

# Desenvolvimento de uma Interface Gráfica para Simulador de Meios Ópticos Guiados

Isaac G. Y. Rabello, Marco Aurélio Jucá e Felipe da Rocha Henriques

**Resumo**—As fibras ópticas revolucionaram a forma como as tecnologias se desenvolvem, tendo em vista seus vastos usos e benefícios. Com sua popularidade em diversos setores da indústria em todo mundo, a pesquisa no desenvolvimento e aprimoramento de fibras ópticas tem sido amplamente incentivada. Este trabalho apresenta o processo de desenvolvimento de um *software* livre e multiplataforma codificado em linguagem *Python* para a simulação em meios ópticos guiados. O objetivo é fomentar a pesquisa nesse meio, promovendo uma ferramenta alternativa às disponíveis no mercado, que geralmente são dispendiosas e incompatíveis com diversos sistemas operacionais modernos.

**Palavras-Chave**—Fibras ópticas, Modelos computacionais, *Software* livre e multiplataforma, *Python*.

**Abstract**—Optical fibers revolutionized the way technologies are developed, due to their vast usages and benefits. With their popularity in several industry sectors in the whole world, research in developing and enhancing optical fibers has been widely encouraged. This work presents the process of developing a free, cross-platform software coded in Python for the simulation of guided optical media. The objective is to promote research in this area by providing an alternative tool to those commercially available, which are often expensive and incompatible with several modern operative systems.

**Keywords**—Optical fibers, Computational models, Cross-platform free software, Python.

## I. INTRODUÇÃO

O protagonismo das fibras ópticas nos sistemas de comunicação modernos é evidente. As numerosas vantagens das fibras como meios guiados, por exemplo a alta capacidade e a baixa perda, viabilizaram seu uso em diversas aplicações, como em sistemas de transmissão de dados e em aplicações de instrumentação [1].

Uma tecnologia muito versátil que pode ser aplicada em fibras ópticas é a inserção de grades de difração no seu interior. As grades podem ser de dois tipos: (i) as grades de Bragg (FBG) apresentam uma perturbação periódica no índice de refração do núcleo ao longo do comprimento da fibra, com um período curto, ao contrário das (ii) grades de período longo (LPG). As grades podem ser aplicadas tanto em comunicação, no tratamento de sinais multiplexados em comprimento de onda [2], na equalização de ganhos de amplificadores em fibra dopada com érbio (EDFA) [3]; quanto em sensoriamento, já

que suas características espectrais apresentam sensibilidade a fatores externos como a temperatura, o índice de refração e a tensão na fibra [4], [5].

A simulação é uma ferramenta importante no planejamento de sistemas de comunicações, bem como no de sistemas de sensoriamento. As opções de simuladores comercialmente disponíveis não são numerosas, especialmente se o foco for instrumentação. Esses programas, em geral, são pagos, de código fechado, e disponibilizados para apenas uma plataforma. Normalmente não é possível, ou não é prático, incluir e controlar efeitos de ruído, aplicar personalizações nas propriedades das grades ou verificar os traços de reflectometria (OTDR) em configurações envolvendo grades. O simulador que está sendo desenvolvido pela nossa equipe visa a minimizar esses desafios. O principal objetivo deste trabalho é apresentar a interface gráfica do simulador e suas principais funções.

## II. O SIMULADOR

O pacote *Python* chamado `GratingSim` está sendo desenvolvido para prover ferramentas que ajudem um desenvolvedor a implementar/simular meios ópticos guiados na construção de outros *softwares*. Um pacote *Python* é um conjunto de módulos, que por sua vez são um conjunto de arquivos contendo classes, funções e variáveis. É uma forma inteligente de distribuir um conjunto de arquivos entre desenvolvedores como uma API (*Application Programming Interface*), em que é possível importar a funcionalidade de um código abstraído o funcionamento interno. Atualmente o pacote contém apenas um módulo, chamado `gs`, que implementa todas as funcionalidades da biblioteca. Além disso, possui um conjunto de arquivos de demonstração para fins de teste.

