ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA211

# XLive: DATA ACQUISITION AND VISUALIZATION AT THE NSLS-II ISS BEAMLINE

B. V. Luvizotto†, K. Attenkofer, E. Stavitski, H. Bassan, Brookhaven National Laboratory (BNL), Upton, USA

*Abstract*

Asynchronous data acquisition at the Inner-Shell Spectroscopy beamline at NSLS-II is performed using custom FPGA based I/O devices ("pizza-boxes"), which store and timestamp data using GPS based clock [1]. During motor scans, incremental encoder signals corresponding to motion as well as analog detector signals are stored using EPICS IOCs. As each input creates a file with different timestamps, the data is first interpolated onto a common time grid. The energy scans are performed by a direct-drive monochromator, controlled with a Power PMAC controller [2]. The motion is programmed to follow the trajectory with speed profiles corresponding to desired data density. The "pizza-boxes" that read analog signals are typically set to oversample the data stream, digitally improving the ADC resolution. Then the data is binned onto a energy grid with data spacing driven by desired point spacing. In order to organize everything in an easy-to-use platform, we developed XLive, a Python based GUI application. It can be used from the pre-experiment preparation to the data visualization and exporting, including beamline tuning and data acquisition.

## INTRODUCTION

The NSLS-II ISS Beamline data acquisition system consists of fly-scanning an energy range by moving the monochomator while collecting data from multiple detectors. Encoders and analog signals are sampled using custom FPGA based I/O devices ("pizza-boxes") – that are also able to trigger detectors and read external triggers using TTL inputs and outputs. The minimum acquisition time for analog inputs is 1 µs. Typically the acquisition time used is around 1 ms (after hardware averaging).

Fast and asynchronous data collection allows the beamline to get to the next level of complexity for spectroscopy measurements: n-dimensional data sets. The system is able to handle multiple detectors at the same time, moving users from 2D data sets (energy and intensity) to more complex data sets (e.g. energy, intensity, sample temperature, electric charge in the sample).

The main goal of XLive, a Python based GUI application, is to handle everything needed by the beamline from the pre-experiment preparation to data acquisition, visualization and processing within a reasonable time, being as simple as possible for users and as flexible as possible to be eventually exported to other NSLS-II spectroscopy beamlines.

† luvizotto@bnl.gov

## DATA ACQUISITION OVERVIEW

The data acquisition system (Figure 1) has three main layers: devices (hardware), EPICS IOCs and data acquisition software (XLive), which can be subdivided in three other layers – Ophyd [3, 5], Bluesky [4, 5] and GUI.

Currently, the beamline gets data from:

- Three ion chambers;
- One PIPS detector;
- Multiple encoders;
- Analog and temperature inputs (PLC);
- SDDs - Silicon Drift Detectors (XIA PXI Crate [6]);
- BPM Cameras (Prosilica GT1290 [7]).

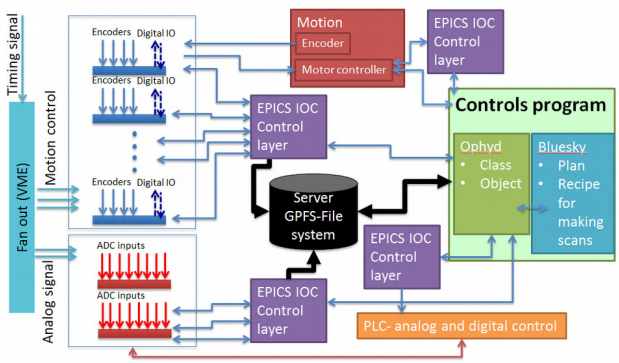


Figure 1: Data Acquisition System overview.

Since the beamline has analog inputs, digital outputs that can be used as triggers and digital inputs available, it is accessible to an assortment of detectors that ca be integrated into the data acquisition system to fulfill users' needs.

Each device is controlled by and share data through an EPICS IOC, which is responsible to control data acquisition and triggers and to generate data files.

Ophyd, a Python library developed by the Data Acquisition, Management and Analysis (DAMA) Group at NSLS-II, is responsible for the EPICS/Python abstraction of the hardware. It contains all the tools to define custom devices, allows direct EPICS PVs reading and writing and defines how each device will work when the system is running a scan.

Bluesky, also a Python library developed by DAMA group, works with objects created using Ophyd. While Ophyd defines the way motors, detectors and other devices will operate, Bluesky is responsible for experiment control and data collection. The beamline defines plans (recipes), containing which detectors and motors will be used and how the data collection will happen. At ISS, depending on the situation, both step and

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA211

fly scans will be used: step scans are used mostly for simple motor / detector scanning and tuning and fly scans are used for real data acquisition.

