PyMoDAQ-Femto Documentation

Release 0.2.0

Weber Sebastien

Jul 26, 2022

CONTENTS:

1 Information **3**

2 Contribution **5**

3 They use it **7**

4 Citation **9**

5 Changelog **11**

6 Index **13**

6.1 Installation . . . 13
6.1.1 Preamble . . . 13
6.1.2 Setting up a new environment . . . 13
6.1.3 Installing PyMoDAQ-Femto . . . 14
6.1.4 Creating shortcuts on **Windows** . . . 14
6.2 Who use it? . . . 14
6.2.1 Institutions using PyMoDAQ . . . 14
6.2.2 What they think of PyMoDAQ? . . . 17
6.3 Contributors . . . 17
6.3.1 Main modules . . . 17
6.3.1.1 Simulator . . . 17
6.3.1.2 Retriever . . . 17
6.3.1.3 Propagator . . . 17
6.3.1.4 Cleaning . . . 18
6.3.2 Documentation . . . 18
6.3.3 You want to contribute? . . . 18

PyMoDAQ-Femto is a 2-in-1 python application dealing with femtosecond laser pulse characterization. It features:

- A user interface called **Simulator** to simulate non-linear traces obtained using most known characterization techniques (FROG, D-Scan, ... with their various non-linear flavour)
- a user interface called **Retriever** to run various retrieval algorithms on simulated or experimental traces (acquired for instance using PyMoDAQ or other means)

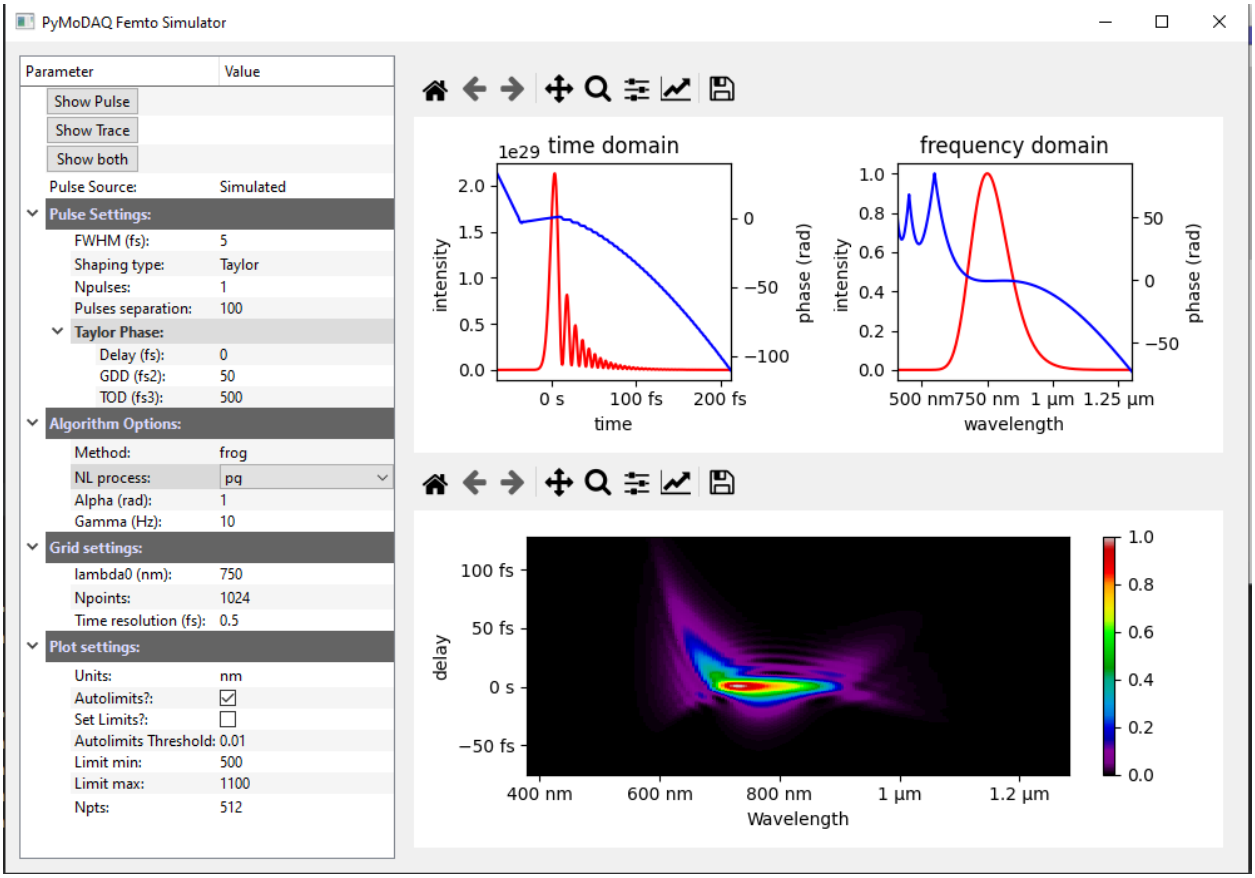

Fig. 1: PyMoDAQ-Femto's Simulator.

