Optimization of Performance of BICM via Design of Mapping and Demapping for 32APSK and 64APSK over Linear and Nonlinear Channels using Bitwise Mutual Information

Neal Becker*
Hughes Network Systems, Germantown MD
Email: *neal.becker@hughes.com

***Abstract*—Modern digital communication systems are often based on Bit-Interleaved Coded Modulation (BICM) architectures. On the transmit side, following the FEC encoder is a mapping of channel bits to modulated complex symbols. On the receive side the received complex symbol values are demapped to bit log-likelihood (LLR) values. As has been shown in previous research, the mapping operation (equivalently, constellation design) can be optimized jointly with the demapping operation on a linear channel to increase capacity. Here, we extend these efforts to increase capacity on non-linear channels - specifically for high-power amplifiers (HPA) typically employed on satellite transponders characterized by AM/AM and AM/PM response. We further consider joint, end-to-end optimization of mapper together with predistortion on the transmit side and also with non-APP demapper optimized for nonlinear channel operation on the receive side.**

I. Introduction

In modern digital communication systems using BICM performance in noise is optimized by selection of the constellation points and bit labeling. The latter refers to the choice of mapping of the M symbol values to the selected constellation points. In [1] joint optimization of mapping (constellation design) and demapping (conversion of received complex symbol values to LLR) was described. In this work training on the bit-wise mutual information (BMI) is described, which achieves joint optimization of constellation shaping and labeling together with demapping to bit log-likelihood ratios (LLR) for input to an FEC decoder. Sionna [2] includes an example called “Autoencoder”[1] which demonstrates use of a single neural-net (NN) dense layer for the transmit mapper and a dense NN as the demapper, using Binary Cross Entropy (BCE) as the loss function. Here minimization of BCE is used as this is a convenient metric commonly available in Machine Learning (ML) libraries and this is equivalent to maximizing BMI. For training over a number of epochs, for each epoch a random batch of transmit bits is generated, mapped to channel symbols, passed through a channel model, and demapped to Log Likelihood Ratio (LLR) values. The loss is computed as the BCE between the transmitted bits and the demapped LLR values and the ML tools of backpropagation are used to optimize the mapping and demapping.

A different approach was used in [3] for optimization of pragmatic capacity of 32APSK constellations. In this work optimization was performed using a classical optimization technique of Simulated Annealing. Results were presented both for linear and for “soft limiter” nonlinear channel models.

In this work we extend the previous results using ML techniques which were applied to AWGN channel to consider the nonlinear channels often employed in satellite communications by adding a nonlinear High Power Amplifier (HPA) characterized by AM/AM and AM/PM distortion. The cascade of transmit pulse shaping filter, memoryless nonlinearity, and receive matched filter introduces intersymbol interference (ISI). This ISI can be mitigated either by predistortion on the transmit side or by compensation techniques on the receive side.

We will consider a variety of alternative transmit and receive architectures designed for the nonlinear channel. In each case we will apply end-to-end optimization to jointly optimize the mapper (constellation design) with the chosen receiver (demapper) and compare them based on their BER performance. As baseline approaches we first examine unoptimized DVBS2x[7] constellation together with a conventional APP receiver designed for linear channel, and then with bivariate demapper described in [4]. Next we again consider the bivariate demapper but this time with the mapper jointly optimized with it. Then we examine the joint optimization of mapper with a convolutional neural net (CNN) demapper which can mitigate ISI by computing LLR based on windows of multiple symbols. Finally we consider joint optimization using iterative symbol-level predistortion as described in [5]. Here we jointly optimize with two different receiver architectures: the bivariate demapper, and a dense neural net demapper.

[1] https://github.com/NVlabs/sionna/blob/main/examples/Autoencoder.ipynb

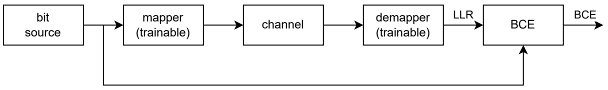


Figure 1: Training Setup Linear Channel

To demonstrate the performance gains available we show BER performance results for 32APSK and 64APSK constellations.

