Python Expert Applications for Large Beam Instrumentation Systems at CERN

J. Martínez Samblas, E. Calvo, M. Gonzalez-Berges, M. Krupa
Beam Instrumentation, CERN, 1211 Geneva 23, Switzerland

Abstract

In recent years, beam diagnostics systems with increasingly large numbers of monitors, and systems handling vast amounts of data have been deployed at CERN. Their regular operation and maintenance poses a significant challenge. These systems have to run 24/7 when the accelerators are operating and the quality of the data they produce has to be guaranteed. This paper presents our experience developing applications in Python which are used to assure the readiness and availability of these large systems. The paper will first give a brief introduction to the different functionalities required, before presenting the chosen architectural design. Although the applications work mostly with online data, logged data is also used in some cases. For the implementation, standard Python libraries (*e.g.* PyQt, pandas, NumPy) have been used, and given the demanding performance requirements of these applications, several optimisations have had to be introduced. Feedback from users, collected during the first year's run after CERN's Long Shutdown period and the 2023 LHC commissioning, will also be presented. Finally, several ideas for future work will be described.

Large Beam Instrumentation Systems

The LHC produces vast amounts of data not only from particle collisions but also from its Beam Instrumentation (BI) systems, which are essential for the operation of the accelerators. This paper delves into two large BI systems:

Diamond Beam Loss Monitors (Diamond BLMs):

- 17 across all accelerators.
- Diamond crystals with gold electrodes polarised at 500V.
- Bunch-by-bunch resolution (1.53 ns).
- Buffer millions of samples per second (650 MSPS).

Beam Position Monitors (BPMs):

- Over 1000 BPMs along the LHC.
- 70 FESA devices.
- Non-destructive electromagnetic sensors.
- Measure the position in the horizontal and vertical planes.
- Turn-by-turn and bunch-by-bunch resolution.

Implementation

Framework > PyQt5 was chosen due to its stability and increasing adoption at CERN's accelerator teams. It offers a modular environment, leveraging the model/view design pattern to separate GUI data from its visual representation.

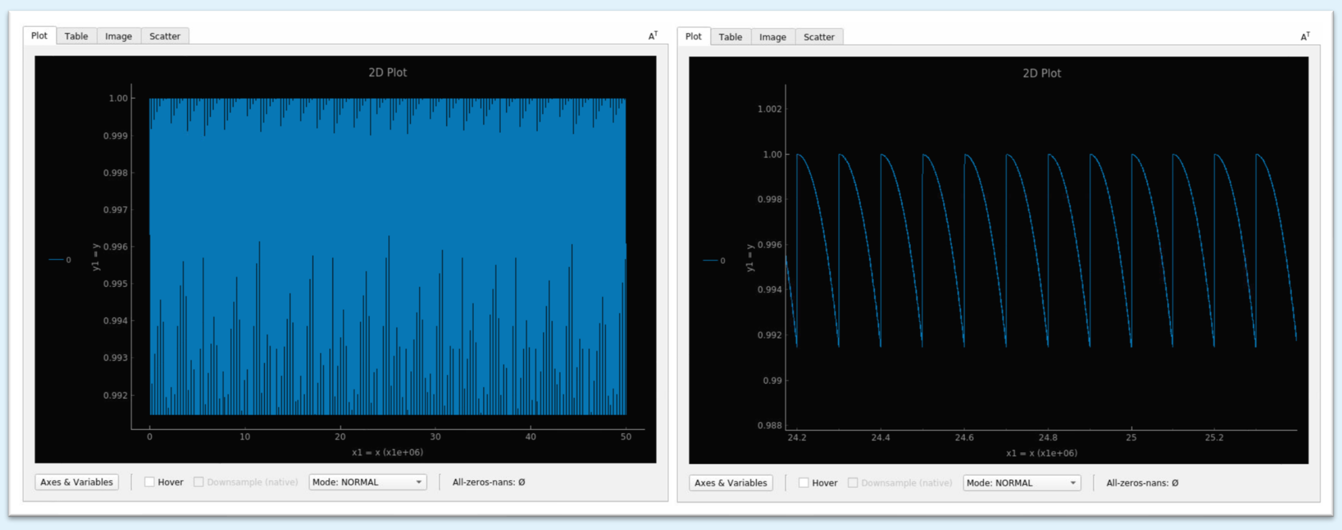
Tables > Custom QTableView with pandas engine as model:

- Filtering, regular expression searching, natural sorting, type conversions, etc.
- x4 less memory than PyQt's default table.
- Loading times grow linearly instead of exponentially.

