International Journal of Satellite Communications and Networking

WILEY

SPECIAL ISSUE PAPER OPEN ACCESS

Review and Analysis of Digital Signal Processing Algorithms for Coherent Optical Satellite Links

Carl Valjus[1,2] | Raphael Wolf[1] | Juraj Poliak[1]

[1]Institute of Communications and Navigation, German Aerospace Center, Oberpfaffenhofen, Germany | [2]Department of Communication Electronics, University of Bremen, Bremen, Germany

Correspondence: Carl Valjus (carl.valjus@dlr.de)

Received: 22 January 2024 | **Revised:** 18 September 2024 | **Accepted:** 21 January 2025

Funding: This study was funded by Deutscher Akademischer Austauschdienst.

Keywords: digital signal processing | optical satellite communications | satellite networks

ABSTRACT

Coherent optical satellite links enable high-throughput communication and high accuracy ranging to and between satellites. Due to the ever-increasing demand for throughput, wavelength division multiplexing of polarization multiplexed optical signals is being considered as a solution to provide high-speed optical satellite links. Fiber-optic systems solve the implementation scalability problem of these systems by shifting design complexity to integrated circuits, thereby massively reducing the system footprint. As a result of the major advances in complementary metal-oxide-semiconductor (CMOS) technology, the implementation scalability of such systems in terrestrial fiber systems has been solved by shifting the system complexity to digital hardware, enabling intradyne reception and complex signal recovery algorithms. While the use of fiber-optic transceivers provides a fast path to high-speed coherent optical satellite links (OSLs), it requires additional mitigation techniques to combat the effects of both the OSL channel and the space environment. To support future satellite networks with Tbit/s optical links, it will be critical to further minimize the size, weight, and power (SWaP), cost and reliability of the transceivers. Thus, the development of custom intradyne optical transceivers for OSLs is emerging as an attractive option as the demand for throughput in satellite networks continues to grow. This would not only enable the use of a more optimized signal processing chain but also enable the use of radiation mitigation techniques optimized for the signal processing architecture and the use of soft-decision forward error correction (FEC) optimized for OSLs. The signal processing of coherent optical satellite receivers can be divided into three key subsystems: timing recovery, carrier synchronization, and equalization. This paper reviews state-of-the-art digital signal processing for optical communication to identify suitable algorithms for timing recovery, carrier frequency and phase compensation, equalization, and polarization demultiplexing with emphasis on high-throughput optical satellite links. Finally, the performance of different digital signal processing algorithms is assessed by numerical simulations considering different optical satellite link scenarios.

1 | Introduction

Current satellite communication systems are mainly based on radio-frequency (RF) transmission, but as the demand for throughput increases, the limited amount of bandwidth available in the RF spectrum will become a bottleneck. One solution that has been proposed is to use free-space optical (FSO) communication systems to transmit data between satellites as well as between ground and satellites [1]. Optical frequencies provide bandwidth in the order of 10 THz, which significantly increases the potential throughput by several orders of magnitude compared with RF systems. However, there are many technical challenges to deploying optical satellite links (OSL), one of the most significant being atmospheric turbulence, which

https://doi.org/10.1002/sat.1553

distorts the phase of the light and causes fluctuations in the intensity of the light, making it difficult to achieve reliable links. Another challenge is the long distance the light must travel to the satellite, resulting in large free-space losses. Coherent communications combined with WDM has been identified as a promising technology to achieve both high spectral efficiency and high sensitivity in OSL [2, 3]. The implementation of coherent communication systems is inherently more complex than, for example, direct detection systems because information is encoded in both the amplitude and phase of the signal. As a result, more complex and demanding optical devices are required in the receiver, and many different signal impairments must be compensated to enable successful decoding of the received signal. As a result of major advances in complementary metal-oxide-semiconductor (CMOS) technology, the implementation scalability of such systems in terrestrial fiber networks has been solved by shifting the system complexity to digital hardware, enabling complex signal recovery algorithms. High-speed analog-to-digital converters (ADCs), digital signal processing (DSP) algorithms, and forward error correction (FEC) algorithms have been integrated on a single chip, enabling intradyne coherent transceivers with a small footprint [4]. As a result, coherent optical communication has revolutionized terrestrial fiber optic networks, enabling Tbit/s data rates over a single fiber.

Terrestrial fiber-optic transceivers could also be used for OSLs, but additional fading mitigation techniques must be applied to the terrestrial transceivers to achieve reliable transmission through the atmosphere. This could be accomplished either by implementing custom automatic repeat request (ARQ) protocols or additional FEC on top of the terrestrial transceivers; in fact, up to 200Gb/s low Earth orbit (LEO) to ground downlinks have been demonstrated using terrestrial transceivers combined with ARQ [5]. On the other hand, implementing custom DSP tailored for OSLs could significantly reduce the size, weight, and power (SWaP), as terrestrial transceivers feature large subsystems that are not required for OSLs, such as chromatic dispersion compensation and FEC optimized for fiber channels. Custom DSP could also be designed to better handle atmospheric turbulence and allow tighter integration with the FEC subsystem, for example, to enable soft-decision FEC codes. Not only would this allow the use of a more optimized signal processing chain, it would also allow the use of radiation mitigation techniques optimized for the signal processing architecture and semiconductor nodes better suited to the space environment.

In general, an intradyne coherent receiver can be divided into three main subsystems, as shown in Figure 1. First, the optical front-end, which is responsible for splitting the incoming beam into two polarizations with a polarization beam splitter (PBS), mixing the received signal with the local oscillator (LO), and converting the optical signal into an electrical baseband signal with balanced photodetectors (BDs). In the subsequent analog stage are the electrical signals amplified by transimpedance amplifiers (TIAs) and sampled by ADCs. Finally, the DSP is used to recover and decode the signal. The DSP stage, which is the focus of this paper, can be further divided into several subsystems. First, a static compensation stage can be used to compensate for known impairments such as timing skews in the analog signal paths. A timing recovery stage is then used to resample and synchronize the signal to the transmitter symbol rate. This is followed by adaptive equalization, which restores the original state-of-polarization (SOP) of the signal and compensates for the channel impulse response. Next, the carrier recovery stage is used to compensate for nonidealities in the optical front-end, such as phase noise and carrier frequency offset between the transmitter and receiver. Finally, the signal can be decoded, and errors can be corrected using FEC. This paper presents a systematic review of state-of-the-art DSP for optical communication to identify suitable signal processing algorithms for different OSL scenarios with a focus on timing recovery, carrier recovery, and adaptive equalization algorithms with a focus on QPSK modulated signals. First, Section 2 gives a brief introduction to the atmospheric channel model and introduces four different OSL scenarios that will be used in this paper to analyze the performance of different DSP algorithms. Sections 3–6 reviews state-of-the-art timing recovery, carrier phase noise compensation, carrier frequency offset compensation, and adaptive equalization for coherent OSL. Although static compensation and FEC algorithms are important to a complete system, they are only briefly discussed in this

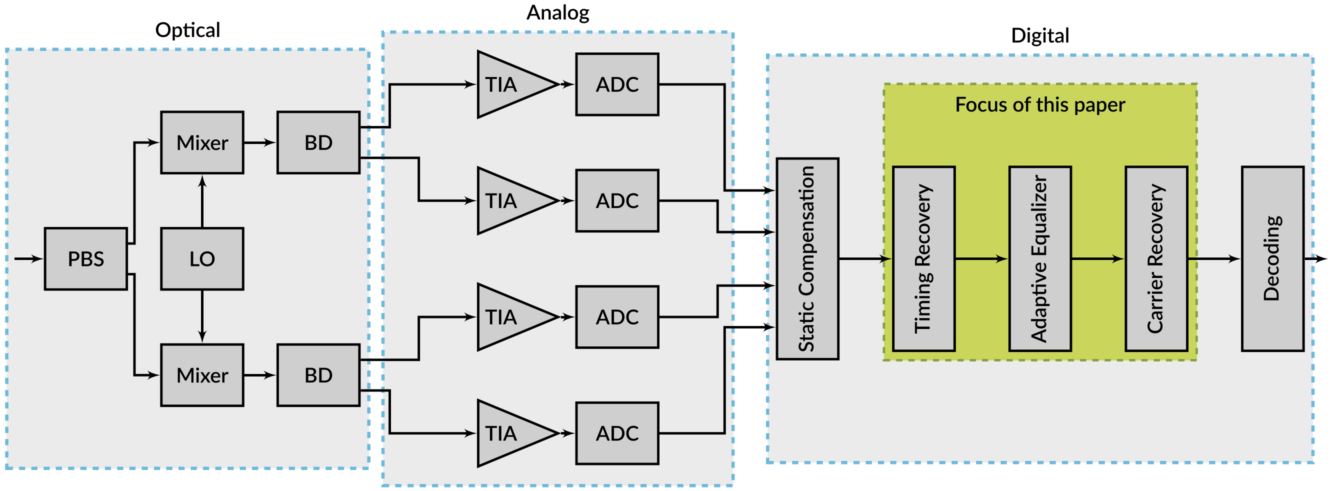

FIGURE 1 | Typical structure of an intradyne coherent receiver. Here, the receiver has been divided into three main subsystems: optical, analog, and digital.

paper. Finally, the potential benefits of different implementation architectures are summarized in Section 7.

2 | Channel Model

Optical links between satellite and ground must propagate through the atmosphere. The refractive index of air changes at small and large scales, distorting the phase and causing fluctuations in the intensity of the light. In addition, pointing errors induced by beam wander and mechanical vibrations will introduce additional intensity variations. Accurate parameterization of atmospheric turbulence depends on many parameters such as weather conditions and elevation angle. Because the focus of this paper is on DSP, we will not delve into the topic of accurate channel modeling but instead consider four broad scenarios to illustrate and analyze the strengths and weaknesses of different signal processing algorithms. The first scenario considered is for an optical inter-satellite link (ISL). In this case, the beam does not propagate through the atmosphere and intensity fluctuations of the received signal are only caused by pointing jitter for which the probability density function (PDF) of the normalized received beam intensity I can be modeled as a beta distribution with the mean intensity $\bar{I}$ given by [6]

$$\bar{I} = \frac{\beta}{\beta + 1} \tag{1}$$

where β is given by the beam divergence θ_0 ($1/e^2$ half-width) and the pointing error jitter rms σ_{jitt} as

$$\beta = \frac{\theta_0^2}{4\sigma_{jitt}^2} \tag{2}$$

The second scenario considers an optical satellite-ground downlink. In this case, the main cause of intensity fluctuations is due to atmospheric turbulence. The intensity variations can in this case be modeled as a lognormal distribution with mean 1 and the variance σ^2 given by

$$\sigma^2 = e^{\sigma_p^2} - 1 \tag{3}$$

where σ_p is the scinintillation index. Typically, aperture averaging can be applied in a downlink to reduce the scintillation index compared with uplinks. Finally, the third and fourth scenarios describe an uplink scenario, where a combined PDF of both pointing errors and atmospheric turbulence has to be considered [7]. The parameters used to describe the four different scenarios are shown in Table 1, and the resulting normalized power PDFs are shown in Figure 2, where the paramters have been chosen by considering the following references [8–11] and where the normalized power has been generated by taking 10^8 random samples from the PDFs. However, for analyzing the performance of signal processing algorithms, the signal-to-noise ratio (SNR) is the main parameter of interest. For a pre-amplified coherent receiver, the SNR can be assumed to be proportional to the received power [12]; hence, in this paper, we assume that the SNR has the same distribution as the received power. For example, this means that when considering Scenario 2, the SNR

TABLE 1 | Parameters for generating the four channel scenarios considered in this paper.

Scenario	Description	σ_p^2	θ_0 [μ rad]	σ_{jitt} [μ rad]
1	Inter-satellite link	—	25	3
2	Downlink	0.029	—	—
3	Uplink	0.15	3	1.7
4	Uplink	0.25	34.5	25.6

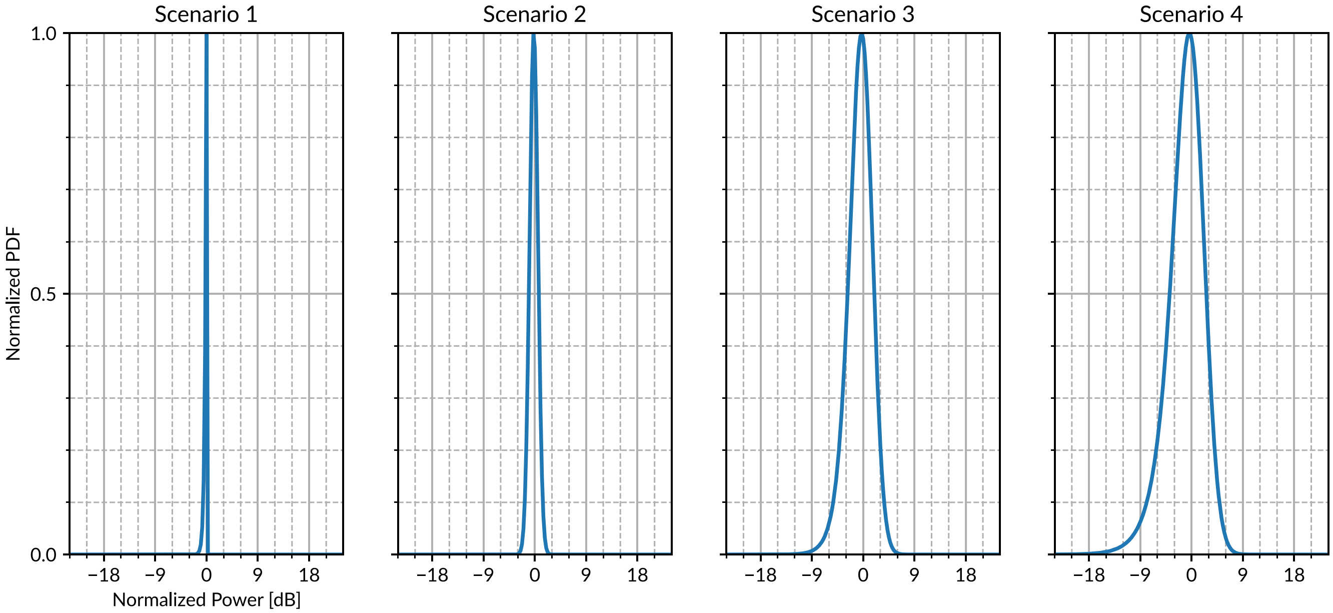

FIGURE 2 | Illustration of the PDF for the four different scenarios studied in this paper.

will mainly be ± 3 dB around a chosen baseline SNR. Where the baseline SNR will be chosen such that an average BER of 10^{-3} is achieved for each scenario in this paper. The average BER is calculated by evaluating the BER for the possible integer SNR dB values through simulations of different DSP algorithms. The BER as a function of SNR can then be approximated, and the average BER for a given scenario is then calculated with the SNR distribution.

It is possible to target a lower BER than 10^{-3} by using for example soft-decision FEC to recover bit errors. However, the average BER assumes an infinite interleaver, but for real applications, a finite interleaver length has to be chosen. As a consequence, a trade-off has to be made between the frame error rate, interleaver length, FEC scheme, and implementation complexity [13, 14]. Though these are important aspects of the practical implementation of a system, this paper aims to provide a broad analysis and review of DSP algorithms; thus, the performance of algorithms will instead mainly be evaluated with respect to the average SNR penalty of the algorithms, that is, how much the average SNR approximately has to be increased to achieve the same average BER as the optimal solution to the problem at hand. An important parameter to determine the appropriate interleaver length is the coherence time of the channel, that is, how long the channel can be assumed to remain constant. The scintillation caused by atmospheric turbulence and pointing errors typically has a coherence time of more than 1 ms. Thus, for a communication system with a high symbol rate on the order of several GBaud, it can be assumed that the channel conditions remain the same for millions of symbols. As most DSP algorithms work with much less symbols, and to simplify the analysis of this paper, we will not model the coherence time in this paper, and instead consider the channel quasi-static. Other impairments such as carrier phase noise will be discussed in the following sections.