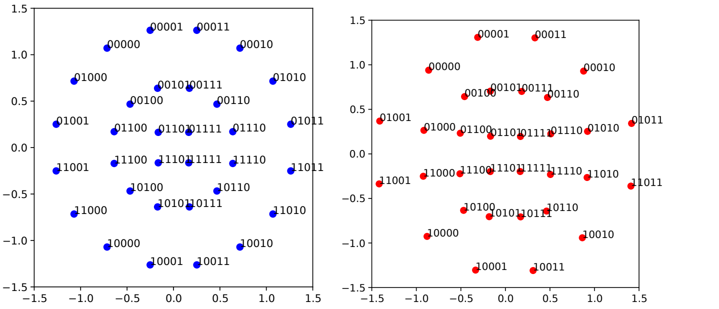
II. Optimization of 32APSK for Linear Channel

Figure 1 illustrates the training setup for the linear channel, where in this case the channel model is AWGN. The best way to understand this learning setup is as a form of self-supervised regression learning, where BCE between the bit source and LLR is optimized using the standard ML techniques of algorithmic differentiation and backpropagation. Neural Networks are commonly trained by starting with randomly distributed weights. As illustrated in "Autoencoder" example in [2], by representing the mapper as a vector of Tensorflow[6] "Variables" we can actually initialize the mapping to a known good mapping that can be subsequently optimized by training rather than a random mapping. This insight is very helpful because otherwise for large constellation sizes the probability of the optimization becoming stuck in a local optimal point becomes large. In the following we will use the DVBS2x [7] constellations as initial starting points for optimization as they are the current state-of-the-art.

Figure 2 illustrates an example of optimization applied to 32APSK from DVBS2x. The original constellation is shown in Fig. 2a while the optimized constellation is shown in Fig. 2b where the "orig" curve is for the DVBS2x constellation and "opt11-12" is the constellation optimized over the range of $E_s/N_0$11–12 dB. Figure 3 shows a comparison of the original constellation points (blue Xs) with the optimized points (red dots) showing how the points moved during optimization. The improvement in pragmatic capacity is demonstrated in Figure 4, where capacity is found using the method described in [8], which shows an improvement of about 0.25 dB at 11.4 dB E_s/N_0 and is confirmed by BLER simulation shown in Figure 5. In this and following examples 3gpp NR rate $R = 2/3$ code is used for coded performance simulations, choosing the largest blocksize $N = 8400$.[2]

III. Optimization of 32APSK for Nonlinear Channel

As an example nonlinear channel we will use the conventional TWTA model [9, Fig. 10] characterized by AM/AM and AM/PM shown in Figure 6. The training setup is shown in Figure 7. Compared with the linear training introduced in [2] Autoencoder example, a sample



(a) Original DVBS2x (b) Optimized

Figure 2: 32APSK Linear

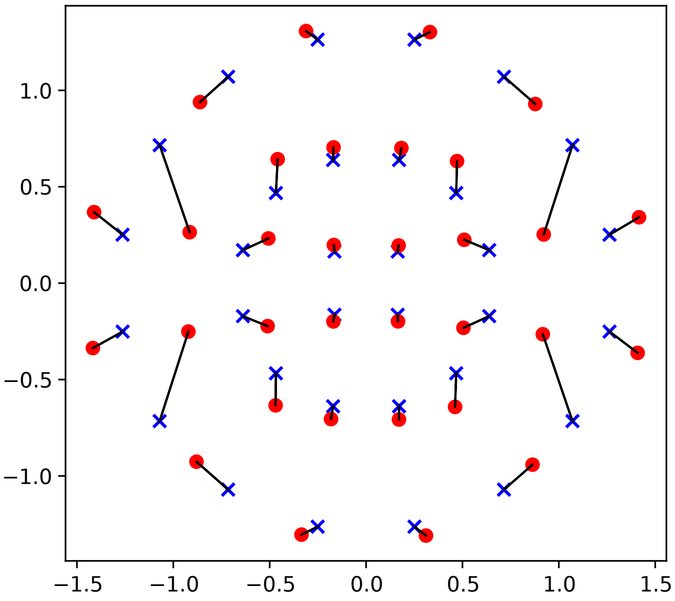


Figure 3: Comparison of 32APSK Original DVBS2x Constellation vs. Optimized for Linear Channel

