Benchmarking of Carrier Phase Recovery Circuits for M-QAM Coherent Systems

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

Börjeson, E., Larsson-Edefors, P. (2021). Benchmarking of Carrier Phase Recovery Circuits for M-QAM Coherent Systems. Optical Fiber Communication Conference, OFC 2021

Benchmarking of Carrier Phase Recovery Circuits for M-QAM Coherent Systems

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

Abstract: We benchmark blind carrier phase recovery DSP circuits in terms of SNR penalty, power dissipation, latency, area usage, and cycle slip probability, to identify optimal implementations for 16, 64, and 256QAM.

1. Introduction

Carrier phase recovery (CPR) is a key component of the receiver DSP used in fiber-optic coherent communication systems. The use of higher-order modulation formats makes the spectral efficiency increase, but comes at a cost of higher susceptibility to phase noise introduced by the carrier and local-oscillator lasers, which makes the requirements on CPR circuits stricter. For short-reach systems, reducing CPR power dissipation becomes especially important. This is because other parts of the DSP, such as chromatic dispersion and PMD compensation, can be simplified or even eliminated, potentially making CPR a dominant portion of DSP power dissipation.

Benchmarking of different CPR algorithms has previously been based on complexity metrics, such as number of operations, and BERs which are obtained from algorithms analyzed in ideal floating-point environments. These types of metrics, however, do not account for the intricacies of an actual circuit implementation of an algorithm and fail to capture fixed-point aspects, arithmetic approximations, and circuit optimizations in general.

Using a combination of FPGA emulation and ASIC analysis, we present SNR performance penalty, power dissipation, latency, area usage, and cycle slip probability for five different CPR implementations. Using a consistent benchmarking methodology, for 16, 64, and 256QAM, we can extend our previous implementation work [1, 2] and identify which CPR circuits are optimal for a particular M-QAM format.

2. Carrier Phase Recovery Algorithms Considered

Of the many CPR algorithms suggested, we have identified five candidates: A **modified Viterbi-Viterbi (mVV)** CPR uses QPSK partitioning of the QAM symbols to facilitate the use of the Mth-power phase estimator [3] for higher-order modulation formats, typically averaging over N consecutive symbols to reduce the impact of AWGN. An alternative approach is the **blind phase search (BPS)** algorithm [4], where the input symbols are rotated with B test phases after which the distance to a valid constellation point is calculated for each rotated symbol. The rotation resulting in the smallest average distance is chosen as the output. In **principal component-based phase estimation (PCPE)** [5], the power iteration method is used to calculate a covariance matrix over N squared input symbols, and the result is used to extract the principal component from which the phase noise can be estimated.

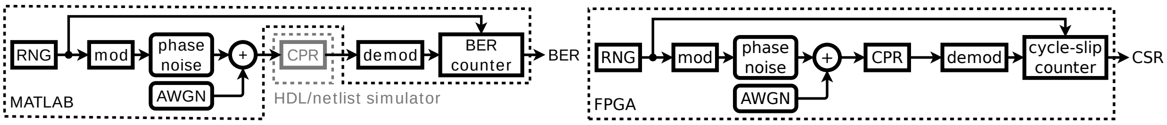
The mVV and PCPE algorithms have a lower complexity compared to BPS, but they have a larger residual phase noise and a larger SNR penalty. Thus, multi-stage CPR approaches have been suggested. In [5], PCPE is followed by a BPS (**PCPE+BPS**) with few test phases and in [3] mVV is followed by a constellation transformation (**mVV+CT**). In CT, the QAM symbols are transformed to QPSK and the Mth power method is used to perform the fine-grain phase estimation; a method that only works if the residual phase noise is small enough.