3 | Timing Recovery

One of the most critical subsystems within the receiver is the timing recovery block, also known as clock recovery. It is necessary to continuously adjust the phase and frequency of the receiver clock such that the received symbols are sampled close to the optimum sampling point. Typically, this is achieved by setting the sampling frequency of the ADC close to the desired sampling frequency.

This is then followed by fine compensation of the sampling frequency offset using digital signal processing [15]. Digital timing recovery algorithms can be broadly divided into two different classes: feedforward- and feedback-based algorithms. In a feedback system, the received signal is first resampled, and then the timing error is estimated by the timing error detector (TED). The error signal is subsequently filtered and used to drive a numerically controlled oscillator (NCO), which adjusts the phase of the resampling filter as depicted in Figure 3a. Both latency of the feedback path and tuning of the filter are key factors in being able to track large clock offsets.

In contrast, a feedforward system first estimates the sampling clock offset of a block of the unaltered received signal and then resamples the signal accordingly as illustrated in Figure 3b. The main advantage of the feedforward approach is that the phase of the signal is obtained immediately, eliminating the need for an acquisition phase. For this reason, feedforward-based approaches are an attractive solution for bursty communication systems such as passive optical networks [16]. Similarly, feedforward-based methods have been proposed as a good solution for FSO links, where fast acquisition after fades is required [17]. However, the phase estimation algorithms required for feedforward timing recovery are generally more complex when compared with feedback algorithms. The impact of different system parameters such as jitter [18], differential group delay [19, 20], loop latency [21, 22], excess bandwidth [23–26], and polarization effects [27] have been studied thoroughly for fiber-optical communication systems. Nevertheless, timing recovery algorithms designed for terrestrial fiber links are optimized for relatively stable channels and do not address the behavior of the algorithms in more dynamic channels, such as the atmospheric channel, where timing recovery stability at low SNR is critical [28]. This section provides an overview of existing timing recovery algorithms and resampling methods. Finally, a comparison and analysis of the different methods will be given with respect to the four different scenarios that have been introduced earlier.

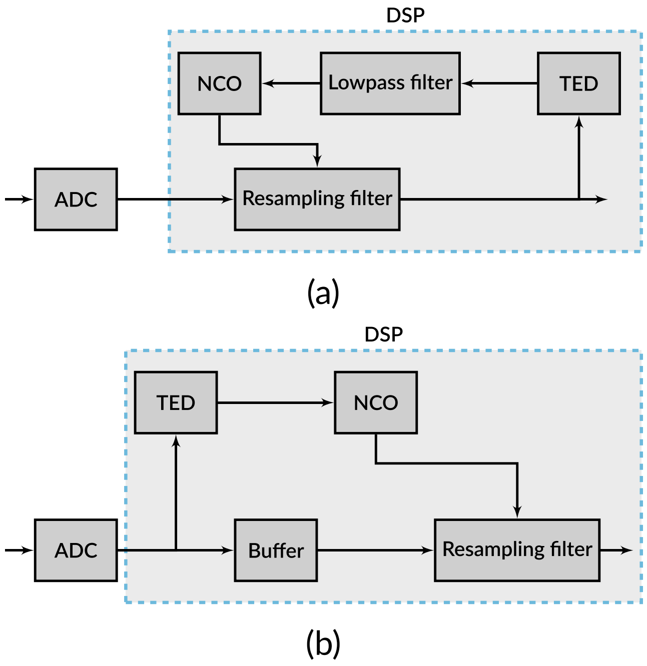

FIGURE 3 | Timing recovery architectures: (a) Feedback-based timing recovery. (b) Feedforward-based timing recovery.

3.1 | Feedback Algorithms

One of the simplest timing recovery algorithms in communication systems is the algorithm first proposed by Gardner that requires an oversampling factor of two samples per symbol [29]. Given the received signal $r[n]$, the error signal ϵ to the loop filter is given for every other sample (each symbol) as

$$\epsilon = \Re[r^*[2n](r[2n-1] - r[2n+1])] \tag{4}$$

where n is the sample index of the received signal. Thanks to its simplicity, the algorithm is widely used in many different types of communication systems, including coherent optical communication. However, the original algorithm proposed by Gardner requires processing the received signal at baud rate. High-speed receivers require parallel implementation of the signal processing algorithms, so instead a modification of the original algorithm is frequently used, in which multiple error signals are added together every clock cycle [30]. The parallel Gardner algorithm is then given by

$$\epsilon = \sum_{n=0}^{N/2-1} \Re[r^*[2n](r[2n-1]-r[2n+1])] \tag{5}$$

where the error signal ϵ is given for a block of N samples. Although the Gardner algorithm is insensitive to the carrier phase of the signal, its performance degrades as the roll-off factor of the received signal approaches zero. In order to achieve better performance with small roll-off factors, the instantaneous power of the signal can be used in place of $r[n]$ in Equation (5) [23] but at the cost of increased complexity and loop latency. A potential problem with the Gardner algorithm is that the amplitude of the error signal ϵ depends on the amplitude of the received signal, which can make it difficult to properly tune the loop filter, especially in a dynamic channel. One solution that has been proposed is to use an adaptive loop gain based on the channel conditions [22]. Another solution proposed by Gu et al. [28] is to use a timing recovery algorithm that is independent of the signal amplitude. The timing error estimator proposed by Gu et al. and hereinafter referred to as the Gu algorithm is given by

$$\epsilon = \arg\sum_{n=0}^{N-2} \mathrm{sign}(r[n]+jr[n+1])\mathrm{sign}(r^*[n]+jr^*[n+1])(-1)^n \tag{6}$$

where $\mathrm{sign}(x)$ is the complex sign function defined as $\mathrm{sign}(x) = \mathrm{sgn}(\Re(x)) + 1j\,\mathrm{sgn}(\Im(x)))$. However, the sign operation makes the algorithm sensitive to carrier phase noise in the case of BSPK and QPSK signals. To show this, the mean output of the TED for different symbol timing offsets is simulated for different carrier phase offsets with QPSK symbols. The resulting plot, commonly referred to as the S-curve is shown in Figure 4, where it can be seen that a carrier phase shift changes the output of the algorithm and thus making it unsuitable for QPSK signals as the carrier synchronization is typically done after the timing recovery. The output of the Gardner TED is also shown in Figure 4, where it can be seen that it is not affected by the carrier phase offset. One possible solution is to only perform one of the sign operations [31] as shown in Equation (7). This maintains the low complexity of the algorithm as one of the multiplication factors remains $\pm 1 \pm 1j$ and results in relatively small changes in the TED output with respect to the carrier phase offset as shown in Figure 4.

$$\epsilon = \arg\sum_{n=0}^{N-2} \mathrm{sign}(r[n]+jr[n+1])(r^*[n]+jr^*[n+1])(-1)^n \tag{7}$$

In this paper, both of the sign operations are instead ignored, this results in a higher complexity algorithm but makes the TED more robust to the carrier phase as can be seen in Figure 4.

It can sometimes be advantageous to perform timing recovery in the frequency domain, especially when other signal processing blocks such as equalizers and resampling filters are implemented with frequency domain algorithms. A common method for performing timing recovery in the frequency domain has been proposed by Godard [32]. The estimated timing error ϵ is given by

$$\epsilon = \sum_{k=0}^{\frac{N}{2}-1} \Im\left[R[k]R^*\left[k+\frac{N}{2}\right]\right] \tag{8}$$

where $R[k]$ is the N-point Fourier transform of the received signal $r[n]$. The output of the Godard TED is similar to the output of the Gardner TED as can be seen in Figure 4. The complexity of the Godard estimator can be reduced by realizing that most of the terms in Equation (8) are negligible, because only the frequencies close to the symbol rate are relevant for estimating the clock offset [33]. The simplified estimate is then given by

$$\epsilon = \sum_{k=\frac{1-\beta}{4}N}^{\frac{1-\beta}{4}N-1} \Im\left[R[k]R^*\left[k+\frac{N}{2}\right]\right] \tag{9}$$

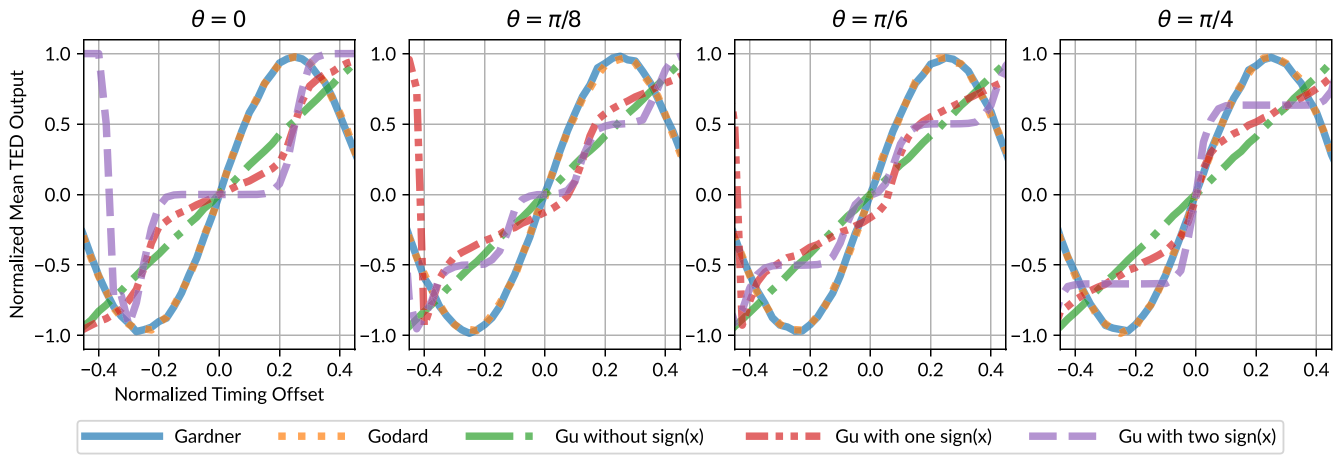

FIGURE 4 | Mean output of the different TEDs as a function of the symbol timing offset for different carrier phase offsets θ.

where β is the roll-off factor. Notably, it has been shown that Equation (9) also works well with sample rates below two samples per symbol [34].

3.2 | Feedforward Algorithms

Feedforward algorithms directly estimates the signal phase shift τ to be used by the resampling filter, unlike feedback algorithms, where the timing error ϵ must be filtered to achieve a good estimate. As a result, a feedforward approach typically require more complex estimation algorithms. Oerder and Meyer showed that it is possible to extract an absolute estimate of the signal phase by calculating the frequency component at the symbol rate of the squared received signal [35]. This yields the estimated signal phase $\hat{\theta}$ for a block of N samples as

$$\hat{\tau} = \frac{1}{2\pi}\arg\sum_{n=0}^{N-1}|r[n]|^2 e^{-\frac{j2\pi n}{k}} \tag{10}$$

where, k is the oversampling factor, $k \geq 4$. Typically, $k = 4$ as it results in $\exp -j2\pi n/k = -i^n$ and significantly reduces the complexity of the implementation. Still, an oversampling factor of four is required that is often hard to achieve in high-data rate systems.

Lee [36] proposed a modification of Equation (10) that reduces the required oversampling rate to only two samples per symbol but at the cost of increased complexity. The estimated phase shift $\hat{\tau}$ for a block of N samples is given by

$$\hat{\tau} = \frac{1}{2\pi}\arg\sum_{n=1}^{N}|r[n]|^2 e^{-jn\pi} + \Re[r[n]r^*[n-1]e^{-j(n-0.5)\pi}] \tag{11}$$

It is important to note that feedforward algorithms assume constant phase given a block of N samples, thereby limiting the frequency drift that can be tracked, especially at low SNR where a larger N is required.

3.3 | Resampling Algorithms

Estimating the correct sampling phase of the signal is only half the solution to a working timing recovery system; the incoming signal must also be resampled to the correct sampling phase. In addition, resampling filters could be used to compensate for the skew between the I and Q signals caused by the different analog signal paths. And while most timing recovery algorithms rely on two samples per symbol, the ADC can utilize lower sample rates if a digital resampling filter is used to upsample prior to the timing recovery circuit. For this reason, resampling can be divided into two stages, the first being a change in sample rate, for example, going from 4/3 samples per symbol to 2 samples per symbol, and a second stage that performs a phase shift of the signal. A sample rate conversion of a factor L/M is typically implemented by first upsampling the signal by a factor L by inserting $L-1$ zeros between each sample, the signal is then filtered and downsampled a factor M as shown in Figure 5a. As most inputs to the filter will be zeros and most of the output will be discarded, it is possible to apply efficient polyphase finite impulse response (FIR) filter structures to achieve the sample rate conversion. Alternatively, L/M upsampling can be achieved by frequency domain operations. By first performing an N point discrete Fourier transform (DFT), then padding the resulting spectrum with zeros, and finally applying a NL/M IDFT, as shown in Figure 5b. However, this requires either a non–power-of-two DFT or an IDFT, which is generally more complex to implement than a power-of-two DFT. Nevertheless, relatively efficient algorithms have been developed for special cases such as IDFTs of size $p2^n$, where p, n are positive integers [37]. To be able to process data continuously, the overlap-and-save method is generally used, that is, each DFT has a certain overlap with the adjacent DFTs, and the corresponding overlapping samples are discarded [38]. A small phase shift τ can also be efficiently implemented in the frequency domain. Given a signal $r(t)$ and its Fourier transform $\mathcal{F}(r(t))$, the time-shifted signal $r(t-\tau)$ is given by

$$\mathcal{F}(r(t-\tau)) = \mathcal{F}(r(t))\exp(-j2\pi f\tau) \tag{12}$$

which allows hardware efficient implementations of the entire timing recovery system with frequency domain operations [34]. In fact, Equation (12) describes the frequency response of the optimal interpolator [39]. Interpolation filters can also be designed for time domain implementation. The most basic way to achieve this is by linear interpolation between adjacent samples, that is, for a time shift τ the resulting signal $\hat{r}[k]$ is given by

$$\hat{r}[k] = \frac{\tau r[k] + (1-\tau)r[k+1]}{2} \tag{13}$$

where $r[k]$ is the input signal. Although linear interpolation provides a simple solution, filters with better performance can be implemented at the cost of higher complexity and are generally required for higher order modulation formats [40]. One of the

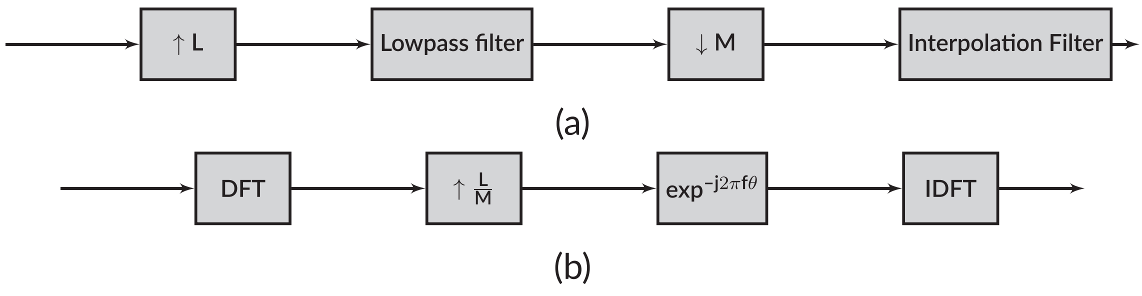

FIGURE 5 | (a) Time domain resampling. (b) Frequency domain resampling.

most common methods is to use a Lagrangian interpolation filter [41], where for an N tap FIR filter the coefficients h_i are given by

$$h_i(\tau) = \prod_{j=-N/2}^{N/2-1} \frac{\tau + i - j}{i - j} \qquad (14)$$

Typically, 14 is efficiently implemented with a Farrow structure [42], which reduces the number of multiplications required to just three per sample for a four-tap filter.

