S-FRSRoot Online Monitoring and Analysis Framework

Ahmed Ali & Stephane Pietri

SFC/Team4

November 17, 2025

Outline

1. Project Overview and Status
2. Schematic Overview
3. S-FRSRoot Framework Lifecycle
4. Framework Adding Subsystem
5. Framework Testbeam Validation
6. Framwork Monitoring
7. MOCADI Simulation Ingestion
8. Gating GUI - PID Simulation Data
9. Framework Documentations
10. Future Work and Strategic Directions
11. Summary and Outlook

Project Overview and Current Status

- This presentation outlines recent progress, schematic design, benchmarks, and future directions for the S-FRSRoot (Super-FRS Root) online monitoring and analysis framework.
- The primary goal is to develop a robust and efficient system for:
  - Real-time data monitoring, advanced data visualization
  - Analysis framework
  - Alarm System
  - Potential integration with other control systems

Schematic Overview

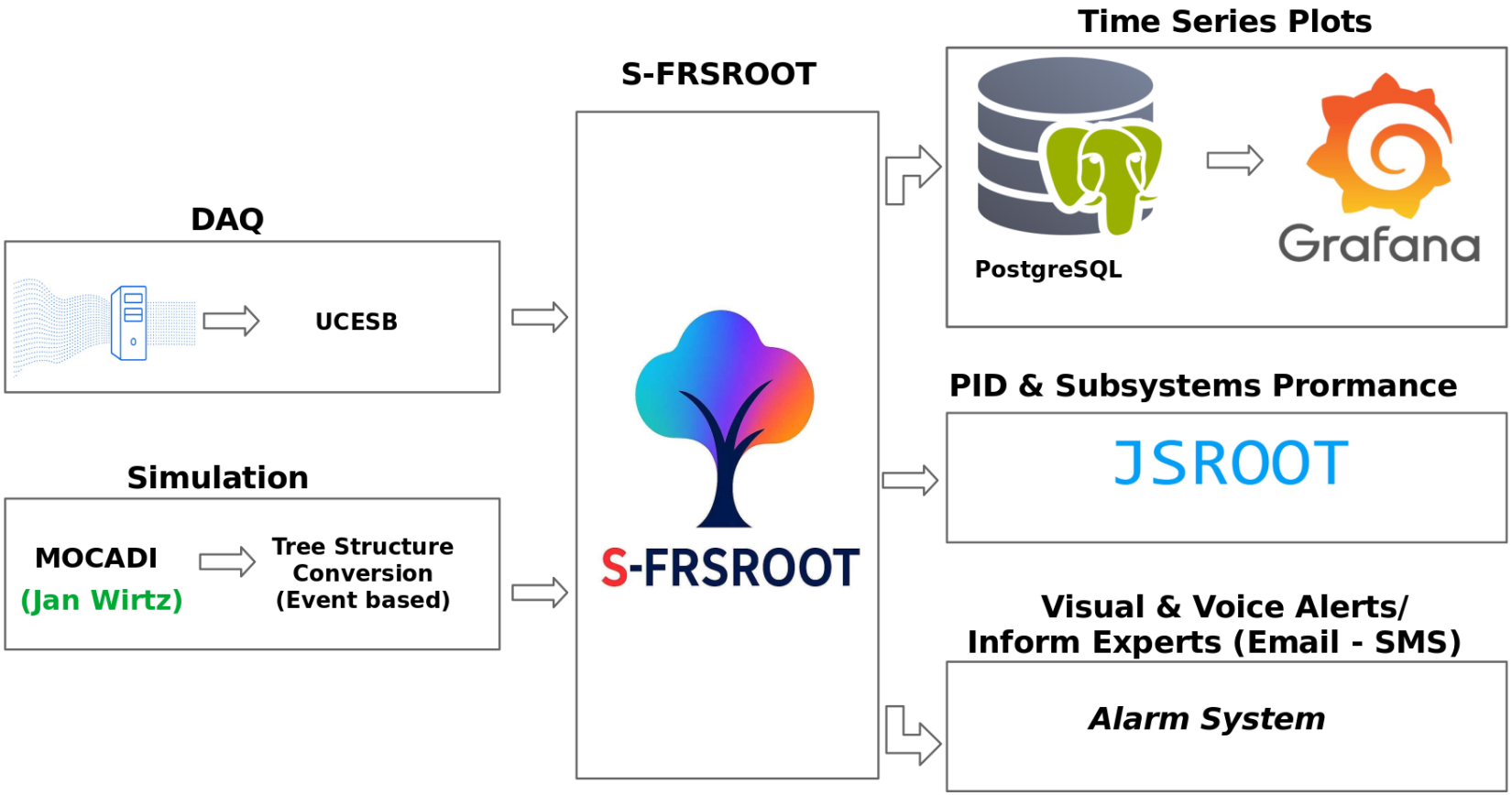


Figure: S-FRSRoot Components

S-FRSRoot Framework Lifecycle

1 Input Specification

Provide detector definitions and mapping files.

