ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA044

REALTA AND pyDART: A SET OF PROGRAMS TO PERFORM REAL TIME ACQUISITION AND ON-LINE ANALYSIS AT THE FERMI FREE ELECTRON LASER

E. Allaria, E. Ferrari[1], E. Roussel[2], L. Vidotto
Elettra Sincrotrone Trieste S.C.p.A., 34149 Basovizza, Italy
[1]now at École Polytechnique Fédérale de Lausanne EPFL, Lausanne, Switzerland
[2]now at PhLAM/CNRS, Villeneuve d'Ascq, France

Abstract

During the optimization phase of the FERMI Free Electron Laser (FEL) to deliver the best FEL pulses to users, many machine parameters have to be carefully tuned, like e.g. the seed laser intensity, the dispersion strength, etc. For that purpose, a new python-based acquisition tool, called REALTA (Real Time Acquisition program), has been developed to acquire various machine parameters, electron beam properties and FEL signals on a shot-by-shot basis thanks to the real time capabilities of the TANGO control system. The data are saved continuously during the acquisition in a HDF5 file. The pyDART (Python Data Analysis Real Time) program is the post-processing tool that enables a fast analysis of the data acquired with REALTA. It allows studying the correlations and dependences between the FEL and electron beam properties and the machine parameters. In this work, we present the REALTA and pyDART toolkit developed for the FERMI FEL.

INTRODUCTION

FERMI is the free electron laser facility in operation at Trieste in Italy. Based on a normal conducting linear accelerator, FERMI produces coherent pulses in the extreme ultraviolet (EUV) and soft - X-ray spectral range [1,2,3]. Stable operation of a FEL in the seeded mode as FERMI, requires a careful control of all the machine parameters. This requires a flexible control system that provides an effective integration of the machine devices. The Tango [4] toolkit is used at FERMI as a control system software allowing to easily include most machine hardware with dedicated device servers [5].

In high gain FELs such as FERMI, each electron bunch is an independent source of radiation whose properties critically depend on the electron beam properties. This leads to the need of a distributed real-time framework integrated into the control system that provides reliable information of electron and machine parameters on a pulse-to-pulse basis [5]. Within this framework, a unique "bunch number" time-stamp is distributed to all of the systems. Most of the measurement detectors (e.g., electron and photon diagnostics) and actuators (e.g., power supplies, RF systems) are synchronized with the bunch trigger and can have the "bunch number" associated to their measurements.

This capability is essential for the users operations of the FEL since it allows sorting and filtering the experimental data based on the required FEL properties. Moreover the real-time capability is very suited for machine studies since it allows to better identify the sensitivity to single parameters and recognize hidden correlation between FEL properties and jittering variables.

For most critical measurements both on the electron beam and on the FEL, it is important to allow operators and machine physicists to take advantage of the real-time capability of FERMI and use it for a post-processing on the data removing signal associate to non-desired shots. This capability was given at FERMI by a Matlab acquisition tool [6] interfaced to the control system via the available Tango bindings. For taking full advantage of the recent upgrade of FERMI to a 50 Hz operation it has been necessary to modify the available acquisition tool. While preserving a similar scheme, in order to increase the flexibility, the new acquisition tool has been programmed using the open source language Python [7] and takes advantage of the permanent development of open packages for data processing such as SciPy [8]. As for the old Matlab version, the new tool is divided in two main programs: one dedicated to the acquisition of real-time data (REALTA) and one to the quick analysis of the data (PyDART) that is necessary for a fast interpretation in control room. Datafiles are saved in the open HDF5 format [9]. HDF5 results in a versatile file format for storing and managing data and is compatible with many software platforms, such as Matlab, Mathematica, IGOR Pro, LabVIEW that can be used for a more deep offline analysis.

REALTA: THE ACQUISITION PROGRAM

REALTA is the new program developed in Python to perform flexible acquisitions of the numerous machine parameters. Using the Python bindings of Tango, it allows communicating with tango devices in a faster way than in Matlab. For the 50 Hz operation of FERMI, this is a critical point for reliable recording of 2D images from CCD cameras.

The acquisition program is controlled by a graphical user interface (GUI) that allows the operator to set various parameters for the acquisition. The GUI has two main panels, one dedicated to the setting of the list of devices to be acquired and one to the type of acquisition to be done.