The disadvantage of implementing the interpolation filter with a Farrow structure is the relatively high latency caused by the critical path, which contains at least $N-1$ multiplications for an N-tap filter. For feedback timing recovery systems, the latency of the interpolation algorithm is critical, as it must be low enough to enable the timing recovery loop to track the clock drift [21]. As an alternative, Fu and Willson proposed a trigonometric interpolation method specifically for timing recovery, using a trigonometric polynomial instead of a polynomial to reduce latency [39]. The time shift τ of $r[n]$ using the trigonometric method is given by

$$r[n+\tau] = \frac{1}{N}\mathcal{R}(c_0 + 2\sum_{k=1}^{N/2-1} c_k e^{-2\pi k\tau/N} + c_{N/2}e^{j\pi\tau}) \qquad (15)$$

with $c_k(k \geq 0)$ given by

$$c_k = \sum_{d=-N/2+1}^{N/2} r[n+d]e^{-j2\pi kd/N} \qquad (16)$$

3.4 | Analysis

Computer simulations were performed to compare the timing recovery methods described above. The timing recovery algorithms described above are simulated using a root-raised cosine filter with a roll-off factor of 0.2, a simulated clock offset and additive white-gaussian noise (AWGN) between the transmitter and receiver with 10^7 symbols per SNR. All other receiver impairments such as clock jitter, carrier phase noise and polarization effects are not considered. The Gardner, Godard, and Gu feedback algorithms are implemented with a parallelization factor of 64 and 256 and a loop latency of 20 clock cycles. For comparison, the Lee algorithm is implemented with block sizes of 512, 1024, 2048, and 4096.

All algorithms are implemented with a 4-tap lagrange interpolation filter except for the Godard algorithm that uses frequency domain interpolation. As shown in [43], as the convergence time of the feedback-based timing recovery algorithms is typically much shorter than the coherence time of the channel, the performance of different algorithms can be estimated by evaluating the algorithms performance at different SNR values. Thus, by evaluating the relationship between the SNR and BER for the different algorithms, it is possible to compare the performance of different algorithms for the four scenarios. For each algorithm, the SNR penalty is calculated as the difference in SNR required to achieve a BER of 10^{-3} relative to a perfectly synchronized system.

The SNR penalty is plotted as a function of the clock frequency offset in Figure 6 for feedback algorithms and in Figure 7 for the Lee implementations. As discussed earlier, the performance of the interpolation filter also plays an important role in digital timing recovery systems. To compare the different interpolation filters, a similar approach is taken by computing the SNR penalty with respect to interpolation using a 2048-point fast Fourier transform (FFT) for different roll-off factors at an average BER of 10^{-3}. The resulting plots for the four different scenarios are shown in Figure 8.

From Figure 6, it can be seen that the Gardner and Godard algorithms appear to be the best options for all scenarios, and

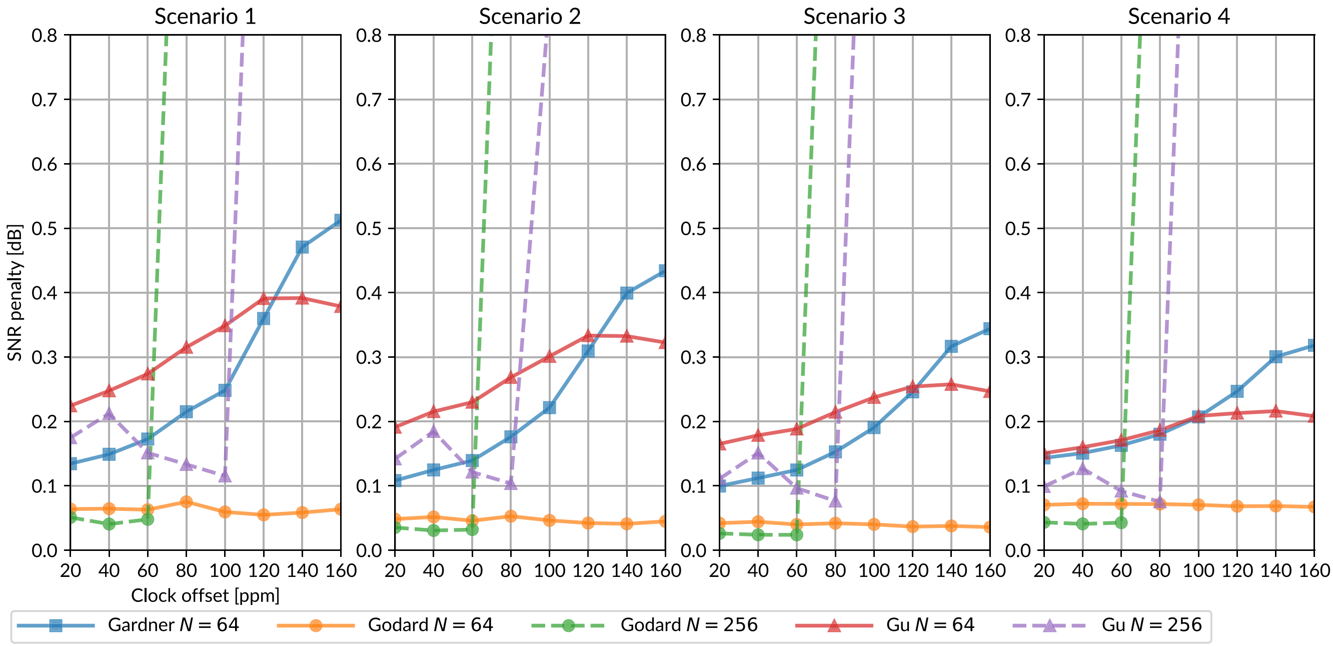

FIGURE 6 | SNR penalty for feedback timing recovery algorithms with a parallelization factor of 64 and 256 with a loop penalty of 20 clock cycles.

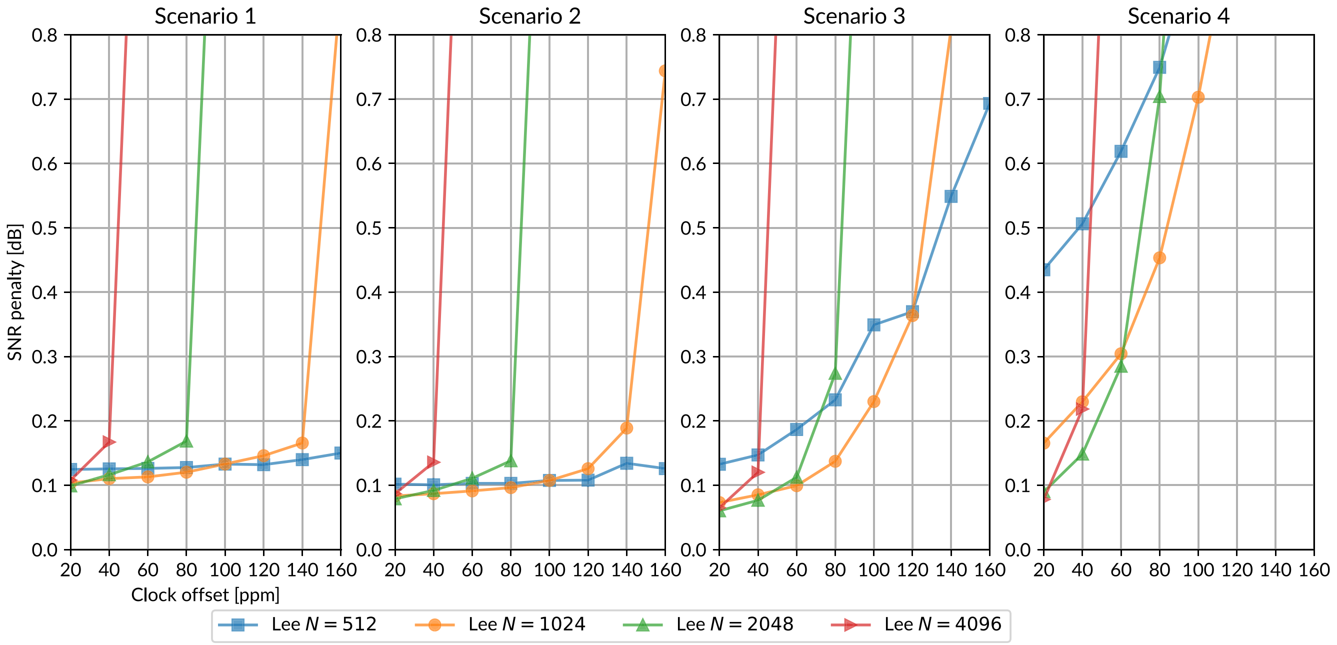

FIGURE 7 | SNR penalty for feedforward timing recovery with the Lee algorithm for different block sizes N.

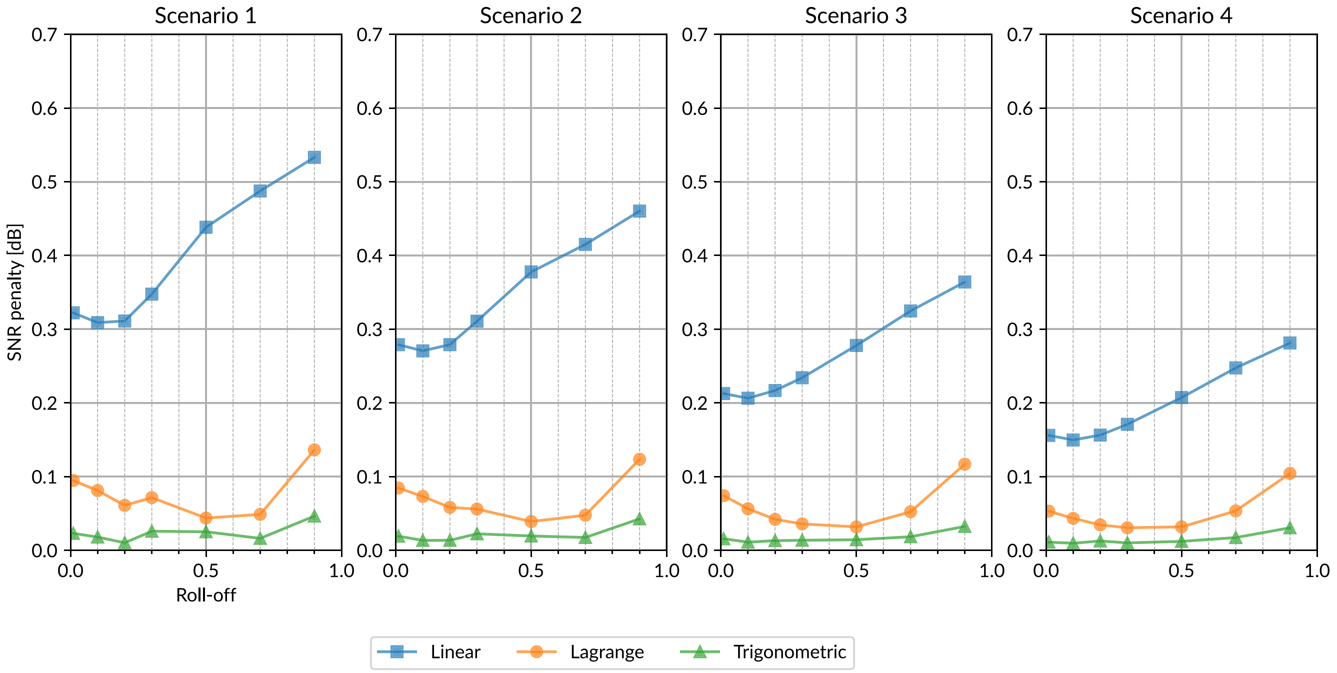

FIGURE 8 | SNR penalty as a function of the roll-off factor of the pulse shaping filter for different time domain interpolation algorithms with respect to frequency domain interpolation.

the more challenging channel in Scenario 4 does not seem to have a major impact on the performance of the timing recovery algorithms. The Gu algorithm performs slightly worse but performs better than the Gardner algorithm as the clock frequency offset increases. Nevertheless, all three feedback algorithms perform relatively well, and a clock offset of less than 100 ppm should be achievable. On the other hand, the implementation of the feedback algorithms in Figure 6 assumes a parallelization factor of 64 and a total loop latency of 20 clock cycles, but in reality, a higher parallelization factor may be necessary to handle the desired data rate. Therefore, Figure 6 also shows the feedback algorithms but with 256 samples processed in parallel instead of 64. In this case, it can be seen that the increase in total loop delay due to the increase in the number of samples processed per cycle results in a much lower clock frequency offset that can be tracked. In fact, the penalty is very large for the Gardner algorithm with a parallelization factor of 256 and latency of 20 and is therefore not included in the plot. For the Godard algorithm, the increase in the parallelization factor also corresponds to an increase in the size of the FFT and thus an increase in the resolution of the estimator, which allows it to perform relatively well up to about 60 ppm. Similarly for the Gu algorithm, increasing the number of samples used for each estimate increases the noise tolerance but limits the tracking range. Figure 7 shows the performance of the Lee algorithm and illustrates the main problem with feedforward timing recovery algorithms; a trade-off must be made between the clock offset that can be tracked and the noise tolerance. For example, the Lee algorithm with a block size of 512 can track clock offsets above 100 ppm in Scenario 1, but the performance degrades rapidly when fading is introduced. To improve estimation accuracy during fades, a larger block size can be used, but this will limit the maximum clock offset that can be tracked, for example, a block size of 2048 allows tracking up to about 60 ppm. Additional filtering of the Lee estimates could potentially improve performance, but because the Lee algorithm is already more complex than the feedback algorithms, it seems like a better option to stick with the feedback-based algorithms. Still, for very high degrees of parallelization, the Lee algorithm may actually be preferred, as its performance is independent of the latency of the algorithm. But the latency of the feedback algorithms could also be reduced, for example by considering a simpler resampling filter. Figure 8 shows that the penalty of using linear interpolation is about 0.3 dB for small roll-off factors in scenarios 1 and 2. However, for scenarios 3 and 4, it can be seen that the penalty from using linear interpolation is lower and closer to 0.2 dB. Thus, for satellite links through the atmosphere, it might very well be worth sacrificing some precision in the interpolation filter to achieve a feedback timing recovery loop with lower latency. Additionally, both Lagrange and trigonometric interpolation requires 3 multiplications per sample with a 4-tap filter implementation, but linear interpolation would require just 1 multiplication per sample and thus significantly reduce the complexity of the entire timing recovery loop. To provide an indication of the clock offset that could be expected in OSLs, the specifications of terrestrial optical communications standards such as 400ZR [44], OpenZR+ [45], OpenROADM [46], and CableLabs P2P [47] can be used for reference. All of them specify that the transmitter must operate within ± 20 ppm of the specified baud rate, so if clocks of similar performance can be achieved onboard satellites, then digital timing recovery can be reliable even in the presence of severe turbulence. However, clock crystals drift with temperature, age and radiation, which is important to account for, particularly for GEO links. Doppler shifts are also important to consider, most notably in the case of ISL. The worst case Doppler shift occurs when two satellites in LEO are traveling in the same orbit but in opposite directions. The maximum orbital velocity for a LEO satellite is at an altitude of 400 km which results in a worst case Doppler shift for an ISL of approximately 50 ppm, which is still well within in the range of what is possible to track with digital timing recovery in an ISL scenario. In the end, digital feedback-based timing recovery algorithms seem to be the best option for OSL with either the Gardner, Gu, or Godard algorithm, depending mainly on whether any subsequent subsystems can also use the FFT required for the Godard algorithm. However, it remains to be proven experimentally that these digital timing recovery algorithms can remain stable over long periods of time in real atmospheric channels. Further, the impact of clock jitter has not been considered in this paper and will have an impact on the system performance especially when considering systems with low roll-off factor or implementations with high loop latency.

4 | Carrier Phase Estimation

Random rotations of the constellation points occur due to phase noise. Typically, the phase noise is modeled as a random walk with a variance as a function of the laser linewidth and symbol period by

$$\sigma^2(\Delta w, T_s) = 2\pi \Delta w T_s \tag{17}$$

where Δw is the combined linewidth of the transmitter and receiver laser and T_s is the symbol duration [48]. Additional rotations will occur due to imperfect carrier frequency offset compensation, which must also be accounted for. Digital phase estimation algorithms have received considerable attention in the terrestrial optical communications community, enabling the use of low-cost distributed feedback (DFB) lasers and higher order modulation formats [49]. However, phase estimation algorithms developed for terrestrial links are generally designed to be optimal in a static channel. Optical links through the atmosphere, on the other hand, require phase estimation algorithms to operate reliably in very low SNR scenarios. There are many different approaches to the phase estimation problem, such as utilizing pilot symbols or choosing modulation formats that are less sensitive to phase noise, and the best solution ultimately depends on many different system parameters. This section gives an overview of several possible phase estimation algorithms with a focus on QPSK modulation. Finally, computer simulations are performed to assess the performance of different algorithms for the four different scenarios.

