Geometric Constellation Shaping for Phase-noise Channels Using a Differentiable Blind Phase Search

Andrej Rode, Benedikt Geiger, and Laurent Schmalen

Communications Engineering Lab (CEL), Karlsruhe Institute of Technology, Kaiserstr. 12, 76131 Karlsruhe, Germany
rode@kit.edu, benedikt.geiger@student.kit.edu

Abstract: We perform geometric constellation shaping with optimized bit labeling using a binary auto-encoder including a differential blind phase search (BPS). Our approach enables full end-to-end training of optical coherent transceivers taking into account the digital signal processing. © 2022 The Author(s)

1. Introduction

Modern high-speed coherent optical communication systems rely on feed-forward carrier phase estimation (CPE), since the high degrees of parallelization and pipelining, required in hardware implementations of coherent receivers, are prohibitive for decision-directed algorithms with feedback. One commonly used state-of-the-art algorithm for CPE is the blind phase search (BPS), which was first introduced in [1]. Classical rectangular quadrature amplitude modulation (QAM) constellations suffer from a penalty in achievable rate and a gap to the capacity, which can be reduced by constellation shaping techniques. In this paper, we consider geometrical constellation optimization to improve achievable rates. Recently, the use of deep-learning in conjunction with auto-encoders has become a popular method to carry out constellation optimization by taking into account the complete transmission system [2], [3], [4]. These methods often use simplified models of the transmission, in particular neglecting the influence of digital signal processing (DSP) algorithms like the BPS, as the latter are often not differentiable, a crucial prerequisite to apply deep learning.

Two approaches to geometrical constellation optimization via deep-learning and auto-encoders for optical systems using a BPS have been presented in [5] and [6]. Both approaches avoid going through the non-differentiable BPS; in [5], the authors employ a cubature Kalman filter to approximate the noise distribution at the output of the BPS algorithm and in [6], the authors model the effect of BPS via training on additive white Gaussian noise (AWGN) and a surrogate residual phase noise (RPN). While both approaches show gains compared to regular QAM, the resulting constellations are still not optimized for the operation of BPS as the end-to-end learning is performed on approximated outputs of the BPS. Additionally, the constellations have been optimized using the mutual information (MI) as target metric. However, most of today's coherent systems employ binary forward error correction (FEC) schemes with a bit-metric decoder (BMD). The performance of constellations optimized with respect to the MI is suboptimal in this context.

In this paper, we propose a novel approach to geometric constellation shaping based on end-to-end deep learning with binary auto-encoders [3]. To circumvent the issue of the non-differentiable BPS, we propose a differentiable version of the BPS algorithm that we use during training. The resulting constellations (including a bit labeling) show superior performance in terms of bitwise mutual information (BMI)[1], in particular in the high signal to noise ratio (SNR) regime. Our investigation paves the way for a fully differentiable DSP chain enabling full end-to-end learning of coherent transceivers.

2. System Model of Binary Auto-encoder

We present an approach for a geometrical shaping (GS) constellation optimization for transmission of an M-ary constellation through a laser phase noise affected optical channel where the BPS algorithm is used for CPE. Our approach is based on an auto-encoder with trainable neural networks (NNs) at the transmitter (Tx) and receiver (Rx). The channel model of the auto-encoder consists of AWGN, laser phase noise, and the BPS algorithm. The BPS algorithm is implemented such that it can be switched between non-differentiable and differentiable mode. With this approach, it can be used in differential mode during training of the NNs and in the original, non-differentiable mode during validation.

The system model of the auto-encoder is shown in Fig. 1. The transmitter and receiver of the auto-encoder (modulation order M) system are implemented following a binary auto-encoder architecture [3], [7] and using PyTorch as software framework. The Tx-NN consists of one fully connected layer, expects one-hot encoded bit vectors $\boldsymbol{b}$ of length m, with $m = \log_2 M$ and performs the bit labeling, i.e. assigns a constellation symbol to each bit sequence. The Rx-NN is implemented as a fully connected feed-forward neural network (FF-NN) with three