3. Evaluation Methodology

The CPR implementations were developed in a hardware description language (HDL) and evaluated in two ways:

MATLAB-HDL co-simulation was used to find the parameter settings resulting in a good tradeoff between SNR penalty and power dissipation. As shown in Fig. 1a, the bitstream and impairments are generated in MATLAB before being fed to an HDL model of the CPR circuit. The result of HDL simulations is fed back into MATLAB for BER calculations. This approach ensures that the results faithfully capture fixed-point aspects and different circuit optimizations. In addition, to estimate ASIC area usage and power dissipation, the HDL models are mapped (synthesized) to 22-nm CMOS netlists, using a 0.72-V, 125-°C characterization, at the slow process corner. The 22-nm netlists are simulated using the MATLAB-HDL model and the resulting switching activity statistics are used for power analysis, using the typical process corner and a 0.8-V, 85-°C characterization.

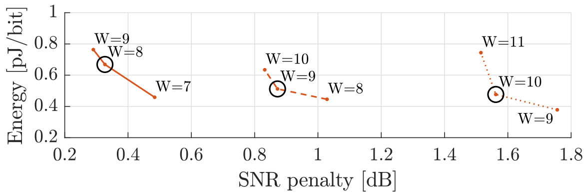
For the benchmarking in Section 4, we want to use parameters representing the best design tradeoff: Fig. 2a shows the tradeoff between SNR penalty and power dissipation for PCPE, considering varying input word lengths.



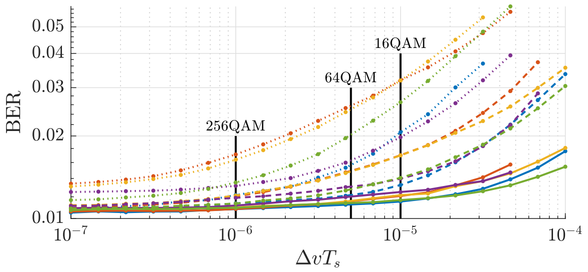
(a) MATLAB-HDL co-simulation.

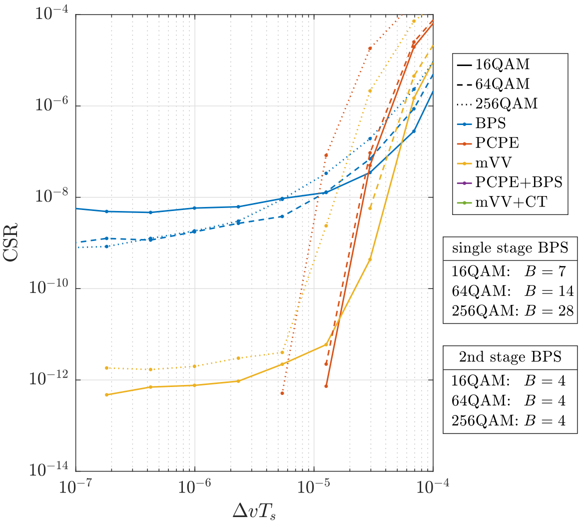
(b) FPGA-based emulation.

Fig. 1: System models used to retrieve (a) BER and switching activities for ASIC power estimation, and (b) CSR.



(a) Word length (W) tradeoff for PCPE. As shown, W has to be incremented by one bit as we increase the modulation order.





(b) Linewidth sensitivity of CPR circuit. The black lines indicate the choice of $\Delta\nu T_s$ for CSR emulations and ASIC netlists.

(c) CSR as a function of the linewidth symbol-duration product. The number of test phases, B, used are shown to the right.

Fig. 2: Simulation results from (a) and (b) MATLAB-HDL co-simulations and (c) FPGA emulator.

The selected tradeoff is circled in black, for each modulation format; here, clearly, a low SNR penalty was prioritized over low power dissipation. Similar analyses are performed for all CPR circuits and parameters. The parameter settings are optimized for a target BER of 10^{-2}, approximating the soft-FEC BER limit, and a linewidth symbol-duration ($\Delta\nu T_s$) of 10^{-5}, $5\cdot10^{-6}$, and 10^{-6} for 16, 64, and 256QAM, respectively.