4.1 | Phase Estimation Algorithms

One of the main challenges with symmetric modulation formats such as QPSK is that there is phase ambiguity, and therefore, a phase reference is required to determine the correct phase of the received symbols. To overcome this, a training symbol or sequence could be transmitted at the beginning of the transmission to provide the receiver with a phase reference. The receiver can then track the random fluctuations by continuously estimating and unwrapping the phase [49]. For M-PSK signals, the phase is often estimated using the Viterbi–Viterbi algorithm [50], where the estimated phase θ for a block of N samples is given by

$$\theta = \frac{\arg \sum_{k=0}^{N-1} r^M[k]}{M} \tag{18}$$

where M is the modulation order (4 for QPSK). Some variations on the original Viterbi–Viterbi algorithm have been proposed, such as introducing weights to the summation in Equation (18) and extending the algorithm to M-QAM constellations [51]. For higher order M-QAM constellations, more sophisticated phase estimation techniques are typically required

[52], such as blind phase search [53] or principal component analysis-based [54]. However, a deeper analysis of phase estimation techniques for M-QAM constellations are beyond the scope of this paper. If an incorrect phase estimate is obtained due to additive noise or fast phase changes, then the absolute phase reference is lost. This is often referred to as a cycle slip and will result in large bursts of errors until a new absolute phase estimate is obtained. To reduce the impact of cycle slips, the data can be encoded in the phase difference between two consecutive symbols, avoiding the need for an absolute phase reference. In this way, the phase difference $\Delta\theta[k]$ can be calculated from the received signal $r[k]$ by

$$\Delta\theta[k] = \arg(r[k]r^{*}[k-1]) \tag{19}$$

This is often referred to as soft differential QPSK and may appear to be an attractive low-complexity solution. However, this approach introduces a relatively large SNR penalty of about 2.5 dB [55]. Instead, it turns out that a better option is to first estimate and compensate the phase of a block of symbols, then make a hard decision as to which constellation point each symbol corresponds to, ignoring the phase ambiguity problem, and finally obtain the data by performing differential decoding on the decided symbols. This method, referred to as hard DQPSK or simply DQPSK, performs much better than soft DQPSK but is still limited by a 0.75 dB penalty [55]. A potential drawback of using hard DQPSK is that it cannot be trivially combined with soft-decision FEC. However, the penalty from using soft DQPSK is already so large that it is typically not worth the benefits of soft-decision FEC [55, 56]. A second option to mitigate cycle slips is to periodically insert pilot symbols into the data stream to allow the receiver to determine the correct phase, that is, if the signal is known to be $x[k]$ at position k, then the phase can be estimated as

$$\theta = \arg r[k]x^{*}[k] \tag{20}$$

The estimate can then be filtered by combining the estimates of several pilot symbols [57]. Because an absolute phase reference is frequently available, the impact of a cycle slip is reduced. Moreover, phase estimation with pilots has a lower hardware complexity, because only the pilot symbols are required for phase estimation. On the other hand, the insertion of pilots introduces a signal overhead that can be described as an SNR penalty P given the pilot rate R_{pilot} as [58]

$$P = 10\log 10(R_{pilot}) \tag{21}$$

4.2 | Analysis

Computer simulations were used to evaluate the performance of the different phase estimation methods for QPSK by calculating the BER in an AWGN channel for 10^7 symbols per SNR value with the phase noise modeled as described in Equation (17). The SNR was set such that a reference system with no phase noise would achieve a BER of 10^{-3}. The SNR penalty was then calculated for each algorithm with respect to the reference system for the four different scenarios. The Viterbi–Viterbi algorithm was implemented with a window N of 32 for differential encoding. Cycle slips play a critical role in the implementation performance of nondifferential Viterbi–Viterbi; hence, a larger averaging window must be used, but at the cost of not being able to track fast phase variations. Therefore, a window of 64 and 256 is used for nondifferential encoding to visualize this behavior. How soon a new reference phase is provided is another important parameter when considering nondifferential encoding in the event of a cycle slip. In this paper, it will be assumed that the nondifferential Viterbi–Viterbi implementations are given a new phase reference every 10^5 symbols. In a practical system, this could be implemented with a short preamble at the start of each frame, so that for sufficiently long frames the overhead from the phase reference is negligible. Thus, in the case of a cycle slip, the nondifferential Viterbi–Viterbi algorithm has to wait for a new phase reference to recover from the cycle slip. Pilot-based phase estimation was implemented with three different pilot rates of $7/8, 15/16, 31/32$, and linear interpolation is used between the pilots to determine the correct phase. The penalty is finally plotted against different values of ΔwT_s for the four different scenarios in Figure 9, where the penalty due to pilots has also been included. The implementation parameters have been chosen to show the strengths and weaknesses of the different algorithms, but it should be noted that the number of symbols used to estimate the phase could be further optimized for specific scenarios. For scenarios 1 and 2, it can be observed that the nondifferential Viterbi–Viterbi algorithm outperforms the others as long as $\Delta wT_s < 3 * 10^{-5}$. For higher values of ΔwT_s and scenarios 3 and 4, the effect of cycle slips becomes more evident. In general, a linewidth between 0.1 MHz and 1 MHz could be expected. Thus, for a 28 GBaud system, ΔwT_s should approximately be between 10^{-6} and 10^{-5}. Although cycle slips are rare, in most cases they would result in the loss of an entire frame, so even though the BER might be lower on average for the Viterbi–Viterbi algorithm, a larger penalty without burst errors may be preferable because it simplifies the design of FEC. It should also be noted that evaluating the cycle slip probability is challenging, as the artifacts of the fixed-point hardware implementation of the algorithm also have a large impact, and properly evaluating the probability in software is often too time consuming [52]. Thus, Figure 9 may show an overly optimistic view of the nondifferential Viterbi–Viterbi algorithm. Instead, using differential encoding might be a better idea, especially for larger values of ΔwT_s, but its performance degrades fast at low SNR, such as during fades, which can be seen in scenario 4, where the penalty is almost 1.5 dB. Pilot-based phase estimation has similar performance to differential encoding but performs much better during fades. In fact, using pilots in scenario 4 could provide a 1 dB improvement over Viterbi–Viterbi with differential coding. Pilot-based phase estimation would also allow the use of soft-decision FEC that results in roughly a 2 dB gain compared with hard-decision FEC but at the cost of increased hardware complexity [59]. In addition, pilot-based phase estimation could be combined with a second blind phase estimator to further improve performance [58]. The low complexity of the implementation of pilot-based phase estimation is also an attractive feature. An adaptive pilot rate, where the pilot rate changes depending on the current channel conditions could be an appealing concept to optimize the system, but the same pilot rate performs similarly for all scenarios as shown in Figure 9. However, because the optimal number of symbols for phase estimation is a function of the ratio between

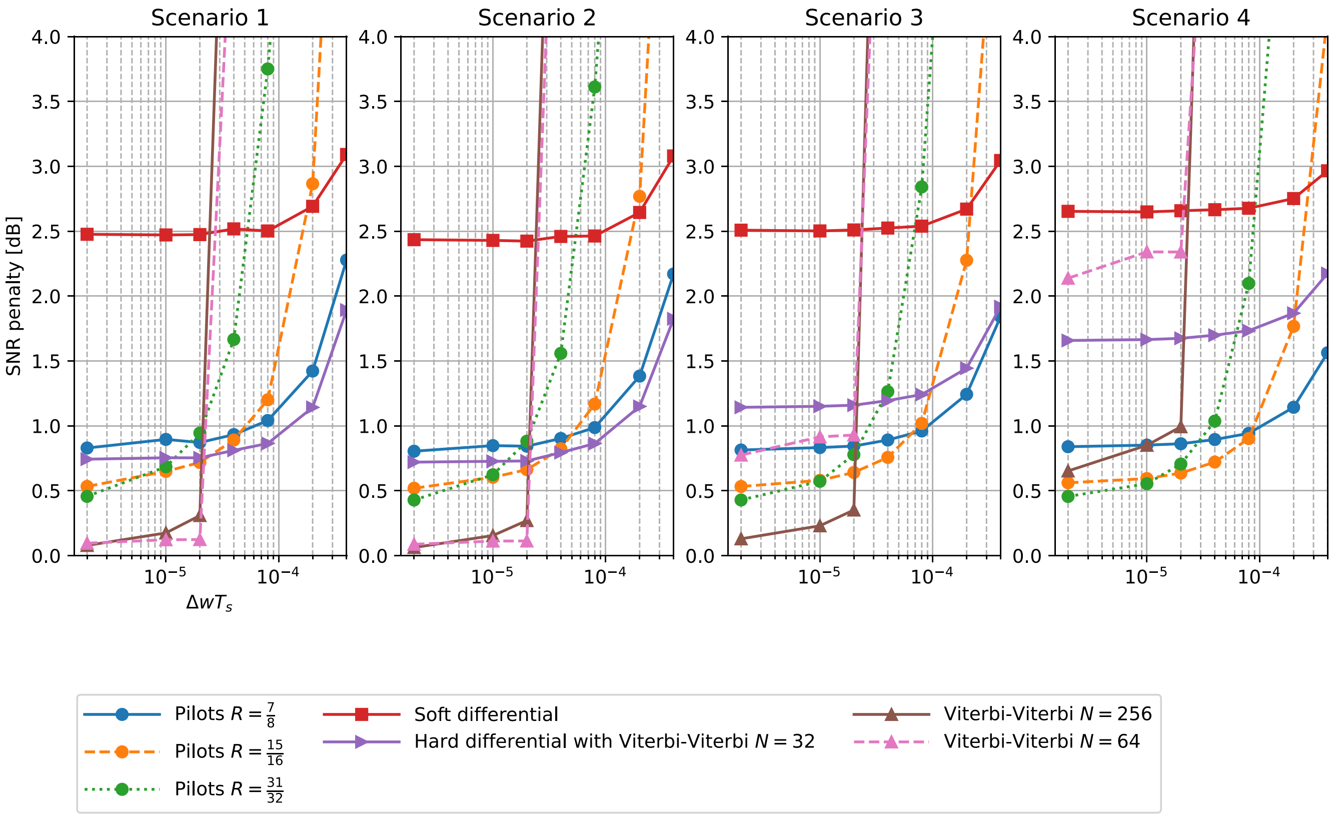

FIGURE 9 | SNR penalty for different phase estimation algorithms including penalty from overhead as a function of the $\Delta w T_s$ at an average BER of 10^{-3}.

SNR and phase noise [60], dynamically adjusting the number of symbols used for phase estimation should improve system performance in a channel with large SNR fluctuations. Further research is necessary to identify practical methods for dynamically adjusting the phase estimation window and to quantify the potential gain. For pilot-based receivers, it should also be noted that it requires knowledge of the position of the pilots. Therefore, frame synchronization is required before phase estimation, which makes frame synchronization more difficult. Timing recovery must also remain stable at low SNR to ensure that the position of the pilots is not lost. Hence, the choice of phase estimation algorithm is a complex issue that is inherently dependent on other subsystems such as frame synchronization, timing recovery, FEC, and lasers.

5 | Carrier Frequency Offset Compensation

The frequency of the transmitter and receiver lasers is rarely perfectly synchronized, and their frequency will drift over time. In addition, due to the movement of the satellites, especially in the case of LEO satellites, an additional frequency offset of up to 10 GHz will occur due to the Doppler effect [61]. In principle, coherent receivers can be realized with either a homodyne or an intradyne receiver architecture. A homodyne system constantly adjusts the phase of the receiver's local oscillator to match the frequency of the transmitter. While a homodyne system architecture is in principle energy efficient, it increases system complexity, is susceptible to phase noise, and is not easily scaled to higher order modulation formats [62]. As a result, terrestrial fiber optic receivers have preferred an intradyne receiver structure, that is, the frequency offset between the transmitter and receiver is compensated via DSP [63]. If a transmitted signal $x_{tx}(t)$ experiences a frequency shift f_Δ at the receiver, then the received signal $x_{rx}(t)$ can be described by

$$x_{rx}(t) = x_{tx}(t)e^{j2\pi f_\Delta t} \tag{22}$$

Thus, by estimating the frequency offset f_Δ, it is possible to compensate for the offset by multiplying the received signal by $e^{-j2\pi f_\Delta t}$. Carrier frequency offset estimation, like many other estimation problems, can be divided into two different categories of algorithms: blind methods and data-aided methods. In general, blind methods have one major drawback, they are typically limited in the frequency offset range that they can estimate. Some data-aided algorithms, on the other hand, can estimate very large frequency offsets, but their estimation range is limited by the bandwidth of the ADC. Therefore, compensating for a large Doppler shift is a challenge for both the ADC and the DSP. Nevertheless, as the movements of the satellites are typically known, the Doppler shift can be precompensated at the transmitter side either by tuning the transmitter laser or by digital precompensation [61]. For example, the ESA specification for terabit/s optical links (ESTOL) specifies that the Doppler shift for ISLs must be precompensated at the transmitter [64].

Even with precompensation of the Doppler shift, there will still be a frequency offset between the transmitter and receiver laser that needs to be compensated. Therefore, this section presents an overview of suitable methods for estimating the carrier frequency offset and compares them using computer simulations.

5.1 | Frequency Offset Estimation Algorithms

By raising the signal to the Mth power, the modulation of signals modulated with M-ary phase shift keying can be removed. The frequency offset can then simply be estimated by the argument of a sample raised to the power of M times the conjugate of the previous sample raised to the power of M. To reduce the effect of noise, multiple estimates are often averaged, and the angle is calculated only for the average. This gives the estimator as [60]

$$f_\Delta = \frac{\arg \sum_{k=1}^{N} r[k]^M (r[k-L]^M)^*}{2\pi ML} \tag{23}$$

where L is a design parameter that represents the L th previous sample relative to sample k. This estimator is limited by the fact that it can only estimate frequencies up to $\pm f_s/2ML$, where f_s is the sampling frequency, that is, a trade-off between frequency estimation range and noise tolerance must be made, as increasing the distance between samples improves the noise tolerance but reduces the frequency offset estimation range. One solution that has been proposed to achieve both a large range and noise tolerance is to perform frequency estimation in multiple stages [65]. In this case, the first stage would perform a coarse estimation with $L = 1$ and the following stages would use the same estimation technique but with a larger L to improve the accuracy. The frequency offset can also be estimated similarly in the frequency domain by [66]

$$f_\Delta = \frac{1}{4} \arg \max_\theta \mathcal{F}[r^M[k]](\theta) \tag{24}$$

However, this method also has a limited frequency range up to $f_s/2M$ and is relatively complex due to the fact that it requires the DFT of $r[k]^M$. Data-aided techniques can be used to enable the possibility of estimating larger frequency offsets. For example, if a sequence $x[k]$ of the received signal $r[k]$ is known, then the frequency offset can be estimated by [67]

$$f_\Delta = \frac{\arg \sum_{k=1}^{N} r[k]x^*[k](r[k-1]x^*[k-1])^*}{2\pi} \tag{25}$$

This method can estimate frequency offsets up to $\pm f_s/2$ but requires a long training sequence to achieve the same accuracy as the blind methods. Another popular data-aided method is to design a preamble, where the structure of the preamble is exploited for frame detection and frequency offset estimation. The advantage of this method is that it is more robust to channel impairments and can therefore be performed before the equalizer stage. A common preamble structure was proposed by Schmidl and Cox [68] where a sequence is repeated twice to create the preamble. The preamble can be found by maximizing a metric based on autocorrelation and then estimating the frequency offset by

$$f_\Delta = \frac{1}{2\pi N} \arg \sum_{k=0}^{N-1} r[k] r^*[k+N] \tag{26}$$

which can estimate frequency offsets up to $\pm f_s/2N$. Rha et al. [69] showed that it is possible to extend the range of Equation (26) by combining three preambles of different lengths. By combining the estimates from each preamble and applying the Chinese remainder theorem (CRT), an estimate with a range up to $\pm f_s$ can be obtained if the length of the preambles is chosen such that the length of the first two is coprime and the length of the third is the product of the first two.

