Cycle-Slip Rate Analysis of Blind Phase Search DSP Circuit Implementations

Citation for the original published paper (version of record):

Börjeson, E., Larsson-Edefors, P. (2020)
Cycle-Slip Rate Analysis of Blind Phase Search DSP Circuit Implementations
2020 Optical Fiber Communications Conference and Exhibition, OFC 2020 - Proceedings
http://dx.doi.org/10.1364/OFC.2020.M4J.3

Cycle-Slip Rate Analysis of Blind Phase Search DSP Circuit Implementations

Erik Börjeson and Per Larsson-Edefors
Dept. of Computer Science and Engineering, Chalmers University of Technology, Gothenburg, Sweden
erikbor@chalmers.se

Abstract: Using FPGA-accelerated simulations, we study the cycle-slip rate of 16QAM blind phase search implementations. While block averaging suffers from degraded BER when compared to sliding-window averaging, it results in lower cycle-slip rates and power dissipation.

OCIS codes: (060.0060) Fiber optics and optical communication; (060.1660) Coherent communications

1. Introduction

Coherent transmission is one of the key technologies used to increase the spectral efficiency of fiber-optic communication systems, especially for long-haul transmission. These systems rely on extensive digital signal processing (DSP) in application-specific integrated circuits (ASICs) to reduce the impact of transmission impairments caused by the optical fiber and other non-ideal system components. Since DSP power can be a significant part of the total system power dissipation, designing capable yet power-efficient DSP is essential to allow for more densely packed equipment and to enable coherent transmission also for shorter-reach systems.

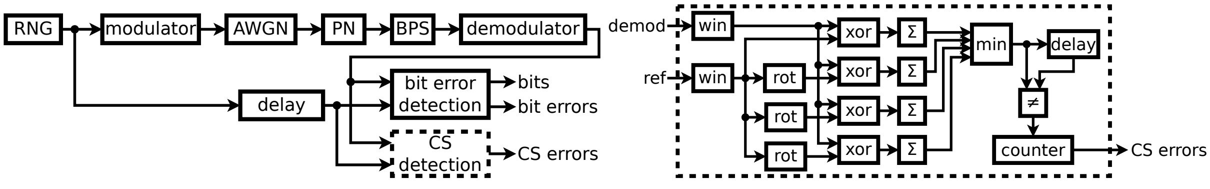
Consider, e.g., the limited linewidth of the carrier and local oscillator lasers which cause phase noise; a time-varying phase rotation of the transmitted symbols. This impairment is handled by the DSP's carrier phase estimation (CPE) unit, using either algorithms that are data aided [1], non data aided (blind) [2], or combinations of the two [3]. The data-aided approaches are typically based around known pilot symbols which are time-division multiplexed with the symbol stream: Since inserting pilot symbols has the disadvantage of reduced spectral efficiency, the pilot overhead needs to be minimized. On the other hand, non-data-aided CPE uses the transmitted data symbols to estimate the current phase. However, blind CPE approaches typically suffer from the problem of cycle-slip (CS) errors, where the received symbol is erroneously rotated by multiples of $\pi/2$. In contrast to common DSP design trade-offs involving bit-error rate (BER) performance versus power dissipation, CS errors are very hard to recover from and can potentially lead to a catastrophic transmission failure. This makes DSP design trade-offs involving CS errors challenging.

Since CS errors are typically few and far between, they are hard to detect using conventional simulation tools, in reasonable run-times. In this work, we use our newly developed system-emulation environment [4], running on a field-programmable gate array (FPGA), to study how different DSP circuit implementation strategies and parameter settings affect the cycle-slip rate (CSR), i.e., the number of CS errors per transmitted symbol, of a blind phase search (BPS) CPE implementation. We also discuss trade-offs related to BER, CSR, and DSP circuits, and how a combination of a pilot-based CPE and BPS can affect the cycle-slip rate.

2. Blind Phase Search

Our BPS implementation is based on the algorithm of Pfau et al. [2]: When applying the BPS algorithm, the input symbols are rotated by a number of test phases, after which the distance to the closest constellation point for each of these rotated inputs is calculated. To reduce the impact of white noise, an average distance over a number of input samples is computed and the test phase with the smallest average distance to a constellation point is selected. Phase noise is removed from the input samples by back-rotation using the estimated phase.

