Non-Data-Aided Feedforward Timing Recovery for Flexible Transceivers Employing PDM-MQAM Modulations

Mohamed Morsy-Osman*, Mathieu Chagnon, Qunbi Zhuge, Xian Xu, and David V. Plant
Department of Electrical and Computer Engineering
McGill University, Montreal, QC, Canada H3A 2A7
mohamed.osman2@mail.mcgill.ca

Abstract: A blind feedforward timing estimator using 2 samples/symbol that is modulation format transparent is modified for PDM signals. When used for interpolator control, sampling frequency offsets up to 5000 ppm are corrected experimentally for PDM-QPSK, -16QAM and -64QAM.

OCIS codes: (060.1660) Coherent communications; (060.2330) Fiber optics communications; (060.4080) Modulation.

1. Introduction
Flexible digital signal processing (DSP)-based coherent transceivers that are modulation format agile are key elements in the envisioned future software-defined optical networks [1]. Therefore, algorithms that are transparent to modulation format are attractive technology candidates for implementation in those transceivers. Among various algorithms, the task of digital timing recovery is obtaining synchronous samples commensurate with the symbol rate from the received asynchronous samples at the ADC sampling rate, i.e. it corrects the sampling frequency offset (SFO) between the transmitter and receiver clocks. Non-data-aided (NDA) (or blind) digital timing recovery algorithms are categorized into feedforward and feedback approaches. Although feedback algorithms usually require fewer computations than feedforward counterparts, they are often not transparent to modulation format. For example, the widely-known Gardner algorithm [2] operates only on BPSK and QPSK signals and also suffers from a long acquisition time until the feedback loop is locked.

In this paper, we modify the NDA feedforward timing error estimator in [3] for PDM signals. The proposed estimator operates at 2 samples/symbol and is transparent to modulation format, tolerant to polarization rotation and relatively tolerant to carrier frequency offset. The timing error estimate obtained is used to control a piecewise-parabolic interpolator in order to compensate for SFO. The proposed timing recovery is experimentally verified on 28 Gbaud PDM-QPSK, PDM-16QAM and 7 Gbaud PDM-64QAM signals with up to 5000 ppm SFO being corrected at small penalties both in back-to-back and transmission scenarios.

2. Principles of the timing error estimator and interpolator control
Assuming CD is compensated first, polarization mode dispersion (PMD) and fiber Kerr nonlinearity are neglected, the Jones vector of the received continuous-time signal \( r(t) \) can be expressed in terms of the Jones vector of the transmitted symbols at the \( k \)th baud position \( \mathbf{s}_k = \begin{bmatrix} s_k^X \\ s_k^Y \end{bmatrix} \) as follows
\[
r(t) = \sum_k \mathbf{J} \mathbf{s}_k h(t - kT - \zeta T)
\] (1)
where \( X \) and \( Y \) denote the modulated orthogonal polarization, \( \mathbf{J} \) is a unitary 2x2 matrix that represents the overall Jones rotation, \( \zeta \in (-0.5,0.5] \) is the normalized symbol timing delay which is the parameter to be estimated. Also, the transmitted symbols on both polarizations \( s_k^X \) and \( s_k^Y \) are independent and can be drawn from any linear and memoryless modulation, e.g. \( M \)-ary QAM with \( M = \{4, 16, 32, 64, \ldots \} \). Next, the real and imaginary parts of each polarization in the waveform given by Eq. (1) are sampled by means of 4 ADCs with a sampling interval \( T_{\text{ADC}} \) that is nominally half the baud duration, i.e. \( T_{\text{ADC}} = T/2 \) in case of zero SFO, leading to the following discrete-time samples
\[
r[n] = r(nT_{\text{ADC}}) = \sum_k \mathbf{J} \mathbf{s}_k h(nT_{\text{ADC}} - kT - \zeta T) = \sum_k \mathbf{J} \mathbf{s}_k h[n - 2k]
\] (2)
where \( h[n] = h(nT_{\text{ADC}} - \zeta T) \) are the samples obtained from the pulse shape \( h(t) \) at a rate \( 1/T_{\text{ADC}} \) after time-shifting by \( \zeta T \). Basically, \( \zeta \) represents a sampling phase offset resulting in the received samples being \( \zeta T \) away from the optimum sampling points. If there is no SFO, \( \zeta \) remains constant over the length of the received samples. However, when there is non-zero (but constant) SFO, the sampling phase offset \( \zeta \) will be linearly time-varying. Over a block of received samples with length \( N \), we can assume that \( \zeta \) stays approximately constant provided that \( N < \approx 1/SFO \) where \( SFO \) is the sampling frequency offset in ppm given by \( SFO = \Delta T_{\text{ADC}} \) where \( \Delta \) is the sampling frequency offset in Hz. Given the above condition is met, an estimate \( \hat{\zeta}_i \) of the normalized timing error within the \( i \)th block of
samples can be formulated by modifying the estimator in [3], which was proposed for a one-dimensional (single-polarization) signal, as follows

