

OptiCommPy: Open-source Simulation of Fiber Optic Communications with Python

Edson Porto da Silva^{1*} and Adolfo Fernandes Herbster^{1*}

¹ Electrical Engineering Department, Federal University of Campina Grande (UFCG), Brazil * These authors contributed equally.

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: ↗

Submitted: 15 February 2024

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

OptiCommPy is an open-source Python package designed for simulating fiber optical communication systems and subsystems. OptiCommPy is freely accessible, providing researchers, students, and engineers with the option to simulate various fiber optical communication systems at the physical layer. Additionally, the toolbox incorporates numerous digital signal processing (DSP) algorithms, particularly essential for coherent optical systems.

Statement of need

Optical fiber communication dominates the transmission of high-speed data traffic, owing to various physical, engineering, and economic factors. Worldwide efforts are continuously being made to research and develop optical communication technologies that can support both current and future Internet infrastructure. The expansion of optical networks necessitates a swift transition from scientific breakthroughs in research labs to telecommunications industry products and solutions. Furthermore, the ever-increasing demand for bandwidth and connectivity places constant pressure on the development of faster and more efficient optical fiber communications (Winzer & Neilson, 2017).

Today, optical communication systems engineering is a multidisciplinary field encompassing various areas of science and technology, including laser science, photonic devices, fiber optics modeling and engineering, digital signal processing, and communications theory. As we approach the limits of information transmission through optical fibers, more sophisticated engineering is required for the construction of optical transmitters and receivers, involving advanced digital signal processing (DSP) (Essiambre et al., 2010), (Savory, 2010). The emergence of high-speed application-specific integrated circuits (ASICs) and advanced DSP algorithms has propelled coherent optical transmission systems to the forefront of high-capacity transmission via optical fibers (Sun et al., 2020).

Whether in the research or development stages, the study of optical communication systems typically necessitates the use of robust computational models to simulate various aspects of the system. For instance, it may be essential to comprehend how information-carrying signals transmitted over fibers will be affected by propagation phenomena such as chromatic dispersion (CD), polarization mode dispersion (PMD), nonlinear effects, and noise (Agrawal, 2002). This information ultimately determines the performance metrics of the transmission system, which play a crucial role in selecting the most suitable technology to become an industrial standard.

Presently, a variety of optical communication simulation toolboxes are accessible. While the majority of these are proprietary software packages (Optiwave, 2023), (VPIphotonics, 2023), (Synopsys, 2023), a few are open-source but are designed to operate within proprietary software environments such as Matlab® (dtu-dsp, 2015), (Paolo Serena, 2021).

41 The module structure of the OptiCommPy package is illustrated in Fig. 1. At the top level,
42 the package is named `optic`, containing five sub-packages: `comm`, `models`, `dsp`, `utils`, and
43 `plot`.

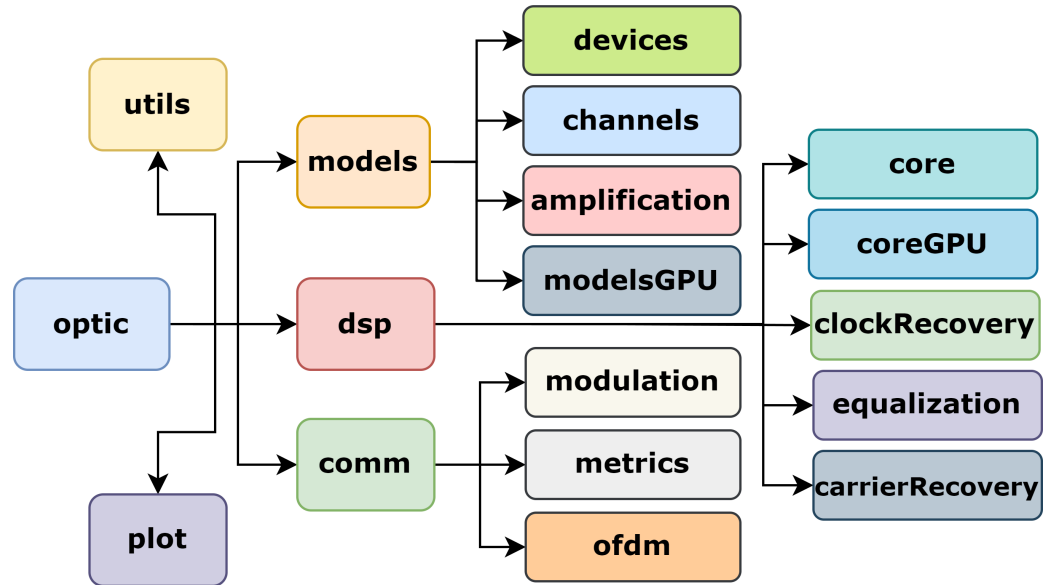


Figure 1: Structure of modules of the OptiCommPy package.