## GUI

The graphical user interface was created to help users follow the expected flow to set up everything and do experiments. It is a tab-based application (Figure 2) where the user should go from the first to the last tab, using features as they are needed.

The first three tabs (“Beamline Status”, “Trajectories” and “XIA”) are used to prepare the beamline, detectors and motors for the experiment, the next two tabs (“Run” and “Run Batch”) are used to run experiments and the last tab (“Processing”) is used for data processing.

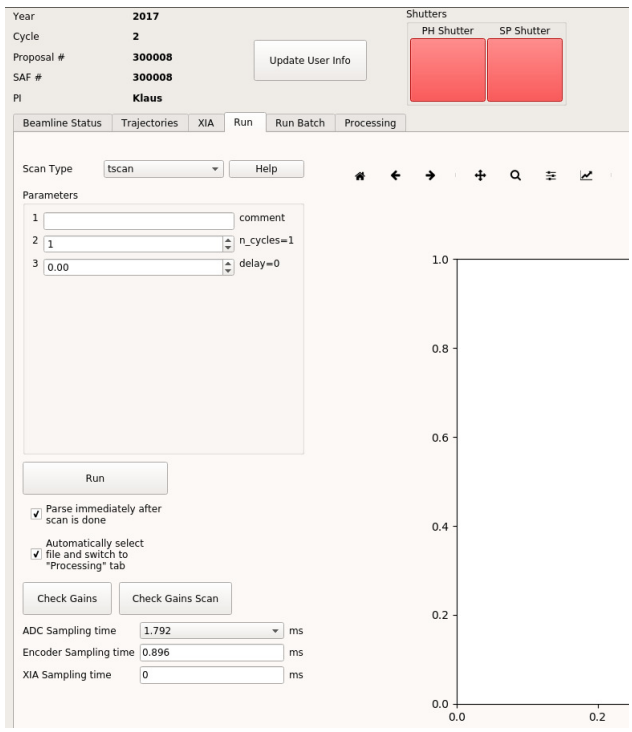


Figure 2: XLive's GUI - tabs and controls.

## BEAMLINE PREPARATION

ISS is a very flexible beamline. Its energy range, for example, is from 4.9 to 36 keV. In order to provide such great flexibility, a few steps have to be performed before running specific experiments. The first three tabs of the GUI are used for beamline setup, trajectory generation and beamline tuning, respectively.

### *Automatic Setup*

In order to properly change the energy range of the experiments, some parameters have to be updated: gas flow and voltage in the ion chambers, filter box position, which high harmonics rejection mirror will be used – bare silicon, silicon coated with Pr or silicon coated with Rh – and BPM Cameras positions (in or out). To make this process easier and more reliable, an automatic routine has been created. All the parameters for each energy range are defined in a JSON file. To setup the beamline for a different energy, the user presses a button in the first tab of the software, “Beamline Status”, a dialog pops-up, describing all the new parameters for this specific energy and the user has the option to confirm or cancel. After confirming, everything will move to the right value automatically in the right sequence.

### *Tuning*

After preparing the beamline for a specific energy, the next step would be to tune the beamline. “Tuning” the beamline means finding the best settings to deliver as many photons as possible by changing some beamline variables (pitch of the monochromator, for example).

Another feature in this tab is scanning a motor while collecting data from a detector. The user can choose any motor and detector, define a range and the step size for this scan, the software will run the scan and live plot it. The user has the option to move the motor to the desired position by clicking on the graph.

### *Trajectory Generation*

In the following step users must specify the energy range to be used to scan their samples for each experiment. In the “Trajectories” tab, the user can choose from four different trajectory types, generate a file containing all the positions of this trajectory and load it into the Power PMAC motor controller. The PMAC is configured to follow the trajectory after receiving a “start” signal.

The trajectory types are:

- Step – user defines energy positions (eV) and velocities (eV/s) for each region (pre-edge, edge and post-edge);
- Sine – user defines first and last positions (eV) and the duration of the scan (s);
- Double Sine – user defines first, last and edge positions (eV) and pre-edge and post-edge durations (s). The motor comes to a full stop in the edge energy;
- Double Sine/Constant Edge (Figure 3) – similar to Double Sine, but the user will also define the edge velocity (eV/s). In this case the motor does not come to a full stop.