\[
\hat{\xi} = \frac{1}{2\pi} \arg \left\{ \sum_{n=0}^{N/2-1} (-1)^n \cdot r^H[n] r[n] + j \sum_{n=0}^{N/2} (-1)^n \cdot r^H[n] r[n+1] \right\}
\]

where \( \arg \{ \} \) and \( \Re \{ \} \) denote the argument and real part of the quantity inside the curly brackets, respectively, and the superscript \( H \) denotes the Hermitian transpose operator. Different from the original estimator in [3], we evaluate the product \( r^H r \) in the first and second terms of Eq. (3) which means that samples on both polarizations are used in the estimator while essentially removing the polarization crosstalk effect which improves the estimation accuracy (note that \( J^H J = I \) where \( I \) is the identity matrix). The estimator in Eq. (3) requires 4 real multiplications per sample per polarization. It needs to be re-computed for each block of \( N \) samples and the difference between the sampling phase estimates of two successive blocks can be used to control a polynomial based interpolator. Out of various interpolating polynomials in the literature, we use the piecewise-parabolic interpolator in [4] where each synchronous sample \( r_{\text{interp}}[n] \) at the output is evaluated from a set of 4 asynchronous received samples at its input as

\[
r_{\text{interp}}[n] = r[m_n + 2] \left[ 0.5 \mu_n - 0.5 \mu_{n+1} \right] + r[m_n + 1] \left[ -0.5 \mu_n + 1.5 \mu_{n+1} \right] + r[m_n] \left[ -0.5 \mu_n - 0.5 \mu_{n+1} + 1 \right] + r[m_n - 1] \left[ 0.5 \mu_n^2 - 0.5 \mu_{n+1} \right]
\]

where \( m_n \) represents the base index of the sample set being interpolated and \( \mu_n \) is the fractional delay that determines the interpolator coefficients. Within the \( i^{th} \) block of \( N \) samples, \( m_n \) and \( \mu_n \) are updated for each interpolator output sample such that \( m_n \) is incremented by 1 and \( \mu_n \) stays the same within the block. As soon as \( m_n \) reaches the last sample in the \( i^{th} \) block, both \( m_n \) and \( \mu_n \) are updated for the next block based on the difference between the estimated sampling phase offsets of the next and current blocks. Similar to [5], this recursion is formulated as follows

\[
m_{n+1} = \begin{cases} m_n + 1, & \text{for } i N < m_n < (i+1) N - 1 \\ m_n + [\mu_n + 1 + \text{SAW}(2 \xi_{i+1} - 2 \xi_i)]_N, & \text{for } m_n = (i+1) N - 1 \end{cases}
\]

\[
\mu_{n+1} = \begin{cases} \mu_n, & \text{for } i N < m_n < (i+1) N - 1 \\ \mu_n + 1 + \text{SAW}(2 \xi_{i+1} - 2 \xi_i) \mod 1, & \text{for } m_n = (i+1) N - 1 \end{cases}
\]

where \( \text{SAW}(x) = 2(0.5 x + 0.5) - 2(0.5 x + 0.5)^2 - 1 \) is the sawtooth function with a period of 2 which wraps the difference between the two timing estimates of consecutive blocks. In the second line of Eq. (5), \( m_{n+1} \) can be either \( m_{n+1} + 2 \), \( m_{n+1} \), or \( m_{n+1} + 1 \) meaning either a sample dropping, a sample overlap, or neither of them, is introduced between the two successive blocks, respectively. Sample dropping (or overlap) occurs when the ADC clock is running faster (or slower) than twice the baud rate. Fig. 1(b) shows the time evolution of \( \mu_n \) and the increment \( m_{n+1} - m_n \) for \( N = 64 \) at two different SFO levels of 1000 (left) and 4000 ppm (right). In the latter case, the increments in \( \mu_n \) from a block to the next are larger and sample dropping happens more frequently.