---

[1]The BMI is often referred to as generalized mutual information (GMI) in the optical communications community. We prefer to use the term BMI due to its easier resemblence with the operational meaning.
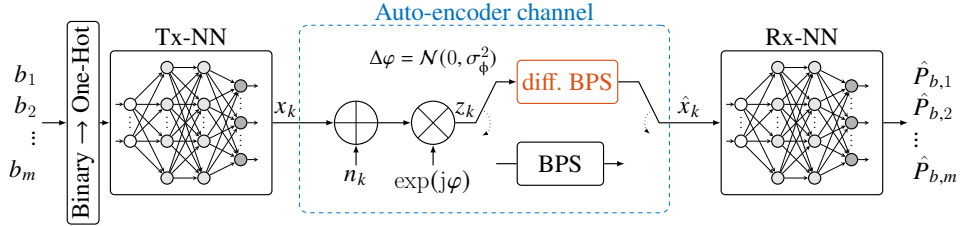
Fig. 1. System model of auto-encoder with differentiable BPS

layers of width 128 and ReLU as activation function. It is used to estimate the probability $\hat{P}_{b,i} = P(b_i = 1)$ that bit $i \in m$ is a 1.

The channel model of the auto-encoder between the Tx and Rx-NN includes AWGN, phase noise and a (differentiable) BPS. The AWGN manifests in a zero-mean complex normal perturbation $n_k \sim \mathcal{CN}(0, \sigma_n^2)$ which is added to the output of the transmitter, where σ_n^2 is the noise variance leading to SNR $= \frac{1}{\sigma_n^2}$. The phase noise, i.e. laser phase noise, is modeled as a Wiener process with variance $\sigma_\varphi^2 = 2\pi(\Delta f \cdot T_S)$, where Δf is the laser linewidth, and T_S the symbol duration which is the inverse of symbol rate $T_S = \frac{1}{R_S}$. The third block of the channel model, the BPS is discussed in more detail, because its understanding is essential for this paper.

3. Differentiable Blind Phase Search Approximation

To start with, we recapitulate the BPS algorithm, as presented in [1]: First, the received complex-valued symbol z_k at time step k is rotated by L test phases $\boldsymbol{\varphi} = (\varphi_0, \ldots, \varphi_{L-1})^T$. Second, the squared error $d_{k,b} = |z_k \exp(\mathrm{j}\varphi_b) - \hat{x}_{k,b}|^2$ between the closest constellation symbol $\hat{x}_{k,b}$ and $z_k \exp(\mathrm{j}\varphi_b)$, the received symbol rotated by the test phase φ_b, is calculated for all L test phases ($b \in \{0, \ldots, L-1\}$). To reduce the impact of noise, a sliding window is used to obtain $s_{k,b} = \sum_{n=-N}^{N} d_{k-n,b}$. Afterwards, the phase is estimated to be $\varphi_{\hat{b}}$, with $\hat{b} = \arg\min_b s_{k,b}$ being the test phase minimizing the sum $s_{k,b}$. Afterwards, a phase unwrap has to be applied to get rid of the phase ambiguities. Finally, the rotated symbol $\hat{x}_{k,\hat{b}}$ can be computed.

In order to determine the correct test phase, the "$\arg\min_b$" operation is applied. Because the $\arg\min_b$ is a non-differentiable operation, the backpropagation algorithm, which is used during deep learning to update the Tx NN, cannot be used in combination with the standard BPS [1], as it requires a differentiable model. We therefore propose to modify the BPS such that it becomes differentiable and can be used during deep learning by the backpropagation algorithm. We use this differentiable BPS during training and switch to the standard BPS during performance evaluation. To obtain a differential BPS, we replace the non-differentiable $\arg\min$ operation by the *softargmin* [8] operation, which is also commonly referred to as *softmin* in the Machine Learning community. The $\mathrm{softmin}(\boldsymbol{x})$ function is applied to a vector $\boldsymbol{x}$ and returns a vector of identical dimension. The ith component of the softmin output is given by

$$\mathrm{softmin}(x_i) := (\mathrm{softmin}(\boldsymbol{x}))_i = \frac{\exp{(-x_i)}}{\sum_j \exp{(-x_j)}} \tag{1}$$

where x_i is the ith element of $\boldsymbol{x}$. To obtain the estimate of the test phase, we compute the the dot product between softmin applied to the vector $\boldsymbol{s}_k := (s_{k,0}, \ldots, s_{k,L-1})$ and the vector of test phases yielding the differentiable phase estimate $\varphi_{\hat{b},\mathrm{diff.}} = \boldsymbol{\varphi}^T \mathrm{softmin}(\boldsymbol{s}_k)$.

