Proactive Fiber Damage Detection in Real-time Coherent Receiver

F. Boitier(1), V. Lemaire(2), J. Pesic(1), L. Chavarría(1), P. Layec(1), S. Bigo(1), E. Dutisseuil(1)

(1) Nokia Bell Labs, Route de Villejust, 91620 Nozay, France, fabien.boitier@nokia-bell-labs.com
(2) Orange Labs, Avenue Pierre Marzin, 22307 Lannion cedex, France

Abstract *We develop an algorithm extension for a coherent receiver, coupled with machine learning to monitor mechanical stress to an optical fiber, for recognizing fiber breaks before they occur. We demonstrate event classification with 95% accuracy over a real-time PDM-QPSK testbed.*

Introduction

To comply with the highest reliability standards, most optical networks have been designed with a protection mechanism which can react to a fiber break by rerouting data to a spare fiber path in <50ms. To meet this small response time, network designers had to almost duplicate all hardware. To save on cost and energy, Pesic *et al.*[1] proposed to anticipate fiber breaks by a few seconds, opening opportunities for dynamic protection mechanisms. These mechanisms are resilient to one fiber cut at a time over a set of N fiber routes (N<1), but share the cost of protecting hardware over the N routes. To enable them, authors of[1] proposed to monitor the State of Polarization (SOP) of an out-of-band unmodulated laser light travelling through the optical fiber with a commercial polarimeter. Simsarian *et al.* recently proposed a low-cost implementation with two photodiodes and a polarizer[2] but limited to a single span. Beside these studies, new software tools were developed for fault detection and localization from receiver monitors[3].

In this paper, we take advantage of digital signal processing to build upon the work of [1] and make it eligible for cost-effective addition into coherent terminals. We developed algorithms which can concurrently decode data and track SOP, avoiding additional hardware in a real-time receiver. We perform proactive detection of fiber damage based on two key ingredients (*i*) flag raising when SOP speed exceeds a certain limit and (*ii*) event classification, which are low complexity enough to be embedded in an ASIC. The latter ingredient reduces probability of false alarm and avoids unnecessary rerouting of traffic.

Fiber Damage Proactive Detection Principle

Coherent technologies with polarization demultiplexing use Digital Signal Processing (DSP) to compensate for SOP fluctuations and other impairments. To this end, Constant Modulus Algorithm (CMA) is the most popular method. Here, we exploit the adaptive Finite-Impulse-Response (FIR) filter of the CMA to compute the SOP while decoding received data (step #1, fig 1a). We obtain the Jones Matrix of each polarization tributary by replacing the four FIR filters of the 2x2 CMA equalizer, by their DC components (resp. the sum of taps) in the frequency domain (resp. in the time domain)[4]. We then derive the SOP in the Stokes coordinates {S_1;S_2;S_3}, e.g. for the ‘y’ tributary, as:

$$\begin{aligned} s_0 &\propto |H_{xx}|^2 + |H_{yx}|^2 \\ S_1 &= \left[|H_{xx}|^2 - |H_{yx}|^2\right] \Big/ s_0 \\ S_2 &= -2\,\mathrm{Re}\left(H_{yx}^* H_{xx}\right) \Big/ s_0 \\ S_3 &= 2\,\mathrm{Im}\left(H_{yx}^* H_{xx}\right) \Big/ s_0 \end{aligned} \qquad (1)$$

where H_{ij} is the sum of taps in the time domain from ‘*i*’ input to ‘*j*’ output tributaries. In step #2, we track the SOP rotation speed, defined as:

$$\frac{\Delta SOP}{\Delta t} = \frac{2}{\Delta t}\sin^{-1}\left(\frac{1}{2}\sqrt{\sum_i (\Delta S_i)^2}\right) \qquad (2)$$

where Δt is the time interval between two measurements of SOP and ΔS_i the associate Stokes parameter differences with $i = \{1,2,3\}$. In step #3, the SOP rotation speed is compared to a threshold which can raise a flag signal to move to step #4, i.e. the recording of pre-trigger and post-trigger samples of a so-called “event” during few seconds, sent at step #5 to the classifier which extracts the class of event. Finally, at step #6, if the event is classified as “risky”, the receiver raises an alarm through signal messaging mechanism to the control plane in step #7. This latter decides whether to reroute the traffic and possibly computes a new route.