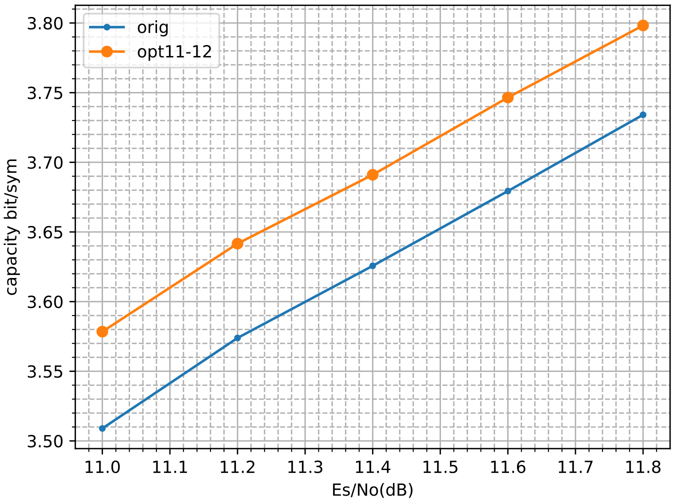


Figure 4: 32APSK 2/3 BICM Linear Capacity Comparison

[2] 3gpp NR code is used in these examples because it is implemented in Sionna allowing very fast simulation runtimes by utilizing GPU.

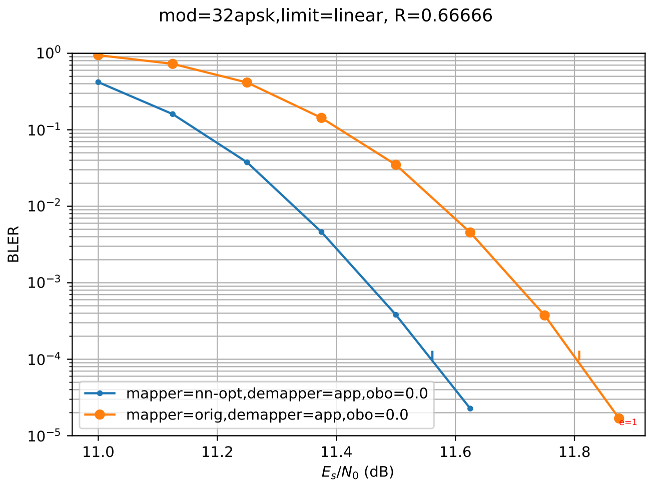


Figure 5: 32APSK Linear BLER Comparison

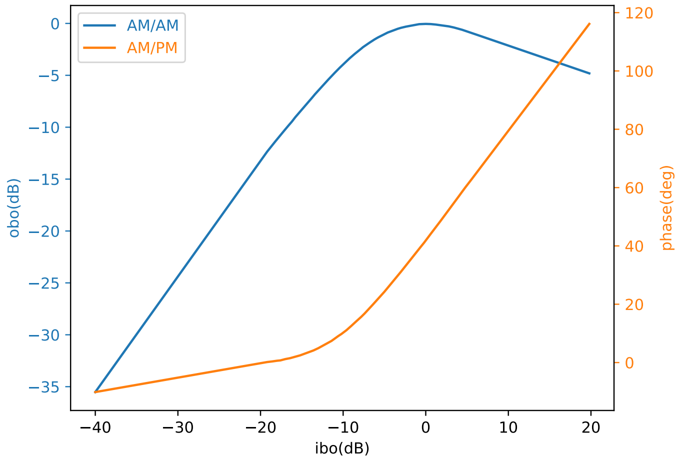


Figure 6: DVB Conventional TWTA AM/AM, AM/PM

interpolator, memoryless nonlinearity (HPA), and sample decimator are added, where the sample interpolator includes a Root-Raised Cosine (RRC) pulse-shaping filter as does the sample decimator, because properly representing the HPA effect requires increasing the sample rate to multiple samples/symbol.

A variety of transmit and receive approaches can be considered for the nonlinear channel. The original DVBS2x constellation mapper at the transmitter with a receiver that implements an ideal AGC function and LLR produced by APP demapper with respect to the original constellation is used as a baseline. This represents an ideal transmitter and receiver that ignores the nonlinear channel effect (other than finding the closest linear mapping of the received constellation to the transmit constellation). To explain the receiver implementing ideal AGC, consider a system model $y = hx + n$, with x, y, h, n being transmit signal, receive signal, channel response, and noise respectively:

$$\begin{aligned} y \cdot x &= h|x|^2 \\ \hat{h} &= \frac{y \cdot x}{|x|^2} \\ \hat{x} &= \frac{y}{\hat{h}} \end{aligned}$$

where we have omitted noise terms for clarity and $\hat{x}$ is the ideal AGC output which would then be applied to the APP demapper input. This receiver applies a single complex-value scaling to the received signal at the input to the APP demapper. This case is labeled “mapper=orig, demapper=app” in subsequent results.