This approximation works well as long as the phase changes only moderately between $(-\pi, \pi)$. If a phase slip occurs, intermediate results $\varphi_{\hat{b},\mathrm{diff.}} \approx \frac{1}{2}[-\pi \cdot \mathrm{softmin}(z_k \exp(-\mathrm{j}\pi)) + \pi \cdot \mathrm{softmin}(z_k \exp(\mathrm{j}\pi))] \approx \frac{1}{2}\left(-\frac{1}{2}\pi + \frac{1}{2}\pi\right) \approx 0$ prevent a proper function of the phase unwrap. We hence propose to use a variant of the softmin function, the *softmin with temperature*, which is defined as

$$\mathrm{softmin}_t(x_i) := (\mathrm{softmin}_t(\boldsymbol{x}))_i = \mathrm{softmin}\left(\frac{x_i}{t}\right) \tag{2}$$

For low temperatures ($t \to 0$) the softmin with temperature approximates the $\arg\min$ closely but leads to vanishing gradients. For high temperatures, we get a softer approximation. We use the softmin with temperature during training with a temperature t that decreases steadily from 1 to 0.001 during training so that we slowly approach the behavior of the non-differentiable original BPS.

4. Results and Evaluation

We compare the performance of GS constellations learned with two differentiable approximations of the BPS: First, the reference model, a binary auto-encoder channel model including only AWGN and RPN, which is modeled as a Gaussian distributed phase rotation $\mathcal{N}(0, \sigma_{\mathrm{RPN}}^2)$ and chacterized by the noise variance σ_{RPN}^2, as in [5]. Second, our approach using a differentiable version of the BPS and therefore including a AWGN, full laser phase noise and differentiable BPS in the auto-encoder channel model. Both approaches are validated using the original, non-differentiable BPS. We use the binary cross entropy (BCE) during training as loss function to minimize the bit error rate (BER) and to maximize the BMI [3].
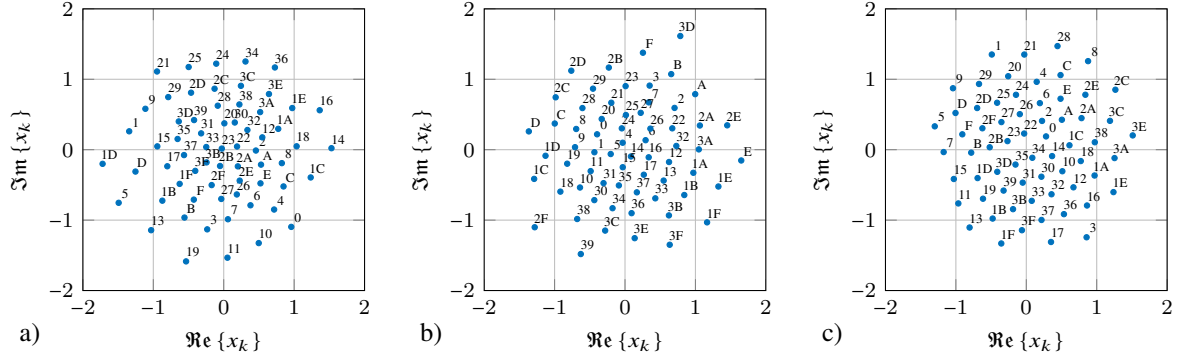
Fig. 2. Geometrically shaped and bit-labeled constellations trained using the differentiable BPS algorithm at a) 17 dB SNR, 100 kHz linewidth b) 17 dB SNR, 600 kHz linewidth c) 20 dB SNR, 300 kHz linewidth

The learned posteriors of the Rx-NN are used to estimate the BMI [5]. The training was performed for $M = 64$ with hyperparameters as defined previously. For the training of both models a SNR of 17 dB has been selected, and for our approach, a fixed linewidth of 100 kHz at 32 Gbaud symbol rate, $L = 60$ test phases in $(-\pi, \pi)$, and a sliding window length of $N = 60$ was chosen, while the reference constellation was trained on the fixed surrogate RPN channel with $\sigma_{\mathrm{RPN}} = 0.005$ as in [6].