Plots > PyQtGraph was employed given its speed in rendering interactive displays:

- For medium-sized curves (~1M samples), the built-in PyQtGraph's downsampling algorithm works responsively.
- For big data (>10M samples), a custom downsampling algorithm had to be implemented. Its major steps are:
    i. Downsampling factor calculation (d).
    ii. Regularisation of d and the plotting range indices.
    iii. Envelope preservation by min-max interleaving each data chunk.
    iv. Caching for fast loading during zoom changes.

Multithreading > Used Qthread from PyQt for data synchronisation and PyJAPC for retrieving the data from the FESA devices.



DAVIT: Downsampling on a 50M samples array. Envelope is preserved when zooming (right plot).

Commissioning and Operation Experience

The Diamond BLM online application has greatly facilitated the commissioning of the LHC and SPS post-winter shutdowns. It efficiently manages detector configurations, minimising errors and simplifying previously manual tasks. The application supports simultaneous execution of commands, and the automatic phase-in algorithm has saved considerable time by ensuring the BCT signal perfectly fits within the correct bunch slot. DAVIT has eased the exploration of the numerous Diamond BLM files, resulting in substantial savings in post-analysis time for MD studies.

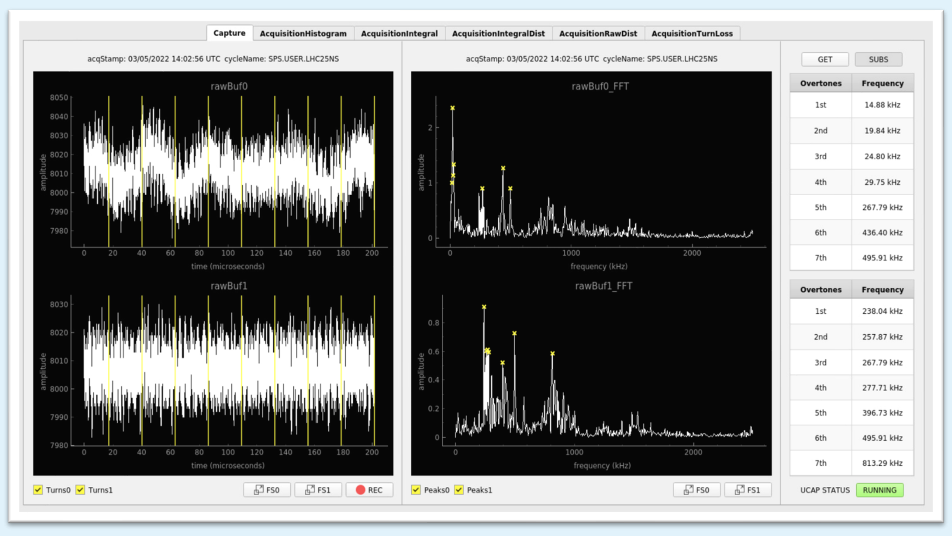
The new LHC BPM suite of applications has aided in identifying broken and anomalous channels, diagnosing intricate performance issues. The tools provide experts with not only a broad overview of the entire BPM system, but also the ability to examine critical low-level details using a user-friendly interface; Most problems are now easily analysed directly through the Expert GUIs.