A second option also uses the original DVBS2x transmit mapper (constellation) but uses the bivariate Gaussian technique [4] to compute the LLR at the receiver. Here the receive PDFs for each constellation point are approximated as 2D Gaussian. This receiver is able to compensate for memoryless distortion, but not ISI. This case is identified here as “mapper=orig, demapper=bivariate” in Figure 8.

Another approach is to use a bivariate demapper on the receive side, but optimize the mapper with the bivariate demapper, using the bivariate demapper during the training of the mapper. This approach is labeled here as “mapper=bivariate-opt, demapper=bivariate” and shows about 0.5 dB additional gain compared with “mapper=orig, demapper=bivariate”, demonstrating the improvement obtained by training the combined mapper and demapper end-to-end.

As mentioned earlier, the cascade of pulse shaping filter, nonlinearity, and receive matched filter introduces ISI. One way to mitigate this ISI on the receive side is to use a demapper based on a Convolutional Neural Net (CNN) which provides a way to include the memory effect in the optimization of the demapper. In this case the mapper and demapper are jointly optimized and are labeled here as “mapper=nn-opt, demapper=cnn,d=x,w=y,N=z” where x, y, z indicate the depth (number of layers), width (kernel width of CNN), and size (#CNN kernels) of the CNN used. The example “mapper=nn-opt, demapper=cnn,d=4,w=3,N=256” shows about 3 dB gain compared with the baseline “mapper=orig, demapper=app”.

It is possible to mitigate the ISI at the transmit side instead of the receive side. One approach to achieve this is the iterative predistortion technique described in [5]. Along with this transmit various receive demappers can be considered of which one possibility is using the bivariate demapper. Here we have jointly optimized the end-to-end system including predistortion and bivariate demapper, and

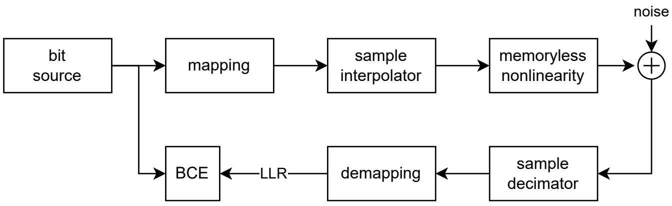


Figure 7: Training Setup for Nonlinear Channel

this case is labeled as “pred+mapper=bivariate-pred-opt, demapper=bivariate”.

A final case is also based on use of the iterative predistortion [5] but this time using a NN demapper. Since ISI has been mitigated at the transmitter, there is no need for a CNN at the receiver and instead a dense NN is used for the demapper. This case is labeled as “pred+mapper=nn-pred-opt, demapper=dense,d=x,N=z” where d, N designate the depth and size of the dense NN used.

Summary curves are shown in Fig. 8. These curves show the E_s/N_0 required to achieve 10^{-4} BLER target as a function of Output Backoff (OBO) for the different cases studied and provide a convenient guide to visualize the optimal OBO setting for each candidate system[3]. The minimum E_s/N_0 for each curve shows the optimal OBO for operation in each case[4] and the resulting E_s/N_0. It can be observed that the 3 techniques which mitigate ISI: using CNN at the receive side “mapper=nn-opt, demapper=cnn,d=4,w=3,N=256”, as well as the 2 examples of iterative predistortion at the transmit side, one using the bivariate receiver “mapper=nn-opt, demapper=cnn,d=4,w=3,N=256” and one using the dense NN receiver “pred+mapper=nn-pred-opt, demapper=dense,d=4,N=256” all show similar results and display about 3 dB gain in noise performance while running at smaller OBO than the baseline case.