The validation between both approaches is performed across three SNR levels and on ten equally spaced linewidths between 50 kHz and 600 kHz, while the other parameters are the same as during training. We simulated 100 validation runs for each setting on both constellations with 10^5 symbols per run. In the plot, we provide the mean together with error bars indicating the standard deviation. The achieved BMI of the
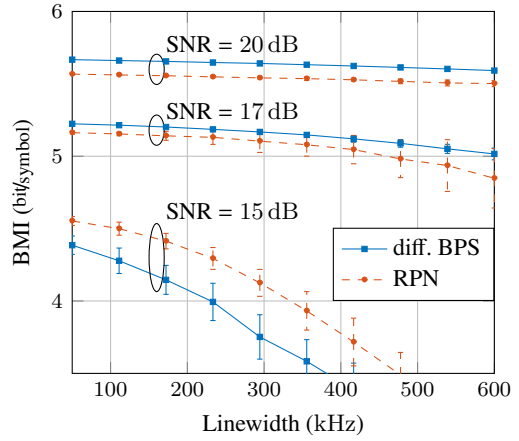

Fig. 3. Validation results with non-differentiable BPS

trained models, training channel is used as label, is shown in Fig. 3. In the low SNR regime, the auto-encoder channel model with AWGN and RPN outperforms our approach. Understanding this behavior is part of our ongoing investigations. At high SNRs, the auto-encoder channel model including the differentiable BPS outperforms the reference auto-encoder model including only AWGN and RPN. Additionally, our approach is more robust (lower variance during validation) which can be observed by comparing the results at an SNR of 17 dB, especially for high laser linewidths. The constellation trained on the differentiable BPS achieves a gain of 0.1 bit/symbol across the full range of linewidths for a SNR of 20 dB. Hence, an accurate modeling of the channel including DSP algorithms like the BPS is important to harness the full potential of GS and increase the achievable rate.

The learned 64-ary constellations with the corresponding bit labels (given as hexadecimal numbers) are shown in Fig. 2. During training, the constellations develop mostly Gray-coded bit labels and a slightly asymmetric shape. This can be observed for example at the lower left of Fig. 2-a) with symbols 5, 13, 19, which have a larger spacing to neighbouring symbols compared to anywhere else in the constellation. Interestingly, this slight asymmetricity improves the BPS performance, by providing constellation points the BPS algorithm can latch-on to.

5. Conclusion

With the proposed differentiable BPS, we obtain geometrically shaped constellations that are well suited for use with binary codes and outperform constellations that have been obtained with surrogate residual phase noise, in particular in the high SNR regime due to the use of a more exact model of the transmission chain. This work shows how part of the receiver can be replaced by its differentiable counter-parts and paves the way for future end-to-end-optimization of complete optical transceivers.

References

1. T. Pfau, S. Hoffmann, and R. Noe, "Hardware-Efficient Coherent Digital Receiver Concept With Feedforward Carrier Recovery for M-QAM Constellations," *J. Lightwave Technol.* **27**, 989–999 (2009).
2. B. Karanov, *et al.*, "End-to-end deep learning of optical fiber communications," *J. Lightwave Technol.*, **36**(20), pp. 4843–4855 (2018).
3. S. Cammerer *et al.*, "Trainable Communication Systems: Concepts and Prototype," *IEEE Trans. Commun.* **68**, 5489–5503 (2020).
4. K. Gümüs *et al.*, "End-to-end learning of geometrical shaping maximizing generalized mutual information," *Proc. OFC* (2020).
5. O. Jovanovic, M. P. Yankov, F. Da Ros, and D. Zibar, "Gradient-free training of autoencoders for non-differentiable communication channels," *J. Lightwave Technol.*, **39**(20), pp. 6381–6391 (2021).
6. O. Jovanovic *et al.*, "End-to-end learning of a constellation shape robust to variations in SNR and laser linewidth," *Proc. ECOC* (2021).
7. R. Jones, M. P. Yankov, and D. Zibar, "End-to-end learning for GMI optimized geometric constellation shape", *Proc. ECOC* (2019).
8. E. Jang, S. Gu, and B. Poole, "Categorical Reparameterization with Gumbel-Softmax," *Proc. ICLR* (2017).