FPGA emulation was used to estimate the cycle-slip rate (CSR), i.e., the number of cycle slips per transmitted symbol. This is because MATLAB-HDL co-simulation is too slow and yields prohibitively long runtimes for CSR analysis. Our FPGA environment [6] allows us to emulate the channel and run CPR implementations onboard the FPGA, as shown in Fig. 1b, resulting in orders-of-magnitude faster runtimes than MATLAB-HDL co-simulation. For the emulation runs, the SNR was held constant at 7.9, 12.0, and 16.4 dB for 16, 64, and 256QAM, respectively. These SNR values correspond to a theoretical BER of 10^{-2}, considering an AWGN channel without phase noise. In addition, the parameter settings resulting in the lowest SNR at the $\Delta\nu T_s$ values marked in Fig. 2b were used.

4. Results

The linewidth sensitivity, shown as the BER at the SNRs described above, is presented in Fig. 2b. For 16QAM, the difference between the CPR approaches is small, but for higher-order QAM the benefit of the 2-stage circuits is clear, especially at higher values of $\Delta\nu T_s$. For 64QAM and $\Delta\nu T_s > 2\cdot10^{-5}$, 2-stage CPR circuits are comparable to 1-stage BPS with $B = 14$. For 256QAM, mVV and PCPE circuits have high BERs and PCPE+BPS outperforms the other CPR circuits at higher $\Delta\nu T_s$. The linewidth sensitivity is relatively stable up to our selected points on the X-axis, illustrating how the laser linewidth requirements differ for the different modulation formats.

Results from our CSR emulations are shown in Fig. 2c (only 1-stage circuits are used, as no cycle slips can occur in a second stage, since this has no unwrapping). BPS has a much higher CSR than PCPE and mVV; this may be due be the shorter optimal averaging window needed to reach a low BER. We have previously shown that the window size has a significant effect on CSR for BPS [7], exposing a tradeoff between low BER and low CSR. A similar CSR floor is seen for mVV, potentially a result from the limited number of input symbols used for estimation. The CSR of PCPE shows steeper slopes and the emulations were stopped when no cycle slips were detected after processing at least 10^{15} symbols, due to the runtimes becoming prohibitively long.

Table 1 presents the results from BER simulations and ASIC synthesis runs. The HDL models were designed for a 32-parallel implementation using a clock rate of 937.5 MHz, resulting in a throughput of 30 GBaud using a single polarization. Note that processing of two polarizations does not necessarily result in a doubling of area usage and power dissipation, as joint CPR is possible [8].

Modulation format	CPR method	Penalty [dB]	Area [mm^2]	Norm. area	Power [mW]	Norm. power	Energy [pJ/bit]	Latency [#cycles]
16QAM W=8	BPS (N=64, B=7)	0.26	0.054	1	136	1	1.14	$6+N/P$
	PCPE (N=96)	0.33	0.048	0.88	80	0.59	0.67	$6+N/P$
	mVV (N=128)	0.31	0.051	0.93	80	0.59	0.67	$5+N/P$
	PCPE+BPS (N_1=96, N_2=32, B=4)	0.47	0.075	1.36	150	1.10	1.25	$9+(N_1+N_2)/P$
	mVV+CT (N_1=128, N_2=32)	0.25	0.096	1.75	168	1.23	1.40	$9+(N_1+N_2)/P$
64QAM W=9	BPS (N=64, B=14)	0.35	0.112	1	302	1	1.68	$6+N/P$
	PCPE (N=96)	0.86	0.057	0.50	92	0.31	0.51	$6+N/P$
	mVV (N=160)	0.84	0.070	0.62	108	0.36	0.60	$5+N/P$
	PCPE+BPS (N_1=96, N_2=32, B=4)	0.51	0.092	0.82	181	0.60	1.01	$9+(N_1+N_2)/P$
	mVV+CT (N_1=192, N_2=32)	0.43	0.128	1.14	218	0.72	1.21	$9+(N_1+N_2)/P$
256QAM W=10	BPS (N=128, B=28)	0.37	0.267	1	738	1	3.08	$6+N/P$
	PCPE (N=256)	1.51	0.082	0.31	114	0.16	0.48	$6+N/P$
	mVV (N=384)	1.29	0.095	0.36	114	0.15	0.47	$5+N/P$
	PCPE+BPS (N_1=192, N_2=32, B=4)	0.54	0.123	0.46	247	0.33	1.02	$9+(N_1+N_2)/P$
	mVV+CT (N_1=384, N_2=32)	0.60	0.166	0.62	254	0.35	1.06	$9+(N_1+N_2)/P$

Table 1: Synthesis and simulation results, where *Norm.* values are normalized to BPS using the same modulation format. W is the word length, P is the parallelization factor and N_n is the averaging window size of the nth stage.