- **UCESB:** A complete unpacker is built from several `.spec` and `.hh` files. The main `.spec` file acts as the top-level blueprint that includes definitions for specific electronics and detector mappings.
- **Main SPEC File:** This is the top-level file for the unpacker. It defines the complete physics event by including other spec files and mapping subevent identifiers to their specific unpacking recipes. The goal is to build a complete event containing all subsystems.
- *Goal: Ensure raw data can be correctly parsed by the framework.*

2 Executable Generation

Compile source code into a UCESB analysis executable, generate subsystems header files.

The framework uses full physics event definition, the specification files, detector mapping, and subsustems header files to build a dedicated executable for processing the raw experimental data stream.

3 Detector Calibration

Subsystem experts provide calibration logic.

- **Calibration Logic:** Algorithms (ROOT scripts/C++) to convert raw signals (ADC, TDC) into physical units.
- **Input Data:** Dedicated Fair tasks handel detector calibration for each subsystem.

4 Calibration Feedback Loop

Iteratively refine calibration logic.

The development team implements calibration logic. Results are reviewed by experts, who provide feedback. This cycle repeats until calibration is validated and optimized.

5 Subsystem Reconstruction

Experts provide their algorithms to derive detector delevrables.

Using calibrated data, higher-level physics quantities (time, dE, position, momentum, etc) are reconstructed. Logic is provided as ROOT scripts or C++ classes.

6 Reconstruction Feedback Loop

Iteratively optimize reconstruction algorithms.

The framework team implements reconstruction logic. Reconstruction results are validated against testbeam data or simulations. Experts review the output and provide feedback for algorithm improvement.

7 Online Monitoring

Define requirements for real-time visualization.

Experts specify key parameters, histograms, and plots (for JSRoot and Grafana) to monitor detector performance and beam conditions in real-time during an experiment.

8 Particle Identification (PID)

Perform final analysis and optimize the PID.

The ultimate goal: uniquely identify each particle by plotting its atomic number (Z) vs. mass-to-charge ratio (A/Q). The clarity of this identification is the benchmark for the success of all preceding steps.