One trade-off that must be considered is that increases to the transmit OBO, while improving noise performance when combined with the mitigations considered here, will increase the out-of-band spectral regrowth in the HPA. To illustrate Fig. 9 shows the spectra of three cases: the “baseline” case at -3 dB OBO, “nn-opt” at -2 dB OBO, and “mapper=bivariate-opt” at -2 dB OBO which are the respective optimal OBO settings for those cases from Fig. 8 not including the use of predistortion[5]. Also the case of predistortion with optimized mapping and dense NN demapping “pred+mapper=nn-pred-opt” operating at -2 dB OBO is included, which represents an optimal OBO for this case in terms of noise performance. Note: Here we use the convention that greater output power levels correspond to positive OBO. These results indicate that spectral regrowth is mostly determined by the OBO, regardless of case considered. We also see that while the considered optimizations allow significant improvements in noise performance achieving optimal noise performance at smaller OBO, there is an increase in Out-Of-Band (OOB) spectral regrowth as a result.

IV. Optimization of 64APSK for Linear Channel

The same techniques applied in Sec. II to optimization of 32APSK can also be applied to 64APSK. In this example we will consider 64APSK with 3gpp NR code rate 3/4. Fig. 10 compares the original DVBS2x 64APSK constellation (Fig. 10a) with the optimized constellation (Fig. 10b). Figure 11 shows a comparison of the original DVBS2x constellation with the constellation optimized for a linear channel, showing the migration of the points. Note that some of the points appear to swap positions, indicating an optimization to the bit labeling. BICM capacities of these constellations are compared in Fig. 12, which shows an improvement of about 0.2 dB.

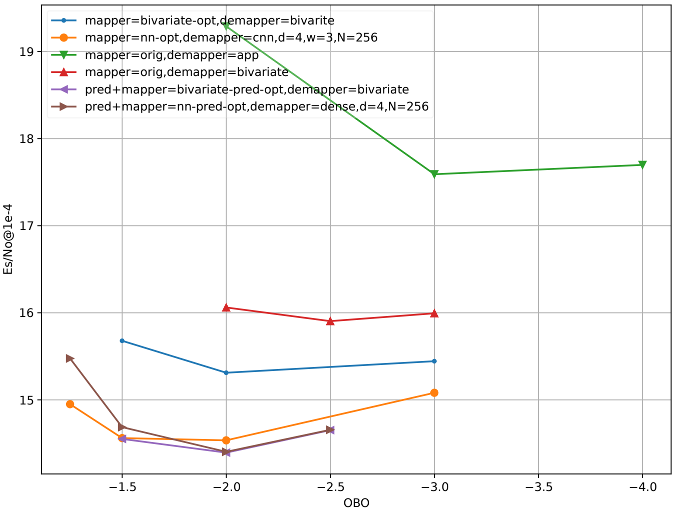


Figure 8: 32APSK Noise Performance vs. OBO Summary Curves

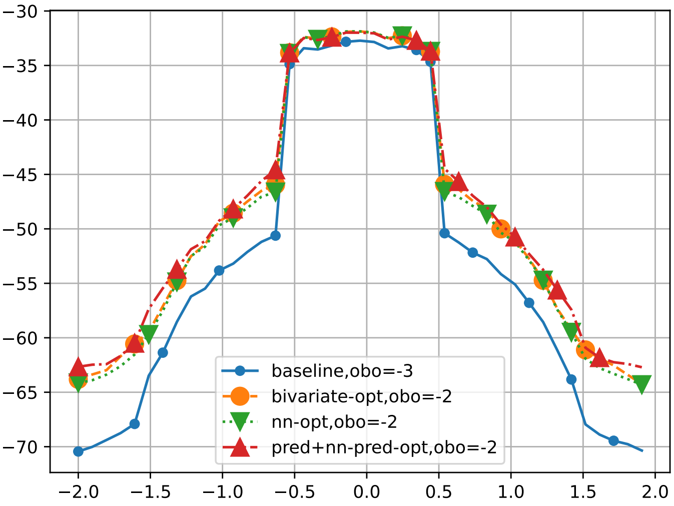


Figure 9: Comparison of Spectra

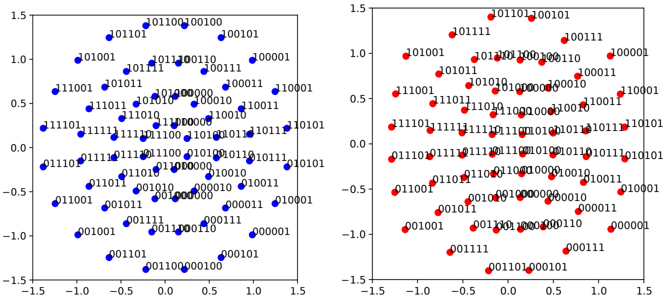
V. Optimization of 64APSK for Nonlinear Channel