The SNR penalty of 1-stage CPR approaches is comparable for the three 16QAM implementations; PCPE and mVV however dissipate much less power than BPS. For 16QAM, PCPE+BPS does not decrease the penalty, due to the fixed-point errors introduced by the additional processing in the 2nd BPS stage. For the higher-order 64 and 256QAM implementations, the penalty of 1-stage PCPE and mVV becomes much higher than for 1-stage BPS, but the power dissipation is significantly lower, due the large number of extra test phases needed to reach a good SNR performance for BPS. For these modulation formats, the 2-stage approaches also start to become a valid alternative when considering the tradeoff between power dissipation and SNR penalty. For 256QAM, the energy per bit of PCPE+BPS is very similar to that of mVV+CT, however, mVV+CT requires significantly more logic resources. This discrepancy between area usage and power dissipation for mVV+CT is due to the lower switching activity, caused by many symbols being set to zero in the partitioning.

If the carrier phase estimation is part of a feedback loop, e.g., in a decision-directed equalizer, the latency can become an issue. For our implementations, the difference in optimum length of the averaging window between the approaches is the main parameter contributing to latency. The 2-stage CPR circuits have a substantially larger latency, caused by the longer pipeline and the two different averaging windows needed.

5. Conclusion

We have shown that different carrier phase recovery implementations differ in terms of tradeoffs between SNR penalty and power dissipation and that 2-stage approaches become effective for 64QAM and above. PCPE followed by a simplified BPS stage is an interesting option for 256QAM, as it offers a good tradeoff between SNR penalty and power dissipation. For 64QAM, 2-stage CPR circuits prove to be good options if low penalty is prioritized, while the 1-stage mVV and PCPE circuits are better when striving for high energy efficiency. For 16QAM, the 1-stage PCPE and mVV result in slightly higher penalties than BPS, but at a much lower power dissipation. The CSR of BPS is largely affected by the choice of averaging window and is higher than mVV and PCPE, of which the latter shows the best resilience to cycle slips at lower laser linewidths.

References

[1] E. Börjeson *et al.*, "VLSI implementations of carrier phase recovery algorithms for M-QAM fiber-optic systems," IEEE JLT **38**, 3616–3623 (2020).

[2] E. Börjeson *et al.*, "Energy-efficient implementation of carrier phase recovery for higher-order modulation formats," IEEE JLT **39**, 505–510 (2021).

[3] S. M. Bilal *et al.*, "Multistage carrier phase estimation algorithms for phase noise mitigation in 64-quadrature amplitude modulation optical systems," IEEE JLT **32**, 2973–2980 (2014).

[4] T. Pfau *et al.*, "Hardware-efficient coherent digital receiver concept with feedforward carrier recovery for M-QAM constellations," IEEE JLT **27**, 989–999 (2009).

[5] J. C. M. Diniz *et al.*, "Low-complexity carrier phase recovery based on principal component analysis for square-QAM modulation formats," Opt. Express **27**, 15,617–15,626 (2019).

[6] E. Börjeson *et al.*, "Towards FPGA emulation of fiber-optic channels for deep-BER evaluation of DSP implementations," in "SPPCom," (2019), p. SpTh1E.4.

[7] E. Börjeson *et al.*, "Cycle-slip rate analysis of blind phase search DSP circuit implementations," in "OFC," (2020), p. M4J.3.

[8] R. R. Müller *et al.*, "Phase-offset estimation for joint-polarization phase-recovery in DP-16-QAM systems," IEEE PTL **22**, 1515–1517 (2010).