POLITECNICO DI TORINO
Repository ISTITUZIONALE

A Python Tool to Control and Virtualize Laser Diodes Characterization Benches

Original
A Python Tool to Control and Virtualize Laser Diodes Characterization Benches / Hrynkevych, O., Bardella, P.. - ELETTRONICO. - Th3A.9:(2021). (2021 Education and Training in Optics and Photonics Conference, ETOP 2021 Washington, DC United States 8-10 settembre 2021) [10.1364/ETOP.2021.Th3A.9].

Availability:
This version is available at: 11583/2952734 since: 2022-01-24T18:52:31Z

Publisher:
Optica Publishing Group

Published
DOI:10.1364/ETOP.2021.Th3A.9

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
Optica Publishing Group (formely OSA) postprint versione editoriale con OAPA (OA Publishing Agreement)

© 2021 Optica Publishing Group. Users may use, reuse, and build upon the article, or use the article for text or data mining, so long as such uses are for non-commercial purposes and appropriate attribution is maintained. All other rights are reserved.

(Article begins on next page)

11 October 2026

Education and Training in Optics & Photonics Conference 2021 © OSA 2021

A Python Tool to Control and Virtualize Laser Diodes Characterization Benches

Oscar Hrynkevych, Paolo Bardella
Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, Torino, Italy
paolo.bardella@polito.it

Abstract: We present a Python tool for the control of the laboratory instruments (thermal electric control, power meter, optical spectrum analyzer) used for the characterization of semiconductor laser sources in the framework of master degree courses available at Politecnico di Torino. This software can be used to control real benches, acquiring and elaborating the measurement data, but it can also work offline emulating the behavior of instruments and laser sources, thus allowing the students to work in a completely virtualized environment. © 2021 The Author(s)

1. Introduction

Together with more theoretical courses and activities oriented to the design of optical components, laboratory activities focused on the experimental characterization of photonic devices play an important role in the education of students attending a master degree with specialization in optics.

At the Electronics and Telecommunications Department of Politecnico di Torino, an educational laboratory has been set-up in the last years, allowing the students to characterize semiconductor laser diodes and directly investigate fundamental concepts such as laser threshold, laser efficiency, and optical spectra.

A typical bench is equipped with an ILX Lightwave 3744B Laser Diode Controller (LDC) for the thermal and electrical tuning of the laser diode, and Agilent 8163A Lightwave multimeter with Agilent 81623 photodiode, and an HP 70950B optical spectrum analyzer. All these instruments are connected via GPIB to a PC. The optical sources characterized by the students are generally DFB laser.

These instruments were originally controlled manually (i.e. directly accessing their front panels) or with a collection of MATLAB® scripts, taking advantage of the Instrument Control Toolbox for the low level communication with the hardware. Acquired data was generally post-processed by the students in MATLAB.

We report here on the creation of a systematic program, called Optical Bench Management (OBM), written in Python [1], able to control all the connected devices from a single Graphic User Interface (GUI), greatly simplifying the acquisition, visualization and analysis of the data. Moreover, this program can also operate in offline mode, realistically emulating the behavior of the instruments and reproducing the expected laser source characteristics. It represents therefore an excellent teaching support, since it allows to show to the class real-time experiments even in the absence of the physical hardware, allowing the students at the same time to practice with the measurements directly from their home, which represents a significant advantage in this period of COVID-19 pandemic.

The choice between the two operation modes (real or virtualized devices) is simply performed by the user at the program start-up; no specific configuration or operation is needed.

2. Software implementation

The source code is internally organized in three main sections, addressing: the communication with the instruments, the realistic simulation of the laser source and the emulation of the instruments behaviour, and finally the GUI.

Concerning the communication with the instruments, great attention has been payed to the development of base abstract classes allowing an easy addition of new instruments connected via GPIB, USB or TCP; each individual instrument is implemented in a class providing the necessary methods and including the communication commands expected by the device. The communication with real GPIB instruments is carried out with methods from the *pyvisa* package.