Framework Adding Sub-system

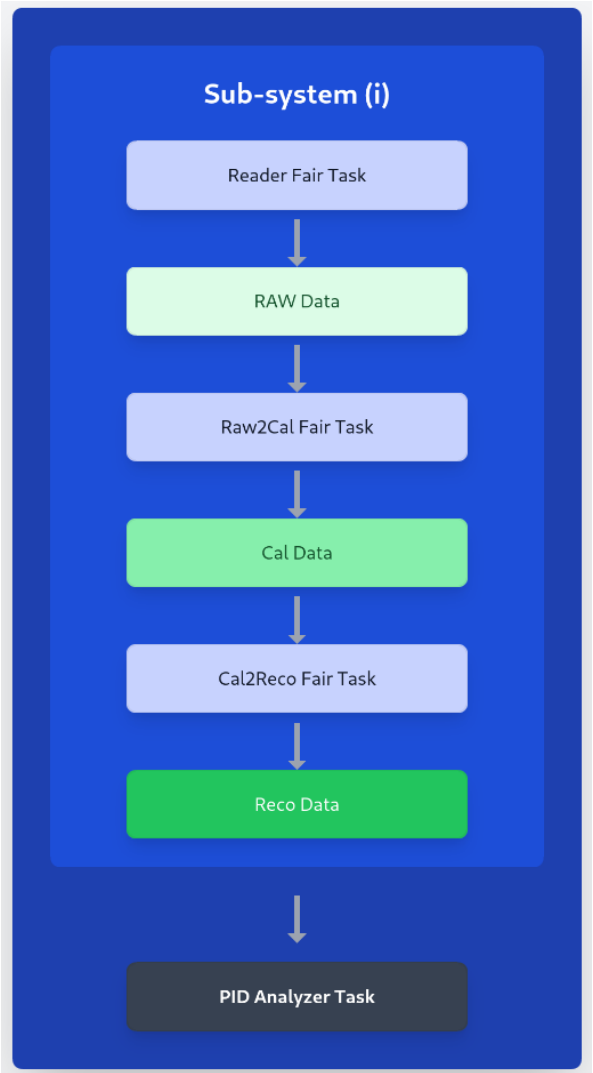
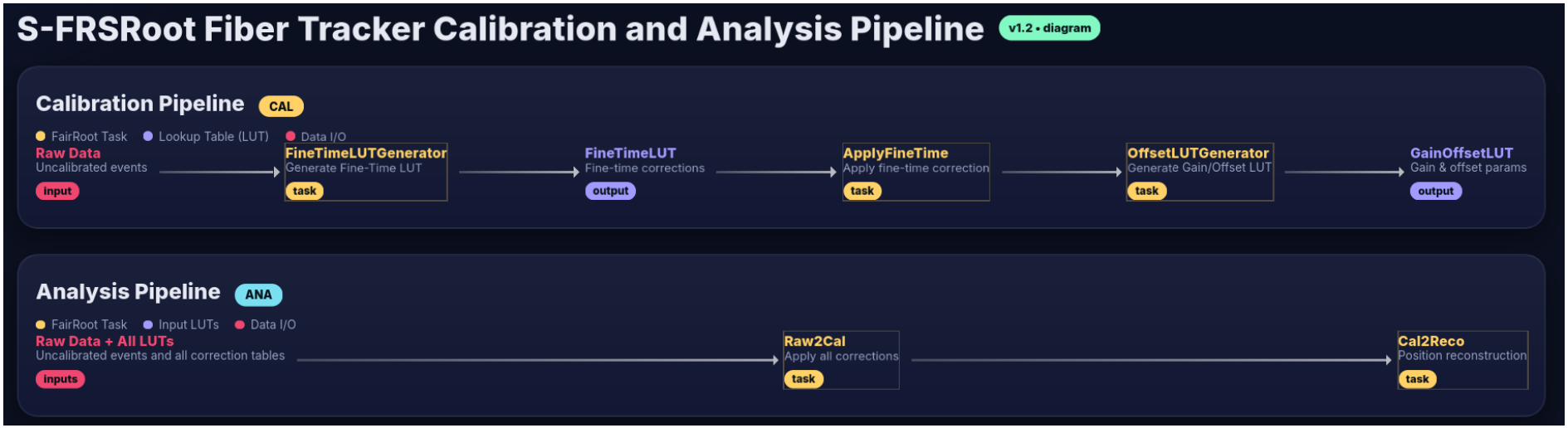
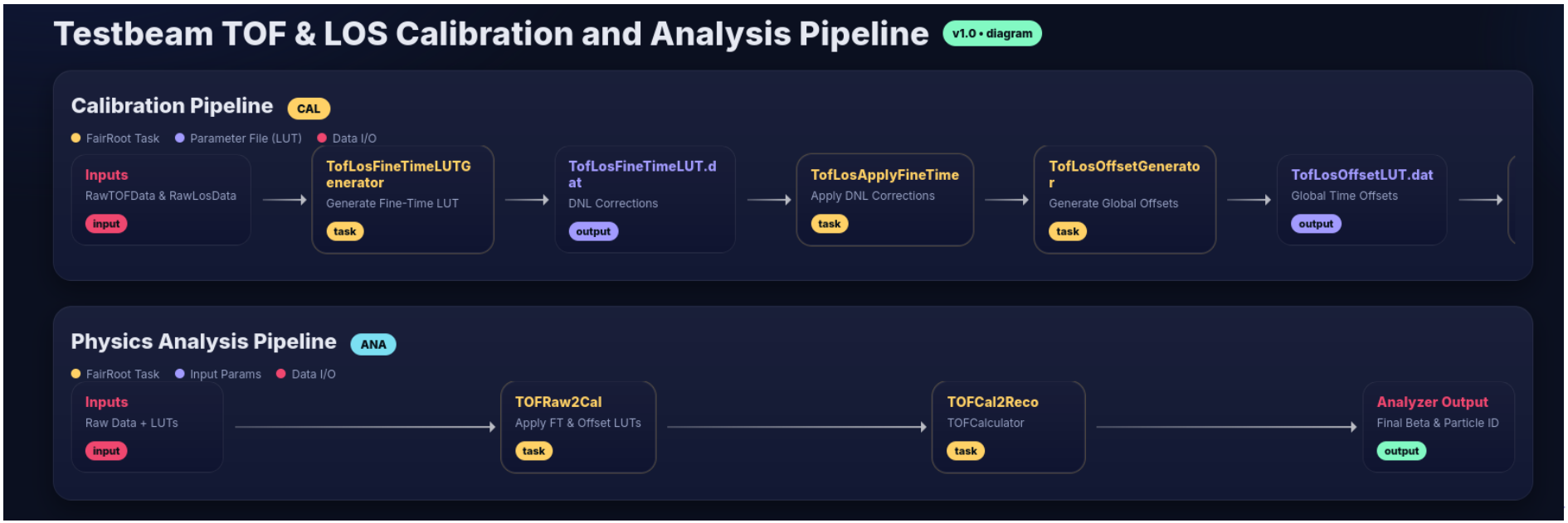


