

Linearized Bregman Iterations for Automatic Optical Fiber Fault Analysis

Lunglmayr, Michael; Amaral, Gustavo C.

DOI

[10.1109/TIM.2018.2882258](https://doi.org/10.1109/TIM.2018.2882258)

Publication date

2018

Document Version

Accepted author manuscript

Published in

IEEE Transactions on Instrumentation and Measurement

Citation (APA)

Lunglmayr, M., & Amaral, G. C. (2018). Linearized Bregman Iterations for Automatic Optical Fiber Fault Analysis. *IEEE Transactions on Instrumentation and Measurement*.
<https://doi.org/10.1109/TIM.2018.2882258>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Linearized Bregman Iterations for Automatic Optical Fiber Fault Analysis

Michael Lunglmayr *Member, IEEE*, and Gustavo C. Amaral

Abstract

Supervision of the physical layer of optical networks is an extremely relevant subject. To detect fiber faults, single-ended solutions such as the Optical Time Domain Reflectometer (OTDR) allow for precise measurements of fault profiles. Combining the OTDR with a signal processing approach for high-dimensional sparse parameter estimation allows for automated and reliable results in reduced time. In this work, a measurement system composed of a Photon-Counting OTDR data acquisition unit and a processing unit based on a Linearized Bregman Iterations algorithm for automatic fault finding is proposed. An in-depth comparative study of the proposed algorithm's fault-finding prowess in the presence of noise is presented. Characteristics such as sensitivity, specificity, processing time, and complexity, are analysed in simulated environments. Real-life measurements that are conducted using the Photon-Counting OTDR subsystem for data acquisition and the Linearized Bregman-based processing unit for automated data analysis demonstrated accurate results. It is concluded that the proposed measurement system is particularly well suited to the task of fault finding. The natural characteristic of the algorithm fosters embedding the solution in digital hardware, allowing for reduced costs and processing time.

Index Terms

Optical fiber measurements, Optical Time Domain Reflectometry, Fault location, Signal Processing, Signal Processing Algorithms

M. Lunglmayr is with the Institute of Signal Processing, Johannes Kepler University, Linz, Austria (e-mail: michael.lunglmayr@jku.at).

G. C. Amaral is with the Center for Telecommunications Studies, Pontifical Catholic University of Rio de Janeiro, RJ, Brazil and with QuTech and Kavli Institute of Nanoscience, Delft University of Technology, Delft, The Netherlands (e-mail: gustavo@opto.cetuc.puc-rio.br).

Copyright (c) 2018 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

I. INTRODUCTION

Optical fiber networks provide the backbone of today's information society. In the year 2014, already about 25 million kilometers of optical fiber had been installed worldwide, carrying over 80% of the long distance data traffic [1]. Even though an excellent media for broad band communications, the mechanical vulnerability of the optical fiber might jeopardize its transmission capacity: fiber bending, fiber breaking, imperfect fiber splices, and corrupted connectors are examples of faults that compromise the power budget in a fiber optical link and, eventually, render the link inoperable. Therefore, a means of detecting, locating, and evaluating the magnitude of such fault events is unavoidable and of great interest to network operators. Consequently, many physical layer monitoring techniques have been developed over the years, with the Optical Time Domain Reflectometry (OTDR) [2] and the Optical Frequency Domain Reflectometry (OFDR) [3] figuring as the most successful ones.

The OTDR, which measures the Rayleigh backscattered power of a high-intensity optical probing pulse to determine a fiber's profile [4], is typically used as the data acquisition system for fiber fault analysis. As the architectures and sizes of the optical networks change with the development of new technologies and increasing demand for higher data rates, the monitoring techniques must also evolve to remain compatible with the networks [5]. An example of the evolution of supervision techniques to meet requirements of new architectures is the wavelength-tunable OTDR, which allows for the supervision of wavelength division multiplexing (WDM) optical networks [6]. An example monitoring system for multiple fibers using an OTDR device is described in [7].

In this context, a natural evolution is the development of mathematical tools capable of automatically identifying the physical layer faults from the raw supervision data. This can eliminate the necessity of human resource allocation to this exhaustive process, which, in turn, reflects into minimized expenses and reduced downtime in the network caused by the scheduling of in-field repair units [8]. Apart from this operational advantage related to optical network management, location and estimation of losses in an OTDR profile can also be used for sensing applications: in [9], a system for measuring the mechanical displacement of a fiber by measuring the losses due to fiber bends has been presented; in [10], fiber losses sensed by a Photon-Counting OTDR are used to measure the pH value of a liquid. Naturally, a methodology able to locate and characterize faults in a fiber is general enough so that it can be adapted to either application.

The previously mentioned fiber faults, or any other fault in a fiber that causes the transmitted power to drop in an individual location, have been shown to manifest as trend breaks, or steps, in the digitized OTDR measurement data [11], [12]. Therefore, the location of a step can be directly associated to the location of the corresponding fiber fault. In other words, detecting faults in an OTDR profile is equivalent to detecting steps in the trend of the digitized data. If an algorithm can be responsible for such identification, automatic location of potential faults could be performed.

The reliable identification of steps in a data series is a matter that attracts substantial interest in areas such as: detection of disease outbreaks [13]; analysis of geophysical signals [14]; surveillance [15]; speech processing [16]; macroeconomics [17]; fraud detection [18]; or even in biological machines [19]. Although this problem is extremely simple in a noiseless scenario, it becomes quite challenging in the presence of noise. In Fig. 1, this feature is exemplified when a trend break is present under different signal-to-noise ratio (SNR) conditions. While in the first case (Fig. 1-a and -b) taking the discrete derivative of the original data and searching for a peak in the outcome is sufficient to precisely identify the fault, as the SNR decreases (Fig. 1-c, -d -e, and -f), the process yields poor results.

For this reason, and due to its relevance, signal processing techniques have been developed [20]–[24] to attack the problem of multiple, unknown, trend break detection in the presence of noise. Specifically in OTDR data acquisition, noise is present due to the power losses along the fiber that causes the measurement SNR to decrease and to the intrinsic scattering noise known as Coherent Rayleigh Noise (CRN) [4]. Also specific to the context of fiber monitoring is the fact that, from the point of view of a network operator, being able to identify and repair a fault before a user complaint is registered is extremely advantageous, the technique must be highly sensitive to the presence of a fault. On the other hand, triggering a repair unit to repair a fault that does not exist is economically harmful, so the technique must also exhibit high specificity in fault detection.

In a recent work [12], the *Adaptive ℓ_1 Filter*, an algorithm based on sparse estimation, was shown to outperform other techniques with respect to the aforementioned expected characteristics. Despite being an excellent method for trend break detection under noise, the Adaptive ℓ_1 Filter reveals some shortcomings with respect to an implementation in an embedded system for automated fiber measurements. It relies on a coordinate descent algorithm, with a greedy candidate selection process similar to the algorithm described in [25]. Even though the simplicity

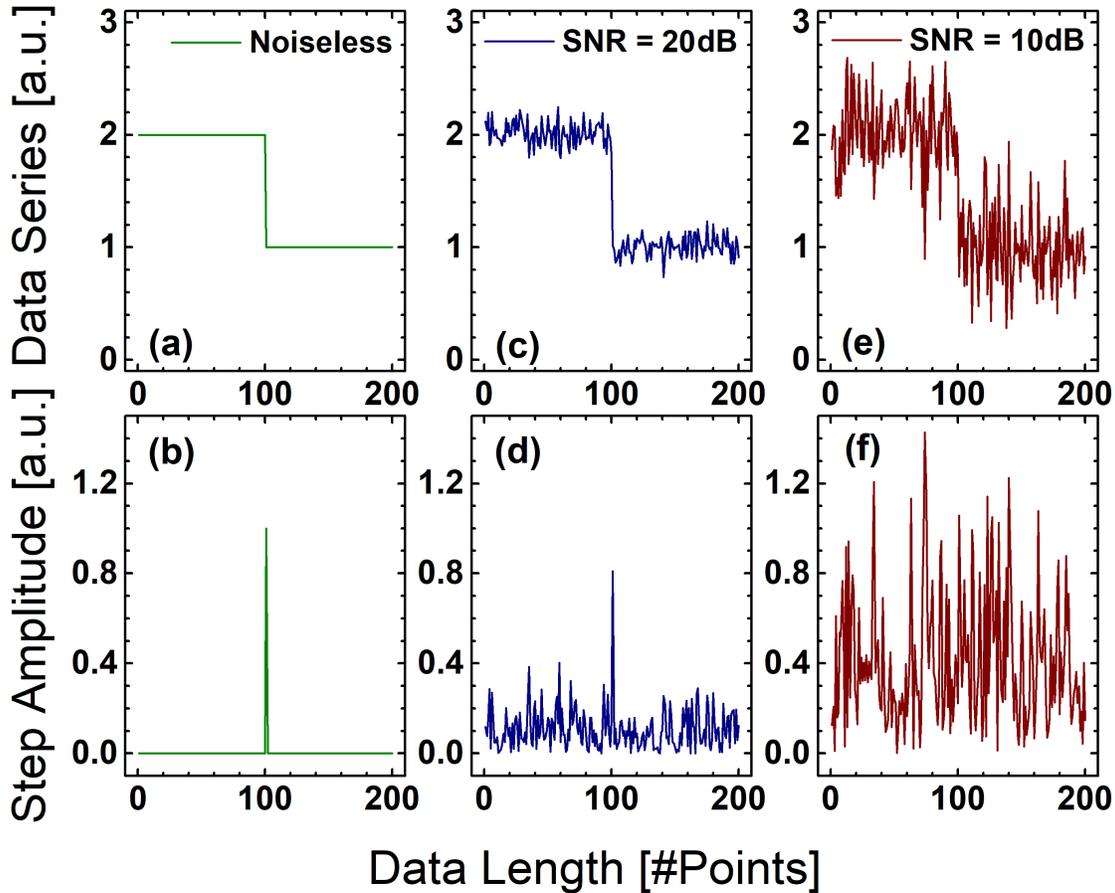


Fig. 1. Trend break detection using a discrete derivative and peak detection method. The result is compared under three signal-to-noise ratio environments: infinite; 20dB; and 10dB. The upper panels show the original data and the lower panels show the results, where a peak identifies the trend break.

of the model used for step detection enables low complexity estimation algorithms, the pre-processing step required for the Adaptive ℓ_1 Filter leads to multiplication-heavy algorithms, increasing its complexity. In addition, as the authors of [25] point out, to ensure stability, the algorithm requires calculating complete solutions paths with different levels of sparsity, which may compromise its processing time. On a different proposal [26], [27], detection of splice faults in an OTDR profile has been demonstrated with a digital signal processing approach based on a Gabor series representation and rank-1 matched subspace detection. Although the excellent detection capability demonstrated, the computational complexity of this approach is cubic in the number of measurement samples thus also limiting its implementation in an embedded measurement system for long fibers.

In the context of sparse estimation, Linearized Bregman Iterations (LBI) offer a low complexity and high precision approach for solving the combined ℓ_1/ℓ_2 minimization problem [28]–[33]. By adapting the LBI algorithm to take into account the characteristics of the trend break detection problem, a signal processing methodology was developed exhibiting better timing and accuracy when compared to state-of-the-art solutions, namely the *Adaptive ℓ_1 Filter*. In this paper, the Linearized Bregman Iterations framework is presented as a candidate for an automatic trend break detection technique. Through an in-depth analysis focused on the particular case of fault detection in noisy OTDR profiles, figures of merit such as sensitivity, specificity, and processing time have been drawn in simulated and real-life environments. For comparison analysis between the proposed algorithm and the Adaptive ℓ_1 Filter, both algorithms have been implemented in the Julia language. The conclusion is that the LBI algorithm is a promising candidate not only for this particular problem, but for the more general issue of trend break detection in noisy data. Furthermore, since the prospect of automated operation promotes the urge of reduced complexity so that a low-cost embedded processing unit such as an FPGA can be responsible for real-time monitoring of the fiber links, this point has also been investigated and the proposed algorithm is shown to offer promising characteristics for FPGA embedding.