3. Experimental setup, offline DSP, results and discussion

The experimental setup is shown in Fig. 1(a). The in-phase (I) and quadrature (Q) signals, having either 2, 4 or 8 levels for various modulations, were generated from two MICRAM digital-to-analog converters (DACs) driven by two Xilinx FPGAs. The symbol clock was set at 28 Gbaud for QPSK and 16QAM and at 7 Gbaud for 64QAM. Then, the DAC outputs were fed to a QAM transmitter to modulate an external cavity laser (ECL) with linewidth less than 100 kHz. PDM was emulated and the signal was launched at the optimum launch power into a recirculating loop containing 4×80 km of SMF-28e+ fiber, each followed by an EDFA having a noise figure of 5 dB. At the output of the loop, noise loading allows varying the OSNR. After filtering and pre-amplification, the signal is fed to a Si-photonic (SiP) coherent receiver (CRx) from TeraXion. An ECL similar to the transmitter ECL was used as an LO. The four CRx outputs were sampled and recorded by one Agilent 80 GSa/s real-time scope (RTS).

Offline processing starts by IQ imbalance and quadrature error correction, resampling to 2 samples per symbol, chromatic dispersion (CD) compensation. Then, the SFO present in the captured data, although being very small at around 30 kHz, was corrected using the proposed timing recovery where \( N \) was set to 16384 samples since the SFO being tracked is very small. Then, the data was resampled depending on the SFO under which the algorithm is desired to be tested, e.g. an SFO of 1000 ppm corresponds to resampling the data from 2 to 2.002 samples per symbol. Next, the SFO introduced into the data was corrected using the proposed timing recovery. Next, laser frequency offset is removed using the periodogram method [6]. Polarization is then demultiplexed using a 31-tap butterfly equalizer whose taps are updated using either the constant modulus algorithm for PDM-QPSK or the least mean squares algorithm in training mode followed by decision-directed mode for PDM-16QAM and 64-QAM [7]. Phase noise was mitigated by a decision-directed phase locked loop (PLL). Finally, the BER is evaluated.

First, we study the impact of varying \( N \) on the performance of the algorithm. In Fig. 1(c), we compare the back-to-back constellations (with no noise loading) of all modulations when \( \text{SFO} = 4000 \) ppm (for different block sizes)
with the case of $SFO = 0$. We notice that under such high SFO, increasing $N$ deteriorates the performance until the algorithm fails if $N > 1/SFO$ which is the case for the green constellation in case of QPSK where $N = 256$. On the other hand, reducing $N$ allows the algorithm to track larger SFO but affects the estimation accuracy especially in presence of noise. For this reason, we plot in the top insets of Fig. 2(a) the BER versus $N$ for all three formats for different SFOs in back-to-back with noise loading at the OSNRs indicated on the figures. We notice that the optimum $N$ decreases as SFO increases however, the algorithm fails when using $N$ below 32. Also, we notice that $N = 32$ always produces the best performance for 64QAM since the OSNR level is the highest among all formats. For all forthcoming results, we fix $N$ at 64 for both QPSK and 16QAM and at 32 for 64QAM. Next, we show in Fig. 2(a) the back-to-back BER versus OSNR for all three modulations at different SFO levels. As a reference, we also plot the theoretical BERs versus OSNR for all formats. Various formats are drawn with different colors whereas different SFOs are distinguished with markers. For QPSK in case of $SFO = 5000$ ppm, we noticed a very small OSNR penalty of 0.15 dB (measured relative to the case of zero SFO at the FEC threshold). For both 16QAM and 64QAM, the OSNR penalty in case of a large SFO of 5000 ppm increases to 1.2 and 5.5 dB, respectively. However, the OSNR penalty for both formats is still very small when the SFO is 1000 ppm and increases when SFO is 3000 ppm to only 0.5 and 1.4 dB for 16QAM and 64QAM, respectively. Finally, we evaluate the performance of the algorithm on transmitted data where we plot in Fig. 2(b) the BER versus distance for all formats. We notice some reduction in the reach for all formats depending on the SFO level. For example, the reach reduction was almost negligible when $SFO = 1000$ ppm. At $SFO = 3000$ ppm, the reach slightly decreases by $\sim 100$ and 70 km for 16QAM and 64QAM, respectively corresponding to $\sim 9\%$ and 16\% relative degradation.

5. References