Currently, the most used one is “Double Sine/Constant Edge” which seems to be the best option mechanically since it has the first, the second and the third derivatives continuous.

### *XIA Configuration*

The third tab, “XIA”, is used to setup the SDD detectors connected to the XIA Crate. The software has a semi-automated routine to calibrate the energy range of each detector. It is also used to define up to twelve ROIs (Regions of Interest) that will be used by scans.

The output file will have a data column with the integration of the values for each ROI defined, making it

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA211



simpler to collect data for multiple energies at the same time.

Currently the crate has 4 boards and each board has 4 inputs for detectors, giving a total of 16 detectors.

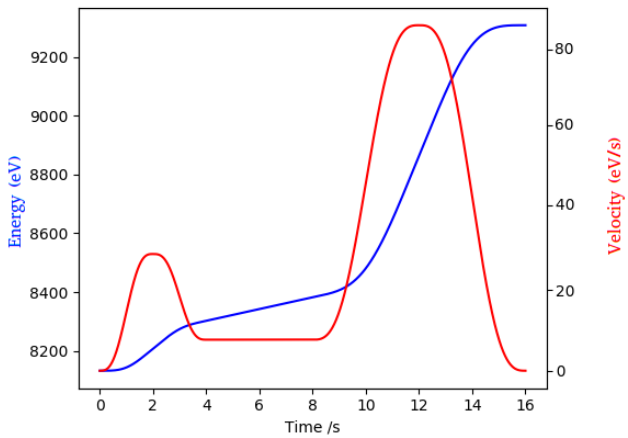


Figure 3: Double Sine/Constant Edge trajectory example (nickel K edge).

## RUNNING EXPERIMENTS

Once the users finish setting up the beamline (trajectory for the controller, detectors, mirrors, gases and filters etc.) they are ready to run the experiment.

There are two different options to run experiments: regular scans and batch mode.

### *Regular Scans*

The "Run" tab contains everything needed to start running scans. The user will first choose between "tscan" (trajectory scan), used for regular experiments acquiring data from the ion chambers, and "tscanxia" (trajectory scan + XIA), experiments that will acquire data from the ion chambers and the SDD detectors from the XIA crate. The next step is passing parameters to the scan: name, number of repetitions and delay between each repetition. Before clicking "Run", the user has the option to choose sampling times for ADCs (from 7 us to 115 ms), SDDs (60 ms minimum, currently) and Encoders. Oversampling is a common practice, since the more points the scan generates, the more reliable the data will be after binning.

### *Batch Mode*

Created to automate scans for multiples samples and energies, the batch mode is being used more and more by users. First, the user defines all the samples that will be automatically scanned, telling the system their X/Y positions in the sample holder stage. The next step is creating scans, where the user defines the trajectory that will be used and the scan type (tscan or tscanxia, currently). Once samples and scans are defined, there are two options for telling the system what to do: step by step, where the user will create every single step of the batch run or using sample loops.

For sample loops the user adds samples and scans by simply dragging and dropping them, chooses if it will run all samples for each scan or all scans for each sample and sets the number of repetitions of the loop.

A typical configuration of batch mode is shown in Figure 4.

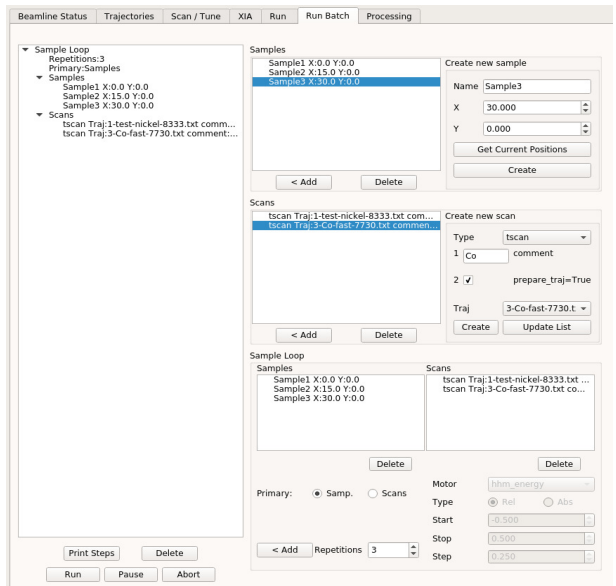


Figure 4: Typical batch mode configuration.

## PROCESSING

Each scan generates a file containing a column for energy and a column for each detector. After running the scans, the last step is opening the data, binning and exporting it. The last tab of the software, "Processing", is used for that (Figure 5). The file typically has thousands or even millions of rows (depending on the length of the scan and sampling time) before binning and around a thousand after binning – typical number for spectroscopy measurements.