† email address: enrico.allaria@elettra.eu

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA044

Selection Setting

With the new version the user can dynamically decide the specific devices (actuators and detectors) to be acquired. This option has been added to allow the use of the program in case of very distinct experiments requiring very different list of detectors to be acquired.

The main panel dedicated to the selection of the devices is reported in Figure 1.

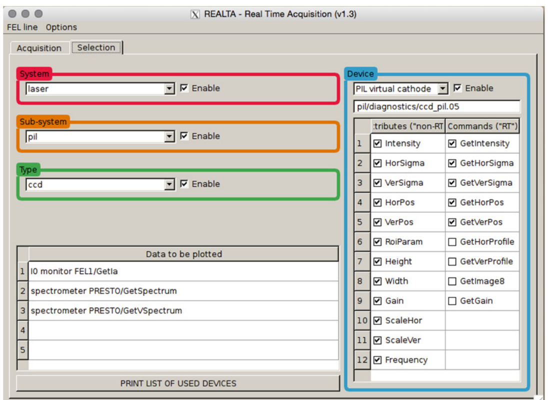

Figure 1 Selection panel for the REALTA GUI.

Most important devices that are critical for the FEL and the accelerator optimization are available in the program and can be selected for being acquired when desired. Each device is identified as part of a specific *system* of FERMI (i.e. the ‘Accelerator’, ‘Laser’, ‘FEL’, …) and a *sub-system* (i.e. a sector of the system). In the case of the “Accelerator”, the first bunch compressor (‘BC1’) and the laser heater (‘LH’) would be two of its sub-systems. Elements are finally identified by their specific functionality; the *type* refers to the feature of the device: e.g. CCD, motor, shutter, etc.. For each device the user can choose which of its attributes should be acquired. Similarly to the previous version, the new acquisition program has two sets of devices and attributes per acquisition.

A first set of devices (generally large) is used to acquire a snapshot of the machine status at the moment of the acquisition. This includes a large set of devices that characterizes the setting of the accelerator and the FEL. Being parameters such as undulator gaps, filter settings, … that are not expected to change over the time of one acquisition, these devices are acquired only once and do not require the real-time capability.

A second set of devices and attributes defines the real time data that the user wants to monitor and/or study as a function of variations on specific machine parameters. These data need to be acquired using the real-time capability of the control system in a way that it is assured that data from different devices are synchronized and refer to the same electron bunches.

Acquisition Setting

The second panel in the REALTA GUI is dedicated to the setting of the acquisition and is reported in Figure 2. From this panel the user can define the number of data points to be acquired and the file name automatically prefixed with the date and time of the acquisition. Due to memory consumption on single devices, in particular images from a CCD, only a reduced number of shots can be acquired in a row. In order to be compatible with the real-time acquisition of a large number of shots, the full acquisition needs to be divided in smaller number of consecutive *points* to be acquired in several *steps*

In addition to allow the synchronous acquisition of the machine data, REALTA also allows to perform scans of specific machine parameters. From the *Acquisition Setting* panel it is possible to select the machine actuator and the range to be scanned (orange area in Fig. 2). For multidimensional parameters scans and complex machine variations, an advanced scan option is also available that uses an external script.

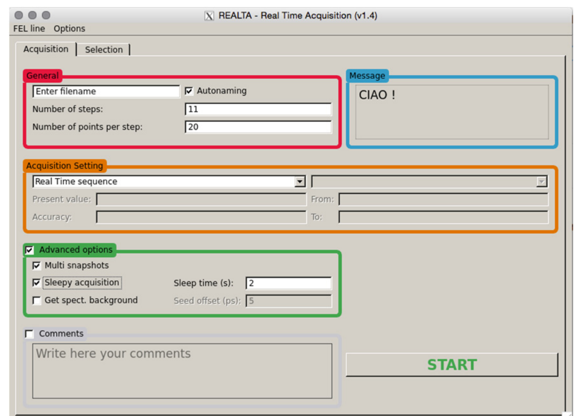

Figure 2 Main panel of the REALTA GUI for defining the acquisition.

More advanced options (green area in Fig. 2) are available and can be used for repeating the snapshot acquisition at every step or for introducing a defined delay between various steps in the acquisition. For measurements with CCDs affected by unavoidable background contamination it is possible to acquire special shots that can be used in the post-processing.