The paper introduces the basic concepts of OTDR operation and the mathematical model of the Linearized Bregman Iterations algorithm in Section II. Section III is responsible for highlighting the modifications on the original Linearized Bregman Iterations algorithm necessary to adapt it to the fault finding problem and thus creating a consistent parameter-free processing algorithm. With the background developed in the previous sections, a testbench experimental result is tested with the proposed algorithm. The results allow the development of a framework of extensive simulated profiles and the results both from real fiber measurements and simulated fiber profiles is detailed in Section IV. Finally, Section V concludes the work summarizing the findings as well as laying the path for future developments.

II. OTDR AND LBI - BASIC CONCEPTS

Optical Time Domain Reflectometry is a single-ended reflectometry measurement technique that consists of measuring the Rayleigh backscattered portion of a light pulse traversing the fiber as a function of time. The OTDR provides the fiber profile by associating the measured data to the pulse's round-trip time, which, combined with the knowledge of the fiber's refractive index and the speed of light, can be translated into distance. The fiber profile is, thus, a descending line in

logarithmic scale with a slope equal to the fiber’s attenuation coefficient [2], [4], [34]. Any event which causes optical power loss is interpreted accordingly, i.e., the aforementioned effects of fiber bending, fiber breaking, imperfect fiber splices, and corrupted connectors can be identified by an abrupt level shift in the signal profile. The goal would be to identify such level shifts and associate their positions and magnitudes to faults. The proposed automated measurement system that provides such results is depicted schematically in Fig. 2, where the Photon-Counting OTDR presented in [11] is used as the data acquisition subsystem. In Fig. 3, an example OTDR fiber profile acquired with this system is presented, indicating the most common effects associated to power loss and/or fiber faults. After the data acquisition step, the Linearized Bregman-based processing unit analyses the resulting OTDR profile and provides the event list to the user.

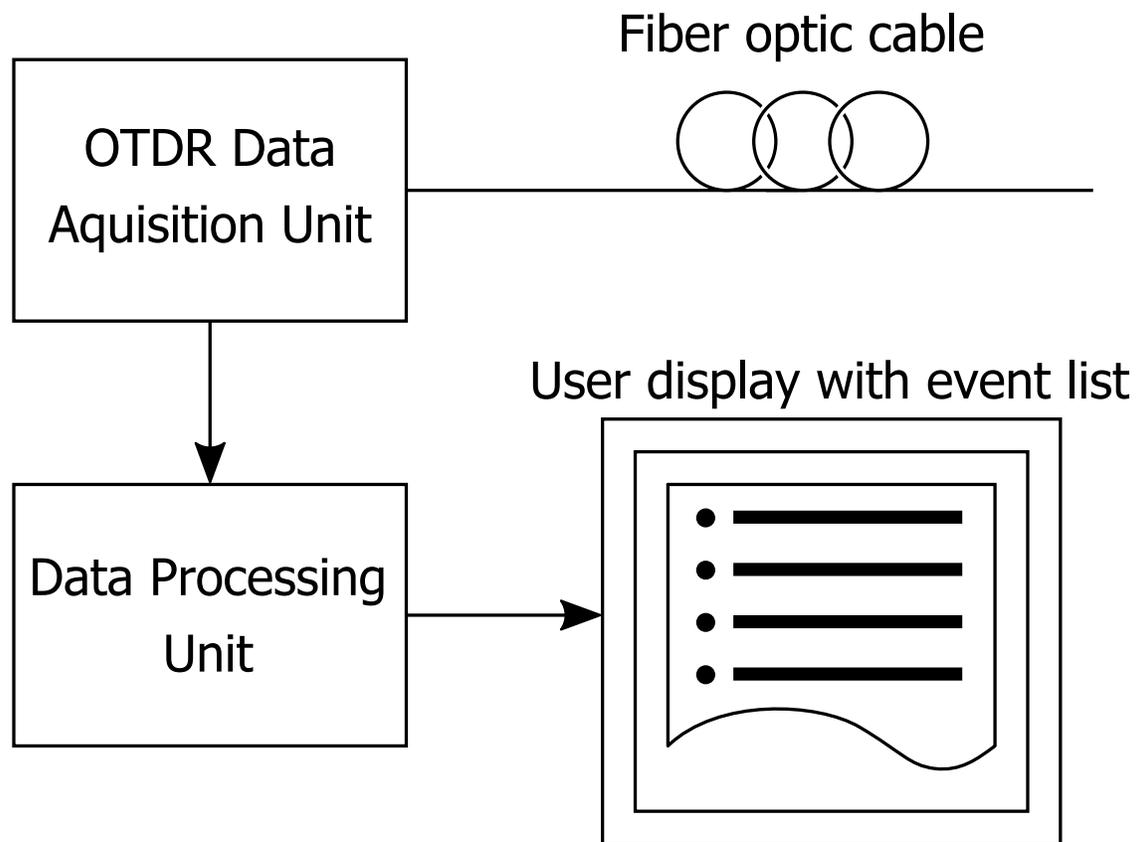


Fig. 2. Block diagram of the proposed measurement system with its desired output: an “event list” containing positions and magnitudes of events along the optical fiber link. The data acquisition subsystem represents the Photon-Counting OTDR of [11].

Given the OTDR fiber profile such as the one in Fig. 3, eventual power losses that could possibly compromise the link’s transmission capacity will be identified by trend breaks. Although some of the power losses are expected – for instance the splitters and connectors necessary for

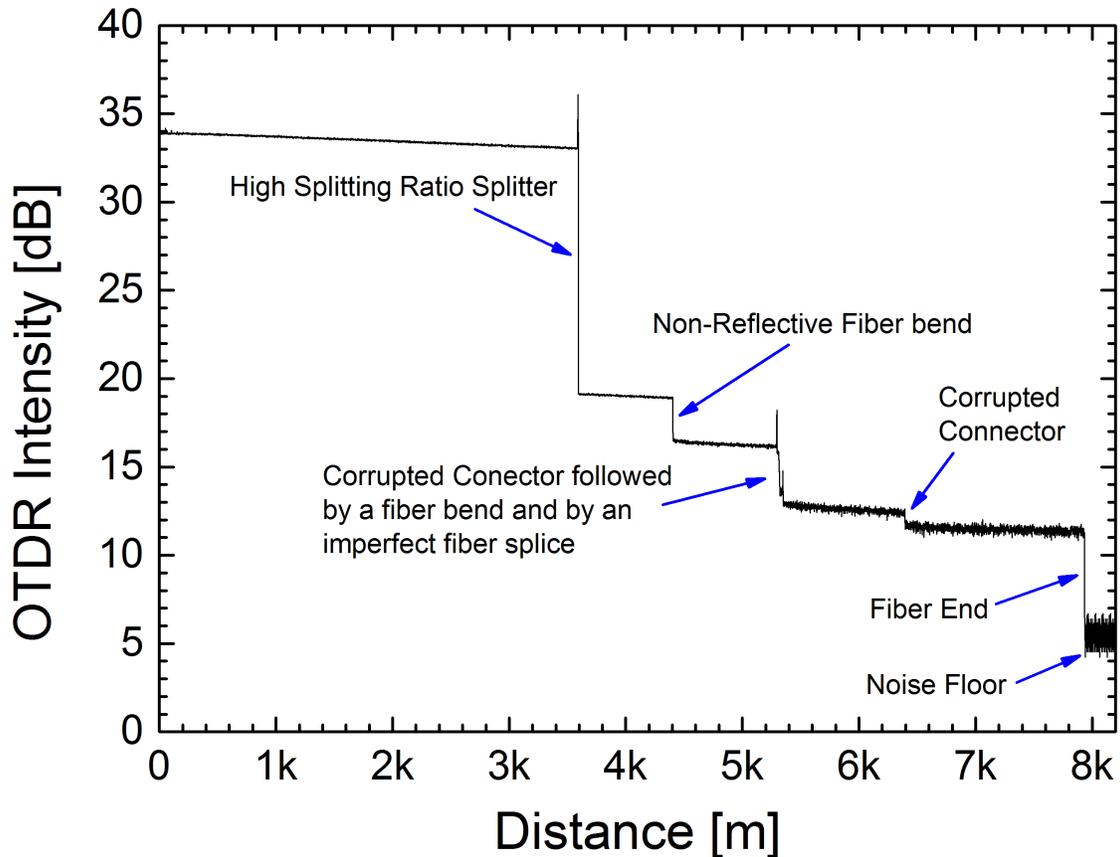


Fig. 3. Example of OTDR fiber profile highlighting the most common fiber fault causes. The profile was acquired with a tunable Photon-Counting OTDR setup, presented in detail in [11].

the network architecture – some are spurious or unexpected and often arise due to the mechanical fragility of the optical fiber [4]. The power balance of the optical link, which takes into account the input optical power and the detector margin for error-free reception, may be able to cope with some of these losses but, eventually, they might render the link inoperable. The task, therefore, is to rapidly detect the presence of such breaks with the highest accuracy possible (within the probing method's limitation). This way, scheduling of in-field repair units can be accelerated and managing costs of the network supervisor can be reduced as well as any down-times experienced by the network user.

It should be noted that the spatial resolution employed by the measuring technique will limit the achievable accuracy of an algorithm that processes the output digitized data. The OTDR, or any other conceivable fiber monitoring apparatus, albeit offering an excellent tool for fiber monitoring, poses a trade-off in terms of its measurable distance and resolution: in order to

investigate distant parts of the fiber, the probe pulse is enlarged – so to carry more optical power – and the spatial resolution is thus reduced. If a higher resolution is required, on the other hand, the probe pulse width is reduced at the expense of reduced reach. For that reason, a signal processing approach will not be able to locate a fault with a precision higher than the physical parameters of the measuring apparatus allow for. Accuracy for automatic fault detection should be interpreted as “as accurate as the physically available spatial resolution”. The signal processing approach deals with points in a data series and it is left to the measurement apparatus to link the digitized data points to a physical position or distance.

The concept behind the automatic identification of fault positions is to interpret the OTDR profile as a weighted sum of step functions and then find the steps and respective weights that best explain the original data series. The profile depicted in Fig. 3 clarifies this point, as, after each highlighted event, there is a sharp decrease in the measured power. Step functions are uniquely defined by their amplitude and the position beyond which they assume a non-zero value. This allows to cast an arbitrary profile, under this interpretation, in the following form:

$$y(z) = \alpha \sum_{i \in F} a_i u(z - z_i), \quad (1)$$

where $y(z)$ is the ideal (noise-free) fiber profile, z denotes the distance inside the fiber, F is the set of fault indices, a_i are the weights (or magnitudes) of the faults, and z_i its positions. The step function is represented by the Heaviside function $u(z)$ and the factor α stands for the linear slope of the fiber attenuation. For a measurement application, $y(z)$ is typically sampled equidistantly and the samples are combined into a vector \mathbf{y} .

Identification of the set F of fault indices is the goal of the fault finding problem, which can be assumed to be sparse in this particular application. Fig. 2 is an example of the sparse characteristic of the problem since the total number of possible fault positions is 8000 (each entry in the dataset, or the spatial resolution of the acquisition system, corresponds to one meter), but the size of the set F is 7. Indeed, from a practical point of view, it makes sense that the number of events should be much smaller than the points in the data series: a high number of fault events translates into high optical losses and, as commented above, the inoperability of the optical link. A remarkable point that arises from this reasoning is that, given the spatial resolution, one has access to all possible positions in the digitized data series where a trend break might appear. Thus, a naive approach would be to check all the possible combinations of step functions until the best approximation to the measured signal is found following (1). Such

a procedure, however, is known to belong to the class of the so-called NP-hard problems [35] and, as such, is computationally intractable. A relaxation of this extremely difficult problem is to use the ℓ_1 norm instead, which allows one to employ algorithms with manageable complexity that output sparse results for a broad range of practical problems [36].

