Development of software tools for optimizing the operation of gaseous detectors and gas systems

Andrés Navarro Pedregal
contact@andresnav.com

EP-DT-FS Gas Team
Supervisors: Gianluca Rigoletti & Pieter Vanslambrouck

August 20, 2024

CERN | EP-DT
Detector Technologies

1. About Me & EP-DT-FS Gas Team

2. Motivation

3. Project

4. Work

5. Conclusions & Future Work

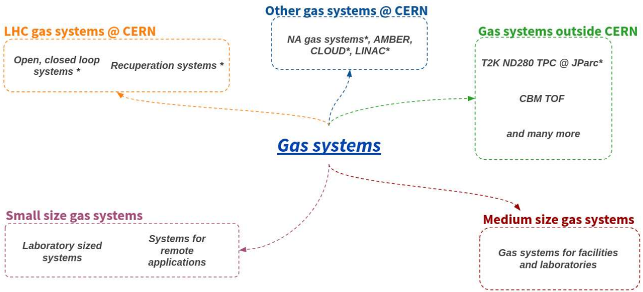
About Me

- **Education:** University Carlos III of Madrid (Sep. 2020 - Feb. 2025)
- **Study Program:** Dual Bachelor in Data Science & Telecommunication Engineering
- **Research:** Telematics Department at UC3M (Jul. 2023 - Present) Developing Autonomous Drone Swarms for Human Rescue
- **Experience:** DevOps, ML & AI, Software Infrastructure, and Microsoldering.

More Info
https://linkedin.com/in/andresnav
https://andresnav.com

About EP-DT-FS Gas Team

- Maintenance and operation of more than 30 gas systems.
- R&D on recuperation systems.
- **R&D for ECO-friendly gas mixtures**.
- A really great team to work on!



* Developed by CERN EP-DT

1. About Me & EP-DT-FS Gas Team
2. **Motivation**
3. Project
4. Work
5. Conclusions & Future Work

RPC gas mixtures at LHC

- Main components: *R*-$134a$, *i*-C_4H_{10}, SF_6.
- RPCs account for **85% of the GHG emission** from particle detectors at CERN.
- 95% of the GWP due to *R*-$134a$, 5% due to SF_6.

In Run 2, **100 000 tCO2e/year** emitted from *R*-$134a$ consumption alone.

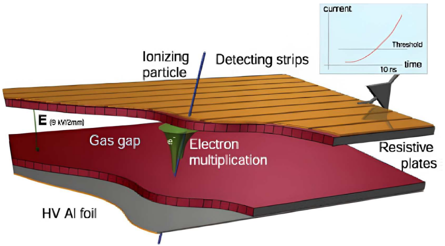


Diagram of a Single Layer RPC

Acronyms:
GHG: Greenhouse Gas
GWP: Global Warming Potential

RPC gas mixtures at LHC

- Main components: $R\text{-}134a$, $i\text{-}C_4H_{10}$, SF_6.
- RPCs account for **85% of the GHG emission** from particle detectors at CERN.
- 95% of the GWP due to $R\text{-}134a$, 5% due to SF_6.

In Run 2, **100 000 tCO2e/year** emitted from $R\text{-}134a$ consumption alone.

30% of the emissions of Andorra.

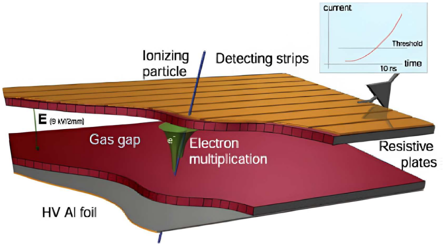


Diagram of a Single Layer RPC

Acronyms:
GHG: Greenhouse Gas
GWP: Global Warming Potential

1. About Me & EP-DT-FS Gas Team

2. Motivation

3. Project

4. Work

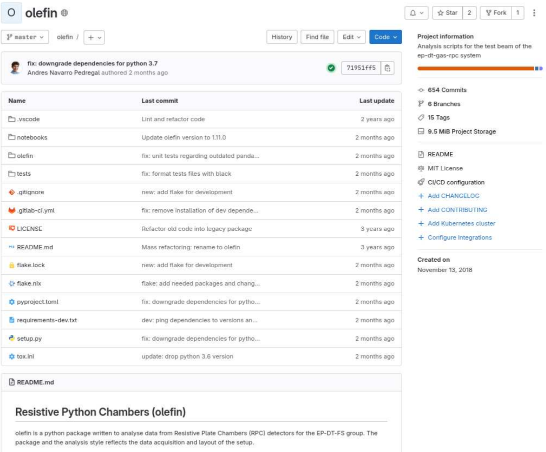
5. Conclusions & Future Work

Olefin

EP-DT
Detector Technologies

Data analysis open-source tool for RPC study.