Both modules can be run as stand-alone application or plugged as an extension to **`PyMoDAQ`__**. All together it produces a framework for complete temporal characterization of shaped ultrashort femtosecond pulses.

CHAPTER

ONE

INFORMATION

GitHub repo: https://github.com/CEMES-CNRS/pymodaq_femto

Documentation: http://pymodaq_femto.cnrs.fr/

Based on PyMoDAQ and the `pypret`__ library the `pyqtgraph` library.

PyMoDAQ-Femto is written by Sébastien Weber: sebastien.weber@cemes.fr and Romain Géneaux: romain.geneaux@cea.fr under a MIT license.

CHAPTER

TWO

CONTRIBUTION

If you want to contribute see this page: *Contributors*

CHAPTER

THREE

THEY USE IT

See *Who use it?*

CHAPTER

FOUR

CITATION

By using PyMoDAQ-Femto, you are being asked to cite the article published in Review of Scientific Instruments **`RSI 92, 045104 (2021)`__** when publishing results obtained with the help of its interface. In that way, you're also helping in its promotion and amelioration.

CHAPTER

FIVE

CHANGELOG

Please see the changelog.

CHAPTER

SIX

INDEX

6.1 Installation

- *Preamble*
- *Setting up a new environment*
- *Installing PyMoDAQ-Femto*
- *Creating shortcuts on **Windows***

6.1.1 Preamble

PyMoDAQ-Femto is written in Python and uses Python 3.7+. It uses the Qt5 library and the excellent pyqtgraph package for its user interface. For PyMoDAQ to run smoothly, you need a Python distribution to be installed. Here are some advices.

On all platforms **Windows**, **MacOS** or **Linux**, Anaconda or Miniconda is the advised distribution/package manager. Environments can be created to deal with different version of packages and isolate the code from other programs. Anaconda comes with a full set of installed scientific python packages while *Miniconda* is a very light package manager.

6.1.2 Setting up a new environment

- Download and install Miniconda3.
- Open a console, and cd to the location of the *condabin* folder, for instance: `C:\Miniconda3\condabin`
- Create a new environment: `conda create -n my_env python=3.7`, where my_env is your new environment name, could be *pymodaq16* if you plan to install PyMoDAQ version 1.6.0 for instance.. This will create the environment with python version 3.7 that is currently the recommended one.
- Activate your environment so that only packages installed within this environment will be *seen* by Python: `conda activate my_env`
- Install, using conda manager, some mandatory packages: `conda install pyqt`

6.1.3 Installing PyMoDAQ-Femto

Easiest part: in your newly created and activated environment enter: `pip install pymodaq_femto`. This will install the latest PyMoDAQ-Femto available version and all its dependencies. For a specific version enter: `pip install pymodaq_femto==x.y.z`.

6.1.4 Creating shortcuts on Windows

Python packages can easily be started from the command line (see section_how_to_start). However, Windows users will probably prefer using shortcuts on the desktop. Here is how to do it (Thanks to Christophe Halgand for the procedure):

- First create a shortcut (see Fig. 6.1) on your desktop (pointing to any file or program, it doesn't matter)
- Right click on it and open its properties (see Fig. 6.2)
- On the *Start in* field ("Démarrer dans" in french and in the figure), enter the path to the condabin folder of your miniconda or anaconda distribution, for instance: `C:\Miniconda3\condabin`
- On the *Target* field, ("Cible" in french and in the figure), enter this string: `C:\Windows\System32\cmd.exe /k conda activate my_env & python -m pymodaq_femto.retriever`. This means that your shortcut will open the windows's command line, then execute your environment activation (*conda activate my_env* bit), then finally execute and start **Python**, opening the correct pymodaq_femto file (here *retriever.py*, starting the Retriever module, *python -m pymodaq_femto.retriever* bit)
- You're done!
- Do it again for each PyMoDAQ-Femto's module you want (to get the correct python file and it's path, see run_module).