To estimate the fault positions within a given OTDR profile, a dictionary of candidate vectors is provided, where the name dictionary is used to emphasize that candidate vectors do not have to be linearly independent [37]. There can be, as it is used in this work, even more candidate vectors than measurement values, in which case it is often referred to as an overcomplete dictionary [37]. The ℓ_1 minimization problem where the dictionary of candidates is provided is known as the basis pursuit problem which, in its matrix form, can be represented as:

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_1 \quad s.t. \quad \mathbf{A}\boldsymbol{\beta} = \mathbf{y}, \quad (2)$$

where \mathbf{A} is the dictionary, with each column of \mathbf{A} being a dictionary or candidate vector. $\boldsymbol{\beta}$ are the coefficients of the dictionary vectors, which represent the amount of contribution, or weight, of a candidate to the fiber profile data series \mathbf{y} .

Based on (1), (sampled) step functions shifted at all possible sample points will be included as candidate vectors in matrix \mathbf{A} . In addition, the estimate of the attenuation coefficient is taken into account by also including a single linear candidate with unitary slope into the dictionary (\mathbf{A} is described in more detail in Sect. III). Ultimately, estimation of the attenuation as well as the level shifts is the estimation of the coefficient vector $\boldsymbol{\beta}$ where the non-zero coefficients identify the step locations and, thus, the position of the faults in the profile.

Linearized Bregman Iterations (LBI) form a class of implementation-efficient algorithms that solve the combined ℓ_1/ℓ_2 problem of the form:

$$\min_{\boldsymbol{\beta}} \lambda \|\boldsymbol{\beta}\|_1 + \frac{1}{2\alpha} \|\boldsymbol{\beta}\|_2^2 \quad s.t. \quad \mathbf{A}\boldsymbol{\beta} = \mathbf{y}. \quad (3)$$

As e.g. shown in [28], [38], [39], Linearized Bregman- based algorithms also give meaningful results in the presence of noise, i.e., even in cases when $\mathbf{A}\boldsymbol{\beta} = \mathbf{y}$ does not have a solution. In such scenarios, it can be shown that the squared error norm of an estimate $\boldsymbol{\beta}^{(k)}$ at iteration k (details on the algorithm are given below) is reduced by following iterations as long as $\|\mathbf{A}\boldsymbol{\beta}^{(k)} - \mathbf{y}\|_2$ is higher than the noise level [28], [38]. An interesting observation is that, adding the ℓ_2 norm in the cost function above, leads to very low complexity solution algorithms [28], [32] that can be implemented very efficiently in digital hardware [40]. Furthermore, it can be noted that, with the

addition of the ℓ_2 norm term, a correct balancing of λ and α allows for the solution of the basis pursuit problem (2) within the framework of the LBI (3) in an efficient manner. For simplicity, we set $\alpha = 1$, leaving only the parameter λ to adjust the weight of the ℓ_1 versus the ℓ_2 norm.

Despite their low complexity, Linearized Bregman Iterations still require matrix vector multiplications. A further simplification, however, called Sparse Kaczmarz, only requires vector operations while still maintaining convergence [41]. The basic algorithm of Sparse Kaczmarz is shown in Algorithm 1, where the Kaczmarz simplification has been used: instead of the full matrix \mathbf{A} , only a single row of \mathbf{A} is used in a single iteration of the algorithm. Although this algorithm represents an adaptation of the original Linearized Bregman Iterations algorithm described in [28], in the following, references to the LBI algorithm should be interpreted as to the Kaczmarz variant. This will also prevent confusion between LBI-based Sparse Kaczmarz algorithms and other algorithms also called Sparse Kaczmarz, e.g., as described in [42].

Algorithm 1 Linearized Bregman-based Sparse Kaczmarz

```

1:  $\boldsymbol{\beta}^{(0)} \leftarrow \mathbf{0}$ 
2:  $\mathbf{v}^{(0)} \leftarrow \mathbf{0}$ 
3: for  $k = 1..N$  do
4:   for  $j = 1..p$  do
5:      $\beta_j^{(k)} \leftarrow \text{shrink}(v_j^{(k)}, \lambda)$ 
6:   end for
7:    $i \leftarrow ((k - 1) \bmod p) + 1$  ▷ cyclic re-use of rows of  $\mathbf{A}$ 
8:    $\mathbf{v}^{(k+1)} \leftarrow \mathbf{v}^{(k)} + \frac{1}{\|\mathbf{a}_i\|_2} \mathbf{a}_i (y_i - \mathbf{a}_i^T \boldsymbol{\beta}^{(k)})$ 
9: end for

```

The LBI-based Sparse Kaczmarz algorithm can be seen as a steepest descent approach using an approximate gradient derived from a single row of \mathbf{A} and a single measurement value (line 8) combined with a *sparsifying* operation (line 5), while the term $\frac{1}{\|\mathbf{a}_i\|_2}$ can be seen as the *step width* of the descent. The sparsifying operation is due to the definition of the shrink function: $\text{shrink}(v, \lambda) = \max(|v| - \lambda, 0) \cdot \text{sign}(v)$, that sets values of v with a magnitude smaller than λ to zero.

In general, a sensible relation when using the Linearized Bregman Iterations is, as commented above, the balance between the ℓ_1 and ℓ_2 norm translated by λ . When adapting the Sparse Kaczmarz algorithm to the OTDR fault detection problem, this parameter turned out to have

a much lower sensibility than it is observed for general ℓ_1/ℓ_2 estimation problems, as will be discussed in the following Section.

III. ANALYSING OTDR FIBER PROFILES WITH THE LINEARIZED BREGMAN ITERATIONS

The first, most crucial aspect of adapting the Bregman methodology to any basis pursuit problem, is the definition of the matrix \mathbf{A} , the dictionary of candidates that will compose the signal of interest as a linear combination. In the specific case of OTDR profiles, the candidates are step functions, which correspond to discontinuous descends of the profile magnitude. As described in Sect. II, due to an intrinsic attenuation of light propagation through optical fibers, the matrix \mathbf{A} must also account for a negative slope. The final form of \mathbf{A} is:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 2 & 1 & 1 & 0 & \cdots & 0 & 0 \\ 3 & 1 & 1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ p-2 & 1 & 1 & 1 & \cdots & 1 & 0 \\ p-1 & 1 & 1 & 1 & \cdots & 1 & 1 \end{bmatrix}, \quad (4)$$

with $n = p - 1$ rows. The structure of \mathbf{A} allows for a significant simplification that can be incorporated into the Sparse Kaczmarz algorithm, preventing the need to store and load this matrix, and is as follows.

When analysing a row \mathbf{a}_i^T of \mathbf{A} , one can see that this row consists of the number i as its first element followed by i one elements and, finally, by $p - i - 1$ zero elements. This allows to formulate the inner product calculation of $\mathbf{a}_i^T \boldsymbol{\beta}$ efficiently as

$$\mathbf{a}_i^T \boldsymbol{\beta} = i\beta_1 + \sum_{s=2}^{i+1} \beta_s, \quad (5)$$

as well as the calculation of the squared norm $\|\mathbf{a}_i\|_2^2$ as

$$\|\mathbf{a}_i\|_2^2 = i^2 + i. \quad (6)$$

Considering that this norm is used as step width in the LBI algorithm ($\frac{1}{\|\mathbf{a}_i\|_2^2} = \frac{1}{i^2+i}$), the squared part i^2 , originating from the first column, contributes most significantly to the norm. This means that the contribution of the first column scales down the step width, limiting the algorithm's convergence speed. For this reason, a scaling factor σ for the first column of \mathbf{A} is used in

$$\mathbf{a}_i^T \boldsymbol{\beta} = \sigma i \beta_1 + \sum_{s=2}^{i+1} \beta_s, \quad (7)$$

as well as in

$$\|\mathbf{a}_i\|_2^2 = \sigma^2 i^2 + i, \quad (8)$$

leading to larger step widths and faster convergence. One point to consider is that, with this scaling factor, the first element of the output of the algorithm is also scaled by σ . Since the algorithm counteracts this scaling and implicitly multiplies by $1/\sigma$ during the estimation process, the first element has to be multiplied by σ at the end of the algorithm for compensation. Considering a future hardware implementation, the value of σ was selected as a power of two, which allows for replacing the multiplication by a shift operation. Furthermore, for large values of the step width $\frac{1}{\|\mathbf{a}_i\|_2^2}$, the value of σ should be chosen as small as possible. However, a compromise between large step widths and the required numerical dynamic range, especially when considering a fixed point implementation, has to be found. For this reason $\sigma = 2^{-10} = 1/1024$ was selected allowing for a fast convergence while still keeping the required dynamic range manageable in a future hardware implementation.

These considerations allowed for the development of the following Algorithm 2. In order to clarify the steps of the algorithm, a flowchart accompanies the pseudo-code's main steps. This flowchart shows the simple iterative nature of the algorithm, consisting mainly of a successive application of the described Sparse Kaczmarz iteration: an approximate gradient step followed by the evaluation of the shrink function. The structure displayed in Algorithm 2 fosters a low complexity implementation (complexities of Algorithm 2 as well as of the Adaptive ℓ_1 Filter [12] are discussed and compared in Sect. IV-C).

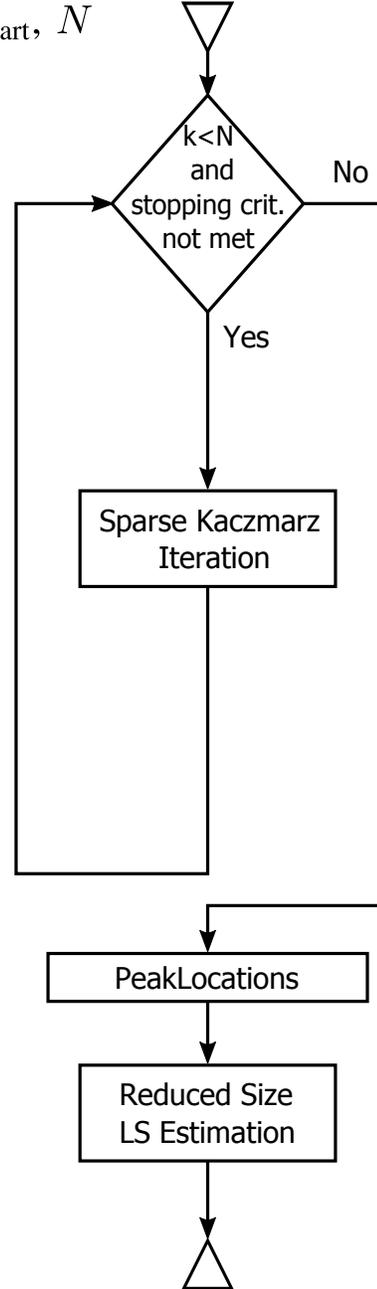
Algorithm 2 LBOTDR: Linearized Bregman Iteration for OTDR Fiber Profile Analysis