5.2 | Analysis

To compare the performance of the different algorithms, computer simulations were performed in an AWGN channel with a QPSK modulated signal, where the mean square error (MSE) of the estimate was calculated over 90% of the valid frequency estimation range for each estimator with 10^6 estimates per SNR value. The full theoretical range is not used because most algorithms suffer accuracy degradation when the frequency offset is close to the limit of the algorithm's capabilities. The considered algorithms and their estimation range are summarized in the Table 2 for any parameter configuration. The algorithms are implemented with different parameter configurations to provide some insight into how the parameter choices affect the performance of different algorithms. The blind time domain algorithm is implemented with $L = 1$, the blind frequency domain algorithm is implemented with a 64-point and 4096-point FFT, the data-aided algorithm is implemented with a preamble size of 128 and 512,the Schmidl and Cox algorithm is implemented with a total preamble size of 128 ($N = 64$) and finally the CRT algorithm is realized with preambles of length $[8,9,72], [11,12,132]$. The theoretical frequency offset estimation range for the implemented algorithms when considering a 10 GBaud and a 28 GBaud system is shown in Table 3. Phase noise is also included in the simulation and modeled as in Equation (17), as it also affects the performance of the algorithms. The resulting MSE for different SNR values is shown in Figure 10 for four different values of Δwt_s. A lower bound to the MSE of any parameter estimation problem is given by the Cramer–Rao bound (CRB). However, because carrier phase estimation and compensation are performed after carrier frequency offset estimation, the CRB has to also consider the carrier phase noise. This results in a complex CRB and no closed-form expression exists [70]. An alternative is to assume that the phase remains constant for the observation window and use the modified CRB [71]. Still, this does not properly illustrate the performance degradation due to strong phase noise. To better illustrate the impact of time-varying phase noise, the approximation of the true CRB as described by Barbieri

TABLE 2 | Frequency offset estimation algorithms and their theoretical frequency offset estimation range.

Algorithm	Range
Blind time domain	$\pm f_s/2ML$
Blind frequency domain	$\pm f_s/2M$
Data-aided	$\pm f_s/2$
Schmidl and Cox	$\pm f_s/2N$
Schmidl and Cox (CRT)	$\pm f_s/2$

TABLE 3 | Frequency offset estimation range for the simulated algorithms when operating at 10 and 28 GBaud.

Algorithm	Range at 10 GBaud	Range at 28 GBaud
Blind time domain $L = 1$	± 1.25 GHz	± 3.5 GHz
Blind frequency domain $N = 64$	± 1.25 GHz	± 3.5 GHz
Blind frequency domain $N = 4096$	± 1.25 GHz	± 3.5 GHz
Data-aided $N = 64$	± 5 GHz	± 14 GHz
Data-aided $N = 512$	± 5 GHz	± 14 GHz
Schmidl and Cox $N = 64$	± 0.078125 GHz	± 0.21875 GHz
Schmidl and Cox (CRT) [8,9, 72]	± 5 GHz	± 14 GHz
Schmidl and Cox (CRT) [11,12, 132]	± 5 GHz	± 14 GHz

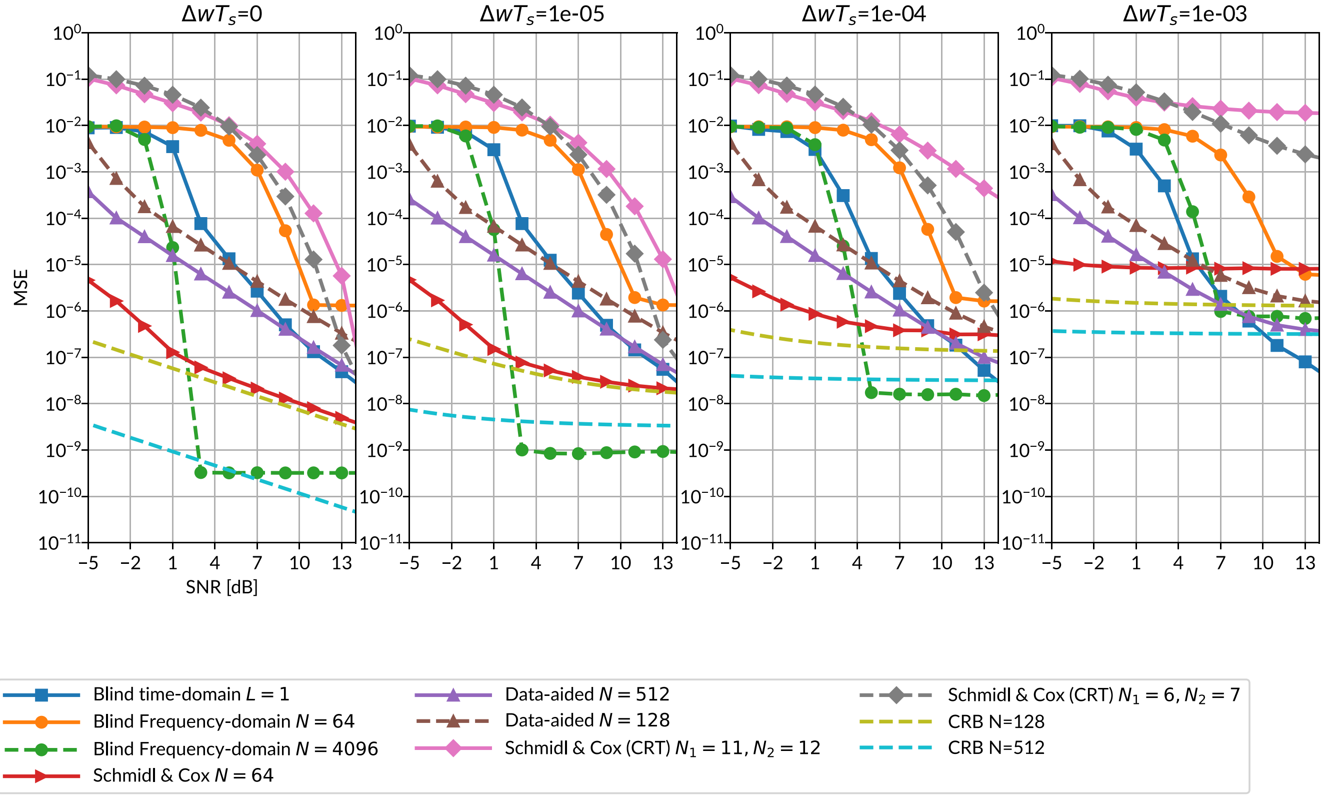

FIGURE 10 | Mean square estimation error (MSE) for the estimated carrier frequency offset plotted against the SNR for different values of phase noise $\Delta w T_s$.

and Colavolpe [70] is evaluated for $N = 128$ and $N = 512$ pilot symbols and also shown in Figure 10.

As shown in Figure 10, the method proposed by Schmidl and Cox may seem to be a good choice and gets very close to the CRB when the phase noise is not strong, but it is important to note that the frequency offset estimation range is quite limited. By extending the Schmidl and Cox algorithm with the CRT, the range is extended, but the algorithm requires a relatively high SNR and is sensitive to phase noise. The data-aided method supports the full frequency estimation range as well and could be an interesting option as it is less sensitive to phase noise. However, the Schmidl and Cox algorithm is robust against polarization impairments and therefore could be used before the polarization demultiplexing stage, which could potentially improve the performance of the equalizer stage. Probably the most important thing to consider is what frequency estimation range is needed and how accurate the frequency needs to be estimated. For example, considering a 28 Gbaud system, the blind methods can actually estimate offsets up to 3.5 GHz, which could be sufficient if large Doppler shifts are precompensated. In addition, the subsequent phase estimation stage can typically handle a small residual carrier frequency offset error, so it is sufficient for the carrier frequency offset compensation to provide only coarse compensation of the carrier frequency offset. In fact, the maximum residual frequency offset Δf_m for the phase estimation stage can be approximated for QPSK as [60]

$$\Delta f_m = \frac{f_s}{8N} \tag{27}$$

where N is the number of samples used for phase estimation and f_s is the symbol rate. Most phase estimation algorithms estimate the phase using less than 100 symbols per estimate, so the maximum residual frequency offset is approximately $\Delta f_m = 10^{-3} f_s$. As the carrier frequency will drift very slowly relative to the baud rate, for example, at worst up to approximately 1 GHz/s due to the Doppler effect in ISLs [72], it would be possible to average multiple estimates over a longer period of time to further improve accuracy and to run the carrier frequency offset estimation algorithm at a much slower rate that would allow hardware efficient implementation of the algorithms. To improve performance during fades, it is likely crucial to avoid updating the estimate when the signal quality degrades below a certain threshold. Therefore, blind carrier offset methods appear to be the best option for OSL, as the range and accuracy offered should meet typical carrier frequency offset requirements. For example, most coherent optical standards for terrestrial networks [44–47] specifies that the carrier frequency offset should be within ± 3.6 GHz. However, when considering coherent receivers with lower baud rates, for example, 10 Gbit/s coherent links, it is clear that carrier frequency offset compensation is a much more difficult problem for intradyne receivers, placing higher requirements on the photonics.

6 | Adaptive Equalizer

Polarization multiplexed optical communication systems use two polarizations to encode information. While this provides spectrally efficient communication, it requires the receiver to be able to split the incoming signal back into the two original polarizations. This can be challenging because the state-of-polarization (SOP) of the signal can undergo random fluctuations as it propagates through the atmosphere, fiber cables, or other optical components in the receiver. The standard way to deal with this problem in terrestrial fiber is to demultiplex the two polarizations with an equalizer stage in the DSP [15]. Given two incoming signals for two polarizations r_X, r_Y, four FIR filters $w_{xx}, w_{xy}, w_{yx}, w_{yy}$ can be used to demultiplex the two polarizations into the desired signals z_X, z_Y. This filter structure is typically referred to as a butterfly equalizer [73] and is illustrated in Figure 11a and can be described by

$$\begin{cases} z_X = w_{xx} * r_X + w_{xy} * r_Y \\ z_Y = w_{yx} * r_X + w_{yy} * r_Y \end{cases} \tag{28}$$

where the number of taps required by the FIR filter is mainly determined by the length of the channel impulse response. As both the SOP and the channel impulse response change over time, the FIR filters must be adaptive and able to track the changes. The implementation of adaptive FIR filters with many taps and high throughput is typically expensive in terms of hardware, and the adaptive filter of terrestrial coherent receivers is typically responsible for 30%–40% of the total receiver DSP power consumption [74, 75]. Therefore, it is crucial to find efficient implementation strategies for the adaptive equalizer. The easiest way to reduce the complexity of the equalizer is to reduce the number of taps in the FIR filters. An efficient way to reduce the number of taps required is to restructure the butterfly equalizer into multiple stages. For example, a two-stage equalizer could be used where the first stage is a 1-tap butterfly equalizer and the second stage consists of two FIR filters, as shown in Figure 11b [76]. This reduces the number of multiplications by nearly 50%. However, this structure is vulnerable to differential group delay (DGD). For example, in [76], it is shown that a 6 ps DGD results in a 1 dB penalty for a 32 GBaud DP-QPSK system; however, if the DGD can be kept below 1 ps, then the penalty from the two-stage equalizer is negligible. In terrestrial fiber networks, the main contribution to DGD is from the polarization mode dispersion in the fiber cable, but for OSL, the dispersion in the fiber should be negligible. Optical amplifiers such as EDFA also induce some polarization mode dispersion, although typically well below 1 ps. However, the amplifiers used for OSL may introduce increased polarization mode dispersion, which would influence the choice of equalizer architecture. The two-stage architecture could also be flipped, that is, two FIR filters are used first, followed by a 1-tap butterfly equalizer, as shown in Figure 11c. The main advantage of this approach is that the N-tap filters typically require 2 samples per symbol, but the 1-tap butterfly equalizer requires only 1 sample per symbol, so by flipping the structure, it is possible to downsample the signal to 1 sample per symbol between the two stages, which would reduce the complexity of the 1-tap butterfly equalizer stage [77].

The implementation of the FIR filters themselves can also be optimized; for short filters, time domain implementations are generally preferred, but as the length of the filter increases, frequency domain implementations become more efficient. [78] Finally, another major contributor to the complexity of adaptive equalizers is the adaptation algorithm. The channel can be estimated either blindly or by inserting known data symbols for

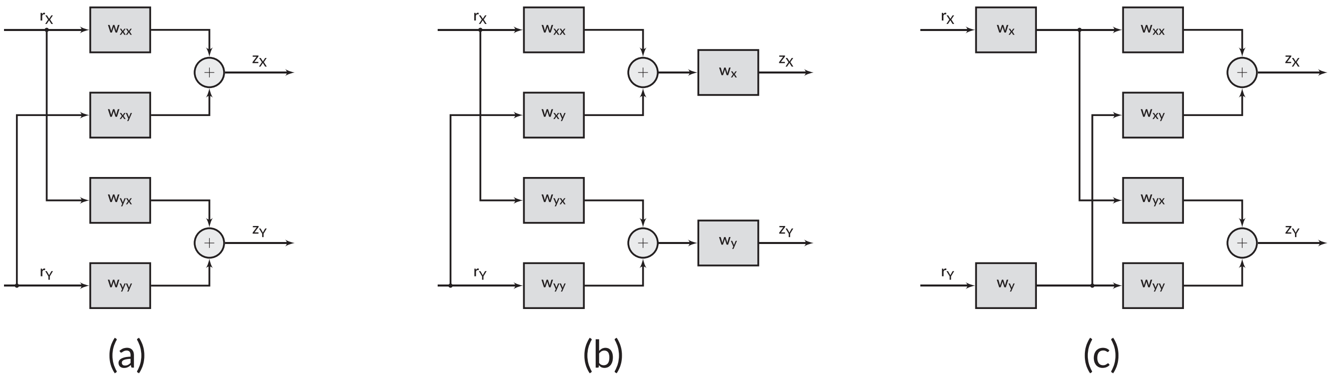

FIGURE 11 | Different types of equalizer structures: (a), N-tap butterfly equalizer, (b) 1-tap butterfly equalizer followed by two N-tap FIR filters, and (c) two N-tap equalizers followed by a 1-tap butterfly equalizer.

channel estimation. Additionally, in turbulent channels, the adaptive equalizer must also be designed to be scintillation tolerant, as it may diverge to a local optimum during fades. A simple solution that has been proposed for this problem is to turn off tracking when the incoming signal power is low [79]. However, if the SOP changes significantly during each fade, then being able to still track the SOP through the fades would be advantageous. The rest of this section will be devoted to discussing state-of-the-art techniques for finding the optimal filter weights. Finally, computer simulations are performed to asses the performance of different algorithms.