Similar to optimization of 32APSK for nonlinear channel (Section III) we consider optimization of 64APSK for the same DVB conventional TWTA model. A summary of results is shown in Figure 13. 64APSK is almost unusable on this channel without nonlinear compensation, as shown by the single point at 10 dB OBO

[3] In these curves the loss in noise performance due to decreased output power is included.

[4] Optimal in requiring minimum E_s/N_0, but there may be other system considerations.

[5] The demapper algorithm for these cases does not affect the transmit spectrum.



(a) Original DVBS2x

(b) Optimized

Figure 10: 64APSK Linear

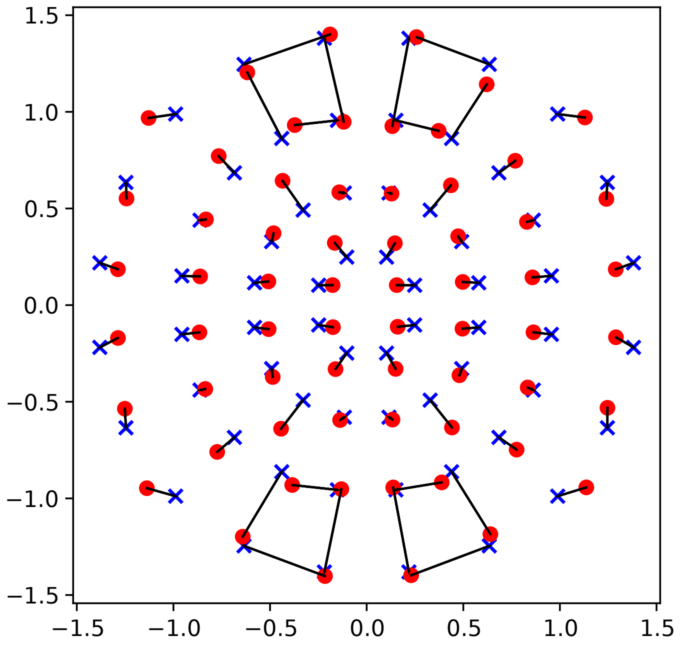


Figure 11: Comparison of 64APSK DVBS2x Constellation vs. Optimized for Linear Channel