Saving and Display

The acquired data are continuously saved step-after-step during the acquisition into a HDF5 file (one standard acquisition step takes 0.2 s to be written on the disk, while about 1 s is required if 2D images are also acquired). Compared to the previous program that kept in memory the full acquired data set, the new program permits less memory consumption.

The HDF5 data file is divided in two groups: one for the snapshot of the machine parameters and one for the data monitored in real-time (Figure 3). For each device, a HDF5 group is created with the description of the device (system, sub-system, type, etc.) written in the attributes of the group. The acquired data are saved in a HDF5 dataset with a description of the data type (single value, 1D profile or 2D image). Compression option is used during the saving process to reduce the final file size that is essential when saving 2D images.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA044

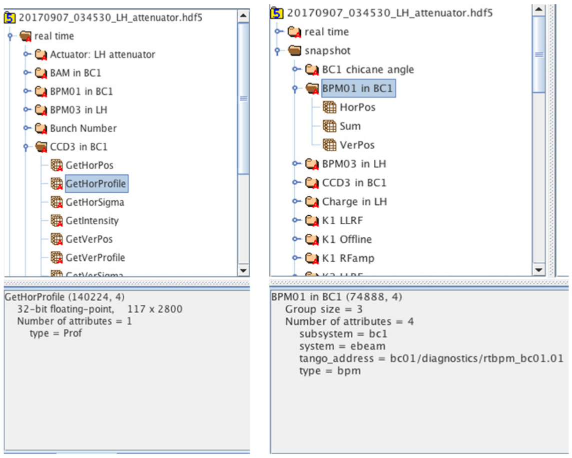

Figure 3: Structure of the HDF5 data file.

When the acquisition is finished, REALTA produces a graph (Figure 4) that summarizes some acquired data. The choice of the displayed data can be controlled in the Selection panel (Figure 1, "Data to be plotted" area) by right clicking on the wanted device. At the end of the acquisition, the data file becomes available for the post-processing tools.

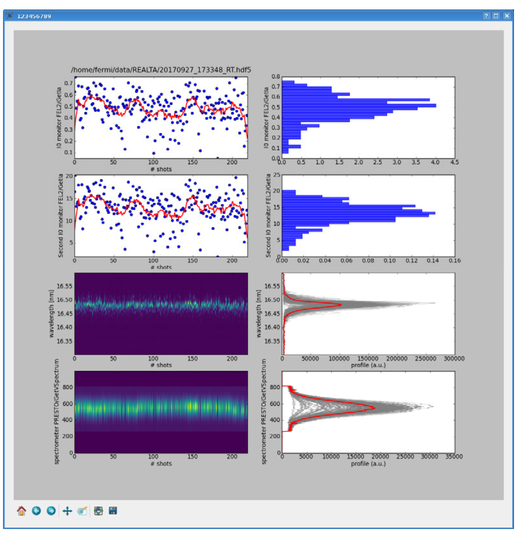

Figure 4: Image with the main information of the acquired data produced by REALTA at the end of the acquisition.

PYDART: THE ANALYSIS PROGRAM

A python-based program, PyDART, has been developed to offer a quick analysis of the acquired data in the control room. A GUI allows loading the files saved by REALTA and looking at the data (Figure 5). The GUI is divided in four panels. The first one ("*RT*") is dedicated to the display of the real-time acquired data that are sorted according to the datatype (single value, 1D profile or 2D image). The second panel ("*Snap*") permits a simple visualization of the machine parameters taken at the beginning of the acquisition. A table with the device names, the attributes and their values is produced and can be sorted by system. The third and fourth panel ("Filters", "Analyze") proposes advanced analysis options and will be detailed below.

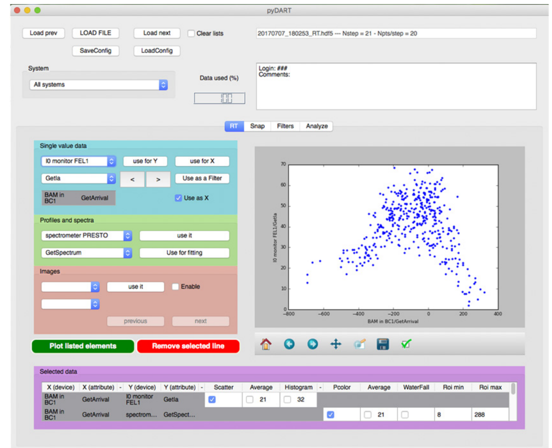

Figure 5: Main panel of the PyDART GUI to load the file and visualize the data.