6.1 | Time Domain Equalization

Let the transmitted signals for a dual-polarization system be given by $s_X[n]$ and $s_Y[n]$, the channel impulse response is given by $h_{XX}, h_{XY}, h_{YX}, h_{YY}$ then the received signals r_X and r_Y can be described as

$$\begin{cases} r_X[n] = \sum_{i=0}^{L_c-1} (h_{XX}[i]s_X[n-i] + h_{XY}[i]s_Y[n-i]) + n_X[n] \\ r_Y[n] = \sum_{i=0}^{L_c-1} (h_{YX}[i]s_X[n-i] + h_{YY}[i]s_Y[n-i]) + n_Y[n] \end{cases} \tag{29}$$

where n_X, n_Y are additive white Gaussian noise and L_c is the length of the channel impulse response. The X-polarization of Equation (29) can be rewritten into matrix form as

$$\mathbf{r}_X = \mathbf{S}\mathbf{h}_X + \mathbf{n}_X \tag{30}$$

where the matrices are defined as:

$$\mathbf{S_X} = \begin{bmatrix} s_X[n] & \dots & s_X[n-L_c+1] & s_Y[n] & \dots & S_Y[n-L_c+1] \\ s_X[n+1] & \dots & s_X[n-L_c] & s_Y[n-L_c] & \dots & S_Y[n-Lc] \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ s_X[n+N-1] & \dots & s_X[n+N-Lc-1] & s_Y[n+N-1] & \dots & S_Y[n+N-Lc-1] \end{bmatrix} \\ \in \mathbb{C}^{N\times 2L_c} \tag{31}$$

$$\mathbf{h_X}^T = \begin{bmatrix} h_{XX}[0] & \dots & h_{XX}[L_c-1] & h_{XY} & \dots & h_{XY}[L_c-1] \end{bmatrix} \in \mathbb{C}^{1\times 2L_c} \tag{32}$$

$$\mathbf{r_X}^T = \begin{bmatrix} r_X[n] & r_X[n+1] & \dots & r_X[n+N-1] \end{bmatrix} \in \mathbb{C}^{1\times N} \tag{33}$$

$$\mathbf{n_X}^T = \begin{bmatrix} n_X[n] & n_X[n+1] & \dots & n_X[n+N-1] \end{bmatrix} \in \mathbb{C}^{1\times N} \tag{34}$$

and where N is the number of received samples. An estimate of the channel $\widehat{\mathbf{h}}_X$ can then be obtained in the least squares sense using the Moore–Penrose inverse such that

$$\widehat{\mathbf{h}}_X = \mathbf{r}_X(\mathbf{S}_X^H\mathbf{S}_X)^{-1}\mathbf{S}_X^H \tag{35}$$

In the same way, h_{YX}, h_{YY} can be estimated. The transmitted signal $\mathbf{S_X}$ must be known to solve Equation (35). This can be achieved by periodically inserting known training sequences into the data stream, which can be used to estimate the channel. Constant-amplitude-zero auto-correlation (CAZAC) sequences are often suggested as good training sequences [80] because of their flat spectrum and cyclic autocorrelation [81]. Given the estimated channel response, an appropriate equalizer can be designed. One way to design the equalizer is to directly invert the channel matrix, but this can result in amplification of the noise. Instead, it is common to design the equalizer according to the minimum mean squared error (MMSE) criterion. Let the system in Equation (29) be described in matrix form as

$$\mathbf{r} = \mathbf{H}\mathbf{s} + \mathbf{n} \tag{36}$$

where the matrices are defined as

$$\mathbf{r}^T = \begin{bmatrix} r_X[n] & r_Y[n] & \dots & r_X[n-L_W+1] & r_Y[n-L_W+1] \end{bmatrix} \in \mathbb{C}^{1\times 2L_W} \tag{37}$$

$$\mathbf{s}^T = \begin{bmatrix} s_X[n] & s_Y[n] & \dots & s_X[n-L_C-L_W+2] & s_Y[n-L_C-L_W+2] \end{bmatrix} \\ \in \mathbb{C}^{1\times 2(L_W+L_C-1)} \tag{38}$$

$$\mathbf{n}^T = \begin{bmatrix} n_X[n] & n_Y[n] & \dots & n_X[n-L_W+1] & n_Y[n-L_W+1] \end{bmatrix} \in \mathbb{C}^{1\times 2L_W} \tag{39}$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_0 & \mathbf{H}_1 & \dots & \mathbf{H}_{L_c-1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_0 & \ddots & \mathbf{H}_{L_c-1} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{H}_0 & \mathbf{H}_1 & \dots & \mathbf{H}_{L_c-1} \end{bmatrix} \in \mathbb{C}^{2L_W\times 2(L_W+L_C-1)} \tag{40}$$

where L_W is the length of the equalizer and the submatrices $\mathbf{H_i}$ are given by

$$\mathbf{H_i} = \begin{bmatrix} h_{XX}[i] & h_{XY}[i] \\ h_{YX}[i] & h_{YY}[i] \end{bmatrix} \in \mathbb{C}^{2\times 2} \tag{41}$$

The output error $\mathbf{e}$ of the equalizer is then given by

$$\mathbf{e} = \begin{bmatrix} s_X[n] \\ s_Y[n] \end{bmatrix} - \mathbf{W}\mathbf{r} \tag{42}$$

where the equalizer $\mathbf{W}$ that minimizes the mean squared error is given by [81]

$$\mathbf{W} = \mathbf{D}(\mathbf{I}\frac{1}{\text{SNR}} + \mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H \tag{43}$$

where the elements of the matrix $\mathbf{D}$ are defined at the i th row and j th column by

$$\mathbf{D}_{i,j}=\begin{cases}1, & \text{if } i+d=j\\ 0, & \text{otherwise}\end{cases},\mathbf{D}\in\mathbb{C}^{2\times 2(L_W+L_C-1)} \tag{44}$$

where d is a design parameter that has to be optimized. However, both of these methods require the inversion of relatively large matrices, which is complex to implement in hardware. To reduce complexity, the MMSE solution can instead be found by iterative search methods. A common method is the least-mean square (LMS) algorithm. Let the equalizer weights at time instance n be given by $\mathbf{W}_{LMS}[n]$ as

$$\mathbf{W}_{LMS}[n]=\begin{bmatrix} w_{xx}[0] & w_{xy}[0] & \dots & w_{xx}[L_W-1] & w_{xy}[L_W-1]\\ w_{yx}[0] & w_{yy}[0] & \dots & w_{yx}[L_W-1] & w_{yy}[L_W-1]\end{bmatrix}\in\mathbb{C}^{2\times 2L_W} \tag{45}$$

where L_W is the number of filter taps in the equalizer. Let the input to the equalizer be given by $\mathbf{r}[n]$ where $\mathbf{r}[n]$ is defined as

$$\mathbf{r}[n]=\begin{bmatrix} r_x[n]\\ r_Y[n]\\ r_x[n-1]\\ r_Y[n-1]\\ \vdots\\ r_x[n-L_W+1]\\ r_Y[n-L_W+1]\end{bmatrix}\in\mathbb{C}^{2L_W\times 1} \tag{46}$$

The output $\mathbf{Z}[n]$ of the equalizer can now be defined as

$$\mathbf{Z}[n]=\begin{bmatrix} z_X[n]\\ z_Y[n]\end{bmatrix}=\mathbf{W}_{LMS}[n]\mathbf{r}[n] \tag{47}$$

The equalizer weights can then be found iteratively by

$$\mathbf{W}_{LMS}[n+1]=\mathbf{W}_{LMS}[n]+\mu\mathbf{e}[n]\mathbf{r}^H[n] \tag{48}$$

where the error vector $\mathbf{e}$[n] is defined as

$$\mathbf{e}[n]=\begin{bmatrix} d_x[n]\\ d_y[n]\end{bmatrix}-\mathbf{Z}[n] \tag{49}$$

where $d_x[n], d_y[n]$ is the desired signal, which can be either a known pilot symbol or the final symbol decision. Using the symbol decision, commonly referred to as decision-directed-least-mean square (DD-LMS), requires that the equalizer weights are already close to the optimum and is therefore generally used for tracking after the equalizer has already converged [82]. Furthermore, it should be pointed out that Equation (48) describes the serial implementation of the LMS algorithm, which cannot be implemented in a high-speed system. Instead, the LMS algorithm has to be implemented blockwise depending on the required parallelization factor, which increases the convergence time and reduces the tracking performance. The LMS algorithm is sensitive to phase noise; however, this can be solved by compensating for the phase noise before calculating the error vector [83]. It is also possible to estimate the equalizer weights using blind methods. Most commonly, this is accomplished using the constant modulus algorithm (CMA), which is similar to the LMS algorithm but immune to phase noise and does not rely upon any pilot symbols [84]. In fact, only the error function in Equation (48) changes in the CMA algorithm. For a QPSK signal, the symbols should lie on a circle of radius R in the complex plane once the equalizer has converged. The CMA algorithm utilizes this property and defines the error function $\mathbf{e}[n]$ in Equation (48) as

$$\mathbf{e}[n]=\begin{bmatrix} R^2-|z_X[n]|^2\\ R^2-|z_Y[n]|^2\end{bmatrix} \tag{50}$$

which can be further extended to higher modulation formats by introducing multiple values of R depending on the received symbol [85]. One of the challenges with the CMA algorithm is that it is possible that the two incoming signals converge to the same output polarization. This mainly occurs when the polarization dependent loss (PDL) in the system cannot be neglected, but could be solved by letting the two polarizations converge sequentially [86].

6.2 | Frequency Domain Equalization

Designing a frequency domain equalizer could also be an attractive option, especially for channels with long impulse responses, as long convolutions can be efficiently realized in the frequency domain. Frequency domain implementations of both the LMS algorithm [87] and the CMA algorithm have been proposed [88], but the error computation of both algorithms must still be performed in the time domain. Thus, although the frequency domain implementation of these algorithms may seem promising in theory, the need to convert between the time domain and the frequency domain within the equalizer loop significantly increases the latency, which severely degrades the performance of the algorithms in real-time implementations [81]. Instead, another option that is more viable for implementing adaptive equalizers in the frequency domain is by utilizing a training sequence analogous to the time domain approach already discussed. However, compared with the implementation in the time domain, implementing Equations (35) and (6.1) in the frequency domain is actually a lot easier, because each frequency component can be computed independently, reducing the main computational problem to matrix multiplication and inversion of several 2×2 matrices [81].

6.3 | Analysis

Computer simulations have been performed to analyze the performance of the different equalization methods, considering both convergence properties and tracking capabilities. A simplified model is used to describe continuous polarization rotations as in [73], where the Jones matrix is given b

$$\mathbf{J}=\begin{bmatrix}\cos\theta(t) & \sin\theta(t)\\ -\sin\theta(t) & \cos\theta(t)\end{bmatrix} \tag{51}$$

where $\theta(t)=\omega t$ is the current SOP and ω is the rotation speed of the SOP.

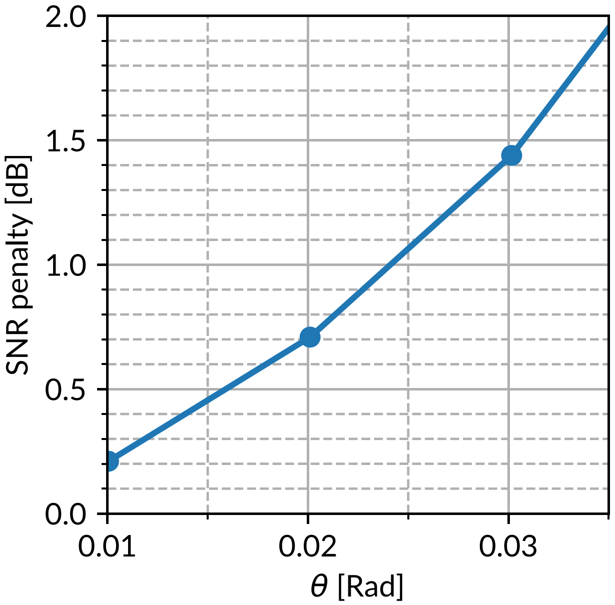

FIGURE 12 | SNR penalty as a function of rotating the state-of-polarization without compensating for the rotation.

First, the penalty from not compensating for changes in θ is investigated to determine how frequently the equalizer must be updated when using a training sequence. The SNR penalty is estimated by running simulations with different values of θ. The resulting plot is shown in Figure 12. It can be seen that there is a penalty of more than 1 dB already at 0.03 radians. Therefore, it is critical to compensate for even relatively small variations in the SOP. In [81], it is shown that a training sequence of 500–1000 symbols is sufficient to obtain a good channel estimate with the MMSE equalizer. Thus, to keep the penalty below 1 dB and the training sequence overhead below 1%, a tracking speed in the region of 10^{-7} radians per symbol can be achieved, or approximately 1–10 krad/s for a 28 Gbaud system. The MMSE equalizer method is not suitable for very fast SOP rotations, but could be a good option for OSL where the SOP rotation speed could be lower. For reference, most coherent optical standards for terrestrial networks [44–47] requires the receiver to support up to 50 krad/s. However, it should also be considered that in order to obtain a good channel estimate during a signal fade, it may be necessary to use a longer training sequence to overcome the additional noise in the signal.

Next, the convergence time of the data-aided-least-mean square (DA-LMS) and CMA algorithms is analyzed. Both algorithms are implemented with a parallelization factor of 64 and a feedback loop latency of 20 clock cycles. Furthermore, the LMS algorithm is implemented with three different pilot rates of $R = 7/8, 15/16, 31/32$. In addition, the LMS equalizer is also implemented with a pilot rate of 1, which would correspond to using a training sequence similar to the MMSE algorithm. Two different equalizer structures are considered, the N-tap butterfly structure shown in Figure 11a and the 1-N-tap structure shown in Figure 11b, where N is set to 11. For the two-stage structure, only the butterfly equalizer is considered adaptive in this case, and the 11-tap stage is set as the matched filter. A roll-off factor of 0.2 is used, and the effect of phase noise is assumed to be compensated for within the equalizer loop. The average BER is calculated by averaging 100 simulations of the algorithms where the optimal system would achieve a BER of about 10^{-3} and with $\theta = \pi/4$ to consider the worst case in terms of convergence speed. The average BER as a function of convergence time is shown in Figure 13. As can be seen in Figure 13, the training sequence-based equalizer converges the fastest, but for both equalizer structures, it still requires close to 10^4 training symbols to converge, which introduces a relatively large overhead, especially compared with using training sequences in combination with the MMSE equalizer. The LMS algorithm combined with pilot symbols converges relatively quickly, especially when considering the two-stage structure. The CMA algorithm generally converges more slowly, and additional care must be taken to avoid converging to the same polarization. Still, it should be possible for the CMA algorithm to converge within 10^5 symbols, or less than 4 μs for a 28 GBaud receiver. Thus, even in the case of complete signal loss, the equalizer should be able to converge relatively quickly compared with the coherence time of the atmospheric channel. It is worth noting that the convergence and tracking properties of both the CMA and LMS algorithms depend strongly on the step size parameter μ. Although there are some theoretical results to estimate the bounds of the step size, there is no easy way to find the step size with the optimal trade-off between convergence time and stability [89]. In this paper, a coarse search has been performed to find suitable values for the step size μ. The CMA algorithm is implemented with $\mu = 0.0005$ and $\mu = 0.0001$, and the DA-LMS is implemented with $\mu = 0.002$ and $\mu = 0.001$ for the one-stage and two-stage equalizer, respectively. A variable step size could also be used [90], in fact this could be an attractive option for OSL through the atmospheric channel where different step sizes could be beneficial depending on the received signal conditions.