No estágio atual do desenvolvimento, a biblioteca é capaz de simular sistemas ópticos compostos por fibras, LPGs, emendas, conectores, acopladores, LEDs, OTDRs e fontes de luz branca.

## III. PACOTE GRÁFICO DO SIMULADOR

Como forma de abranger o usuário final, não necessariamente programador *Python*, a equipe decidiu desenvolver um pacote gráfico. Desta maneira, qualquer pesquisador da área de óptica poderia usufruir das simulações e modelos computacionais do pacote `GratingSim` usando uma Interface Gráfica de Usuário (GUI).

Esta interface consome a API provida pelo pacote `GratingSim` e permite que o usuário interaja com ele pela manipulação de componentes gráficos como botões e caixas de texto. Além disso, é possível obter resultados visuais como

Isaac Rabello, Coordenação de Telecomunicações, CEFET/RJ, Petrópolis - RJ, Engenharia Elétrica, USP, e-mail: isaac.rabello@usp.br; Marco Aurélio Jucá, Coordenação de Telecomunicações, CEFET/RJ, Petrópolis - RJ, Programa de Pós-Graduação em Engenharia Elétrica, UFJF, Juiz de Fora - MG, e-mail: marco.junior@cefet-rj.br; Felipe Henriques, Programa de Pós-graduação em Instrumentação e Óptica Aplicada (PPGIO), Programa de Pós-graduação em Ciência da Computação (PPCIC), Coordenação de Telecomunicações, CEFET/RJ, Petrópolis - RJ, e-mail: felipe.henriques@cefet-rj.br.

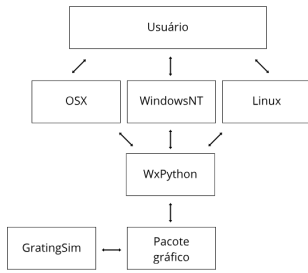


Fig. 1. Diagrama de funcionamento do pacote gráfico.

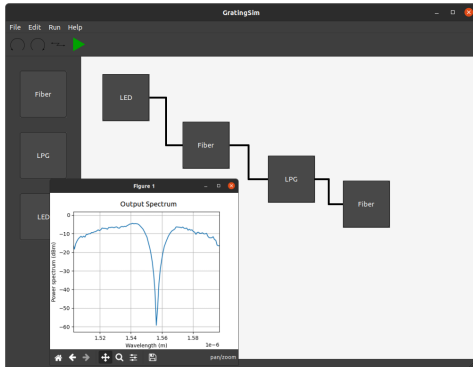


Fig. 2. Tela principal da aplicação com simulação simples.

imagens, gráficos ou qualquer coisa capaz de ser representada na tela.

Para facilitar o desenvolvimento de uma interface gráfica, pode-se usar um *framework* GUI, que abstrai a criação de componentes gráficos (*widgets*) e a interação com o usuário (por meio de mouse e teclado, por exemplo) como uma API.

Dentre as diversas opções de *frameworks* GUI implementados em *Python* a equipe trouxe a debate *WxPython*, *Tkinter* e *PyQT* por serem os mais usados, robustos e com uma boa documentação [6].

Por fim, a opção escolhida foi o *WxPython*, por ser de código aberto, compatível com sistemas operacionais das famílias WindowsNT, OSX e Linux, e contar com uma grande variedade de *widgets* nativos ao Sistema Operacional em uso.

A Fig. 1 mostra o esquema de interações entre o usuário, o sistema operacional e os demais componentes presentes no simulador.

Tendo em mente o conforto do usuário final, a equipe dispôs os elementos da interface de forma similar à de outros softwares de uso comum.