Data Selection and Display

Once the file is loaded, the single value data (cyan area in Fig. 5) can be quickly visualized in a scatter plot either in function of the number of shots or in function of an other machine variable. The latter option can be very useful to identify correlation between FEL and/or machine variables. For the 1D profiles, a superposition of the distribution is quickly displayed, while for the 2D image, a single image is shown with the possibility to browse in the various shots.

More advanced plots can be obtained by adding the selected data to the plot table in the bottom. The selected data are then processed according to the selected plot options. For single value data, a simple scatter plot, a plot with the data averaged over a given number and the data distribution are available (Figure 6).

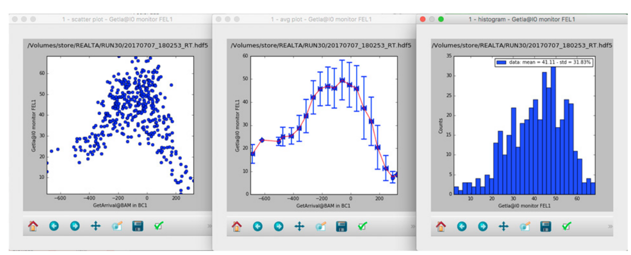

Figure 6: Advanced single value plots: scatter plot highlighting correlation between FEL properties and machine parameters, average plot and data distribution.

In case of profiles, a different list of advanced plot options is available: a false color image representation of the consecutive profiles, an averaged image over a given number of shots of the profiles and a waterfall superposition of the profiles. For the false color images, a second image is automatically computed in which each profile is normalized to its maximum value to enhance the shape of the profile (Figure 7). As for the single value data, profiles can be ordered as they have been acquired or as a function of a second variable.

16th Int. Conf. on Accelerator and Large Experimental Control Systems ICALEPCS2017, Barcelona, Spain JACoW Publishing
ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA044

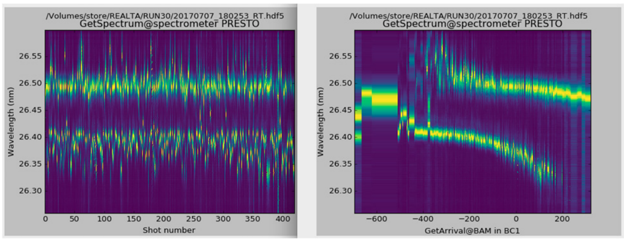

Figure 7: FEL spectra profiles ordered as a function of the acquisition time and of the arrival time of the beam in the first bunch compressor.

Data Filtering

A filter tool is available and can be activated when selecting a single value datatype. The possibility to filter data permits to improve the signal-to-noise ratio and get rid of bad or unwanted shots. The filter settings can be specified in the dedicated panel (Figure 8). The previously selected data for filter appears in the table. The band-pass filter can be set by specifying either extreme values or by defining number of RMS sigmas. A fast plot of the data used as filter is represented below the table with the discarded shots in red. Once a filter configuration has been defined, it can be applied to the data selected for further process. The percentage of used data is also updated in the main PyDART window.

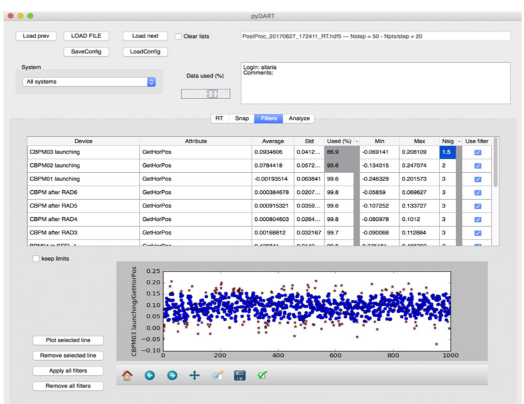

Figure 8: The filter panel of the PyDART GUI.