Next, the tracking performance of the different iterative algorithms is analyzed where DA-LMS with $R = 1$ is not included as the tracking performance depends on how often the preamble is inserted just as for the MMSE equalizer. The simulation setup remains the same, but now with a time varying θ that rotates at a constant speed. The tracking performance is simulated for different rotation speeds with 10^7 symbols per SNR value, allowing the performance given the four different scenarios to be evaluated. Finally, the SNR penalty is calculated with respect to an optimal system with $\theta = 0$. The resulting plot is shown in Figure 14, where the overhead penalty is not included for DA-LMS, as it is assumed that the pilots would also be used for phase estimation in this case. As shown in Figure 14, all algorithms are able to track polarization rotation speeds above 10^{-6} radians per symbol before performance starts to degrade. DA-LMS with $R = 7/8$ and CMA can even track over 10^{-5} radians per symbol or close to 300 krad/s in the case of a 28 GBaud receiver. The difference between the two different equalizer architectures is relatively small, the two-stage architecture has slightly better performance up to a certain rotation speed, and the single-stage equalizer is able to track the slightly faster rotations. The second stage of the two-stage architecture is also instantiated with optimal filter tap coefficient, which gives it better performance compared with the one-stage equalizer at slower rotation speeds. However, this performance gap would change depending on how the step size parameter is chosen, as discussed earlier. It should also be noted that in a real system, the optimal filter response

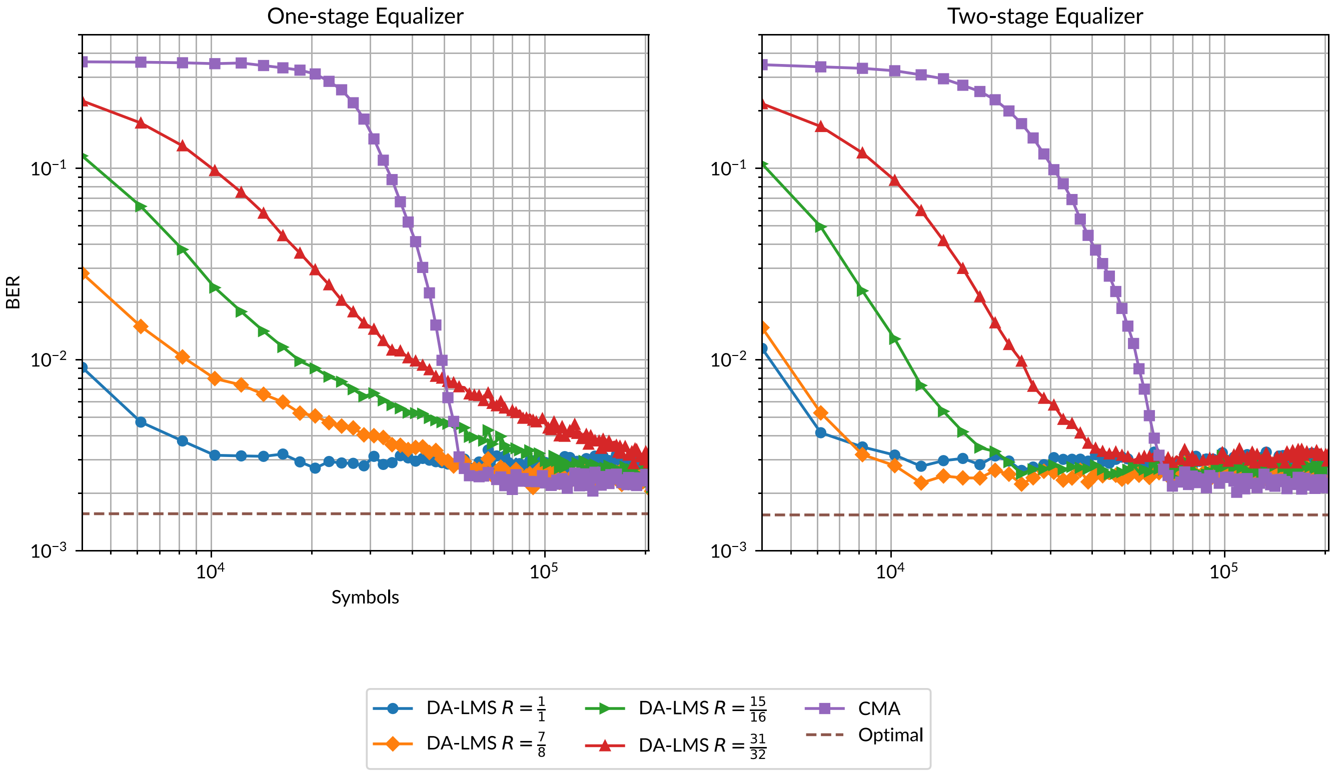

FIGURE 13 | BER as a function of the number of symbols processed by the equalizer.

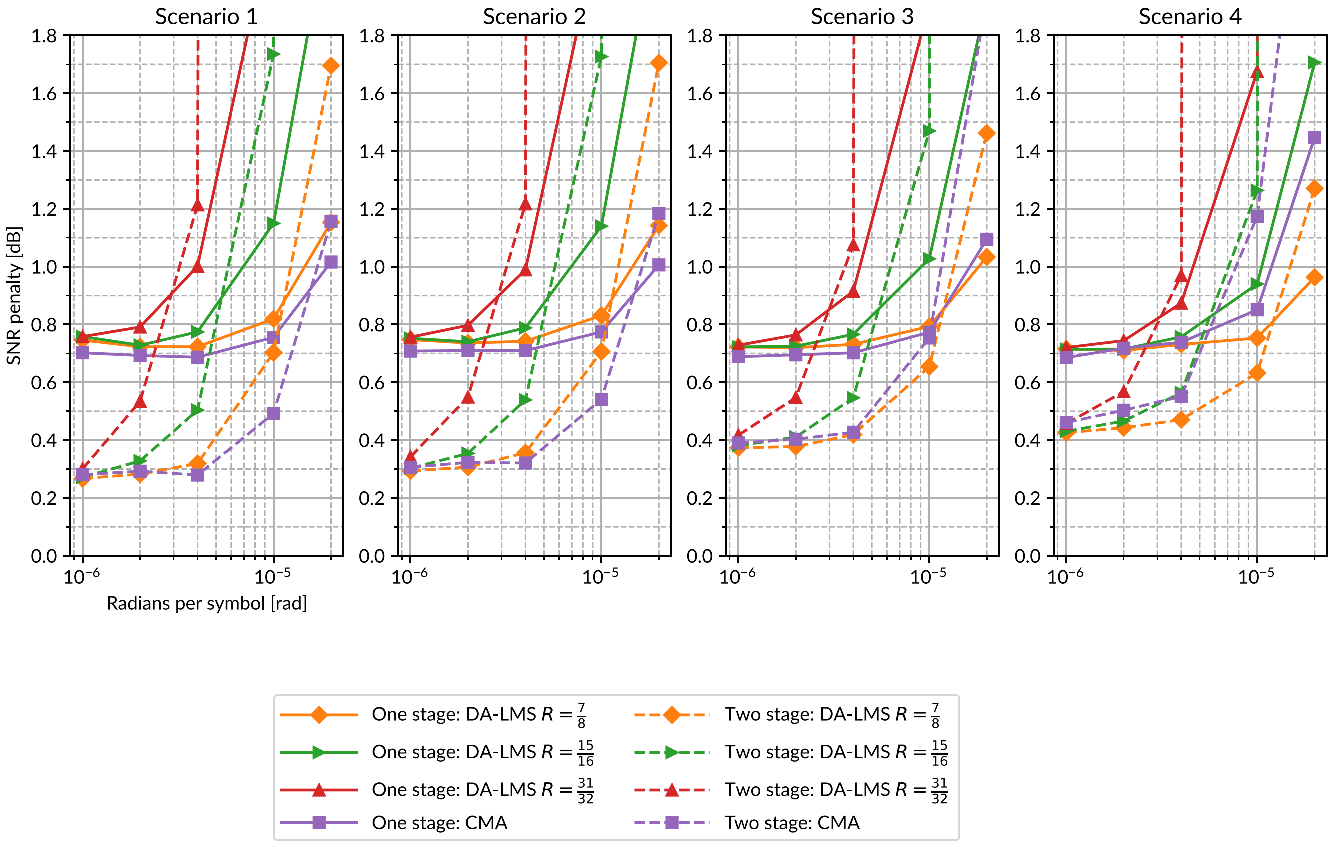

FIGURE 14 | SNR penalty as a function of the polarization rotation speed normalized by the symbol rate.

of the second stage of the two-stage equalizer also has to be estimated, but would not need to be updated as rapidly as it does not need to track the SOP. Therefore, the results in Figure 14 should be considered as a first estimate of the capabilities of the different algorithms and experimental testing is required to precisely quantify the performance of the two different equalizer structures. It may also be interesting to note that the difference between the four scenarios is quite small; scenarios 3 and 4, which feature deep fades, show only a slightly larger penalty compared with the static case in scenario 1. However, it should be noted that a relatively simple model of the polarization rotations was used here, and more complicated models as well as experimental studies would be required to ensure that the equalizers can track the desired polarization rotation speeds and do not get stuck in local optimums during signal fades. Finally, all three algorithms analyzed here, DA-LMS, CMA, and MMSE, could be potential candidates for OSL. MMSE has the lowest implementation complexity when implemented in the frequency domain, but is not suitable for tracking fast changes. DA-LMS and CMA perform quite similarly if pilot-based phase estimation is used and the pilot symbol penalty can be ignored. In addition, the LMS algorithm requires phase noise compensation within the equalizer feedback loop, which would increase latency and thus reduce tracking capabilities in a real-time receiver. On the other hand, DA-LMS has a much lower implementation complexity compared with the CMA algorithm, because only the pilot symbols are used to update the equalizer taps. Additionally, the probability of the equalizer diverging to a local optimum during deep fades has not been analyzed here but could have a major impact on the system, especially for the CMA equalizer, this would however require a more complex simulation to quantify. In conclusion, the design of the adaptive equalizer is a complex topic with many trade-offs, and it is crucial to properly characterize the system parameters such as DGD, PDL and polarization rotation speed in order to successfully design the equalizer.

7 | Conclusion

In this paper, we have outlined several potential DSP algorithms for the most important subsystems in coherent receivers for OSL. An extensive body of research has been dedicated to the design of DSP for terrestrial optical networks, and in fact many algorithms and approaches terrestrial receivers can be directly applied to OSL. However, there is still a considerable amount of research to be done to find the right implementation strategies and trade-offs for OSL. Data-aided receivers, in which pilot symbols are used to guide the receiver's DSP algorithms, appear to be an attractive choice for OSL providing the required performance at low SNR, and fast acquisition compared with blind receivers, which gives them an advantage in channels with deep fading channels. Additionally, data-aided DSP algorithms are significantly less complex and therefore can be realized with a much smaller footprint making them an attractive option for all OSL scenarios due to the imposed SWaP constraints onboard the satellites. Furthermore, data-aided algorithms provides an appealing path to scale toward higher modulation formats and flexible receivers as the DSP structure does not depend on the modulation format used, which is often the case for blind algorithms that tend to be modulation format specific. It is also important to stress the synergies and dependencies between different algorithms and subsystems, especially when considering data-aided algorithms. For example, data-aided algorithms for phase noise compensation and adaptive equalization could use the same pilot symbols but rely on the timing recovery and frame synchronization system to ensure correct alignment of the pilot symbols and will fail otherwise. Therefore, evaluating the complete DSP chain under realistic conditions is critical to understand the impact of possible design trade-offs. The selected implementation hardware platform will also play a major role in the design of the DSP algorithms. For example, an FPGA implementation of the DSP will be severely limited by the available resources and unable to operate at as high frequencies as an ASIC implementation, reducing the performance of feedback algorithms. Finally, the design of future DSP for coherent OSLs will be influenced by the capabilities of other key system components, for example, photonic-integrated circuits, ADCs, and optical amplifiers. While the use of COTS fiber optic transceivers provides a fast path to high-speed coherent OSLs demonstrations, to support future satellite networks with Tbit/s optical links, it will be critical to further minimize the SWaP, cost and reliability of the transceivers by tailoring the DSP architecture to the essentials. A customized architecture creates new opportunities, for instance in interfacing the physical, link, and network layer algorithms and their optimization as a whole. With the introduction of mega constellations and the integration of satellite networks with terrestrial networks, the development of custom intradyne optical transceivers for OSLs is therefore emerging as the most attractive option to meet the ever-increasing throughput demands.

Acknowledgments

The authors would like to thank Gianluca La Torre and Samuele Raffa for fruitful discussions on channel modeling for optical satellite links. Open Access funding enabled and organized by Projekt DEAL.

Conflicts of Interest

The authors declare no conflicts of interest.

References

1. D. Giggenbach, E. Lutz, J. Poliak, R. Mata-Calvo, and C. Fuchs, “A High-Throughput Satellite System for Serving Whole Europe With Fast Internet Service, Employing Optical Feeder Links,” in *Broadband Coverage in Germany 9th ITG Symposium Proceedings* (2015), 1–7.

2. R. Mata-Calvo, J. Poliak, J. Surof, et al., “Optical Technologies for Very High Throughput Satellite Communications,” in *Free-space Laser Communications XXXI*, Vol. 10910, SPIE. International Society for Optics and Photonics, (2019): 189–204.

3. D. J. Geisler, “Coherent Architectures for Free-Space Optical Communications,” in *2020 Conference on Lasers and Electro-optics* (2020), 1–2.

4. T. Pfau, H. Zhang, J. Geyer, and C. Rasmussen, “High Performance Coherent ASIC,” in *European Conference on Optical Communication*, (2018): 1–3.

5. C. M. Schieler, K. M. Riesing, B. C. Bilyeu, et al., “On-Orbit Demonstration of 200-Gbps Laser Communication Downlink From the TBIRD CubeSat,” in *Free-space Laser Communications XXXV*, Vol. 12413, SPIE. International Society for Optics and Photonics, (2023): 1241302, https://doi.org/10.1117/12.2651297.

6. M. Toyoshima, T. Jono, K. Nakagawa, and A. Yamamoto, “Optimum Divergence Angle of a Gaussian Beam Wave in the Presence of Random Jitter in Free-Space Laser Communication Systems,” *Journal of the Optical Society of America* 19, no. 3 (2002): 567–571, https://opg.optica.org/josaa/abstract.cfm?URI%3Djosaa-19-3-567.

7. D. Giggenbach, S. Parthasarathy, A. Shrestha, F. Moll, and R. Mata-Calvo, “Power Vector Generation Tool for Free-Space Optical Links PVGeT,” in *IEEE International Conference on Space Optical Systems and Applications*, (2017): 160–165.

8. D. Giggenbach, A. Shrestha, C. Fuchs, F. Moll, and K. Saucke, “Reference Power Vectors for the Optical LEO Downlink Channel,” in *2019 IEEE International Conference on Space Optical Systems and Applications, ICSOS 2019*, (2019), https://elib.dlr.de/130776/.

9. R. Barrios, S. Dimitrov, R. Mata Calvo, and D. Giggenbach, “Link Budget Assessment for GEO Feeder Links Based on Optical Technology,” *International Journal of Satellite Communications and Networking* (2020), https://elib.dlr.de/97153/.

10. F. Moll, D. Giggenbach, C. Schmidt, and C. Fuchs, “Analysis of Power Scintillation and Fading Margin in the LEO-Ground Downlink With the OSIRISv1 Laser Terminal on Flying Laptop and the DLR Optical Ground Station Oberpfaffenhofen,” in *Environmental Effects on Light Propagation and Adaptive Systems v 2022*, Proceedings of SPIE, (SPIE Press, 2022), https://elib.dlr.de/191905/.

11. A. M. Carrillo Flores, D. Giggenbach, M. T. Knopp, D. Orsucci, and A. Shrestha, “Effects of Pointing Errors on Intensity Losses in the Optical LEO Uplink,” in *2022 International Conference on Space Optics, ICSO 2022 Edited by* B. Cugny, N. Karafolas, and K. Minoglou, (2022), https://elib.dlr.de/191052/.

12. K. Kikuchi and S. Tsukamoto, “Evaluation of Sensitivity of the Digital Coherent Receiver,” *Journal of Lightwave Technology* 26, no. 13 (2008): 1817–1822.

13. O. Griebel, A. M. Sauter, U. Wasenmüller, L. Steiner, J. Poliak, and B. Matuz, “Physical Layer Forward Error Correction for Free-Space Optical Links,” in *2024 Spie Photonics West*, (2024), https://elib.dlr.de/203318/.

14. J. Poliak, H. Bischl, B. Matuz, et al., “Erasure Correcting Codes for High-Throughput Optical Ground-to-Satellite Links,” in *Free-space Laser Communications XXXVI 2024*, Proceedings of SPIE - The International Society for Optical Engineering, Vol. 12877, (2024), https://elib.dlr.de/203319/.

15. B. Mukherjee, I. Tomkos, M. Tornatore, P. Winzer, and Y. Zhao, *Springer Handbook of Optical Networks*, (Cham, Switzerland: Springer Nature, 2020).

16. J. Zhang, Z. Jia, M. Xu, H. Zhang, L. A. Campos, and C. Knittle, “High-Performance Preamble Design and Upstream Burst-Mode Detection in 100-Gb/s/λ TDM Coherent-PON,” in *2020 Optical Fiber Communications Conference and Exhibition* (2020), 1–3.

17. P. Conroy, J. Surof, J. Poliak, and R. Mata-Calvo, “Demonstration of 40 GBaud Intradyne Transmission Through Worst-Case Atmospheric Turbulence Conditions for Geostationary Satellite Uplink,” *Applied Optics* 57, no. 18 (2018): 5095–5101.

18. C. R. S. Fludger, T. Duthel, P. Hermann, and T. Kupfer, “Jitter Tolerant Clock Recovery for Coherent Optical Receivers,” in *Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference* (2013), 1–3.

19. D. Zibar, J. C. R. F. de Oliveira, V. B. Ribeiro, et al., “Experimental Investigation of Digital Compensation of DGD for 112 Gb/s PDM-QPSK Clock Recovery,” in *2011 37th European Conference and Exhibition on Optical Communication* (2011), 1–3.