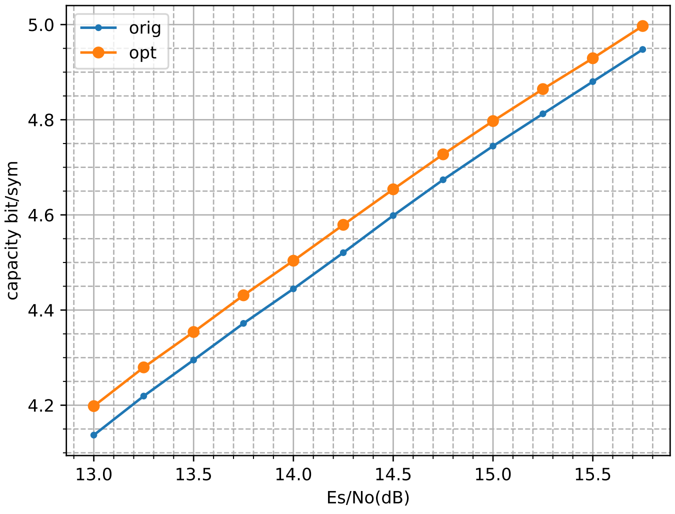


Figure 12: 64APSK 3/4 BICM Linear Capacities

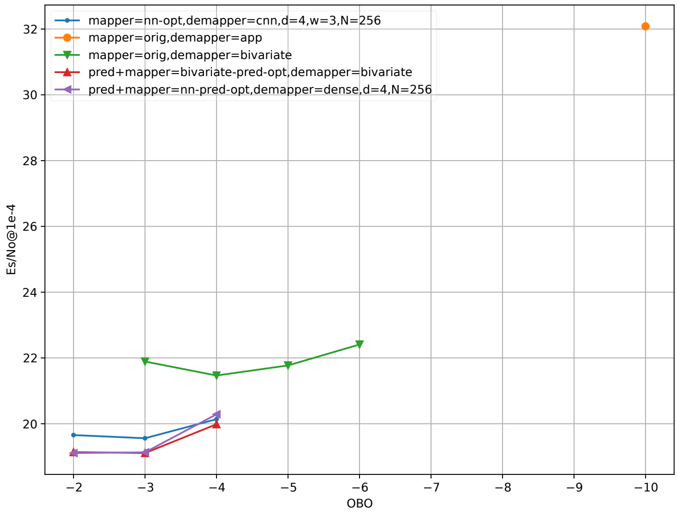


Figure 13: 64APSK Nonlinear Channel Noise Performance vs. OBO Summary Curves

(mapper=orig,demapper=app)[6]. Using the same original DVBS2x transmit constellation with a bivariate demapper provides a large improvement (mapper=orig,demapper=bivariate). An optimized mapper coupled with a CNN demapper offers further improvement and ability to run with higher power (OBO) (mapper=nn-opt,demapper=cnn,d=4,w=3,N=256). Adding iterative predistortion shows a small additional improvement. Iterative distortion is considered both with bivariate demapper (pred+mapper=bivariate-pred-opt,demapper=bivariate) and with a dense NN demapper (pred+mapper=nn-pred-opt,demapper=dense,d=4,N=256) both showing similar performance. One might hypothesize that the NN learned the bivariate demapping, although we have not attempted to confirm this.

VI. Complexity

We have seen the performance advantages of the iterative predistortion on the transmit side, and both bivariate and CNN on the receiver side but implementation complexity is also an important consideration.

The iterative predistortion technique passes the transmit symbol sequence through a model of the transmit pulse shaping filter, nonlinearity, and receive filter. For a transmit and receive filters of symbol span and samples/sym $N_{\text{sym}}, N_{\text{sps}}$ the number of multiplications/sym is $N_{\text{sym}} N_{\text{sps}}$. These multiplications are with fixed coefficients which can often reduce hardware complexity. The nonlinear model can be composed of a $|\cdot|$ or $|\cdot|^2$ block, a table lookup and a single complex multiply per sample. The iterative predistorion would have I iterations, where I could be 5.

The bivariate receiver can be considered to have two parts. There is a “training” which comprises the estimation

[6] With smaller OBO the BLER curves flare badly.

of the 2nd order statistics of each of the received constellation points. This training could be done in non-real time and so will not be considered the dominant component of complexity. For convenience repeating here the bivariate conditional PDF for the symbol s_k (from which bitwise LLR is derived [10]) is

$$\begin{aligned}\log P(y|s_k) = \\ \frac{-1}{2(1-\rho_k^2)} \Bigg\{ \frac{\left(\Re(y) - \mu_{k,\mathrm{real}}\right)^2}{\sigma^2_{k,\mathrm{real}}} + \frac{\left(\Im(y) - \mu_{k,\mathrm{imag}}\right)^2}{\sigma^2_{k,\mathrm{imag}}} \\ - 2\rho_k \left(\frac{\Re(y) - \mu_{k,\mathrm{real}}}{\sigma_{k,\mathrm{real}}}\right)\left(\frac{\Im(y) - \mu_{k,\mathrm{imag}}}{\sigma_{k,\mathrm{imag}}}\right) \\ - \log\left(2\pi\sqrt{1-\rho_k^2}\sigma_{k,\mathrm{real}}\sigma_{k,\mathrm{imag}}\right)\Bigg\}\end{aligned}$$

Assuming 2nd order statistics for each constellation point have been pre-computed, a small number of multiplications and divisions are needed for each hypothesized transmit symbol s_k per received symbol.[7]

The CNN demapper can mitigate ISI and is an alternative to predistortion on the transmit side. In the example CNN in the results we have $d = 4, w = 3, N = 256$. There are d CNN layers, with each kernel having a width of w and there are N such kernels. Therefore the number of multiplications is dwN.