6.2 Who use it?

- PyMoDAQ is used as the core acquisition program of several experiments at CEMES/CNRS and the main interface of its HC-IUMI Ultrafast Electron Microscope
- The attolab platform at CEA Saclay started using it in 2019

6.2.1 Institutions using PyMoDAQ

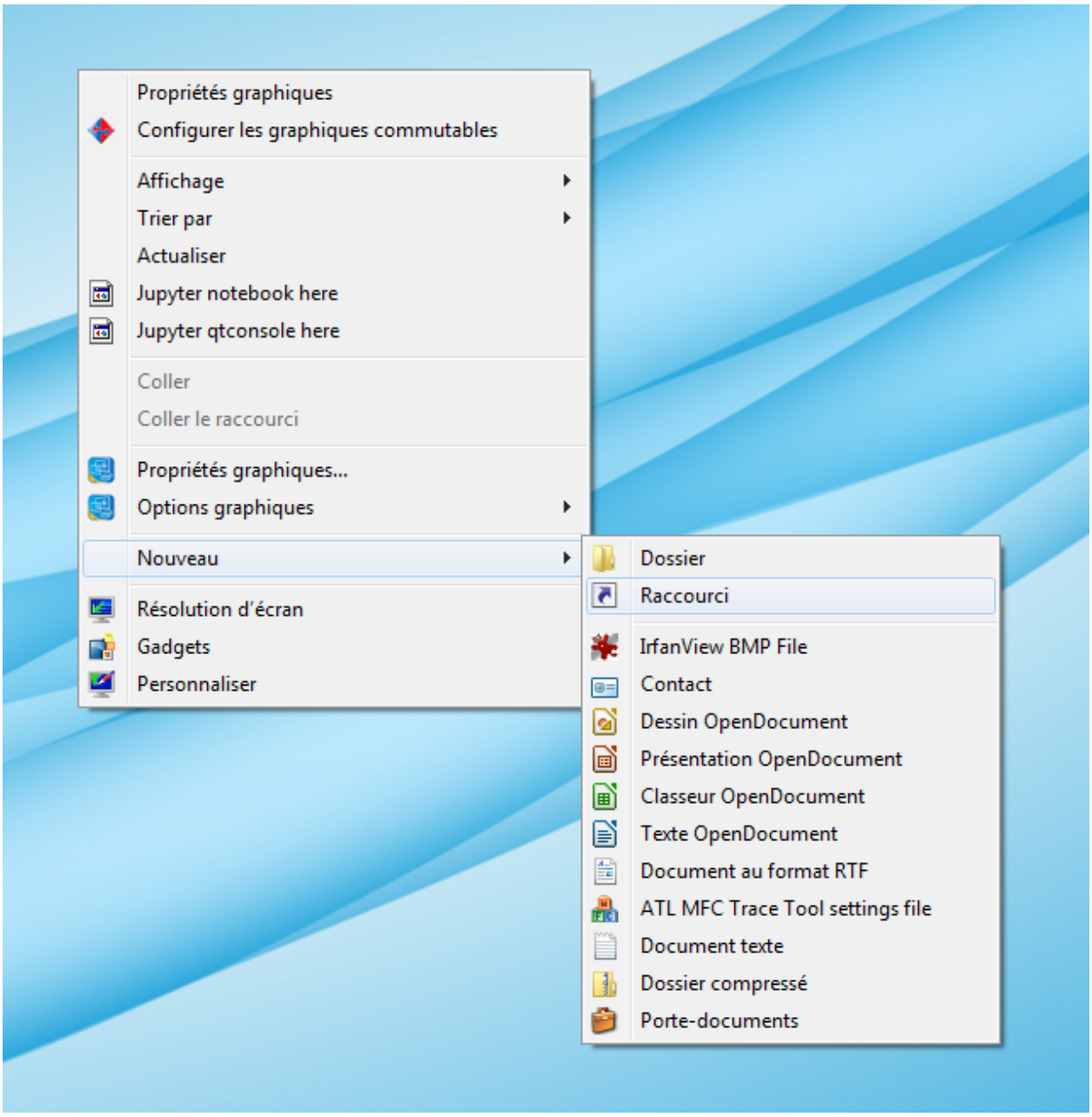

Fig. 6.1: Create a shortcut on your desktop

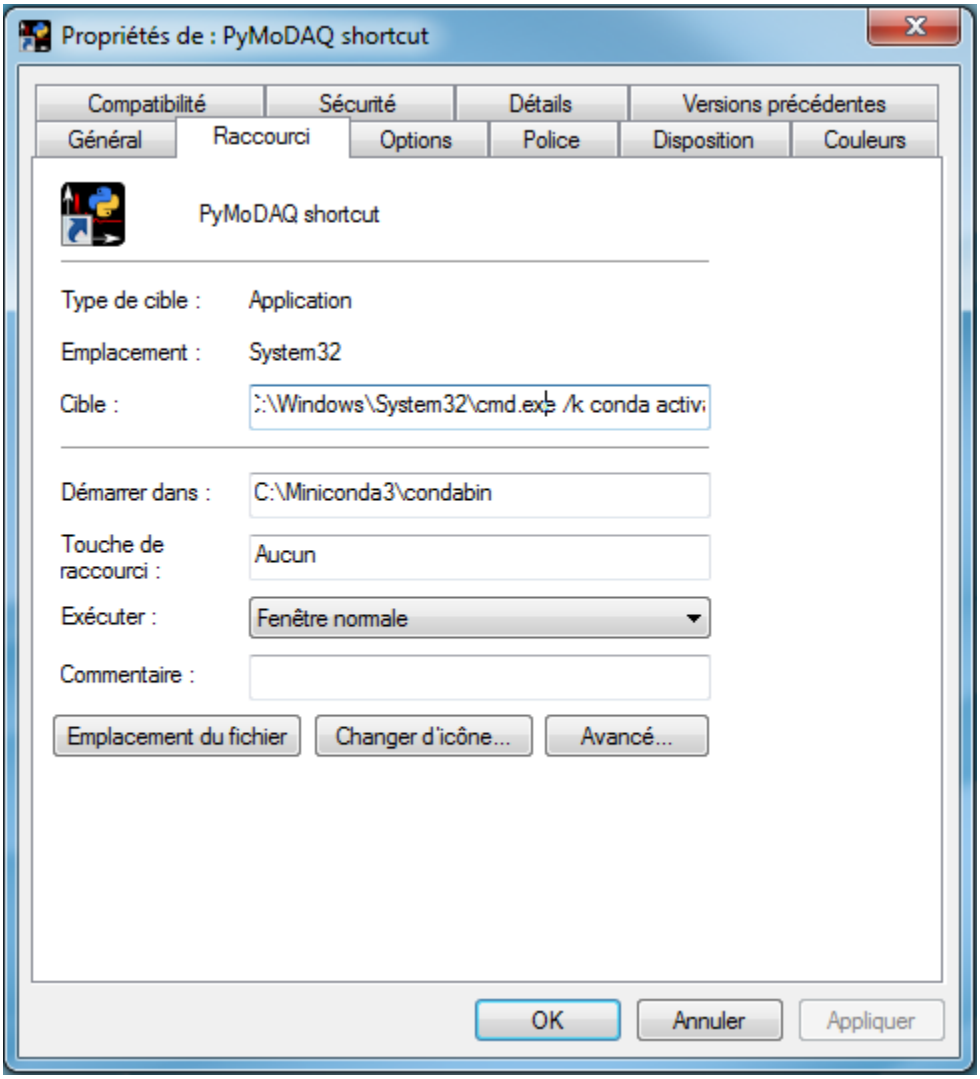

Fig. 6.2: Shortcut properties

6.2.2 What they think of PyMoDAQ?

- *"The use of PyMoDAQ has really accelerated our experimental development by allowing to develop a modular acquisition system involving very different motorized stages or piezoactuators. It is now running everyday on our experiments, 100% reliable"*, Dr Arnaud Arbouet, Senior Researcher CEMES/CNRS
- *Pymodaq is a python framework for data acquisition. If your specific device driver is not yet implemented, that is the only thing you will have to do. Pymodaq take care of the rest. Graphical user interface, synchronization of the instruments and so on, is already implemented. Once you have implemented your driver, you can release it for the community. That is how Pymodaq will get more and more complete. Of course you need to invest a bit of your time to get used to it, but it is worth it!*, Dr David Bresteau, Researcher at CEA Saclay, Attolab platform.

Note: If you are using PyMoDAQ and would like to help to promote the project, please send your feedback to sebastien.weber@cemes.fr and we will include your message or logo on this page. If you wish to contribute, see *Contributors*.