Input: Measurement vector \mathbf{y} , λ , $\boldsymbol{\beta}_{\text{start}}$, $\mathbf{v}_{\text{start}}$, N
Output: Estimated $\hat{\boldsymbol{\beta}}$, N_c

```

1:  $\boldsymbol{\beta}^{(0)} \leftarrow \boldsymbol{\beta}_{\text{start}}$ 
2:  $\mathbf{v}^{(0)} \leftarrow \mathbf{v}_{\text{start}}, k \leftarrow 1$ 
3: while  $k < N$  and stopping criterion not met do
4:   for  $j = 1..p$  do
5:      $\beta_j^{(k)} \leftarrow \text{shrink}(v_j^{(k)}, \lambda)$ 
6:   end for
7:    $i \leftarrow ((k - 1) \bmod m) + 1$ 
8:    $\|\mathbf{a}_i\|_2^2 \leftarrow \sigma^2 i^2 + i$ 
9:    $e \leftarrow (y_i - \sigma i \beta_1^{(k)} - \sum_{s=2}^{i+1} \beta_s^{(k)})$ 
10:   $d \leftarrow \frac{1}{\|\mathbf{a}_i\|_2^2} e$ 
11:   $\mathbf{v}_1^{(k+1)} \leftarrow \sigma i d$ 
12:  for  $j = 2..(i + 1)$  do
13:     $v_j^{(k+1)} \leftarrow v_j^{(k)} + d$ 
14:  end for
15:   $k \leftarrow k + 1$ 
16: end while
17:  $\beta_1^{(N)} \leftarrow \sigma \beta_1^{(N)}$ 
18:  $\mathbf{l} \leftarrow \text{PEAKLOCATIONS}(\boldsymbol{\beta}^{(N)}, t_p)$ 
19:  $\mathbf{A}' \leftarrow \mathbf{A}(\mathbf{l}, :)$ 
20:  $\boldsymbol{\beta}' \leftarrow (\mathbf{A}'^T \mathbf{A}')^{-1} \mathbf{A}' \mathbf{y}$ 
21:  $\hat{\boldsymbol{\beta}} \leftarrow \mathbf{0}$ 
22: for  $j = 1..\text{length of } \mathbf{l}$  do
23:    $\hat{\beta}_{l_j} \leftarrow \beta'_j$ 
24: end for
25:  $N_c \leftarrow k$ 

```



Due to the structure of matrix \mathbf{A} , the operations that require access to its rows can be directly encoded in the algorithm without the necessity to load \mathbf{A} , allowing Algorithm 2 to perform the LBI with reduced complexity. The function `PEAKLOCATIONS`, as shown in Algorithm 3, returns a vector containing those positions where the sign of the difference between consecutive positions changes, which is used to detect local maxima in the estimated $\boldsymbol{\beta}$. The detection of

peaks works, in Algorithm 2, as an efficiency booster since it lessens the sensitivity of the algorithms to the parameter λ . In addition, as Linearized Bregman-based algorithms show the tendency to require more iterations when using larger λ values [33], [39], this allows for fast convergence while still maintaining sparse estimation results. After detecting the peaks, their respective positions are used to select the corresponding columns of \mathbf{A} to form matrix \mathbf{A}' . Then, a least squares estimation is performed in the reduced subspace (spanned by the peak positions) of the larger column space of \mathbf{A} . We describe the stopping criterion of Algorithm 2 in the next subsection.

Algorithm 3 PeakLocations

Input: β, t_p

Output: Vector of peak locations l

```

1:  $S \leftarrow \emptyset$ 
2: for  $j = 2..(p - 1)$  do
3:   if  $\text{sign}(\beta(j + 1) - \beta(j)) \neq \text{sign}(\beta(j) - \beta(j - 1))$  then
4:      $S \leftarrow S \cup \{j\}$ 
5:   end if
6: end for
7: Remove all indices  $i$  from  $S$  where  $\beta_i < t_p$ 
8:  $S \cup \{1\}$  ▷ always add slope coefficient
9: form vector  $l$  of all elements of  $S$  in ascending order

```

In PEAKLOCATIONS, the first element of the estimated β is always included in the output ensuring that the analysis always considers the fiber profile's slope represented by the first column of \mathbf{A} . This phenomenological observation corresponds to the fact that a non-zero slope must always be present since no fiber exhibits zero attenuation. Furthermore, in the implementation with the Julia language, the least squares estimation of line 20 in Algorithm 2 is performed via a matrix inverse. In a hardware implementation, the same operation can be efficiently performed via Kaczmarz-like ℓ_2 estimation algorithms [43]. Since the algorithm used for results comparison also requires such a least squares estimation step, this Kaczmarz-like ℓ_2 estimation has not been included for comparison's sake. It should be noted, however, that the LBI algorithm provides the advantage that existing hardware elements can be easily re-used due to the fact that, by setting λ to zero, the Sparse Kaczmarz algorithm becomes the ordinary Kaczmarz algorithm.

An important aspect of an algorithm directed to the problem of automatic multiple trend break detection is the convergence criteria: the algorithms' processor must reach a state at which the results are considered to be trustworthy and further processing would only minorly change the estimation result while still consuming processing resources. Inspecting the pseudo code presented in Algorithm 1, one can see that the Sparse Kaczmarz can be implemented without any convergence criterion other than the selection of the number of iterations N that shall be ran by the algorithm. Even though, for some applications, an *a priori* determined N suffices for a convergence criteria, it is often helpful to add a convergence criterion based on the quality of the estimate, which allows reduced number of iterations until convergence and guarantees a robust and consistent output.

A. Stopping Criterion

In Fig. 4, a magnified view of the OTDR profile of Fig. 3, focusing on the region between 5.25 km and 5.4 km, is presented. Within this region, several faults of different magnitudes can be found very close to each other, almost to the limit of the measurement apparatus's spatial resolution. This particular set of faults is hard to be detected by an analysis algorithm. To ensure high quality estimation results such exceptional cases would have to be considered when setting the maximum number of iterations N . For scenarios that do not exhibit such an exceptional fault configuration, on the other hand, running the algorithm for the same number of iterations might lead to an unnecessary expense of processing time. Needless to say, this information is not available *a priori* to the algorithm, so choosing N based on the characteristics of the profile is not possible. For that reason, a simple but effective convergence criteria is introduced as an addition to the original Linearized Bregman Iterations by resorting to a practical implementation factor.

In practice, faults smaller than a given value – usually expressed in dB to follow the OTDR logarithmic intensity representation – are negligible, i.e., their presence, even though potentially indicating a process that can lead to a jeopardizing fault, is not relevant enough to trigger a repairing unit. In addition, due to the presence of noise, it becomes extremely challenging to detect trend breaks with a magnitude comparable to the noise level. This simple concept may be used to introduce a *minimum detectable loss* for the analysis algorithm: the minimum loss the algorithm should be able to detect, on average, over all the sampled data points. Throughout the result analysis, the minimum detectable loss was arbitrarily set to 0.125 dB, an extremely strict

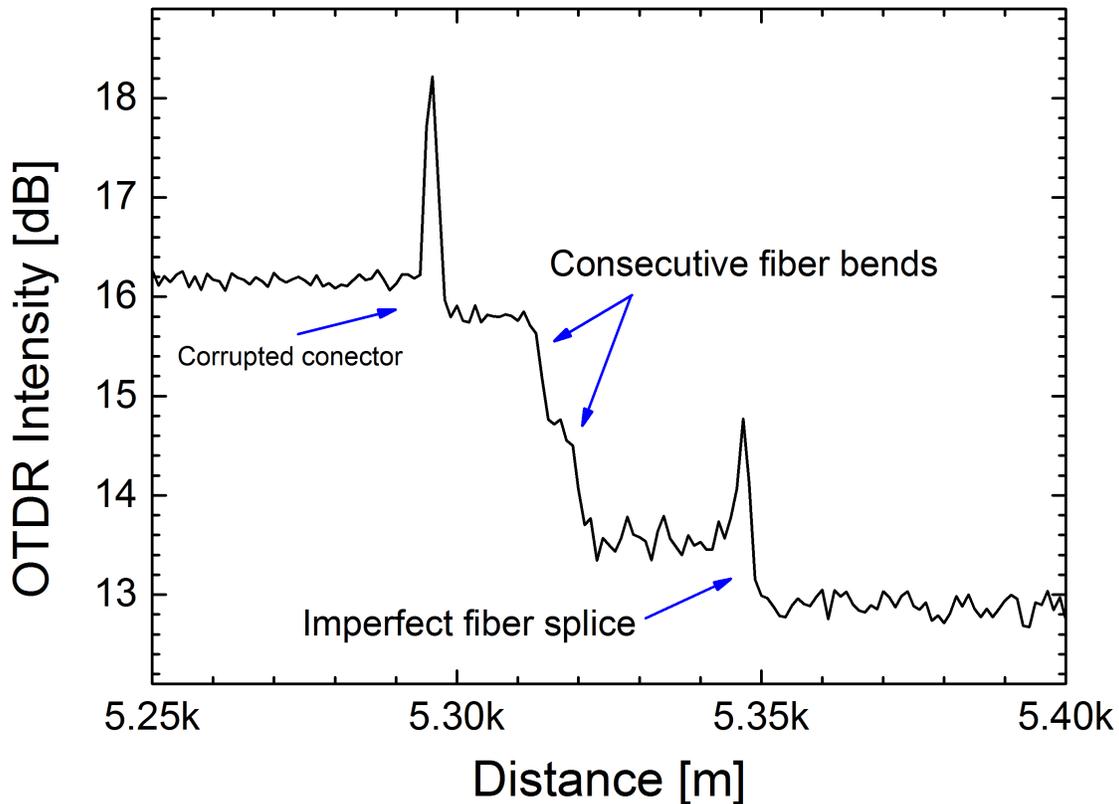


Fig. 4. Magnified view of an OTDR profile where multiple faults can be distinguished within very close spacing from each other. Such events are hard to be detected, and a fixed number of iterations might create a pitfall in this case.

value that, depending on the user's will, can be raised for faster convergence. It is important to note that setting a value for the minimum detectable loss does not guarantee that the algorithm will find all trend breaks with magnitudes higher than its value but, rather, provides a compromise between the algorithm's performance and its elapsed time.

B. Using multiple λ values

Linearized Bregman-based algorithms use the parameter λ in the cost function to trade sparsity (considered by the ℓ_1 norm) for model fitting error (considered by the squared ℓ_2 norm part). In general, a selection of λ is application-specific and no closed form solutions for choosing its optimal value exist. Even though smaller values of λ lead to faster convergence, it comes at the expense of lower sparsity in the results. In order to combine these two apparently contradicting characteristics, a set of λ values together with an approach called hot-starting [44] has been implemented.

Algorithm 2 starts with an empirically determined value of $\lambda = 0.5$ and runs until convergence is reached. Using $\lambda = 0.5$ is also beneficial for a fractional precision fixed point implementation of the LBI algorithm, as has been described in [40]. As previously commented, this pre-set value of λ yields high-quality results due to the PEAKLOCATIONS function, which will be demonstrated in Sect. IV. A new value of λ is then selected from a set S_λ and Algorithm 2 runs again with hot-started vectors β and v . S_λ is constructed as a grid from the initial value of $\lambda = 0.5$ up to λ_{max} – a λ value that would already yield a trivial (all-zero) solution of (3) [25], [45]. The hot-starting approach allows re-using the already calculated result of the LBI in a consecutive run with a higher lambda so that a significantly small number of iterations may still lead to convergence. Even though the vector β of a previous run can be directly used in the consecutive run, vector v must be adjusted to the new λ value. This can be performed through [44]:

$$v_j = (|\beta_j| + \lambda)\text{sign}(\beta_j), \quad (9)$$

for all $j = 1, \dots, m$. In order to constrain the processing time taken for the evaluation of higher λ values, a maximum number of iterations $N = 0.1N_c$ is allowed for each subsequent run after the first, where N_c is the total amount of iterations of the first run that led to convergence.

Selecting the model that best explains the original signal while maintaining the expected sparsity is performed by the so-called Bayesian Information Criterion (BIC) [46]:

$$\text{BIC}(\mathbf{y}, \hat{\beta}) = \|\hat{\beta}\|_0 \log(p) + p \log(\|\mathbf{y} - \mathbf{A}\hat{\beta}\|_2^2/p). \quad (10)$$

The β vector with the smallest BIC value is output as $\hat{\beta}_{\text{best}}$ in Algorithm 4. In the following Section, the performance achieved by Algorithm 4 is presented and analysed.