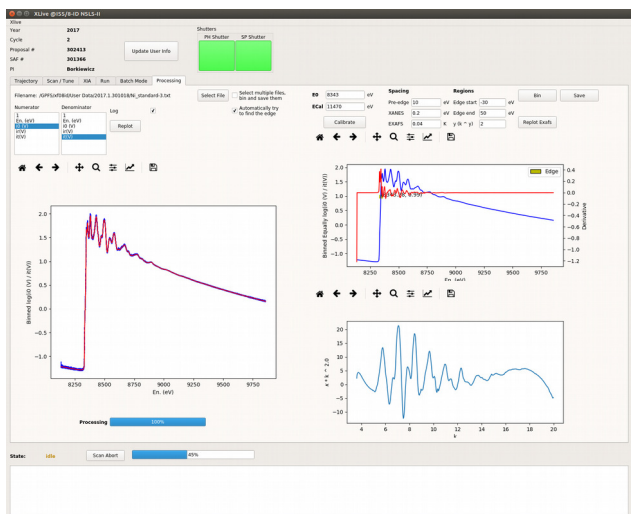

Figure 5: "Processing" tab showing the results for a nickel foil.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA211

The graph on the left shows the pre-processed data (in blue) and the final binned data (in red). The graph on the top-right shows the data binned with regular spacing (usually 2 eV, in blue) and its derivative (in red). The traces shown in the second graph are used to define the edge energy (manually or automatically). Finally, the last graph is a preview of the EXAFS.

The binning is done using a Gaussian Filter, by a convolution of the original data and a Gaussian function. A binning output example is shown in Figure 6 below.

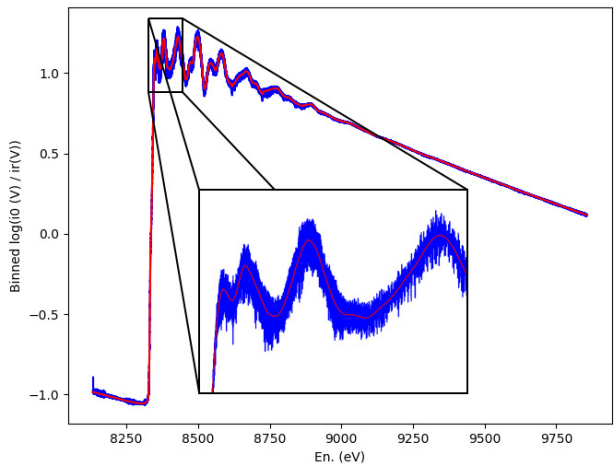


Figure 6: Binning output using a Gaussian Filter.

After binning, the user has the option to export the data to a file with .dat extension that can be read by Athena [8], an interactive graphical utility for processing EXAFS data.

## CONCLUSION

Handling everything a beamline needs from experiment preparation to data acquisition and processing can be complex. XLive is able handle multiple operations and provides the best option to make its usability better and to make it easier to port to other beamlines. The integration with Bluesky and Ophyd is fundamental for its flexibility.

The system has been used for several months with satisfactory results and reliability with no major issues. Users usually learn in reasonable time how to operate it.

The next steps will be to integrate tools for data analysis to the system, reducing the need for third-party software and rewrite part of the code to make it even easier to port to other beamlines.

## REFERENCES

[1] R. Kadyrov *et al.*, "Encoder Interface for NSLS-II Beamline Motion Scanning Applications", ICALEPCS'15, Melbourne, Australia, October 2015, WEPGF080.

[2] Delta Tau Power PMAC, http://www.deltatau.com/dt_powerpmac/powerpmachome.aspx

[3] Ophyd, https://nsls-ii.github.io/ophyd

[4] Bluesky, https://nsls-ii.github.io/bluesky

[5] A. Arkilic, D. B. Allan, T. A. Caswell, L. Li, K. Lauer and S.Abeykoon, "Towards Integrated Facility-Wide Data Acquisition and Analysis at NSLS-II", Synchrotron Radiation News, vol. 30:2, pp. 44-45, 2017. DOI: 10.1080/08940886.2017.1289810

[6] XIA PXI Crate, http://www.xia.com/PXI_Crate.html

[7] Prosilica GT 1290, https://www.alliedvision.com/en/products/cameras/detail/Prosilica%20GT/1290.html

[8] Athena, http://bruceravel.github.io/demeter/documents/Athena/index.html