A tela principal da aplicação é bem simples e tem sua maior parte composta por uma área de trabalho na qual o usuário pode organizar os componentes para as simulações da forma que desejar num esquema “arrasta e solta”. Há também uma lista de todos os dispositivos disponíveis (fontes de luz, grades de difração, etc) e uma barra de ferramentas que apresenta diversas maneiras de interagir com os dispositivos dispostos na área de trabalho. A Fig. 2 mostra a tela principal da interface do simulador com um sistema simples sendo simulado. Na janela menor, há uma simulação do espectro em questão, que foi confirmada usando outros simuladores.

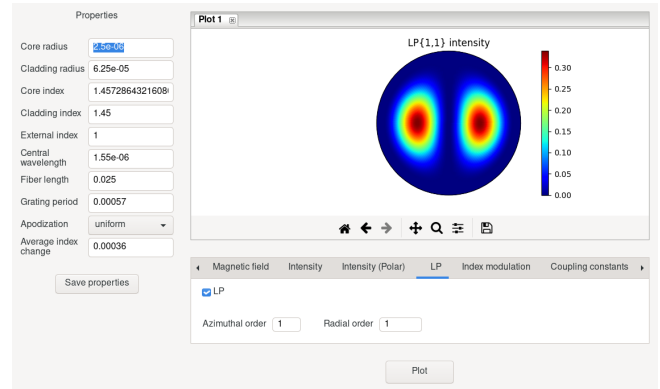


Fig. 3. Tela de propriedades de um dispositivo em um ambiente GTK.

Quando um componente qualquer é clicado, uma nova tela se abre onde é possível editar propriedades e plotar gráficos relacionados ao dispositivo em questão, como pode ser visto na Fig. 3.

#### IV. CONCLUSÕES

O projeto apresentado aqui tem grande potencial de auxiliar a comunidade de pesquisa em óptica. A meta é oferecer um pacote em *Python* para simulações em meios ópticos guiados contendo uma vasta gama de dispositivos e funções, e uma GUI multiplataforma para uma interação mais intuitiva com o usuário final. Pretende-se publicar um projeto completo que possa ser usado em situações reais.

O desenvolvimento está nas fases iniciais, apesar de já ser utilizável para alguns casos. A próxima etapa é continuar o desenvolvimento e escrever uma documentação que explique como usar a biblioteca para que outros desenvolvedores possam usá-la. Além disso, a criação de um *website* que promova o projeto é uma possibilidade para ajudar o crescimento da comunidade.

Com o crescimento da popularidade, pretende-se acolher uma comunidade de usuários que, por se tratar de um *software* livre, coopere com o projeto sugerindo melhorias, correções e novas funcionalidades, de forma a construir um *software* cada vez melhor e mais completo.

#### AGRADECIMENTOS

Os autores agradecem ao CEFET/RJ e à UFJF pelo apoio recebido durante a realização deste trabalho.

#### REFERÊNCIAS

- [1] G. Keiser, *Fiber Optic Communications*. Singapore: Springer, 2021.
- [2] K. O. Hill and G. Meltz, “Fiber Bragg grating technology fundamentals and overview,” *J. Lightwave Technol.*, vol. 15, no. 8, pp. 1263–1276, Aug. 1997.
- [3] A. Vengsarkar, N. Bergano, C. Davidson, J. Pedrazzani, J. Judkins, and P. Lemaire, “Long-period fiber-grating-based gain equalizers,” *Opt. Lett.*, vol. 21, no. 5, pp. 336–338, Mar. 1996.
- [4] A. Othonos, “Bragg gratings in optical fibers: Fundamentals and applications,” in *Optical Fiber Sensor Technology: Advanced Applications - Bragg Gratings and Distributed Sensors*, K. T. V. Grattan and B. T. Meggitt, Eds. New York, NY: Springer, 2000, ch. 2, pp. 79–187.
- [5] S. W. James and R. P. Tatam, “Optical fibre long-period grating sensors: characteristics and application,” *Meas. Sci. Technol.*, vol. 14, no. 5, pp. R49–R61, Mar. 2003.
- [6] J. Hunt, *Advanced guide to Python 3 programming*. Cham, Switzerland: Springer, 2019.