware, still figuring as an excellent candidate for implementation in smaller FPGA chips (such as the CYCLONE V), where instantiating a single unit was beneficial. Furthermore, since the software version of the split-profile LBI used to extract the results of Figure 5 also performed sequential processing of the sub-profiles, the timing comparison between this and the sequential hardware could be made on the same grounds: while the software version required close to 300 s to process a 50,000 point long dataset, the sequential hardware elapsed in about 1.1 s, a close to 300-fold reduction.

Quite noticeable as well in Figure 7 are the discrete jumps taken by the black (original LBI) and blue (simultaneous split LBI) curves. As commented before, in case the dataset contained more points than could be stored by a particular hardware, batch processing was required. Whenever the number of necessary batches increased, the lines performed a jump. Above the marked value of 170,240, for instance, the blue curve jumps since the number of necessary batches is two; it then remains stationary since, even though the dataset length is increasing, only two batches are necessary. When the dataset is long enough such that three batches are necessary, it jumps again, and so forth.

The above described behavior is also observed in the black curve, although for this curve, a slope break is also observed, which must be discussed. For small dataset lengths, the original LBI processed the data without batches and, therefore, followed the quadratic complexity curve that was already been discussed in Section 2 and Figure 5. The fact that the slope of the timing curve in LOG-LOG scale was close to two was a consequence of its quadratic behavior. When the dataset increased such that batch-processing became necessary (above the marked value of 463,536), the batching had the effect of splitting the original profile into smaller sub-profiles, thus artificially inducing a split-profile character. In the asymptotic regime, the slope then acquired the characteristics of the split-profile curves (blue and red) with a LOG-LOG slope close to one, indicating a linear increase of the total processing time.

5. Materials and Methods

To evaluate the capabilities of the proposed architecture, the following methods were used: a commercially available state-of-the-art synthesis software (Intel Quartus Prime) was used to evaluate the maximum clock frequency and the device occupation of the design for different sizes (i.e., the number of used block RAMs in parallel).

6. Conclusions

The profile-split methodology ensured comparable algorithm performance while overcoming the quadratic scaling of the elapsed time, which became a severe hindrance for long instance sizes. As it stands, no information from any of the splits was necessary so that another split could be processed, and thus, simultaneous processing of each of the split profiles was a possibility. The results from Section 3 stipulated optimized parameters for individual hardware units implementing Algorithm 3, in particular with respect to the size, or number of data points, assigned to each of these units. This, in turn, enabled an analysis, in Section 4, of the impact of implementing multiple units that simultaneously processed data inside an embedded unit, such as an FPGA, with limited resources in terms of memory and logical elements. The limited resources created a constraint on the number of total parallel units that could be instantiated and put the advantages of this methodology to a more stringent test. The processing times of a few seconds for datasets as long as 100,000 points associated with an MCC value of 0.84 placed the hardware-embedded split-profile linearized Bregman iterations algorithm as the candidate of choice for trend break detection problems.

Author Contributions: All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: Financial support from Brazilian agency CNPq is acknowledged by F. Calliari. This work was supported by the COMET-K2 "Center for Symbiotic Mechatronics" of the Linz Center of Mechatronics (LCM) funded by the Austrian federal government and the federal state of Upper Austria.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FPGA	Field Programmable Gate Array
LBI	Linearized Bregman Iterations
TBD	Trend Break Detection
TP	True Positives
TF	False Positives
TN	True Negatives
FN	False Negatives
MCC	Matthews Correlation Coefficient
SNR	Signal-to-Noise Ratio
OTDR	Optical Time Domain Reflectometry
BIC	Bayesian Information Criterion