Experimental setup

Fig. 1(b) shows the experimental setup. A polarization division-multiplexed QPSK signal carrying pseudo-random data at 7 GBaud is amplified into an Erbium Doped Fiber Amplifier (EDFA), possibly mixed with additional Amplified Spontaneous Emission (ASE) noise. The signal propagates over few meters of optical fiber before reaching the receiver, equipped with a dual-stage EDFA preamplifier followed by a 1.4 nm-broad filter at the signal wavelength (193.150 THz)

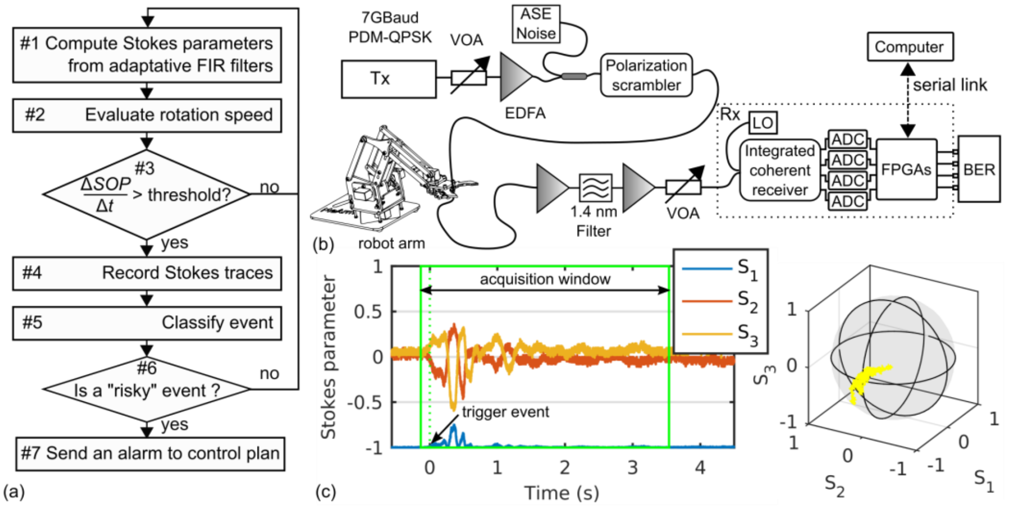

Fig. 1: (a) Flow-chart of the proactive fiber break detection; (b) Experimental setup with programmable mechanical events driven by a robot arm (c) Example of Stokes parameters vs time for a "shaking" event. Green rectangle shows the acquisition window. Green dashed line indicates the trigger signal raised in step #3 of (a). Yellow curve on the Poincaré sphere shows the SOP variation for this event.

against excess ASE noise. The real-time receiver board, presented in detail in [5] consists in an integrated coherent receiver followed by four 5-bits Analog to Digital Converters (ADCs) operating at two samples per symbol followed by FPGAs. All DSP is performed in one FPGA. CMA was implemented as a 5-taps fractionally-spaced blind adaptive filter operating on 128 samples in parallel. Real and imaginary part of each tap are coded on 8 bits and updated at the 109 MHz FPGA clock rate – every 64th symbol.

We emulate about-to-happen fiber damage by creating a multiplicity of mechanical stress events to the fiber with the claw of a robot arm controlled by an Arduino™. An important milestone of our work was to improve the accuracy of SOP measurement by (*i*) limiting the extraction rate of H_{ij} coefficients to one extraction every mth clock cycle such that consecutive samples are impaired by truly decorrelated contributions of noise (we came up with m=64) and then (*ii*) averaging each resulting H_{ij} coefficient over a time-window of optimized duration. The larger the window, the smaller the variance of measurements, but also the lower is the ability to track fast varying effects. For this milestone, we temporarily replace the real-time-board with a

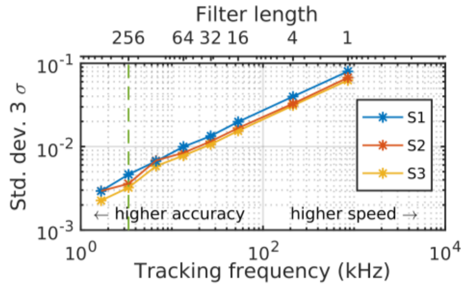

Fig. 2: Standard deviation of the Stokes parameters vs. the obtained tracking frequency (inversely proportional to the averaging filter length).