44 The `comm` sub-package comprises three modules designed for implementing various digital
45 modulation and demodulation schemes (Proakis, 2001), including pulse amplitude modulation
46 (PAM), quadrature amplitude modulation (QAM), phase-shift keying (PSK), and on-off keying
47 (OOK). Evaluating the performance of these diverse digital communication schemes is made
48 possible through different metrics, such as bit-error-rate (BER), symbol-error-rate (SER), error
49 vector magnitude (EVM), mutual information (MI), and generalized mutual information (GMI)
50 (Alvarado et al., 2018), all available within the `comm.metrics` module.

51 The `models` sub-package contains the majority of the mathematical/physical models used to
52 build OptiCommPy simulations. Within the `models.devices` module, one can access models
53 for a range of optical devices, encompassing optical Mach-Zehnder modulators, photodiodes,
54 optical hybrids, optical coherent receivers, and more. These functions serve as fundamental
55 building blocks for constructing simulations of optical transmitters and receivers. In the
56 `models.channels` module, a collection of mathematical models for the fiber optic channel is
57 provided, spanning from basic additive white Gaussian noise (AWGN) and linear propagation
58 models to more sophisticated non-linear propagation models rooted in variants of the split-
59 step Fourier method (SSFM) (Agrawal, 2002). In particular, it includes an implementation
60 of the Manakov model to simulate nonlinear transmission over a fiber optic channel with
61 polarization-multiplexing (Marcuse et al., 1997). Certain computationally intensive models,
62 such as the Manakov SSFM, have a CuPy-based version (Okuta et al., 2017) accessible via
63 `models.modelsGPU`, designed specifically for execution with CUDA GPU acceleration (NVIDIA
64 et al., 2020).

65 The `dsp` sub-package is a collection of DSP algorithms ranging from basic signal processing
66 operations, such as linear finite impulse response filtering, to more advanced algorithms,
67 e.g. blind adaptive equalization. Currently, the `dsp` sub-package contains three special-
68 ized modules: `dsp.clockRecovery` provides algorithms for clock and timing synchronization;
69 `dsp.equalization` contains implementations of common digital equalization algorithms used
70 in optical communications; `dsp.carrierRecovery` provides algorithms for carrier frequency
71 and phase recovery. These sub-packages covers all the basic DSP functionalities required in

most of the modern coherent optical transceivers.

Finally, the `utils` and the `plot` sub-packages provide functions that implement a few general utilities and custom plotting functions to visualize signals (eyediagram plots, constellation plots, etc).

Several types of analysis can be conducted to characterize transmission performance across various system parameters. For instance, one can generate performance curves that depict BER and Q-factor as functions of received optical power at varying transmission distances, as illustrated in Fig.~3.

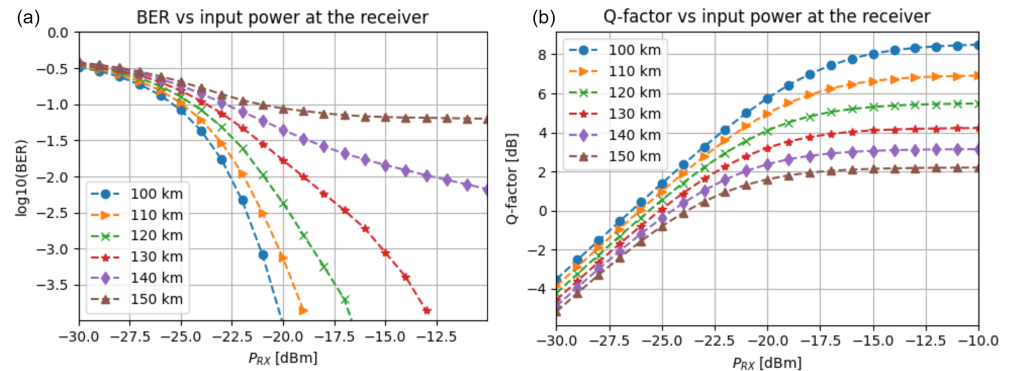


Figure 2: Performance metrics for different transmission distances and received optical powers, characterizing the increasing penalty from chromatic dispersion with the distance in a 10 Gb/s OOK transmission system. (a) BER vs received optical power for different transmission distances; (b) Q-factor vs received optical power for different transmission distances.