IV. RESULTS

In order to provide an extensive analysis of the automatic fault location capabilities of the proposed LBI, an equally extensive testbench of fibers must be available. Since different fiber profiles with different combinations of events (fault magnitudes and positions) that could provide such an extensive testbench are not easily available, it is necessary to resort to simulation to enrich and broaden the analysis. Therefore, the result analysis is divided into two parts: first, the algorithm is tested against a testbench fiber profile, which has been manipulated to present challenging characteristics in terms of fault location; second, a methodology that is able to

Algorithm 4 Model Selection with different λ values

Input: $S_\lambda = \text{grid}(0.5 : \lambda_{max})$
Output: Estimated λ_{best} , $\hat{\beta}_{first}$, $\hat{\beta}_{best}$

```

1:  $\hat{\beta}_{first}, N_c \leftarrow \text{LBOTDR}(\mathbf{y}, \lambda=0.5, \beta_{start}=\mathbf{0}, \mathbf{v}_{start}=\mathbf{0}, N = N_{max})$ 
2:  $\hat{\beta} \leftarrow \hat{\beta}_{first}$ 
3:  $b_{best} \leftarrow \text{BIC}(\hat{\beta}_{first}, \mathbf{y})$ 
4:  $\hat{\beta}_{best} \leftarrow \hat{\beta}_{first}$ 
5: for  $\lambda \in S_\lambda$  do
6:   for  $j \in 1, \dots, p$  do
7:      $v_j = (|\beta_j| + \lambda)\text{sign}(\beta_j)$ ,
8:   end for
9:    $\hat{\beta} \leftarrow \text{LBOTDR}(\mathbf{y}, \lambda, \beta_{start}=\hat{\beta}, \mathbf{v}_{start}=\mathbf{v}, N=0.1N_c)$ 
10:  if  $\text{BIC}(\hat{\beta}, \mathbf{y}) < b_{best}$  then
11:     $b_{best} \leftarrow \text{BIC}(\hat{\beta}, \mathbf{y})$ 
12:     $\hat{\beta}_{best} \leftarrow \hat{\beta}$ 
13:  end if
14: end for

```

create simulated fiber profiles that mimic all the relevant characteristics of real-world profiles is presented. This way, an extensive testbench of simulated profiles based on this procedure is created, which, in turn, enables a statistically relevant analysis of the algorithm's performance.

A. Experimental Results

The complexity of a fiber profile may differ greatly, from a perfect fiber with no faults or events – for instance, a recently installed fiber without any splitters –, to a fiber with multiple events of different magnitudes separated by arbitrary distances. The OTDR profile presented in Fig. 3 is a good example of the latter kind of profile since it contains a variety of events: extremely high magnitude faults; low magnitude faults in noisy regions of the profile; and both low and high magnitude faults within a short spatial interval. This specific fiber, however, has been manipulated so that such features could be observed and, due to the challenge it poses in terms of precisely locating the events from a fault location algorithm perspective, has been used to attest the success of the proposed LBI algorithm in a pessimistic real-world environment.

The monitoring method employed for the acquisition of this profile was the Wavelength Tunable Photon-Counting OTDR presented in detail in [11].

In Fig. 5, the filtering results using the LBI algorithm are presented. Simultaneously, the comparison between positions (Real Pos.) and magnitudes (Real Mag.) of faults located by an operator that has access to the OTDR trace and the estimated positions (Est. Pos.) and magnitudes (Est. Mag.) retrieved by the algorithm is detailed in Table I. It can be concluded that the estimated values are extremely close to the real values, with the maximum error in fault location being 2 meters and the maximum error in magnitude estimation being 0.2 dB. Therefore, even in a pessimistic environment, the proposed LBI algorithm excelled in determining the fault positions with both sensitivity and specificity, since no other events than the ones presented in Table I were identified. The algorithm elapsed time for this profile was 150 seconds, and the chosen λ value after the BIC model selection was 0.5.

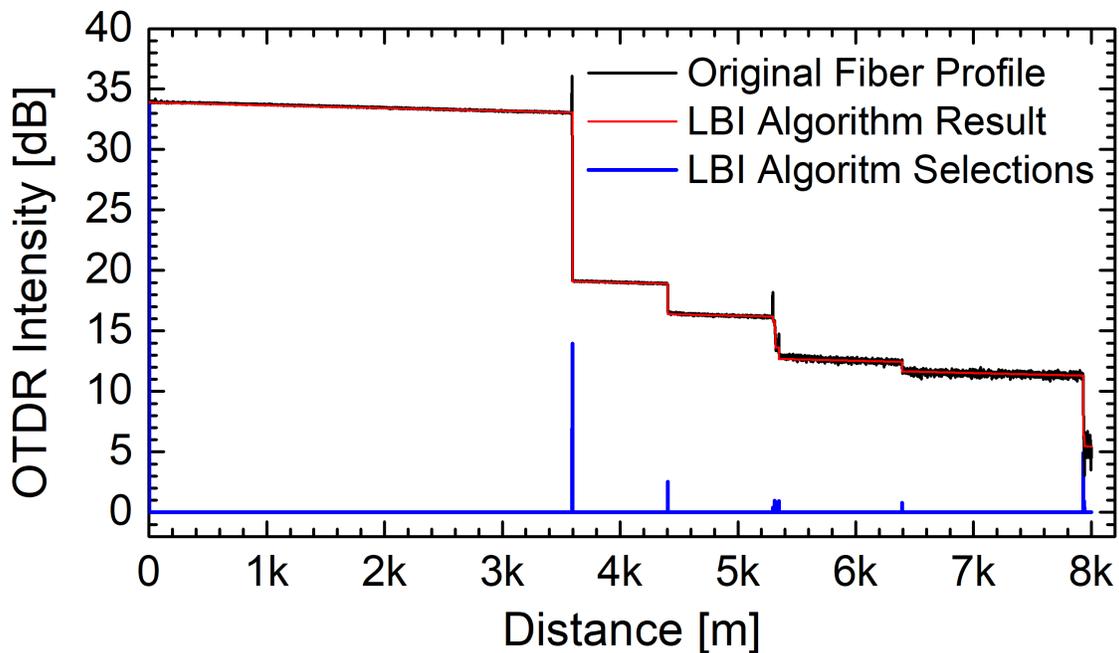


Fig. 5. LBI Algorithm results for the testbench fiber profile. The original fiber profile is depicted in black, while the filtered result is presented in red. In blue, the vector $\hat{\beta}_{\text{best}}$, that carries the information of position and magnitude of the selections of the algorithm, is depicted for reference.

It is interesting to evaluate the results of Fig. 5: they not only depict an extremely close estimation of the filter result with respect to the original signal, but it also calls attention to the fact that, given a sparse β vector, a noiseless fiber profile can be simulated through $A\beta$. Indeed, the

TABLE I
EVENT LIST RESULTS FOR THE TESTBENCH FIBER PROFILE

Real Pos.	Est. Pos.	Real Mag.	Est. Mag.
3593 m	3594 m	13.9 dB	13.9 dB
4404 m	4404 m	2.4 dB	2.5 dB
5298 m	5297 m	0.4 dB	0.4 dB
5313 m	5313 m	1.1 dB	1.0 dB
5317 m	5319 m	1.2 dB	1.2 dB
5349 m	5348 m	0.6 dB	0.9 dB
6395 m	6395 m	0.7 dB	0.8 dB
7932 m	7932 m	5.8 dB	5.6 dB

only striking difference between the red curve ($\mathbf{A}\beta$) and the black curve (\mathbf{y}) of Fig. 5 is the absence of the noise component in the former. Thus, if one chooses an arbitrary vector β_{sim} , a simulated noiseless fiber profile \mathbf{y}_{sim} can be created through $\mathbf{y}_{\text{sim}} = \mathbf{A}\beta_{\text{sim}}$. The noise parcel must, therefore, be included in the model of the simulated noiseless fiber profile $\mathbf{y}_{\text{sim}} = \mathbf{A}\beta_{\text{sim}}$ so that the simulation approaches the real measurement. The main sources of noise in an OTDR profile are the uncertainty of data acquisition [12], which can be modelled as a Poisson random variable, and the intrinsic coherent noise of coherent OTDR measurements [47], which can be modelled as an additive Gaussian random variable.

The uncertainty on data acquisition can be interpreted as the uncertainty of the number of photons measured by the photodetector. This counting process scales with the square root of the number of counts and reflects the lost of precision in the measurement as the signal power is reduced. This effect can be observed in positions of the fiber profile that are either far from the OTDR probe input or are positioned after drastic power drops – as can be seen e.g. in Fig. 5. Given a position X in \mathbf{y}_{sim} , with C counts, the counting noise is given by \sqrt{C} , and the signal-to-noise ratio (SNR) can be calculated as $\frac{C}{\sqrt{C}}$. Using the well-known relation $\text{SNR} = \frac{P_s}{P_n}$, (P_s and P_n being the signal and noise power, respectively) the counting noise variance for each position can be calculated and included in the simulated profile.

The noise originated from the CRN, on the other hand, can be modelled as an additive white Gaussian noise with variance calculated as given by [47]:

$$\sigma_{CRN} = \left(\frac{V_g}{4\Delta z \Delta \nu} \right)^{1/2}, \quad (11)$$

where v_g is the group velocity inside the fiber (assumed to be 2×10^8 m/s with an index of refraction approximately equal to 1.5), Δz is the fiber length, and $\Delta\nu$ is the linewidth of the optical probing source. If one considers a probing source with spectral bandwidth as wide as the DWDM channel width ($\Delta\lambda = 0.8$ nm), it translates into $\Delta\nu = 100$ GHz as specified in **ITUT G.694.1** [48], which causes the contribution of the CRN to be feeble. Since one of the goals of this study is to stress the fault-finding capabilities of the methodologies in noisy environments, a narrow linewidth probing source with $\Delta\nu = 100$ kHz has been considered throughout the numerical simulations, which increases the contribution of the CRN.

An example of a simulated fiber profile y_{sim} is depicted in the third panel of Fig. 6. In order to illustrate the recovery of all the characteristics expected in a fiber profile by the simulation, β_{sim} was chosen to be equal to the estimated $\hat{\beta}_{\text{best}}$ of the LBI algorithm after processing the testbench fiber profile of Fig. 5. The three panels of Fig. 6 depict: the original testbench fiber profile (top); the noiseless y_{sim} constructed from the resulting β after processing of this profile (middle); and the final y_{sim} after introduction of the previously described noise components (bottom). In summary, the simulated fiber profiles used throughout the simulated testbench were generated by creating random sparse vectors β_{sim} and applying matrix \mathbf{A} to generate controlled fiber profiles y_{sim} to which noise is included such that it simulates real-world fiber profiles. A striking resemblance to the original profile can be observed (first and third panels of Fig. 6), validating the method for simulating fiber profile.

A comment should be made regarding the reflection peaks that are often present in the original OTDR measurement at fault positions but have not been reconstructed in the simulated profile. Even though the detection of such peaks could be included in the analysis algorithm [49], the main interest is in the detection of the trend breaks which indicate the presence of faults. These peaks, as far as the LBI algorithm is concerned, are treated as outliers, when not included in the model, and do not compromise its performance. In order to focus on the trend break detection, the inclusion of such peak signatures was not considered throughout the construction of the simulated profiles.