Note: If you wish to communicate with users of PyMoDAQ, a mailing list exists: pymodaq@services.cnrs.fr

---

6.3 Contributors

Here is a list of the main contributors:

6.3.1 Main modules

6.3.1.1 Simulator

- Sébastien Weber, Research Engineer at CEMES/CNRS

6.3.1.2 Retriever

- Sébastien Weber, Research Engineer at CEMES/CNRS

6.3.1.3 Propagator

- Romain Géneaux, Researcher at CEA Saclay, Lidyl

6.3.1.4 Cleaning

- Sébastien Weber, Research Engineer at CEMES/CNRS
- Romain Géneaux, Researcher at CEA Saclay, Lidyl

6.3.2 Documentation

- Sébastien Weber, Research Engineer at CEMES/CNRS
- Romain Géneaux, Researcher at CEA Saclay, Lidyl

6.3.3 You want to contribute?

If you're willing to help, you can clone the up-to-date GitHub repo: https://github.com/CEMES-CNRS/pymodaq_femto using git command line or GitHub Desktop. I advise to create a dedicated conda environment for this and install PyMoDAQ-femto's package as a developer:

- `conda create -n dev_env`
- `conda activate dev_env`
- `cd` to the location of the folder where you downloaded or cloned the repository.
- install the package as a developer using the command `pip install -e .`.

Any change on the code will be *seen* by python interpreter. When ready, you can ask to push your code into the main development branch. A simpler way is to raise *Issues* on PyMoDAQ-femto's github page.