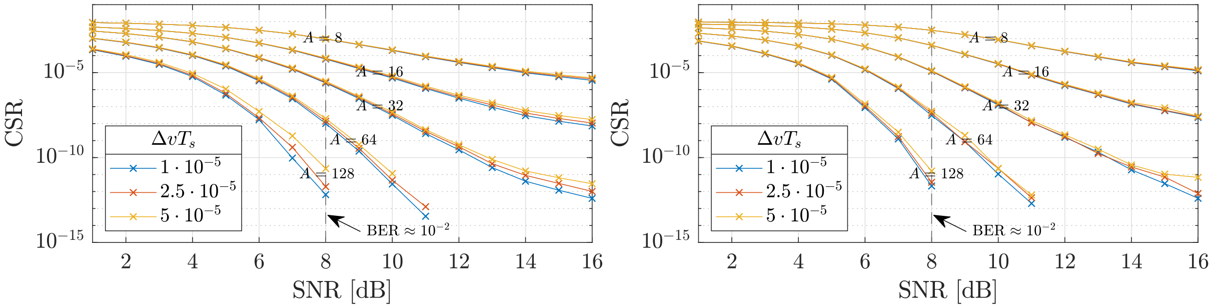
During DSP implementation, algorithmic simplifications can significantly reduce power dissipation; typically at a cost of BER penalties: Our BPS circuit implementation [5] uses several such DSP-hardware-centric modifications. In this work, however, the block averaging (BA) method solely used in our previous design is complemented by the method of sliding-window averaging (SA), used in the original algorithm [2], to allow us to discern the effects of averaging on the cycle-slip rate. The circuit implementation has three main design parameters, affecting both the output quality and power dissipation: The input resolution (word length), which also affects the internal resolution, the number of test phases, and the size of the averaging window.



(a)

(b)

Fig. 1: Block diagram of (a) FPGA environment and (b) CS detection implementation.



(a)

(b)

Fig. 2: CSR as a function of SNR for BPS implementations using (a) block and (b) sliding-window averaging.

3. Emulation Environment

We emulate a fiber-optic communication system using our FPGA environment [4], whose high real-time performance allows us to *digitally* simulate long transmissions in a short time frame. A block diagram of the emulation environment is shown in Fig. 1a. The input data are generated using a pseudo-random number generator (RNG) with a periodicity of 2^{64} [6] and modulated using a 16QAM modulation format. In our simulations, we assume that linear fiber impairments, such as chromatic dispersion and polarization-mode dispersion, are totally compensated for by other DSP components, and we neglect fiber nonlinearities. Gaussian number generators [7] are used to emulate not only additive white Gaussian noise (AWGN), but also the Wiener process of the phase noise (PN). The resulting symbol stream is processed by our BPS implementation prior to demodulation. Finally, as shown in Fig. 1a, the two outputs, one from the demodulator and one being a delayed version of the original input bits, are fed to a bit error detection unit that counts the number of processed bits and the number of errors.

The two outputs are also fed to a cycle-slip (CS) detection unit, shown in Fig. 1b, where detection of CS errors is performed by matching a set window (win) of the demodulated bitstream with $\pi/2$ rotated versions of the original (rot). The matching is performed by bitwise XOR of the two bit vectors and counting the number of 1s in the result (Σ). If the best match differs from the previous match, a cycle slip has occurred and the CS error counter is incremented.

The calculation of very low CSRs requires processing of a large number of input bits. For our experiments we use a Xilinx VCU110 development board [8], which allows us to process approximately $3 \cdot 10^{10}$ bits/s when running our emulation environment on the on-board Virtex UltraScale FPGA. This speed enables us to calculate BERs and CSRs as low as 10^{-14} in a matter of a few hours. Previous findings suggest that the output results are in agreement with results calculated using data and impairment generation in a floating-point environment [4].

4. Results

Early exploratory simulations showed that the design parameter having the largest impact on the CSR is the averaging window size (A). Thus, we focus on simulations where we vary A, while keeping the other parameters constant. Based on our previous work [5], we set the number of test phases to 8 and use an input word length of 8 bits for all implementations described here. The averaging window is increased in powers of two, due to the design of BA-based circuit, and the result is shown in Fig. 2a for three representative linewidth symbol-duration products ($\Delta v T_s$).

The implementations using a larger A show better resilience to CS errors, indicating that AWGN is the main source of these errors. When studying the result at an SNR of 8 dB, i.e., close to the soft-decision forward error correction (FEC) threshold in the order of a BER of 10^{-2}, the CSR decreases multiple orders of magnitude for each increase in window size. However, a small CSR penalty can be seen for higher $\Delta v T_s$, especially for larger values of A, where the phase changes can grow larger inside the span of the window. An interesting aspect is that the CSR curves level out

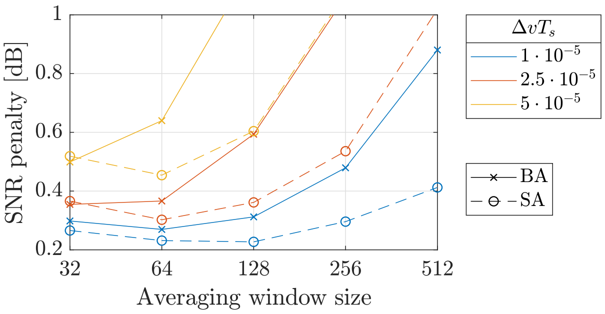


Fig. 3: SNR penalty compared to the theoretical minimum.