B. Controlled Test Results

The simulated testbench was created by randomly selecting positions and magnitudes of faults in profiles with sampled data points spanning from 5000 to 15000. On a standard OTDR device, and assuming the maximum dynamic range achievable, a profile with 15000 points corresponds

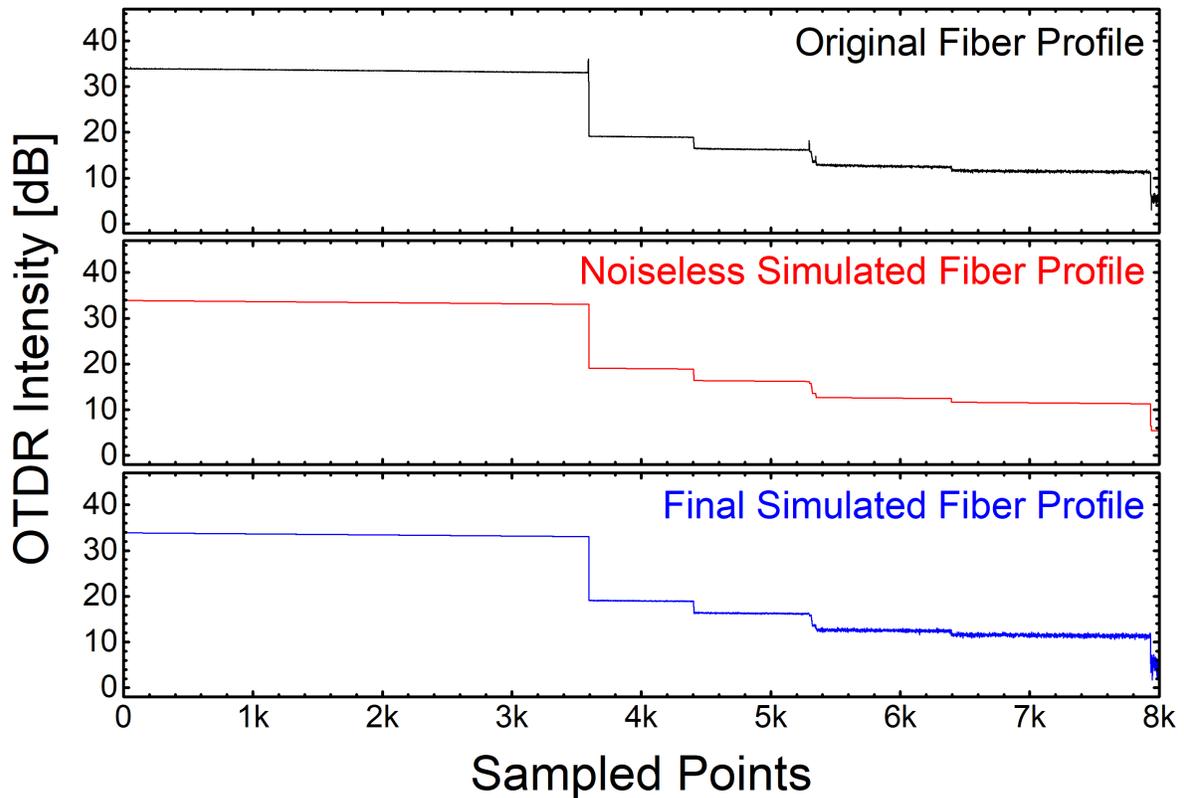


Fig. 6. Generation of a simulated profile that exhibits the characteristics of real-world fiber profiles. In the first panel, the original profile of Fig. 5 is depicted as a reference, where the noise components can be clearly identified. In the second panel, the noiseless reconstructed fiber profile is presented, where the $\hat{\beta}_{\text{best}}$ resulting from the processing of the original profile has been used. Finally, in the third panel, the simulated profile after inclusion of the noise components is presented.

to 100km, which is a sensible distance for long-haul optical networks and was thus chosen as the maximum number of points in the testbench. On the other hand, shorter fibers, within the tens of kilometers range, must also be contemplated, so 5000 sampled data points is set as the minimum number of points. Steps of 1000 data points from the minimum to the maximum number of data points are taken in order to provide a detailed dependence of the algorithm on the size of the original data. For each length, 1000 profiles were randomly selected and 5 faults of magnitudes out of a uniform distribution from 0.5 to 5 dB were randomly placed in the original data stream.

In order to provide a comparative analysis, the profiles have been analysed with the LBI Algorithm (Algorithm 4) and with the Adaptive ℓ_1 Filter, where the latter is taken as a reference. Gauging the success of the algorithms involves three parameters, as previously mentioned and informally set forward by [12]:

- sensitivity, the capacity to find the positions which correspond to faults.
- specificity, the capacity to neglect all the positions which do not correspond to faults.
- celerity, the capacity to be both sensitive and specific in the least amount of time.

An analysis based on the computational complexity will also be drawn in the next section, since it is of the utmost importance when adapting the algorithm to an embedded processing unit, one of the sought for goals of automatic fault finding [50].

Instead of analysing the results directly in light of the figures of merit described in the previous paragraph, a visually interesting and intuitive analysis is performed and presented in Fig. 7. That is the squared error norm between the original β used to create the fiber profile and the estimated coefficient vector $\hat{\beta}$ averaged over all the random profiles for a given data size. This metric is useful since it considers errors in the estimation of the magnitudes of the faults as well as in the estimation of their positions. Since β is a sparse vector, lack of both sensitivity and specificity by the algorithms are easily identifiable from a deviation from the perfect zero-error mark of $\|\beta - \hat{\beta}\|_2^2$. Additionally, since the noise components of each fiber profile were simulated to mimic the practical reality, longer fibers will often exhibit positions with lower SNR than shorter fibers. Therefore, the slight increase in the averaged squared error norm observed when the data size increases is expected.

TABLE II
CONTINGENCY TABLE – FAULT FINDING

	Fault Present			Fault Absent		
Fault Found	TP (ℓ_1) 75,389	TP (LB) 74,814	TP (LB $\lambda = 0.5$) 74,811	FP (ℓ_1) 777,800	FP (LB) 30525	FP (LB $\lambda = 0.5$) 56808
Fault Neglected	FN (ℓ_1) 1599	FN (LB) 2174	FN (LB $\lambda = 0.5$) 2177	TN (ℓ_1) 109,145,212	TN (LB) 109,892,487	TN (LB $\lambda = 0.5$) 109,866,204
Measures	Sens. (ℓ_1) 97.92%	Sens. (LB) 97.18%	Sens. (LB $\lambda = 0.5$) 97.17%	Spec. (ℓ_1) 99.29%	Spec. (LB) 99.97%	Spec. (LB $\lambda = 0.5$) 99.95%

In Fig. 7, the results considering the output of the proposed LBI algorithm (black) are depicted along with those after the first run (red), with $\lambda = 0.5$. This attests that, as commented in Section III, running the algorithm based on the stopping criteria and $\lambda = 0.5$ yields high-quality results. Also, as can be observed in Fig. 8, the total average processing time increases substantially when the search for the best λ is performed while the gain in accuracy is not extreme. Therefore, an

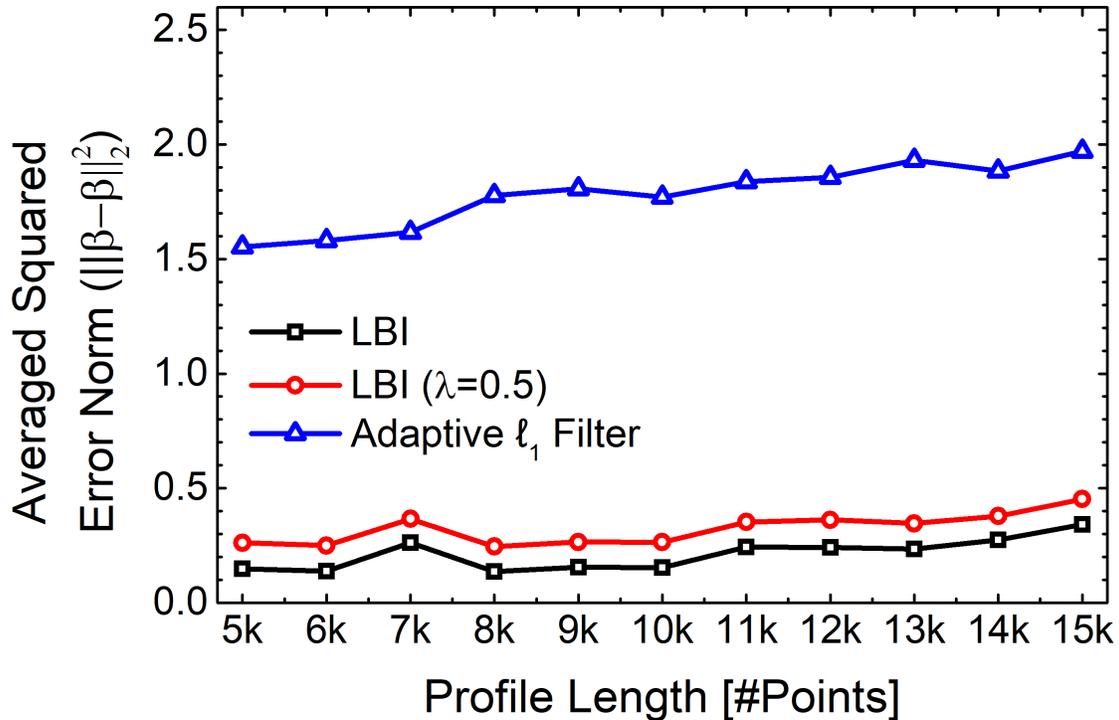


Fig. 7. Averaged squared error norm between the recovered sparse vector $\hat{\beta}$ and the original sparse vector β ($\|\beta - \hat{\beta}\|_2^2$). The average is between the 1000 different profiles randomly sorted for each of the data sizes. While the black curve correspond to the final output of the LBI algorithm, the red curve corresponds to the result of the first run, with $\lambda = 0.5$.

operator facing the compromise between longer processing times and slight accuracy gains may opt to stop Algorithm 4 in line 4. Fig. 8 also shows that the average processing time for both algorithms is comparable for the range of sampled data points considered in this testbench. It is clear, however, that the slope of the blue curve increases faster, indicating that the Adaptive ℓ_1 Filter algorithm performs slower than the proposed LBI algorithm for bigger data sets.

The trends presented in Figs. 7 and 8 indicate that both algorithms were successful in recovering the original sparse vector used to create the simulated fiber profiles in the presence of noise while maintaining comparable processing times. Fig. 7 in particular, however, shows a clear dominance of the LBI filter over the Adaptive ℓ_1 Filter with respect to the proximity of the estimated to the original profile. This is qualitatively translated into an error more than 5 times smaller even when disregarding the λ selection step of Algorithm 4.

Even though the results of Fig. 7 are clear, a means of assessing the figures of merit of sensitivity and specificity and analyse the performance of the algorithms with greater precision is desirable. In this context, contingency tables are specially interesting for the analysis of binary

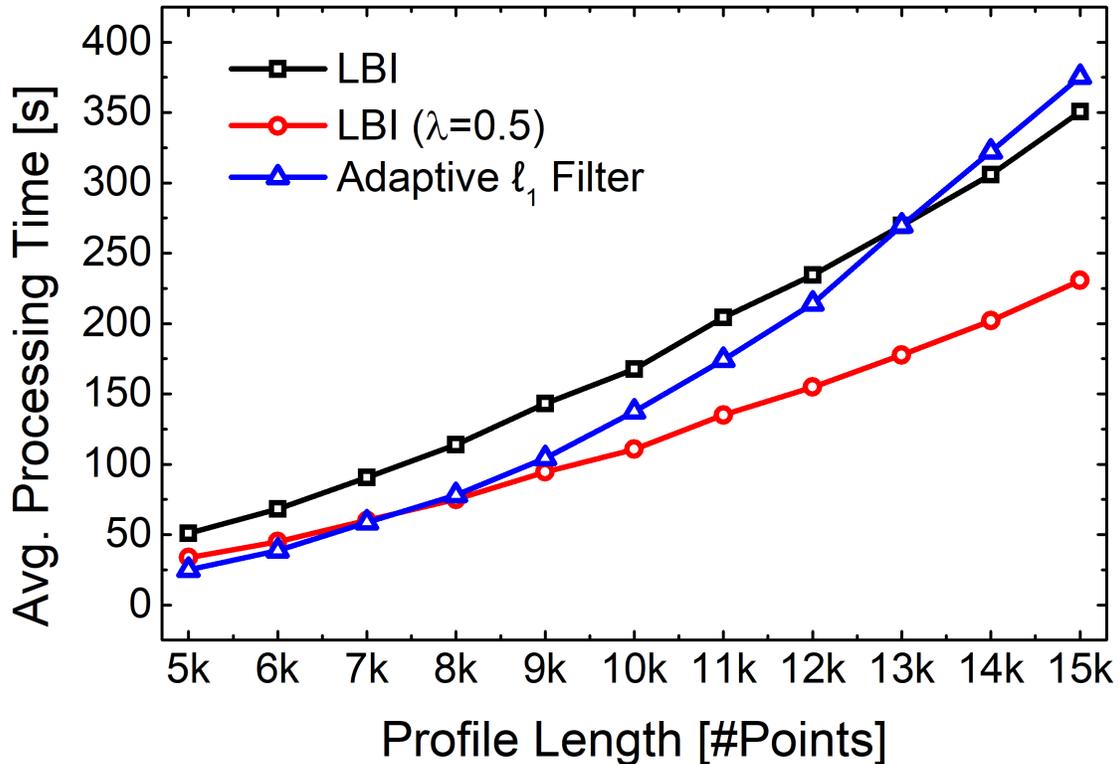


Fig. 8. Algorithm elapsed time averaged over the 1000 different profiles randomly sorted for each profile length. For long profiles, the Linearized Bregman Iterations reach convergence faster than the Adaptive ℓ_1 Filter.

classification tests. Additionally, the outputs of a contingency table analysis are exactly the figures of merit of interest. By interpreting the trend break detection problem as a *break/no break* – and, thus, a binary — classification test, the contingency table for the results of the algorithms can be constructed. In Table II, the absolute values of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) are presented for the LBI and for the Adaptive ℓ_1 Filter Algorithms.