'classical laboratory' setup - a coherent mixer, 4 balanced photodiodes and an oscilloscope - and offline processing. We load the system with ASE noise to meet realistic system conditions, at an optical signal to noise ratio of 10.5 dB in 0.1 nm. Fig. 2 reports the measured evolution of the standard deviation σ of the Stokes parameters as a function of window duration, highlighting a square root-law relationship, which seems to confirm the Gaussian nature of the extracted H_{ij} distribution. In the real-time testbed, we set the filter length to 256, to achieve a good compromise between the SOP accuracy and the need to capture a rich-enough signature of the events, hence operate at high speed. Then the three Stokes parameters are computed and rounded to 8-bits sign integers and sent to a computer via a serial interface. The computer collects new Stokes parameters at a frequency $f_{col.}$ = 1920 Hz. The SOP rotation speed is computed with Δt = #samples/$f_{col.}$ where #samples is set at 600 for all acquisitions. We set the SOP rotation speed threshold in step #3 to 0.7 rad/s. Should that limit be exceeded, 256 pre-triggered and 7872 post-triggered samples are collected in a database for classification. Fig. 1(c) shows an example of the evolution of the Stokes parameters as a function of time during one event and its representation on the Poincaré sphere. Thanks to the programmable robot arm, we emulate four types of events, referred to as "classes", namely "bending", "shaking", "small hit", "up and down". We then train the receiver to recognize them and assess the accuracy of the prediction when events occur randomly.

Classification of the data

In the training phase, we elaborate a function f such as $Y = f(X)$ where Y represents the class, and X is a vector of “explanatory variables” i.e. mathematical combinations[7] of all samples from the database and of the time when they were recorded. Once f is trained, any forthcoming event should be categorized as one of the predefined classes. In the experiments, we collect 16548 events divided into training and a test phases according to the 10-fold cross validation[6] process, whatever their class (stratified random sampling). To improve the efficiency of the learning process, we rotate the SOP at fiber input between events with a polarization scrambler, such that function f rapidly captures a common denominator from all event of the same class regardless the SOP before the event takes place. Fig. 3 shows an example of Stokes parameter traces of event successfully classified with a probability superior to 99.99% for each class.

The vector X is obtained following the 3-steps method: (*i*) P “explanatory variables”, are elaborated[7] (*ii*) then using a mathematical criterion[7] Q variables (a subset of the P variables) are selected for their informativeness, i.e. they bring information to predict Y, (*iii*) finally R explanatory variables (a subset of the Q variables) are kept as meaningful using a forward backward selection and a maximum a posteriori approach for variable selection[8]. The classifier uses a naïve Bayes classifier[9] relying on the R variables.

In Tab. 1 we give typical examples of what the classifier tool brought up as “explanatory variables” of interest. Tab. 2 gives the values of P, Q and R and the performances of the classifier using two criteria: the Area Under the receiver operating characteristic Curve (AUC)[10] and the Accuracy (ACC, the rate of good classification). The performances of the classifier tool is seen to grow with *P* until it reaches a limit where more variables do not bring any additional significant improvement. For *P*=1000 (*R*=18.2 ± 1.2) the classifier exhibits excellent results (AUC=99% and the prediction accuracy ACC is as high as 95%) with excellent robustness (ratio train/test results close to 1). This good ratio guarantees similar accuracy with random events.

Tab. 1: Examples of explanatory variables *of X*.