References

1. Yin, W.; Osher, S.; Goldfarb, D.; Darbon, J. Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing. *SIAM J. Imaging Sci.* **2008**, *1*, 143–168. [CrossRef]
2. Cai, J.F.; Osher, S.; Shen, Z. Linearized Bregman iterations for compressed sensing. *Math. Comput.* **2009**, *78*, 1515–1536. [CrossRef]
3. Cai, J.F.; Osher, S.; Shen, Z. Linearized Bregman iterations for frame-based image deblurring. *SIAM J. Imaging Sci.* **2009**, *2*, 226–252. [CrossRef]
4. Osher, S.; Mao, Y.; Dong, B.; Yin, W. Fast linearized Bregman iteration for compressive sensing and sparse denoising. *Commun. Math. Sci.* **2010**, *8*, 93–111.
5. Goldstein, T.; Osher, S. The split Bregman method for L1-regularized problems. *SIAM J. Imaging Sci.* **2009**, *2*, 323–343. [CrossRef]
6. Choi, K.; Fahimian, B.P.; Li, T.; Suh, T.S.; Lei, X. Enhancement of four-dimensional cone-beam computed tomography by compressed sensing with Bregman iteration. *J. X-ray Sci. Technol.* **2013**, *21*, 177–192. [CrossRef]
7. Lorenz, D.A.; Wenger, S.; Schöpfer, F.; Magnor, M. A sparse Kaczmarz solver and a linearized Bregman method for online compressed sensing. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 1347–1351.
8. Lunglmayr, M.; Huemer, M. Efficient linearized Bregman iteration for sparse adaptive filters and Kaczmarz solvers. In Proceedings of the 2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM), Rio de Janerio, Brazil, 10–13 July2016; pp. 1–5.
9. Riley, W.J. Algorithms for frequency jump detection. *Metrologia* **2008**, *45*, S154–S161. [CrossRef]
10. Babcock, H.P.; Moffitt, J.R.; Cao, Y.; Zhuang, X. Fast compressed sensing analysis for super-resolution imaging using L1-homotopy. *Opt. Express* **2013**, *21*, 28583–28596. [CrossRef]
11. Lunglmayr, M.; Amaral, G.C. Linearized Bregman Iterations for Automatic Optical Fiber Fault Analysis. *IEEE Trans. Instrum. Meas.* **2018**, *68*, 3699–3711. [CrossRef]
12. Calliari, F.; Amaral, G.C.; Lunglmayr, M. FPGA-Embedded Linearized Bregman Iterations Algorithm for Trend Break Detection. *arXiv* **2019**, arXiv:1902.06003.
13. Amaral, G.C.; Garcia, J.D.; Herrera, L.E.; Temporao, G.P.; Urban, P.J.; von der Weid, J.P. Automatic fault detection in wdm-pon with tunable photon counting otdr. *J. Light. Technol.* **2015**, *33*, 5025–5031. [CrossRef]
14. Von der Weid, J.P.; Souto, M.H.; Garcia, J.D.; Amaral, G.C. Adaptive filter for automatic identification of multiple faults in a noisy otdr profile. *J. Light. Technol.* **2016**, *34*, 3418–3424. [CrossRef]
15. Urban, P.J.; Vall-Llosera, G.; Medeiros, E.; Dahlfort, S. Fiber plant manager: An OTDR-and OTM-based PON monitoring system. *IEEE Commun. Mag.* **2013**, *51*, S9–S15. [CrossRef]
16. Hornsteiner, A. Fiber Optic Technology Trends in Data Transmission: Digitalization of data advance the need for constant upgrading of data networks. *Opt. Photonik* **2017**, *12*, 20–24. [CrossRef]
17. Candès, E.; Romberg, J.; Tao, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory* **2006**, *52*, 489–509. [CrossRef]
18. Han, P.Z.; Li, P.H.; Yin, P.W. *Compressive Sensing for Wireless Networks*; Cambridge University Press: Cambridge, MA, USA, 2013.
19. Calliari, F.; Herrera, L.E.; von der Weid, J.P.; Amaral, G.C. High-dynamic and high-resolution automatic photon counting OTDR for optical fiber network monitoring. In Proceedings of the 6th International Conference on Photonics, Optics and Laser Technology, Funchal, Portugal, 25–27 January 2018; Volume 1, pp. 82–90.
20. Bühlmann, P.; Rütimann, P.; van de Geer, S.; Zhang, C.H. Correlated variables in regression: Clustering and sparse estimation. *J. Stat. Plan. Inference* **2013**, *143*, 1835–1858. [CrossRef]
21. Bertsimas, D.; King, A.; Mazumder, R. Best subset selection via a modern optimization lens. *Ann. Stat.* **2016**, *44*, 813–852. [CrossRef]
22. Saavedra, R.; Tovar, P.; Amaral, G.C.; Fanzeres, B. Full Optical Fiber Link Characterization With the BSS-Lasso. *IEEE Trans. Instrum. Meas.* **2018**, *68*, 4162–4174. [CrossRef]
23. Barnoski, M.; Rourke, M.; Jensen, S.; Melville, R. Optical time domain reflectometer. *Appl. Opt.* **1977**, *16*, 2375–2379. [CrossRef]

24. Amaral, G.C.; Herrera, L.E.; Vitoreti, D.; Temporão, G.P.; Urban, P.J.; der von Weid, J.P. WDM-PON monitoring with tunable photon counting OTDR. *IEEE Photonics Technol. Lett.* **2014**, *26*, 1279–1282. [CrossRef]

25. Schwarz, G. Estimating the dimension of a model. *Ann. Stat.* **1978**, *6*, 461–464. [CrossRef]