In emulation mode, the interaction between the instruments has been reproduced accurately. Several threads allow the simultaneous emulation of each instrument on different TCP ports in *localhost*, not only for what concerns the communication, but also including acquisition delays and set-up times. An interrupt-based protocol is used for data retrieval between the instruments. The LDC together with the OSA and the power meter have open communication channels both with the laser and with the GUI.

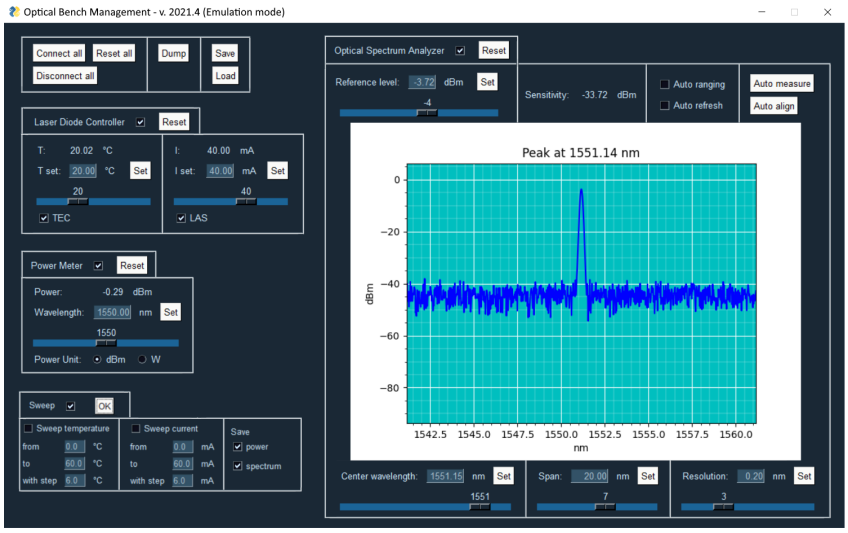

Fig. 1. Main program interface, with the panels for the LDC, the power meter, and the sweep analysis (left) ad the OSA (right).

The emulated laser periodically receives the target current and temperature from the LDC and sends to the instruments its current internal state (actual temperature, emitted power, peak wavelength). These quantities are calculated using phenomenological polynomial models fitted on data obtained *a priori* from measurements of real lasers. The coefficients of the models are saved in encrypted files distributed to the students: in this way we can differentiate the results of the various groups home-works simply distributing to each group a different set of laser parameters.

Finally, the implemented GUI relies on the *Tk* widget toolkit for the interaction with the user and on *matplotlib* for the plots. The GUI is modular and can be easily adapted to include additional instruments. It is possible to control individually the main features of each instrument, performing single acquisitions or executing batch measurements with a thermal or electrical sweep. When necessary, the acquired data is elaborated using the standard *numpy* package. The current set-up and all the acquired data can be stored in text files for subsequent elaborations and discussions. Attention has been paid to the accessibility and usability of the GUI which was designed for inclusivity [2]. The main window of OBM is shown in Fig. 1.

The choice of the Python language was driven by many important aspects, in particular: the intrinsic simplicity of the language, the availability of libraries for the control of the hardware at a consolidated stage of development, and the possibility to easily distribute a compiled version of the program for the various operating systems the students could be using.

3. Conclusions and future works

We developed a tool for the control of instruments typically used for the characterization of laser diodes, with the possibility to completely emulate a measurement bench with realistic results.

Future works will include the possibility to customize the GUI hiding or displaying elements according to the specific course, e.g. simplifying its layout for basic courses, and increasing the control possibilities offered to the students in more advanced classes.

Acknowledgements

The Authors thank Gianfranco Albis, manager of the students laboratories (LED) of the Electronics and Telecommunications Department for his help and support.

References

1. G. Van Rossum, F. L. Drake Jr, *Python tutorial*, Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands (1995)
2. P. M. Langdon, *Inclusive Designing: Joining Usability, Accessibility, and Inclusion*, Springer International Publishing (2014)