	BA		SA	
	Area [µm^2]	Power [µW]	Area [µm^2]	Power [µW]
Estimation	28,000	79	143,000	334
Averaging	2,140	3	110,000	255
Compensation	20,400	35	25,263	37
Total*	58,700	149	182,000	415

*Including pipeline registers and clock gating logic.

Table 1: Area and power dissipation for BPS circuits implemented using a 22-nm 0.8-V CMOS technology.

at high SNRs, which suggests that the BPS algorithm is sensitive to CS errors even under conditions with very low AWGN. CSR results from simulations using the SA method are shown in Fig. 2b and display similar properties as BA. However, SA performs worse than BA for low SNRs, as only one CS error can occur per block in the BA method; SA has no such limitations. This behavior is less prominent at higher SNRs, where the probability of having multiple CS errors in one block is very low.

Fig. 3 shows the SNR penalty at a BER of 10^{-2} for the two averaging methods. The SA implementations have their minimum at larger window sizes and also show a lower penalty than BA. To study how the choice of averaging affects the power dissipation and area of ASIC circuits, we synthesized both SA and BA designs using a 22-nm cell library. Synthesis results for a 32-GBd 16QAM BPS unit with $A = 64$ and a clock rate of 1 GHz are shown in Table 1. The power dissipation of the SA version is almost three times higher than for BA, uncovering a clear trade-off between output quality and energy efficiency. (Note, however, that CPE complexity is relatively low if compared to, e.g., dynamic equalization [9].) Fig. 3 also shows that higher $\Delta v T_s$ results in a shorter average to minimize the penalty, However, the CSR at the FEC limit, shown as dashed lines in Fig. 2, is higher for smaller averaging windows, resulting in a trade-off between low BER and low CSR.

One way to improve phase estimation is to precede the blind CPE with a pilot-symbol-aided CPE stage, which can work as a coarse phase tracker [3]. When using the two types of estimators in tandem, the pilot overhead can be kept to a minimum while still reducing the low-frequency phase noise. However, in this study, high CSRs were detected for low SNRs in combination with short averaging windows. Since the window size is limited by the high-frequency portion of the phase noise, this approach will most probably only have a minor impact on the CSR, even though the BER can be increased.

5. Conclusion

Using FPGA-based emulation, we have shown how the cycle-slip rate (CSR) of a blind phase search (BPS) circuit implementation is affected by averaging method and window size. Compared to sliding-window averaging, power-efficient block averaging was found to have a slight BER penalty of < 0.1 dB, while it in fact improves the cycle-slip rate. Cycle-slip errors are caused mainly by AWGN in conjunction with the short averaging windows needed for high BER performance, suggesting that inserting pilot-based carrier-phase estimation before the BPS will not have a major impact on the CSR, at least not for the $\Delta v T_s$ values considered here.

Acknowledgement: This work was financially supported by the Knut and Alice Wallenberg Foundation and Vinnova.

References

[1] A. Spalvieri *et al.*, "Pilot-aided carrier recovery in the presence of phase noise," IEEE Trans. Commun. **59**, 1966–1974 (2011).
[2] T. Pfau *et al.*, "Hardware-efficient coherent digital receiver concept with feedforward carrier recovery for M-QAM constellations," IEEE JLT **27**, 989–999 (2009).
[3] M. Magarini *et al.*, "Pilot-symbols-aided carrier-phase recovery for 100-G PM-QPSK digital coherent receivers," IEEE PTL **24**, 739–741 (2012).
[4] E. Börjeson *et al.*, "Towards FPGA emulation of fiber-optic channels for deep-BER evaluation of DSP implementations," in "SPPCom," (2019), p. SpTh1E.4.
[5] E. Börjeson *et al.*, "ASIC design exploration of phase recovery algorithms for M-QAM fiber-optic systems," in "OFC," (2019), p. W3H.7.
[6] G. Marsaglia, "Xorshift RNGs," J. Stat. Softw. **8**, 1–6 (2003).
[7] R. Gutierrez *et al.*, "Hardware architecture of a Gaussian noise generator based on the inversion method," IEEE TCAS II **59**, 501–505 (2012).
[8] Xilinx Inc., *Virtex Ultrascale FPGA VCU110 Development Kit* (2019). https://www.xilinx.com/products/boards-and-kits/dk-u1-vcu110-g.html.
[9] D. A. Morero *et al.*, "Design tradeoffs and challenges in practical coherent optical transceiver implementations," IEEE JLT **34**, 121–136 (2016).