Figure: S-FRSRoot Adding Sub-system

Framework Adding Sub-systems





Correlation plots Testbeam 2025 Jul

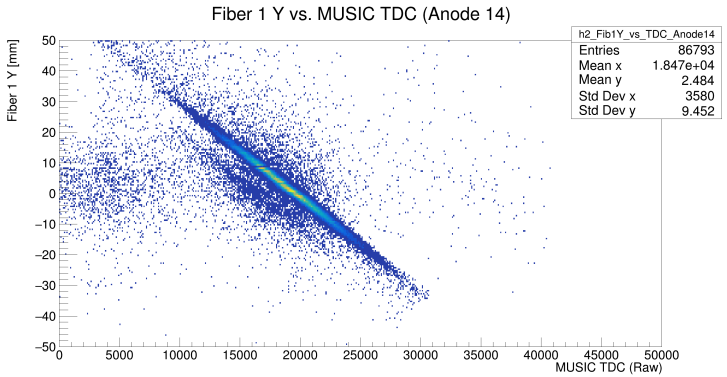


Figure: Fiber tracker vs Music

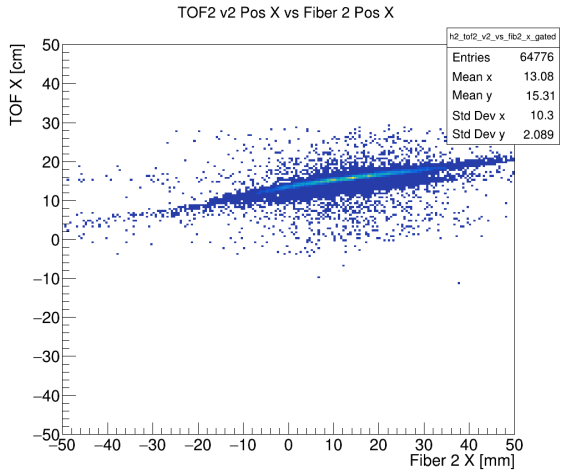
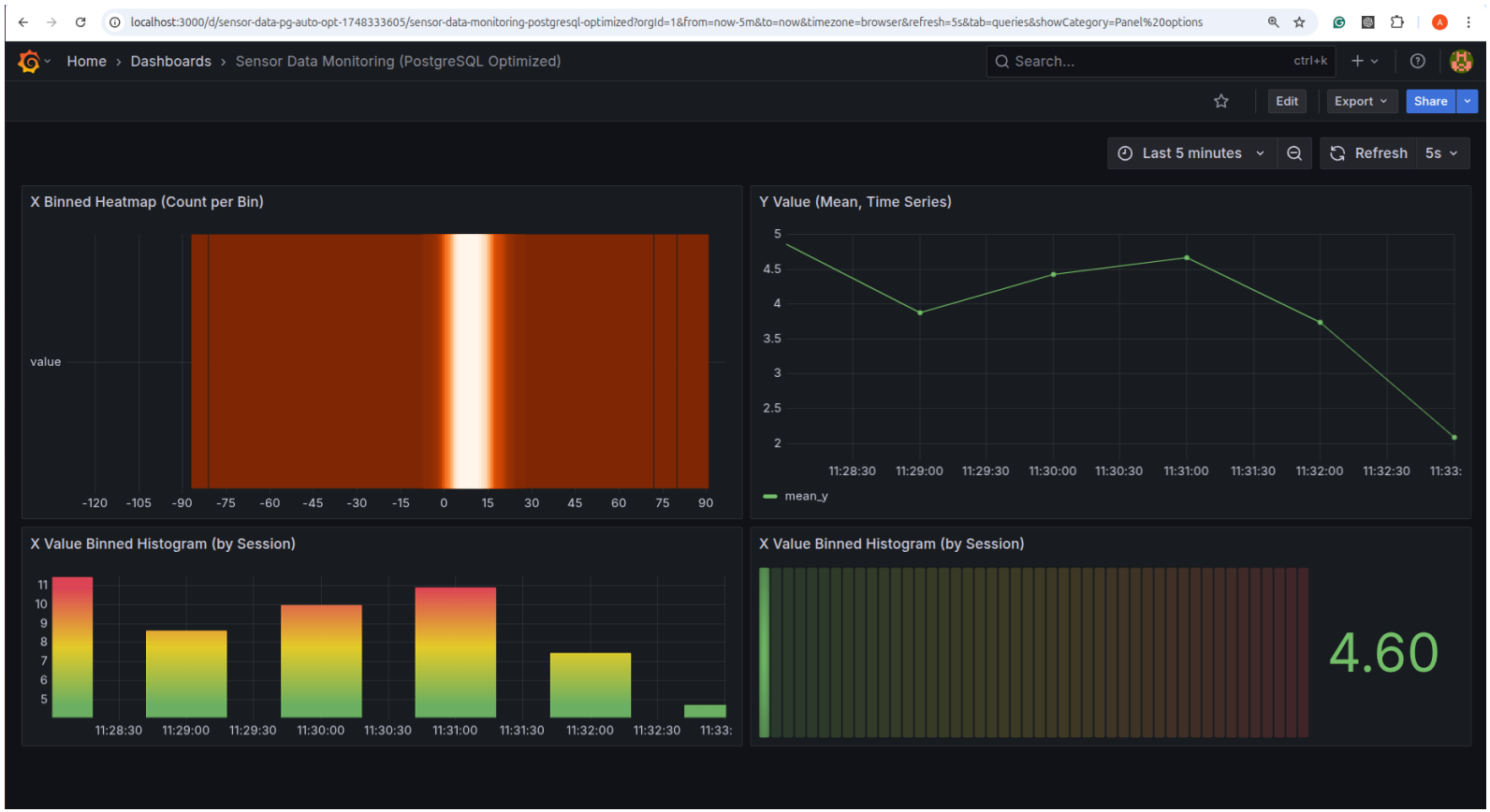


Figure: Fiber vs TOF

Successful Integration: with PostgreSQL-Grafana

Key achievements in system integration:

- The PostgreSQL-Grafana monitoring stack is now **fully integrated with S-FRSROOT**.
- The system is capable of processing and visualizing **real experimental data from the fiber detector**.



MOCADI → S-FRSROOT PID Analysis

Goal:

- Identify a $^{132}\mathrm{Sn}$ secondary beam using MOCADI simulation data within the FairRoot framework.

Input Data:

- MOCADI 4.2 data (U@1500MeV/u on C target)
- Converted to a ROOT `TTree` with an event-based structure: `EventData` objects holding `SubsystemData` from 6 SAVE points.

PID Method:

- **Z:** Taken from ΔE-like data at FHF1 (SAVE #4).
- **A/Q:** Calculated from $B\rho$ and β:

$$\beta = L/(\Delta t \cdot c) \quad \text{(ToF: FMF2} \rightarrow \text{FHF1)}$$

$$\frac{A}{Q} = \frac{B\rho}{\beta\gamma} \cdot \frac{1}{3.10715}$$

FairRoot Pipeline:

1. **SimEventSource**: Reads `EventData` from the input ROOT file.
2. **Digitizers**: Tasks for FMF2 and FHF1 extract hits from `EventData`.
3. **PID Task**: Calculates Z vs. A/Q.
4. **Output**: Histograms saved to `pid_analysis.root`.

Results & Conclusion:

- The uncalibrated PID plot (Z vs. A/Q) clearly resolves the two expected structures:
  - A Sb-like island ($Z \approx 51$)
  - A Sn-like horizontal band ($Z \approx 50$)
- The modular simulation pipeline successfully processes the event-based data and validates the simulation topology.