Profile Analysis

The profiles saved in the data files can be post-processed once they have been selected for analysis. Post-processing settings can be controlled in the last panel (Figure 9). During the post-processing step, the profile are analyzed in a given region-of-interest, and an asymmetric background is calculated by specifying the left and right region inside the ROI for background calculation and is subtracted to the profile. A smoothing of the data can also be activated. Finally, a Gaussian fit can be applied to the cleaned profile to determine beam sizes and positions in case of transverse profiles or spectral bandwidth and central wavelength in case of FEL spectra. The raw, cleaned and fit profiles can be visualized in a fast plot in the bottom part of the panel. The results of the analysis can then be saved in the HDF5 file. For secured storage of the data, a new file containing a copy of the initial data file is created and the results of the analysis are append to the real time HDF5 group and are identified with the prefix "Pp".

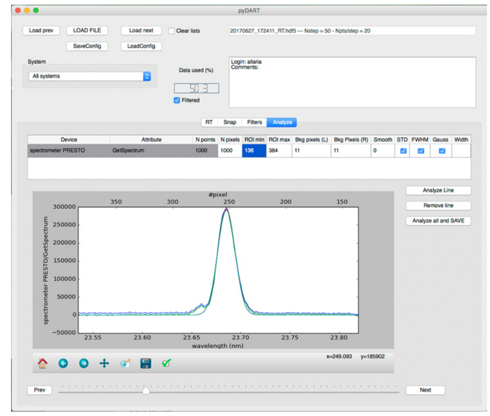

Figure 9: The analysis panel of the PyDART GUI.

The post-processed file can be then loaded in the PyDART GUI and the results of the analysis can be selected for data visualization or filtering (Figure 10).

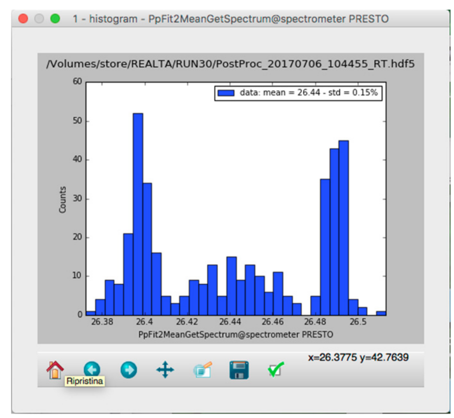

Figure 10: Histogram distribution of the central wavelength calculated from Gaussian fit of the FEL spectrum during a two-color operation.

CONCLUSION

The past experience at FERMI has shown the importance of a reliable tool for data acquisition dedicated to machine physics experiments and FEL optimization. The existing matlab based program has been recently replaced by a new set of python-based programs. The new programs have improved the usability of the software by general users and it allows an online setting of acquired and monitored machine parameters. In order to facilitate the off-line analysis of the acquired data, most important information of the acquired files is stored in a database that can be used to find the interesting files.

ACKNOWLEDGEMENT

Authors are grateful to all the FERMI team for the valuable support and acknowledge W.M Fawley, G. Kourousias and F. Bille for useful discussions and suggestions. We also thank M. Popoff, Homebrew maintainer, for his

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA044

assistance on laying out the architecture of the Python softwares.

REFERENCES

[1] E. Allaria *et al.*, Nat. Photonics **6**, 699 (2012).

[2] E. Allaria *et al.*, Nat. Photonics **7**, 913 (2013).

[3] E. Allaria *et al.,* Journal of Synchrotron Radiation **22**, 485 (2015).

[4] TANGO Controls, http://www.tango-controls.org/

[5] M. Lonza *et al.*, "Status report of the FERMI@Elettra control system", *Proceedings of ICALEPCS2011*, Grenoble, France, 2011, TUDAUST02.

[6] E. Allaria, W.M. Fawley and E. Ferrari, "A tool for real time acquisitions and correlation studies at FERMI", *Proceedings of FEL2014*, Basel, Switzerland, 2014, THP070.

[7] G. van Rossum, Python tutorial, Technical Report CS-R9526, Centrum voor Wiskunde en Informatica, Amsterdam, (1995).

[8] E. Jones, T. Oliphant, P. Peterson et al. SciPy: Open Source Scientific Tools for Python, 2001, http://www.scipy.org/

[9] The HDF Group, https://support.hdfgroup.org/HDF5/