To perform this analysis, the entries of the contingency table were defined as follows: the TPs correspond to faults found by the algorithms which are present in the original sparse vector; the TNs correspond to positions that were not selected by the algorithms and that did not originally contain faults; the FPs are faults found by the algorithms in positions where there were originally no faults and the FNs correspond to positions not selected by the algorithm but that originally contained faults. Throughout the analysis, matches were considered with a zero error margin, i.e., matches are only considered if the estimated position is exactly the same as the original position. Also, for the construction of this contingency table, the results from all the simulation

runs have been taken into account without discrimination with respect to data size, as in Fig. 7.

Regarding the Sensitivity and Specificity, both algorithms exhibit extremely high values for both figures of merit. The *Accuracy*, defined as $\frac{TP+TN}{TP+FP+FN+TN}$, is also extremely high with values of 99.97% and 99.29% for the proposed LBI algorithm and the Adaptive ℓ_1 Filter, respectively, indicating that both algorithms are extremely well suited for the ascribed trend break detection problem. A closer analysis, however, shows that the small differences in Accuracy may have a deeper meaning which is clouded by the high number of TNs, so it is useful to compare the absolute numbers of FPs and FNs. To put the absolute numbers in perspective, an overall number of 110 Million potential fault positions have been simulated altogether for the presented results.

While the LBI failed to identify 575 more faults than the Adaptive ℓ_1 Filter, which corresponds to an increase of $\sim 35\%$ in the FNs, the Adaptive ℓ_1 Filter wrongly identified 747,275 more points as faults than the LBI, an increase of $\sim 2,400\%$ in the FPs. This analysis shows that, even though the Adaptive ℓ_1 Filter is slightly less likely to neglect a trend break that is present in the data stream, it comes at the expense of flooding the result with several candidates which do not correspond to actual trend breaks. At the same time, the equilibrium between FPs and FNs seems to be more closely achieved by the LBI algorithm. Additionally, the magnitudes of the faults neglected by the LBI were studied, showing that the algorithm fails to identify small faults, i.e., those that correspond to a 0.5 dB power drop in the profile – the minimum value used in the simulation procedure. This indicates that the LBI is more sensitive and specific to faults that can indeed pose threats to the link operation.

To get a better impression on the relative error distribution of the results summarized in the FP (False Positive) class, the histogram of FP incidence with respect to its distance from the nearest real fault is depicted in Fig. 9 for both Algorithm 2 (with and without the selection of λ 's) and the Adaptive ℓ_1 Filter. For this analysis, the following evaluation was employed: for each processed simulated fiber profile, the minimum absolute value difference between entries of the output vector $\hat{\beta}$ and the original vector β were determined, i.e., the False Positive's distance to the nearest true fault position.

The histograms clearly exhibit different widths, translating the overall proximity of a False Positive result to a real fault. It is interesting to note that the widths of the LBI and LBI ($\lambda = 0.5$) histograms already differ, indicating that the impact of the λ selection stage is majoritarilly on the FPs, as already evidenced by the Contingency Table. The most striking result of Fig. 9, however,

is the major width difference between the LBI algorithm and the Adaptive ℓ_1 Filter histograms, corroborating the previous analysis that the latter, even though having a higher TP rate, floods the selection result with False Positives. In this context, the *Precision* figure of merit, defined as $\frac{TP}{TP+FP}$, is relevant as it translates the confidence of an operator on a positive result output from the analysis algorithm. Even with a lower number of TPs, the confidence on a positive result for the LBI ($\sim 71\%$) is much higher than for the Adaptive ℓ_1 Filter ($\sim 8\%$).

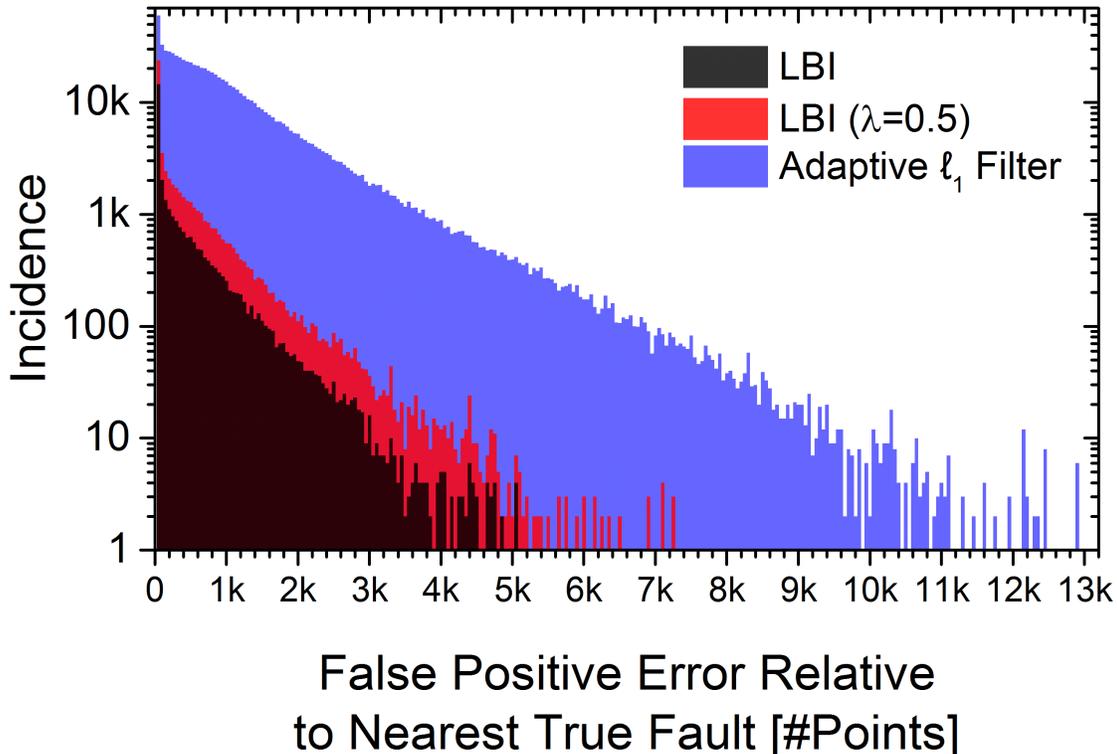


Fig. 9. Histogram of False Positive incidence with respect to the distance from nearest real fault. The histogram bin width is 50 points and the logarithmic scale is used for clearer visualization of the results.

C. Complexity

The complexity analysis for the Adaptive ℓ_1 Filter has been presented in [12]. There, the authors state that “a model that can be described by m components will take $O(pm)$ operations” [per iteration], where p , in this document, stands for the number of rows of the matrix \mathbf{A} . The number m refers to the number of non-zero elements in a current estimate of β during the iterations. The number of iterations for the Adaptive ℓ_1 Filter is not preselected, the algorithm

typically runs until convergence, i.e., until the non-zero positions do not change between two consecutive iterations.

The presented Algorithm 2 on the other hand only requires $O(N)$ operations per iteration. More specifically, as can be seen in Algorithm 2, its main processing steps (lines 9 and 10 of Algorithm 2) only require 2 multiplications and $i + 2$ additions at iteration k with $i = (k - 1) \bmod p + 1 \in \{1, \dots, p\}$. On average this means that this step only requires 2 multiplications (the value of $1/\|\mathbf{a}_i\|_2^2$ can be pre-calculated and stored in memory) and approximately $p/2$ additions. Based on the same considerations one can see that the lines 11 to 13 add a single multiplication and approximately another $p/2$ additions per iteration on average. Combining this with the p subtractions of the shrink operation, this means that a single iteration of the Algorithm can be performed with about $2p$ addition-equivalents on average, as well as with 3 multiplications.

The Adaptive ℓ_1 Filter is typically calculated multiple times for a set of λ values, usually processed in descending order. This allows using the results of a previous calculation using a higher λ value to warm-start a following calculation with a lower value. This is a reasonable approach for an active set-type algorithm, where a large value of λ , due to the high sparsity of the expected result, leads to a fast convergence of such algorithms. Starting the Adaptive ℓ_1 Filter with a small λ value would result in low sparsity, i.e. a huge active set, leading to a high computational complexity due to a large number of required iterations.

For Linearized Bregman-based algorithms, an opposite behaviour can be observed. In terms of convergence speed, λ should be chosen as small as possible. This is because, although the complexity of a single LB Iteration does not depend on the sparsity of the result, using higher λ values typically increases the number of required iterations until convergence. Useful values of λ are generally within the range of $[0, \lambda_{\max})$, where λ_{\max} is the value where the zero vector is the solution of (3). For λ_{\max} one can typically calculate upper bounds (e.g. as described in [25], [45]). However, as evaluated in this section, successful sparse estimation of trend breaks in OTDR profiles can be achieved with values close to 0, giving LB based algorithms an additional advantage in terms of computational complexity over the Adaptive ℓ_1 Filter.

Finally, as also stated in [12], the Adaptive ℓ_1 Filter requires a normalization of the columns of the \mathbf{A} matrix to achieve a good performance. Such normalization requires additional multiplications compared to when using \mathbf{A} directly, where columns with only ones and zeros enable performing scalar products with additions. Compared to the Adaptive ℓ_1 Filter, the LB algorithm can use the matrix \mathbf{A} directly without any pre-processing.

D. Digital Hardware Implementation and Improvements on the LBI

When analysing Algorithm 2, one can see that this algorithm is perfectly suited for implementation in digital hardware: the main computational complexity is spent on parallel additions. When considering an implementation on an FPGA, by storing the vectors β in parallel block RAMs, one can access the values of the vectors in parallel. Providing such an access for hundreds of values in parallel poses no problem to even medium level FPGAs, e.g., as described in [51] for Altera Cyclon IV FPGAs. Adding up a large number of values in parallel can also be easily performed by using pipelined adder trees [52]. This allows for a high level of parallelization for the LB based algorithm enabling an efficient development of an unified measurement and analysis hardware in a single embedded unit.

For LB based sparse Kaczmarz, recently a variety of improvements have been published such as efficient Linearized Bregman Iterations [33], mikrokicking [44], or using a sparsity-adaptive step width [53]. Such approaches significantly allow reducing the complexity of LB based Sparse Kaczmarz algorithms. To provide a general comparison on an algorithmic level, it was decided not to include such enhancements in this work. However, each of the mentioned enhancements can be easily applied to the LB iterations for OTDR analysis, allowing reducing the complexity even further.