VII. Online Learning

So far we have only considered cases where we have a known model for the nonlinear amplifier, given by it's AM/AM and AM/PM characteristics; along with the assumption that the operating point (drive level) is perfectly known. There are applications in which these elements are not known accurately, in some cases because they may vary over time, temperature, etc. There is therefore a need for transmit and receive systems that are able to adapt to these unknown conditions.

The bivariate demapper lends itself naturally to adaptation, as it depends only on the second order statistics $p(y|x_i)$ for each of the points in the transmit constellation x_i. This is straightforward to estimate from known training sequences. Also as noted this training need not be performed in real-time and could be updated slowly, as the nonlinear characteristics should only vary slowly over time, temperature, etc. A possible architecture based on this could use a fixed mapper (optimized based on assumed nonlinear model) together with an adaptive bivariate receiver.

Sionna [2] includes an example in Autoencoder called "Trainable End-to-end System: RL-based Training" which demonstrates joint optimization of the mapper and demapper for the case where the channel is non-differentiable, as in the case where the nonlinearity is not accurately known. This technique can probably be applied to the case of an imperfectly known nonlinearity and so can probably be used with online learning, although this idea has not been pursued here.

VIII. Observations and Conclusions

The simulated cases show there is considerable room for performance optimization compared to the baseline cases. In Sec. II it was shown that the existing SOA DVBS2x 32APSK constellation can be modified to obtain increased capacity on the linear channel through the optimization of BCE. Sec. IV shows similarly optimization of DVBS2x 64APSK constellation on linear channel also achieving increased capacity.

On the nonlinear channel characterized by the example TWTA from DVB a variety of techniques are demonstrated to optimize mapping and demapping. Even when considering the use of an advanced receiver demapping technique (bivariate), substantial performance improvements are demonstrated when jointly optimizing the mapping together with demapping. We can observe that ISI is effectively reduced either by using iterative predistortion at the transmitter (to control ISI) together with bivariate demapper at the receiver (which does not address ISI), or using a memoryless mapper at the transmitter (which does not address ISI) together with a CNN for the demapper (which does handle ISI), with both approaches producing similar results.

End-to-End optimization of capacity including use of the simple iterative predistortion technique [5] together with mapping design and demapping design was investigated. The end-to-end capacity optimization using other predistorter designs is an interesting possibility left for future research.

IX. Acknowledgments

The author would like to thank R. I. Seshadri for his help in producing this work.

References

[1] S. Cammerer, F. A. Aoudia, S. Dörner, M. Stark, J. Hoydis, and S. ten Brink, "Trainable communication systems: Concepts and prototype," *IEEE Trans. Commun.*, vol. 68, pp. 5489–5503, Sept 2020.

[2] J. Hoydis, S. Cammerer, F. Ait Aoudia, A. Vem, N. Binder, G. Marcus, and A. Keller, "Sionna: An open-source library for next-generation physical layer research," *arXiv preprint*, Mar. 2022.

[3] F. Kayhan and G. Montorsi, "Optimal constellations for the pragmatic receiver in the DVB-S2 standard," in *31st AIAA International Communications Satellite Systems Conference*, Oct. 2013.

[4] B. Beidas and R. I. Seshadri, "Forward error correction decoder input computation in multi-carrier communications system." US Patent 9203680, Dec 2015. Patent granted on December 1, 2015.

[5] B. Beidas, S. Kay, and N. Becker, "System and method for combined predistortion and interference cancellation in a satellite communications system." US Patent 8,355,462, Jan 2013. Patent granted on Jan 15, 2015.

[7] The final log term may be omitted without large performance loss.

[6] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[7] DVB, "Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications part 2: DVB-S2 extensions (DVB-S2X)," Feb. 2020. DVB Document A083-2.

[8] R. I. Seshadri, *Capacity-based parameter optimization of bandwidth constrained CPM.* West Virginia University, 2007.

[9] DVB, "Implementation guidelines for the second generation system for broadcasting, interactive services, news gathering and other broadband satellite applications; part 2: S2 extensions (DVB-S2X)," Feb. 2020. DVB Document A171-2.

[10] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 927–946, 1998.

[11] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, pp. 563–575, Dec 2017.

[12] Y. Qian and F. Liu, "Neural network predistortion technique for nonlinear power amplifiers with memory," in *2006 First International Conference on Communications and Networking in China*, pp. 1–5, 2006.