A collection of examples of utilization of OptiCommPy to build several different simulation setups, including advanced setups with non-linear fiber propagation models, WDM transmission, and coherent detection can be found in <https://github.com/edsonportosilva/OptiCommPy/tree/main/examples>.

Acknowledgements

This work was supported by the National Council for Scientific and Technological Development (CNPq), Brazil, grant 406684/2021-9.

References

- Agrawal, G. P. (2002). *Fiber-optic communication systems* (3rd ed.). John Wiley & Sons.
- Alvarado, A., Fehenberger, T., Chen, B., & Willems, F. M. J. (2018). Achievable Information Rates for Fiber Optics: Applications and Computations. *J. Light. Technol.*, 36(2), 424–439. <https://doi.org/10.1109/JLT.2017.2786351>
- dtu-dsp. (2015). *Robochameleon: A matlab coding framework and component library for optical communication systems* (Version 0.1). <https://github.com/dtu-dsp/Robochameleon>
- Essiambre, R. J., Foschini, G. J., Winzer, P. J., Kramer, G., & Goebel, B. (2010). Capacity Limits of Optical Fiber Networks. *J. Light. Technol.*, 28(4), 662–701. <https://doi.org/10.1109/JLT.2009.2039464>
- Marcuse, D., Menyuk, C. R., & Wai, P. K. A. (1997). Application of the Manakov-PMD equation to studies of signal propagation in optical fibers with randomly varying birefringence.

- 99 *J. Light. Technol.*, 15(9), 1735–1745. <https://doi.org/10.1109/50.622902>
- 100 NVIDIA, Vingelmann, P., & Fitzek, F. H. P. (2020). *CUDA, release: 10.2.89*. <https://developer.nvidia.com/cuda-toolkit>
- 101
- 102 Okuta, R., Unno, Y., Nishino, D., Hido, S., & Loomis, C. (2017). CuPy: A NumPy-compatible library for NVIDIA GPU calculations. *Proceedings of Workshop on Machine Learning Systems (LearningSys) in the Thirty-First Annual Conference on Neural Information Processing Systems (NIPS)*.
- 103
- 104
- 105
- 106 Optiwave. (2023). *OptiSystem*. <https://optiwave.com/products/optisystem/>
- 107 Paolo Serena. (2021). *Optilux, the optical simulatr toolbox*. <https://optilux.sourceforge.io/>
- 108 Proakis, J. G. (2001). *Digital communications* (4th ed.). McGraw Hill.
- 109 Savory, S. J. (2010). Digital coherent optical receivers: Algorithms and subsystems. *IEEE J. Sel. Top. Quantum Electron.*, 16(5), 1164–1179. <https://doi.org/10.1109/JSTQE.2010.2044751>
- 110
- 111
- 112 Sun, H., Torbatian, M., Karimi, M., Maher, R., Thomson, S., Tehrani, M., Gao, Y., Kumpera, A., Soliman, G., Kakkar, A., Osman, M., El-Sahn, Z. A., Doggart, C., Hou, W., Sutarwala, S., Wu, Y., Chitgarha, M. R., Lal, V., Tsai, H.-S., ... Kandappan, P. (2020). 800G DSP ASIC design using probabilistic shaping and digital sub-carrier multiplexing. *Journal of Lightwave Technology*, 38(17), 4744–4756. <https://doi.org/10.1109/JLT.2020.2996188>
- 113
- 114
- 115
- 116
- 117 Synopsys. (2023). *Synopsys OptSim for optical communication*. <https://www.synopsys.com/photonic-solutions/optsim.html>
- 118
- 119 VPIphotonics. (2023). *VPItransmissionMaker™ optical systems*. <https://www.vpiphotonics.com/Tools/OpticalSystems/>
- 120
- 121 Winzer, P. J., & Neilson, D. T. (2017). From scaling disparities to integrated parallelism: A decathlon for a decade. *Journal of Lightwave Technology*, 35(5), 1099–1115. <https://doi.org/10.1109/JLT.2017.2662082>
- 122
- 123