Gating - GUI

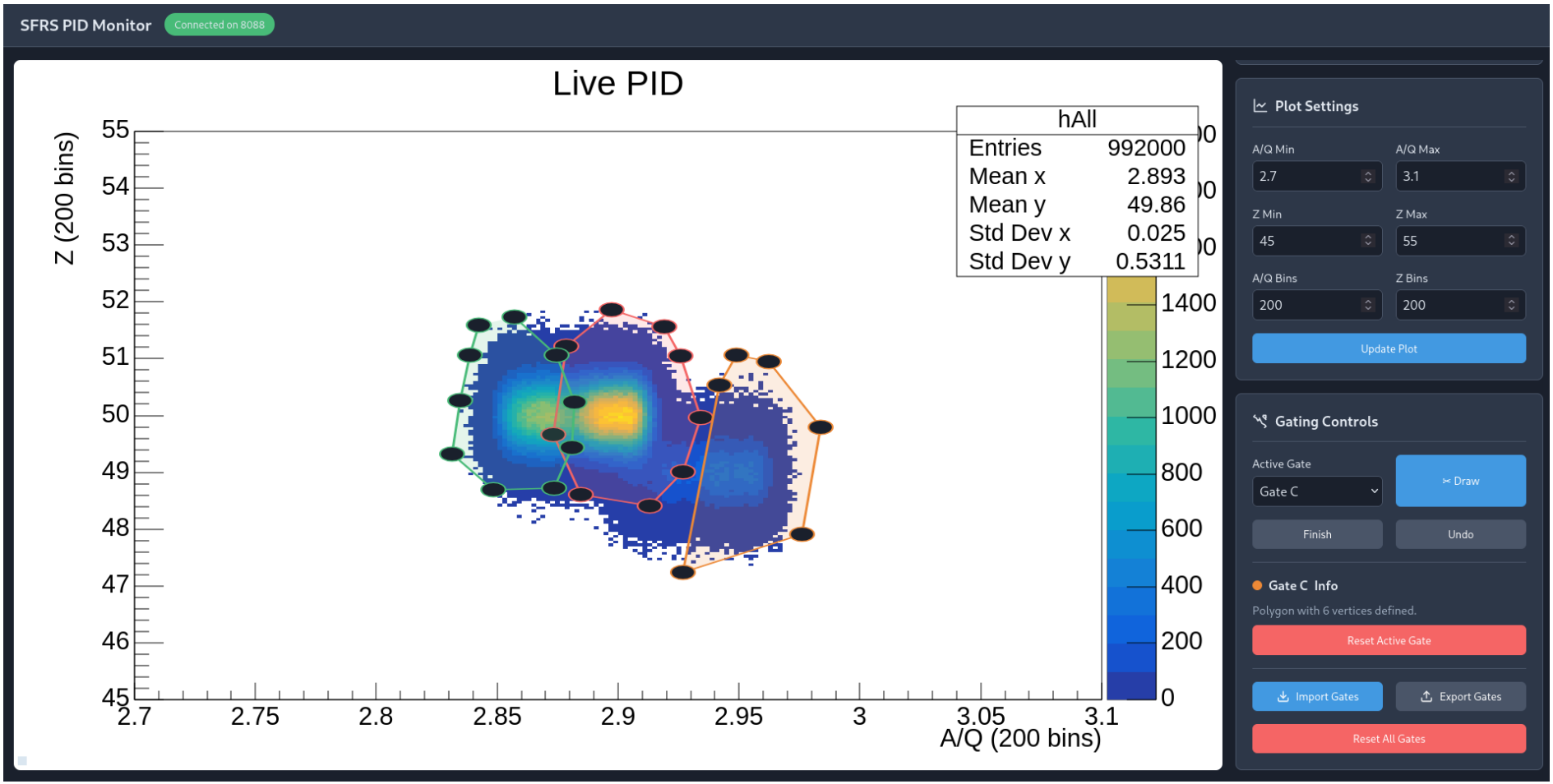


Figure: Gating GUI - Simulation Data

Gating - JSROOT

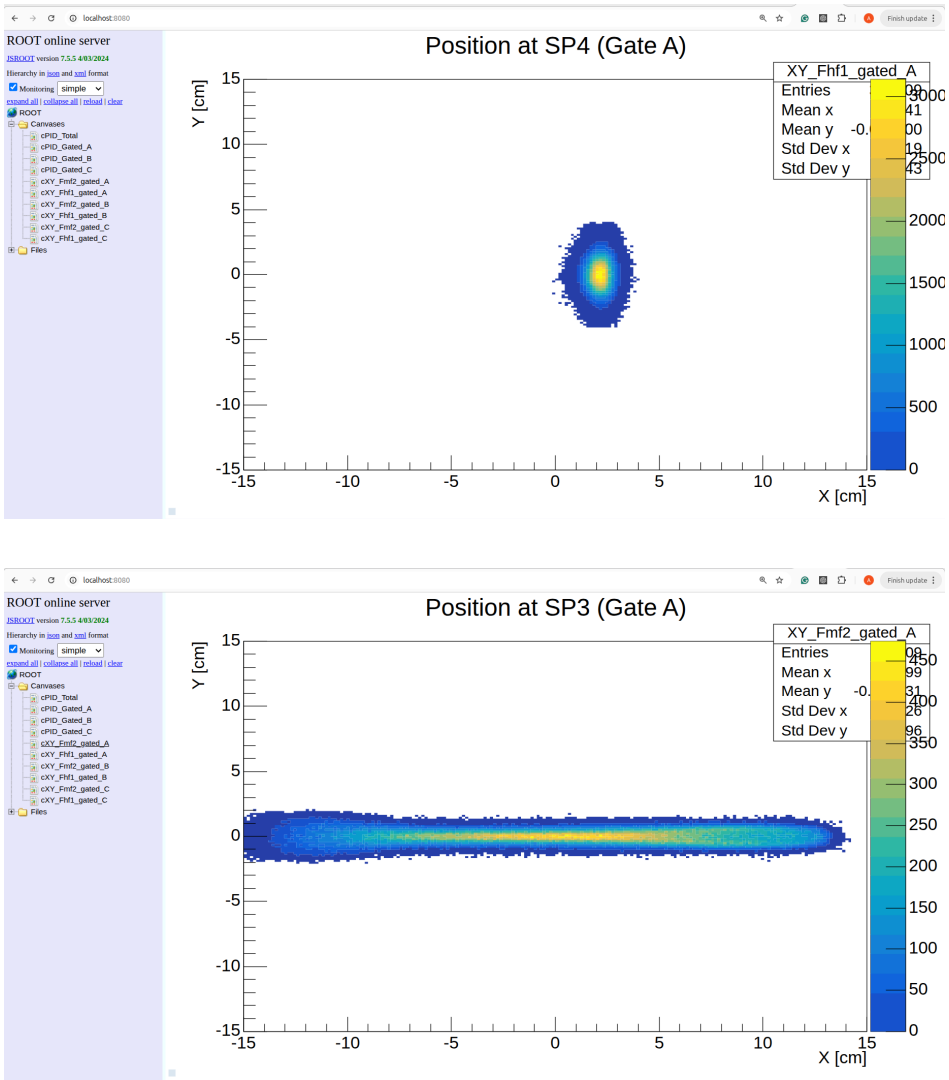


Figure: Gating JSRoot Monitoring - PID Simulation Data

Documentations

- Sub-System Documentation Example
  `https://web-docs.gsi.de/~aali/sfrs/fib/index.html`
- Alarm system Schematic Diagram
  `https://web-docs.gsi.de/~aali/sfrs/alarm.html`
- S-FRSRoot lifecycel
  `https://web-docs.gsi.de/~aali/sfrs/S-FRSROOT_lifeCycel.html`
- UCESB guide
  `https://web-docs.gsi.de/~aali/sfrs/ucesb_guide.html`
- Mocadi conversion (Event based format)
  `https://web-docs.gsi.de/~aali/sfrs/mocadiConversion.html`

Next Steps

- **Priority 1: Stress test**
  - Benchmark using FRS data
- **Priority 2: Readiness for commissioning**
  - S-FRSRoot preparation for commissioning activities from Day1

Summary and Outlook

- **Significant Progress:** Development of the S-FRSRoot online monitoring and analysis framework is advancing well.
  - Key functionalities include low-level data conversion to reconstructed hits.
  - Default visualization via JSROOT.
  - Enhanced time series plotting and interactive visualization with integrated PostgreSQL DB and Grafana.
- **Milestones:** Validation using testbeam Jul 2025, MOCADI Simulation Data Ingestion
- **Future Vision:** (1) Benchmark using FRS data , (2) S-FRSRoot preparation for Commissioning activities from Day1

Questions?

Thank you for your attention.