Online Applications

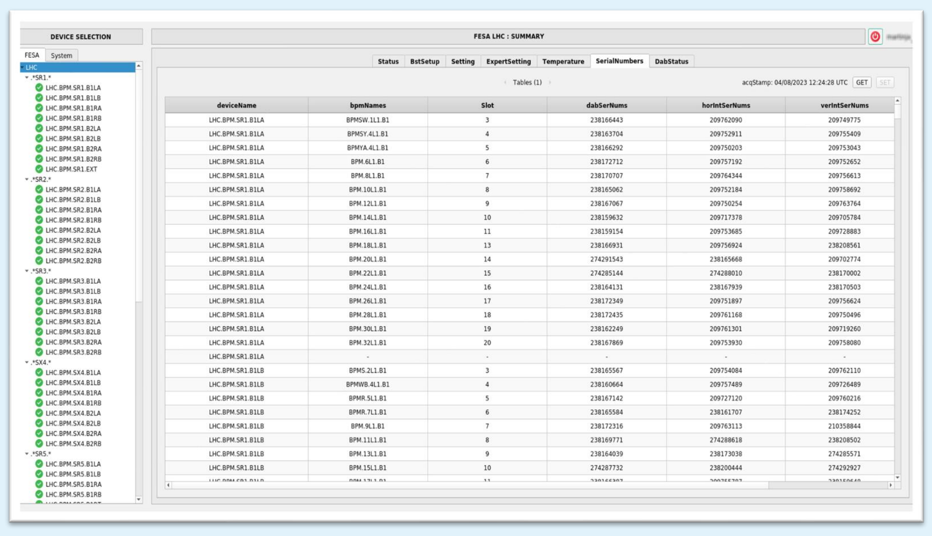
Online applications handle real-time data, streaming directly from devices via FESA. These applications serve during both commissioning and operational periods, and they have the capacity to monitor and control all devices simultaneously.

The **Diamond BLM application** offers comprehensive visualisations of raw buffers, including detailed bunch-by-bunch and FFT representations. FFTs are computed on independent nodes through UCAP. The app also incorporates an algorithm for phase-in calibration using the BCT signal.

The **LHC BPM application**, part of the new Python ecosystem, provides summary views, detailed plots, multi-device control, and serial numbers history tracking for the extensive range of BPMs.



Diamond BLM Expert Application: Capture & FFTs.



LHC BPM Expert Application: Device Summary.

Offline Applications

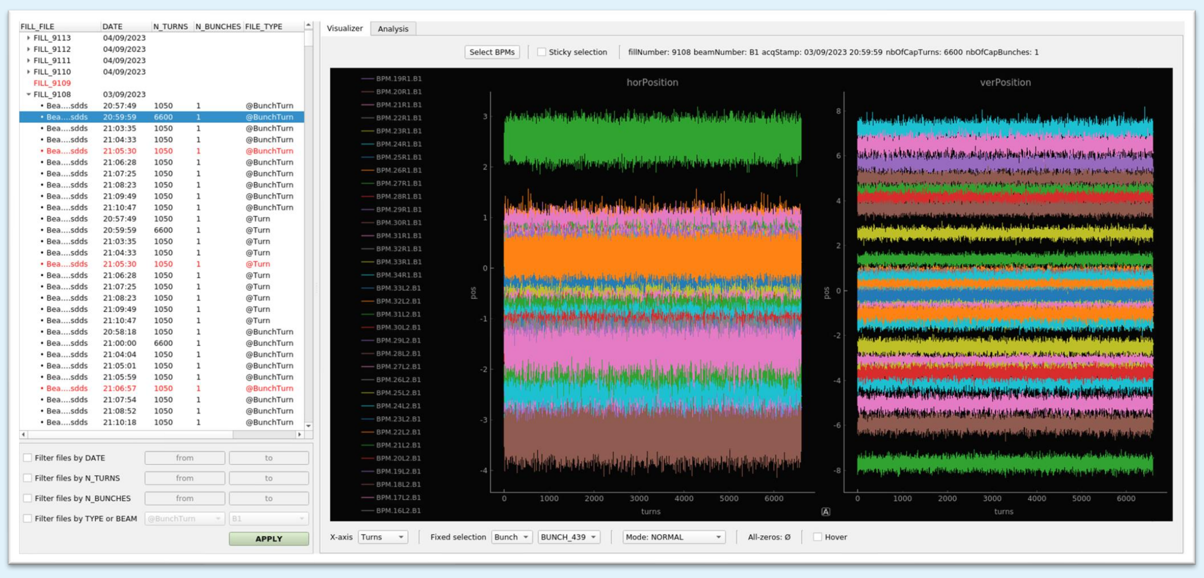
Offline applications process a variety of logged data (*e.g.* HDF5, SDDS, NXCALS).