V. CONCLUSION

The increasing demand for high-rate communication links can be met with new generation optical fiber networks. An important aspect of such networks, for both the user and the network operator, is robustness, which translates into shorter downtimes for the users and reduced operational costs for the operator. In this work, a measurement and processing system for fault analysis and detection for fiber optic links is presented, using a Linearized Bregman Iterations based algorithm especially designed for this problem. In-depth performance evaluation has been presented to support the claims.

Application of the Linearized Bregman Iterations method by means of a tailored algorithm has been successful in real and simulated fiber profiles. By conjugating a simple procedure to generate arbitrary noiseless profiles and an analysis of the main noise contributions that affect the data stream permitted a close reconstruction of fiber profiles that share the same characteristics as found in practice. With such a methodology at hand, an extensive testbench was created, where fibers with distinct characteristics could be analyzed by the proposed algorithm and a reference

trend break detection algorithm found in the literature. Analysis of the results was performed evidencing the suitability of the algorithms and highlighting the dominance of the LBI algorithm with respect to the equilibrium between sensitivity and specificity with high accuracy.

A timing comparison indicated that LBI based algorithms converge faster for long fiber profiles. In addition, a smoother slope was observed in the processing time of the proposed algorithm as the instance size increase, hinting to the fact that it exhibits reduced complexity. To further investigate this point, a complexity analysis of both algorithms was conducted confirming the lower complexity of the LBI based algorithm. Finally, a brief rationale points toward the possibility of embedding the algorithm in an FPGA with expected results such as: significant reduction of processing time due to the parallelizability in dedicated hardware; and the development of an unified measurement and analysis hardware in a single embedded unit.

In conclusion, the results show that the novel algorithm outperforms the comparison algorithm in terms of estimation performance as well as computation speed, while still maintaining the algorithmic structure that can be efficiently implemented in digital hardware. Therefore, the proposed measurement system consisting of a Photon-Coupling OTDR as data acquisition subsystem and the LBI algorithm as the data processing unit enables reliable automatic fault detection for optical fiber links.

ACKNOWLEDGMENT

The authors would like to thank F. Calliari, J. D. Garcia, and M. Souto for their assistance with the Julia programming. G. C. Amaral is indebted to J. D. Garcia for enlightening discussions. Financial support from brazilian agencies CAPES, FAPERJ, and CNPq are acknowledged by G. C. Amaral. 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.

REFERENCES

- [1] S. Kumar and M. J. Deen, *Fiber optic communications: fundamentals and applications*. Wiley, 2014.
- [2] M. K. Barnoski, M. D. Rourke, S. M. Jensen, and R. T. Melville, “Optical Time Domain Reflectometer,” *Applied Optics*, vol. 16, 1977.
- [3] W. Eickhoff and R. Ulrich, “Optical frequency domain reflectometry in single-mode fiber,” *Applied Physics Letters*, vol. 39, no. 9, pp. 693–695, 1981.

- [4] D. Derickson, *Fiber Optic - Test and Measurement*, 1st ed. Prentice Hall, 1998.
- [5] P. J. Urban, G. C. Amaral, G. Żegliński, E. Weinert-RaŹczka, and J. P. von der Weid, "A tutorial on fiber monitoring for applications in analogue mobile fronthaul," *IEEE Communications Surveys & Tutorials*, 2018.
- [6] 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.
- [7] S. Hann, J. sang Yoo, and C.-S. Park, "Monitoring technique for a hybrid PS/WDM-PON by using a tunable OTDR and FBGs," *Measurement Science and Technology*, vol. 17, no. 5, p. 1070, 2006. [Online]. Available: <http://stacks.iop.org/0957-0233/17/i=5/a=S22>
- [8] International Electrotechnical Commission, "Non-connectorized single-mode OTDR reflecting device for category c – controlled environments," IEC 61753-041- 2, May 2014.
- [9] L. Wu, Q. Wang, M. Guo, C. Du, and Y. Zhang, "Characterization of displacement sensing based on fiber optic microbend losses," *Instrumentation Science & Technology*, vol. 44, no. 5, pp. 471–482, 2016. [Online]. Available: <https://doi.org/10.1080/10739149.2016.1170033>
- [10] C. Saunders and P. J. Scully, "Sensing applications for POF and hybrid fibres using a photon counting OTDR," *Measurement Science and Technology*, vol. 18, no. 3, p. 615, 2007. [Online]. Available: <http://stacks.iop.org/0957-0233/18/i=3/a=010>
- [11] G. Amaral, J. Dias Garcia, L. Herrera, G. Temporaio, P. Urban, and J. von der Weid, "Automatic fault detection in WDM-PON with tunable photon counting OTDR," *Lightwave Technology, Journal of*, vol. 33, no. 24, pp. 5025–5031, 2015.
- [12] 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.
- [13] M. M. Wagner, F.-C. Tsui, J. U. Espino, V. M. Dato, D. F. Sitting, R. A. Caruana, L. F. McGinnis, D. W. Deerfield, M. J. Druzdel, and D. B. Fridsma, "The emerging science of very early detection of disease outbreaks," *Journal of Public Health Management and Practice*, vol. 7, no. 6, pp. 51–59, 2001.
- [14] M. Basseville and A. Benveniste, "Desgin and comparative study of some sequential jump detection algorithms for digital signals," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 31, no. 3, pp. 521–535, 1983.
- [15] C. Clavel, T. Ehrette, and G. Richard, "Events detection for an audio-based surveillance system," in *Proceedings of the IEEE International Conference on Multimedia and Expo, 2005. ICME 2005*. IEEE, 2005, pp. 1306–1309.
- [16] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [17] C. F. Karney, "Long-time correlations in the stochastic regime," *Physica D: Nonlinear Phenomena*, vol. 8, no. 3, pp. 360–380, 1983.
- [18] R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review," *Statistical Science*, pp. 235–249, 2002.
- [19] B. C. Carter, M. Vershinin, and S. P. Gross, "A comparison of step-detection methods: How well can you do?" *Biophysical Journal*, vol. 94, pp. 306–319, 2008.
- [20] R. L. Brown, J. Durbin, and J. M. Evans, "Techniques for testing the constancy of regression relationships over time," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 149–192, 1975.
- [21] W. S. Rea, M. Reale, C. Cappelli, and J. A. Brown, "Identification of changes in mean with regression trees: an application to market research," *Econometric Reviews*, vol. 29, no. 5-6, pp. 754–777, 2010.
- [22] M. Storath, A. Weinmann, and L. Demaret, "Jump-sparse and sparse recovery using Potts functionals," *IEEE Transactions on Signal Processing*, vol. 62, no. 14, 2014.
- [23] E. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier analysis and applications*, vol. 14, no. 5-6, pp. 877–905, 2008.
- [24] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky, " ℓ_1 Trend Filtering," *SIAM review*, vol. 51, no. 2, pp. 339–360, 2009.

- [25] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, 2010. [Online]. Available: <http://www.jstatsoft.org/v33/i01/>
- [26] F. Liu and C. J. Zarowski, "Events in fiber optics given noisy OTDR data. Part I. GSR/MDL method," *IEEE Transactions on Instrumentation and Measurement*, vol. 50, no. 1, pp. 47–58, Feb 2001.
- [27] —, "Detection and location of connection splice events in fiber optics given noisy OTDR data. Part II. R1MSDE method," *IEEE Transactions on Instrumentation and Measurement*, vol. 53, no. 2, pp. 546–556, April 2004.
- [28] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 460–489, 2005. [Online]. Available: <https://doi.org/10.1137/040605412>
- [29] J. Cai, S. Osher, and Z. Shen, "Convergence of the linearized Bregman iteration for l_1 -norm minimization," *AMS Mathematics of Computation*, no. 78, pp. 2127–2136, 2009.
- [30] —, "Linearized Bregman iterations for compressed sensing," *AMS Mathematics of Computation*, no. 78, pp. 1515–1536, 2009.
- [31] W. Yin, "Analysis and generalizations of the linearized Bregman method," *SIAM Journal of Imaging Sciences*, vol. 3, no. 4, pp. 856–877, 2010.
- [32] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for l_1 -minimization with applications to compressed sensing," *SIAM Journal of Imaging Sciences*, vol. 1, pp. 143–168, 2008.
- [33] 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.
- [34] G. P. Agrawal, *Fiber-Optic Communication Systems*, 2nd ed. John Wiley & Sons, 2002.
- [35] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM journal on computing*, vol. 24, no. 2, pp. 227–234, 1995.
- [36] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.
- [37] E. Candès and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.
- [38] M.-J. Lai and W. Yin, "Augmented l_1 and nuclear-norm models with a globally linearly convergent algorithm," *CoRR*, vol. abs/1201.4615, 2012. [Online]. Available: <http://dblp.uni-trier.de/db/journals/corr/corr1201.html#abs-1201-4615>
- [39] D. A. Lorenz, F. Schoepfer, and S. Wenger, "The linearized Bregman method via split feasibility problems: Analysis and generalizations," *SIAM Journal on Imaging Sciences*, vol. 7, no. 2, pp. 1237–1262, 2014.
- [40] M. Lunglmayr, B. Hiptmair, and M. Huemer, "Scaled linearized Bregman iterations for fixed point implementation," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2017, pp. 1–4.
- [41] D. A. Lorenz, S. Wenger, F. Schöpfer, and M. Magnor, "A sparse kaczmarz solver and a linearized Bregman method for online compressed sensing," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 1347–1351.
- [42] H. Mansour and O. Yilmaz, "A sparse randomized Kaczmarz algorithm," in *Proceedings of the IEEE Global Conference on Signal and Information Processing*, Dec 2013, pp. 621–621.
- [43] M. Lunglmayr, C. Unterrieder, and M. Huemer, "Approximate least squares," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 4678–4682.
- [44] M. Lunglmayr and M. Huemer, "Microkicking for fast convergence of sparse Kaczmarz and sparse LMS," in *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec. 2017, p. 5.

- [45] R.-B. Chen, K.-H. Cheng, S.-M. Chang, S.-L. Jeng, P.-Y. Chen, C.-H. Yang, and C.-C. Hsia, "Multiple-instance logistic regression with LASSO penalty," *The R Journal*, vol. 9, no. 1, pp. 446–457, 06 2017.
- [46] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, pp. 461–464, 1978.
- [47] K. De Souza, "Significance of coherent Rayleigh noise in fibre-optic distributed temperature sensing based on spontaneous Brillouin scattering," *Measurement Science and Technology*, vol. 17, no. 5, p. 1065, 2006.
- [48] International Telecommunication Union, "Spectral grids for WDM applications: DWDM frequency grid," <https://www.itu.int/rec/T-REC-G.694.1-201202-I/en>, 2012, [Online; accessed 15-May-2018].
- [49] M. Souto, J. D. Garcia, and G. C. Amaral, "11 adaptive trend filter via fast coordinate descent," in *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2016 IEEE*. IEEE, 2016, pp. 1–5.
- [50] 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," in *Proceedings of the 6th International Conference on Photonics, Optics and Laser Technology - Volume 1: PHOTOPTICS*, INSTICC. SciTePress, 2018, pp. 82–90.
- [51] Altera, "Cyclone IV device handbook," Tech. Rep., March 2016. [Online]. Available: https://www.altera.com/en_US/pdfs/literature/hb/cyclone-iv/cyclone4-handbook.pdf
- [52] ———, "Advanced synthesis cookbook," Tech. Rep., Jul. 2011. [Online]. Available: http://www.altera.com/literature/manual/stx_cookbook.pdf
- [53] M. Lunglmayr and M. Huemer, "Sparsity-enabled step width adaption for linearized Bregman based algorithms," to appear in *IEEE Statistical Signal Processing Workshop (SSP)*, Jun. 2018, p. 5.