$\sigma_{S1}(t)$ where 396.6 ms < t < 660.2 ms
$\sigma_{S2}(t)$ where 396.6 ms < t < 660.2 ms
$\sigma_{S2}(t)$ where 263.5 ms < t < 395.6 ms
$\sigma_{S2}(t)$ where t > 1983.1 ms
$\sigma_{S1}(t)$ where 1718.5 ms < t< 1983.1 ms

Tab. 2: Number of constructed (*P*), informative (*Q*), used variables (*R*) and classification results obtained (mean ± standard deviation over the 10 test-folds, *± 0)

P	*Q*	*R*		After training	After test
10	9*	6.2 ± 0.9	AUC	0.93 ± 0.01	0.92 ± 0.01
			ACC	0.76 ± 0.02	0.76 ± 0.02
100	99*	11.9 ± 0.8	AUC	0.98 ± 0.01	0.98 ± 0.01
			ACC	0.91 ± 0.01	0.90 ± 0.01
1000	999*	18.2 ± 1.2	AUC	0.99 ± 0.01	0.99 ± 0.01
			ACC	0.95 ± 0.01	0.95 ± 0.01
10000	9999*	24 ± 1.3	AUC	0.99 ± 0.01	0.99 ± 0.01
			ACC	0.97 ± 0.01	0.96 ± 0.01

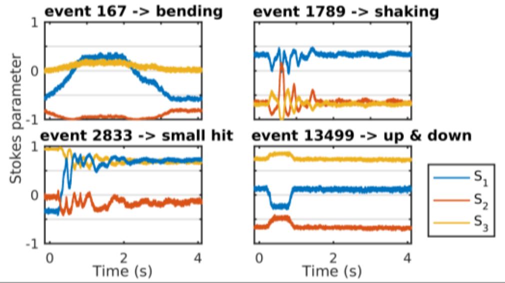

Fig. 3: Recording of Stokes parameter of sample events classified with a probability of 0.99997174 ± 10^{-9} for P=10000.

Another interesting result in Tab. 2 is the small number *R* of variables needed for classification, all expected to be computed in real time in a commercial transceiver. This number confirm that the classifier is implementable with relatively low complexity. Moreover, most of variable are easy to preprocess online for incremental learning[11] and fast predictions.

Conclusions

We demonstrated SOP monitoring as add-on feature to a real-time coherent receiver. We elaborated an extension of DSP with limited extraction rate, followed by optimized averaging. We showed that a naïve Bayes classifier could successfully recognize “bending”, “shaking”, “small hit”, “up and down” events with >95% reliability. In addition, our approach is low complexity for an implementation in a coherent transceiver.

References

[1] J. Pesic et al., “Proactive restoration of optical links based on the classification of events,” Proc. ONDM, (2011).

[2] J. E. Simsarian et al., “Shake Before Break: Per-Span Fiber Sensing with In-Line Polarization Monitoring,” Proc. OFC, M2E.6 (2017).

[3] K. Christodoulopoulos et al., “Exploiting network kriging for fault localization,” Proc. OFC, W1B.5 (2016).

[4] F. N. Hauske et al., “Optical performance monitoring in digital coherent receivers,” J. Lightwave Technol., Vol. **27**, p. 3623 (2009).

[5] E. Dutisseuil et al., “34 Gb/s PDM-QPSK coherent receiver using SiGe ADCs and a single FPGA for digital signal processing,” Proc. OFC, OM3H.7 (2012).

[6] https://en.wikipedia.org/wiki/Cross-validation_(statistics)

[7] M. Boullé, “Towards Automatic Feature Construction for Supervised Classification,” Proc. ECML/PKDD, (2014).

[8] M. Boullé, “Compression-Based Averaging of Selective Naive Bayes Classifiers,” J. of Machine Learning Research, Vol. **8**, p. 1659 (2007).

[9] P. Langley et al., “An analysis of Bayesian classifiers,” Proc. of National Conference on Artificial Intelligence, Vol. **415**, (1992).

[10] T. Fawcett, “ROC graphs: note and practical considerations of researcher,” Machine Learning, Vol. **31** (2004).

[11] V. Lemaire et al., "A survey on supervised classification on data streams", Lecture Notes in Business Information Processing, Vol. **205**, p 88 (2015).