The **LHC BPM Capture Analysis Tool** is used to analyse, explore and filter BPM capture data. It integrates three anomaly detection algorithms to diagnose malfunctioning BPMs:

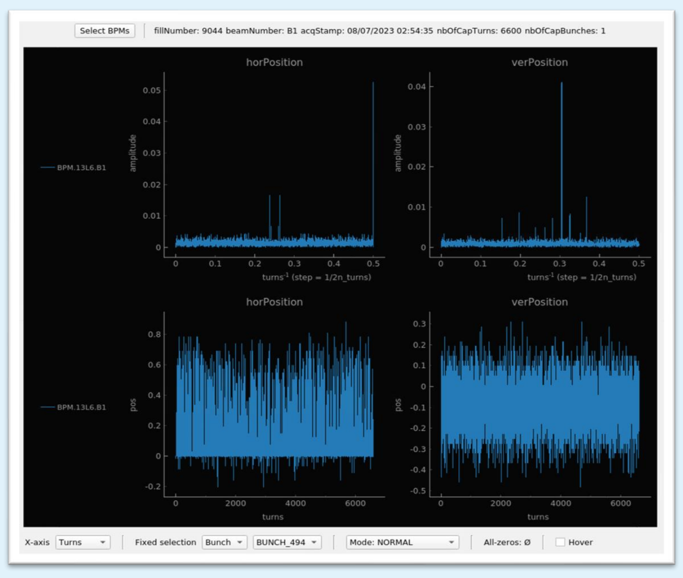
- **Spike detection** > Identifies unexpected peaks in the position data by calculating the Z-Score and checking for points that exceed a threshold.
- **N-zeros** > Any BPM showing all-zeros, or a large range of zeros, in any of their planes indicate potentially corrupted data.
- **Tune comparison** > FFTs are calculated after DC offset removal. The tune is calculated as the highest peak of the FFT and it should be consistent for all BPMs within the same plane; Any BPM with a differing tune is flagged as an outlier.

The LHC BPM Python ecosystem also includes auxiliary tools such as the **FIP Expert GUI**, which manages FIP control operations like BPM calibration, and the **Memory Check Tool**, which assesses the memory integrity of FECs.

The **Data Analysis and Visualisation Tool (DAVIT)** is a general-purpose program built to handle big data that exceeds FESA's limits. It allows for high-performance visualisations and offers versatile manipulations using pandas dataframes, such as merging, transposing and slicing.



LHC BPM Capture Analysis Tool: H&V Positions for a 6600 turns file.



Anomaly Detection: Tune Analysis.

Future Work and Conclusions

All applications will necessitate regular maintenance to keep compatibility with FESA changes as well as updates to other libraries. The LHC BPM Capture Analysis Tool is scheduled to be upgraded with two new anomaly detection algorithms based on Isolation Forest and SVD dominance analysis. DAVIT might also be expanded to handle new file types (*e.g.* Parquet) and data sources (*e.g.* PostMortem).

Managing Beam Instrumentation systems is challenging due to the substantial number of devices and the vast data volumes they continuously collect. The suite of applications introduced in this paper successfully addresses this challenge by delivering rapid overview information, facilitating data manipulations, enabling automatic and swift anomaly detection, and providing comprehensive visualisations. Performance techniques, such as downsampling and multithreading, have been explored and will be used for future Python developments in other accelerator systems.

Beam Instrumentation Group
SY-BI-SW (CERN)
javier.martinez.samblas@cern.ch
THPDP061 (Contribution ID)
ICALEPCS CAPE TOWN SOUTH AFRICA 2023