20. F. N. Hauske, N. Stojanovic, C. Xie, and M. Chen, “Impact of Optical Channel Distortions to Digital Timing Recovery in Digital Coherent Transmission Systems,” in *12th International Conference on Transparent Optical Networks* (2010), 1–4.

21. L. Barletta, M. Magarini, F. Scardoni, and A. Spalvieri, “Impact of Loop Delay on the Performance of Gardner Timing Recovery,” *IEEE Photonics Technology Letters* 25, no. 18 (2013): 1797–1800.

22. C. Hebebrand, A. Napoli, A. Bianciotto, S. Calabr, B. Spinnler, and W. Rosenkranz, “Digital Clock Recovery With Adaptive Loop Gain to Overcome Channel Impairments in 112 Gbit/s CP-QPSK Receivers,” in *36th European Conference and Exhibition on Optical Communication* (2010), 1–3.

23. M. Yan, Z. Tao, L. Dou, et al., “Digital Clock Recovery Algorithm for Nyquist Signal,” in *Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference* (2013), 1–3.

24. R. M. Ferreira, J. D. Reis, S. B. Amado, et al., “Performance and Complexity of Digital Clock Recovery for Nyquist UDWDM-PON in Real Time,” *IEEE Photonics Technology Letters* 27, no. 21 (2015): 2230–2233.

25. N. Stojanovic, Y. Zhao, and C. Xie, “Feed-Forward and Feedback Timing Recovery for Nyquist and Faster Than Nyquist Systems,” in *Ofc 2014*, (2014): 1–3.

26. C. Li, S. Cui, J. Liu, et al., “Time-Domain Low-Complexity Clock Recovery for Non-Integer Oversampled Nyquist Signals With a Small Roll-Off Factor,” *Optics Express* 31, no. 11 (2023): 18599–18612, https://opg.optica.org/oe/abstract.cfm?URI%3Doe-31-11-18599.

27. F. N. Hauske, C. Xie, N. Stojanovic, and M. Chen, “Analysis of Polarization Effects to Digital Timing Recovery in Coherent Receivers of Optical Communication Systems,” in *Itg Symposium on Photonic Networks*, (2010): 1–3.

28. Y. Gu, S. Cui, C. Ke, K. Zhou, and D. Liu, “All-Digital Timing Recovery for Free Space Optical Communication Signals With a Large Dynamic Range and Low OSNR,” *IEEE Photonics Journal* 11, no. 6 (2019): 1–11.

29. F. Gardner, “A BPSK/QPSK Timing-Error Detector for Sampled Receivers,” *IEEE Transactions on Communications* 34, no. 5 (1986): 423–429.

30. X. Zhou and X. Chen, “Parallel Implementation of All-Digital Timing Recovery for High-Speed and Real-Time Optical Coherent Receivers,” *Optics Express* 19, no. 10 (2011): 9282–9295.

31. J. Tang, H. Chen, S. Cui, et al., “High Receiver Skew-Tolerant and Hardware-Efficient Clock Recovery for Short-Reach Coherent Transmission,” *Optics Express* 30, no. 15 (2022): 27064–27079.

32. D. Godard, “Passband Timing Recovery in an All-Digital Modem Receiver,” *IEEE Transactions on Communications* 26, no. 5 (1978): 517–523.

33. N. Stojanovi and X. Chuan, “Clock Recovery in Coherent Optical Receivers,” in *Optical Fiber Communications Conference and Exhibition*, (2015): 1–3.

34. B. Baeuerle, A. Josten, M. Eppenberger, D. Hillerkuss, and J. Leuthold, “Low-Complexity Real-Time Receiver for Coherent Nyquist-FDM Signals,” *Journal of Lightwave Technology* 36, no. 24 (2018): 5728–5737.

35. M. Oerder and H. Meyr, “Digital Filter and Square Timing Recovery,” *IEEE Transactions on Communications* 36, no. 5 (1988): 605–612.

36. S. J. Lee, “A New Non-Data-Aided Feedforward Symbol Timing Estimator Using Two Samples per Symbol,” *IEEE Communications Letters* 6, no. 5 (2002): 205–207.

37. K. Li, W. Zheng, and K. Li, “A Fast Algorithm With Less Operations for Length-$N = q \times 2^m$ DFTs,” *IEEE Transactions on Signal Processing* 63, no. 3 (2015): 673–683.

38. G. Bi and S. K. Mitra, “Sampling Rate Conversion in the Frequency Domain [DSP Tips and Tricks],” *IEEE Signal Processing Magazine* 28, no. 3 (2011): 140–144.

39. D. Fu and A. N. Willson, “Trigonometric Polynomial Interpolation for Timing Recovery,” *IEEE Transactions on Circuits and Systems I: Regular Papers* 52, no. 2 (2005): 338–349.

40. S. M. Bilal and C. R. S. Fludger, "Interpolators for Digital Coherent Receivers," in *Photonic Networks; 19th ITG-Symposium*, (2018): 1–3.

41. L. Erup, F. M. Gardner, and R. A. Harris, "Interpolation in Digital Modems. II. Implementation and Performance," *IEEE Transactions on Communications* 41, no. 6 (1993): 998–1008.

42. C. W. Farrow, "A Continuously Variable Digital Delay Element," in *IEEE International Symposium on Circuits and Systems*, (1988): 2641–2645 vol.3.

43. C. Valjus and R. Wolf, "Comparison of Timing Recovery Algorithms for Optical Feeder Links," in *Photonic Networks; 24th ITG-Symposium*, (2023): 1–5.

44. "Implementation Agreement 400ZR,", (2020), Standard, The Optical Internetworking Forum.

45. "Open ZR+ MSA Technical Specification 3.0," (2023), Srandard, The Optical Internetworking Forum.

46. "Open ROADM MSA Optical Specification 6.0," (2023), Standard, OpenROADM.

47. "Point-to-Point Coherent Optics Physical Layer 2.0 Specification," (2020), Standard, CableLabs.

48. E. Ip and J. M. Kahn, "Feedforward Carrier Recovery for Coherent Optical Communications," *Journal of Lightwave Technology* 25, no. 9 (2007): 2675–2692.

49. M. G. Taylor, "Phase Estimation Methods for Optical Coherent Detection Using Digital Signal Processing," *Journal of Lightwave Technology* 27, no. 7 (2009): 901–914.

50. A. J. Viterbi and A. M. Viterbi, "Nonlinear Estimation of PSK-Modulated Carrier Phase With Application to Burst Digital Transmission," *IEEE Transactions on Information Theory* 29, no. 4 (1983): 543–551.

51. S. M. Bilal, C. R. S. Fludger, V. Curri, and G. Bosco, "Multistage Carrier Phase Estimation Algorithms for Phase Noise Mitigation in 64-Quadrature Amplitude Modulation Optical Systems," *Journal of Lightwave Technology* 32, no. 17 (2014): 2973–2980.

52. E. Brjeson and P. Larsson-Edefors, "Benchmarking of Carrier Phase Recovery Circuits for M-QAM Coherent Systems," in *Optical Fiber Communications Conference and Exhibition* (2021), 1–3.

53. T. Pfau, S. Hoffmann, and R. Noe, "Hardware-Efficient Coherent Digital Receiver Concept With Feedforward Carrier Recovery for M-QAM Constellations," *Journal of Lightwave Technology* 27, no. 8 (2009): 989–999.

54. J. C. M. Diniz, Q. Fan, S. M. Ranzini, et al., "Low-Complexity Carrier Phase Recovery Based on Principal Component Analysis for Square-QAM Modulation Formats," *Optics Express* 27, no. 11 (2019): 15617–15626.

55. Kuschnerov, "Low Complexity Soft Differential Decoding of QPSK for Forward Error Correction in Coherent Optic Receivers," in *36th European Conference and Exhibition on Optical Communication*, Vol. 1 (2010), 1–3.

56. F. Yu, N. Stojanovic, F. N. Hauske, et al., "Soft-Decision LDPC Turbo Decoding for DQPSK Modulation in Coherent Optical Receivers," in *2011 37th European Conference and Exhibition on Optical Communication* (2011), 1–3.

57. A. Spalvieri and L. Barletta, "Pilot-Aided Carrier Recovery in the Presence of Phase Noise," *IEEE Transactions on Communications* 59, no. 7 (2011): 1966–1974.

58. C. S. Martins, F. P. Guiomar, and A. N. Pinto, "Hardware Optimization of Dual-Stage Carrier-Phase Recovery for Coherent Optical Receivers," *OSA Continuum* 4, no. 12 (2021): 3157–3175.

59. G. Tzimpragos, C. Kachris, I. Djordjevic, M. Cvijetic, D. Soudris, and I. Tomkos, "A Survey on FEC Codes for 100 G and Beyond Optical Networks," *IEEE Communications Surveys and Tutorials* 18 (2014).

60. A. Leven, N. Kaneda, U.-V. Koc, and Y.-K. Chen, "Frequency Estimation in Intradyne Reception," *IEEE Photonics Technology Letters* 19, no. 6 (2007): 366–368.

61. S. Almonacil, R. Boddeda, T. Allain, D. R. Arrieta, and S. Bigo, "Digital Pre-Compensation of Doppler Frequency Shift in Coherent Optical Satellite Communications," in *European Conference on Optical Communications*, (2020): 1–4.

62. S. Schaefer, W. Rosenkranz, and M. Gregory, "Comparison of Homodyne and Intradyne Detection for High-Order Modulation Schemes in Optical Intersatellite Communication Systems," in *2015 IEEE International Conference on Space Optical Systems and Applications* (2015), 1–5.

63. K. Kikuchi, "Fundamentals of Coherent Optical Fiber Communications," *Journal of Lightwave Technology* 34, no. 1 (2016): 157–179.

64. "ESA Specification for Terabit/Sec Optical Links (ESTOL),", (2023), Standard, European Space Agency.

65. D. Huang, T.-H. Cheng, and C. Yu, "Accurate Two-Stage Frequency Offset Estimation for Coherent Optical Systems," *IEEE Photonics Technology Letters* 25, no. 2 (2013): 179–182.

66. M. Selmi, Y. Jaouen, and P. Ciblat, "Accurate Digital Frequency Offset Estimator for Coherent PolMux QAM Transmission Systems," in *2009 35th European Conference on Optical Communication* (2009), 1–2.

67. M. Morelli and U. Mengali, "Feedforward Frequency Estimation for PSK: A Tutorial Review," *European Transactions on Telecommunications* 9, no. 2 (1998): 103–116.

68. T. M. Schmidl and D. C. Cox, "Robust Frequency and Timing Synchronization for OFDM," *IEEE Transactions on Communications* 45, no. 12 (1997): 1613–1621.

69. H. Y. Rha, C. J. Youn, E. S. Nam, and H.-W. Choi, "Simple Full-Range Carrier Frequency Offset Estimation for High Speed CO-OFDM," *Optics Express* 21, no. 20 (2013): 23896–23906, https://opg.optica.org/oe/abstract.cfm?URI%3Doe-21-20-23896.

70. A. Barbieri and G. Colavolpe, "On the Cramer-Rao Bound for Carrier Frequency Estimation in the Presence of Phase Noise," *IEEE Transactions on Wireless Communications* 6, no. 2 (2007): 575–582.

71. A. N. D'Andrea, U. Mengali, and R. Reggiannini, "The Modified Cramer-Rao bound and Its Application to Synchronization Problems," *IEEE Transactions on Communications* 42, no. 234 (1994): 1391–1399.

72. I. P. Vieira, T. C. Pita, and D. A. A. Mello, "Modulation and Signal Processing for LEO-LEO Optical Inter-Satellite Links," *IEEE Access* 11 (2023): 63598–63611.

73. S. J. Savory, "Digital Filters for Coherent Optical Receivers," *Optics Express* 16, no. 2 (2008): 804–817, http://www.osapublishing.org/oe/abstract.cfm?URI%3De-16-2-804.

74. C. Fougstedt, O. Gustafsson, C. Bae, E. Börjeson, and P. Larsson-Edefors, "ASIC Design Exploration for DSP and FEC of 400-Gbit/s Coherent Data-Center Interconnect Receivers," in *Optical Fiber Communication Conference 2020* (Optical Society of America, 2020), Th2A.38.

75. T. Kupfer, A. Bisplinghof, T. Duthel, C. Fludger, and S. Langenbach, "Optimizing Power Consumption of a Coherent DSP for Metro and Data Center Interconnects," in *Optical Fiber Communications Conference and Exhibition*, (2017): 1–3.

76. K. Matsuda, R. Matsumoto, and N. Suzuki, "Hardware-Efficient Adaptive Equalization and Carrier Phase Recovery for 100-Gb/s/λ-Based Coherent WDM-PON Systems," *Journal of Lightwave Technology* 36, no. 8 (2018): 1492–1497.

77. X. Zhang, X. Li, T. Zeng, et al., "Real Time Low-Complexity Adaptive Channel Equalization for Coherent Optical Transmission Systems," *Optics Express* 28, no. 4 (2020): 5058–5068.

78. B. Spinnler, F. N. Hauske, and M. Kuschnerov, "Adaptive Equalizer Complexity in Coherent Optical Receivers," in *European Conference on Optical Communication*, (2008): 1–2.

79. K. Matsuda, S. Koshikawa, T. Yoshida, N. Suzuki, and T. Ando, "FPGA Implementation of Scintillation Tolerant Adaptive DSP for 4 Gbps Coherent Reception," in *Free-space Laser Communications XXXII*, Vol. 11272, SPIE. International Society for Optics and Photonics, (2020): 149–157.

80. D. Falconer, S. L. Ariyavisitakul, A. Benyamin-Seeyar, and B. Eidson, "Frequency Domain Equalization for Single-Carrier Broadband Wireless Systems," *IEEE Communications Magazine* 40, no. 4 (2002): 58–66.

81. M. Kuschnerov, M. Chouayakh, K. Piyawanno, et al., "Data-Aided Versus Blind Single-Carrier Coherent Receivers," *IEEE Photonics Journal* 2, no. 3 (2010): 387–403.

82. B. Spinnler, S. Calabr, and M. Kuschnerov, "Pilot-Assisted Channel Estimation Methods for Coherent Receivers," in *Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference* (2013), 1–3.

83. J. Yang, E. Sillekens, W. Yi, P. Bayvel, and R. I. Killey, "Joint Estimation of Dynamic Polarization and Carrier Phase With Pilot-Based Adaptive Equalizer in PDM-64 QAM Transmission System," *Optics Express* 29, no. 26 (2021): 43136–43147, https://opg.optica.org/oe/abstract.cfm?URI%3Doe-29-26-43136.

84. R. Johnson, P. Schniter, T. J. Endres, J. D. Behm, D. R. Brown, and R. A. Casas, "Blind Equalization Using the Constant Modulus Criterion: A Review," *Proceedings of the IEEE* 86, no. 10 (1998): 1927–1950.

85. I. Fatadin, D. Ives, and S. J. Savory, "Blind Equalization and Carrier Phase Recovery in a 16-QAM Optical Coherent System," *Journal of Lightwave Technology* 27, no. 15 (2009): 3042–3049.

86. K. Kikuchi, "Performance Analyses of Polarization Demultiplexing Based on Constant-Modulus Algorithm in Digital Coherent Optical Receivers," *Optics Express* 19, no. 10 (2011): 9868–9880.

87. B. Farhang-Boroujeny and K. S. Chan, "Analysis of the Frequency-Domain Block LMS Algorithm," *IEEE Transactions on Signal Processing* 48, no. 8 (2000): 2332–2342.

88. C. K. Chan, M. R. Petraglia, and J. J. Shynk, "Frequency-Domain Implementations of the Constant Modulus Algorithm," in *Twenty-third Asilomar Conference on Signals, Systems and Computers, 1989*, Vol. 2, (1989): 663–669.

89. B. Farhang-Boroujeny, *Adaptive Filters: Theory and Applications*, (Wiley, 2013).

90. R. H. Kwong and E. W. Johnston, "A Variable Step Size LMS Algorithm," *IEEE Transactions on Signal Processing* 40, no. 7 (1992): 1633–1642.