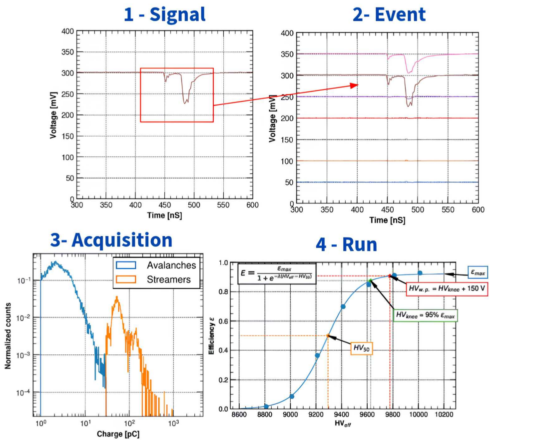
- Developed in **Python with Pandas and Numpy** for ease of use.
- Flexible and modular for high adaptability.
- Customizable and performant.



https://gitlab.cern.ch/gasteam/olefin

Architecture

1. **Signal:**
   Single waveform →baseline, charge, height, peak times.
2. **Event:**
   N waveforms from same RPC and trigger event →cluster size, event charge, signal classification.
3. **Acquisition:**
   N. triggers →efficiency, st. prob., mean prompt charge, rate.
4. **Run:**
   Set of acquisitions →working point, rate, currents.

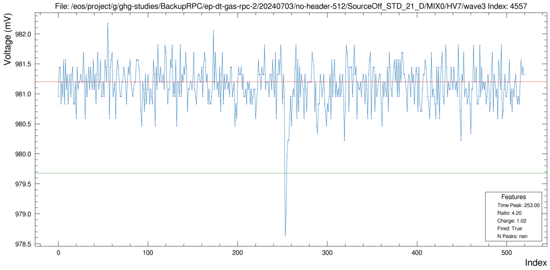


Overview of Olefin architecture

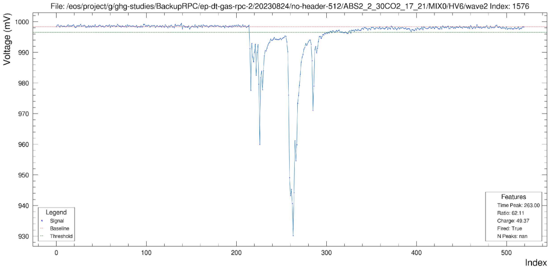
STUDIES OF NEW GAS MIXTURES

With the study of new gas ECO-friendly mixtures, there are new challenges in the analysis:

- The actual implementation only gives a rough overview of the run.
- No ability to study **complex characteristics of the signals** such as the **after peaks**.



Signal with no disturbances



Signal with disturbances

Objectives

1. Research on the **backend storage form** of Olefin to support new features.
2. Understanding the **influence of background** radiation on the efficiency.
3. Study the addition of **new features** based on the shape and complexity of the signals.

1. About Me & EP-DT-FS Gas Team

2. Motivation

3. Project

4. **Work**

5. Conclusions & Future Work

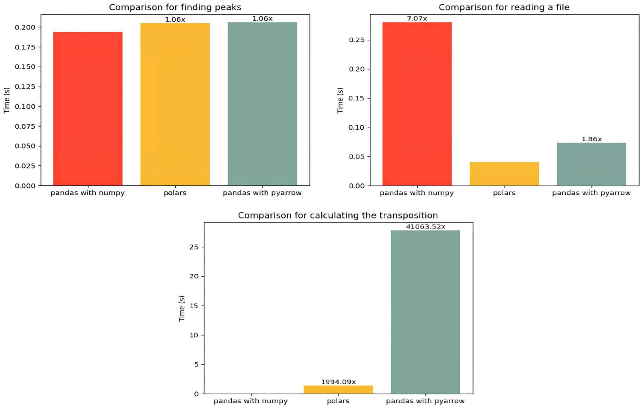
Backend Analysis

Current implementation → Pandas with Numpy.
Requirement → **Lists inside columns to add new features.**
Candidates → Pandas with Numpy (upgraded), Pandas with PyArrow, ROOT, and Polars.

Pandas with Numpy (upgraded)	Pandas with PyArrow	ROOT	Polars
Advantages **Easy to use and already implemented.** Strong support for data analysis. Well-suited for in-memory operations.	High performance for columnar data. Native support of Pandas. **Support for lists inside columns natively.**	Extensive features for high-energy physics. Efficient handling of complex data structures. **Large support at CERN.**	High performance for columnar data. Multiprocessing built-in. **Support for lists inside columns natively.**
Disadvantages **Not performant for very large datasets.** Limited to in-memory operations.	**Makeover of the current implementation.** Less native support for data analysis libraries. Primarily focused on data storage and transfer.	Makeover of the current implementation. Niche implementation for CERN. **Does not support lists inside columns.**	**Makeover of the current implementation.**

Comparison of different backends

Backend Analysis



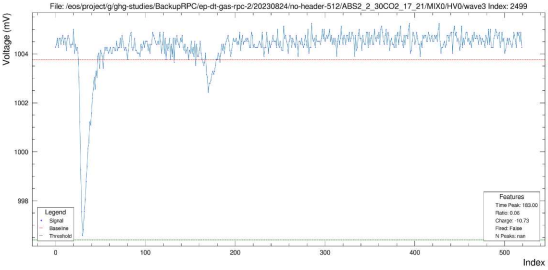
Performance tests for different backends

- No noticeable difference in performance when finding peaks.
- Trade-offs on switching backends are not worth it.

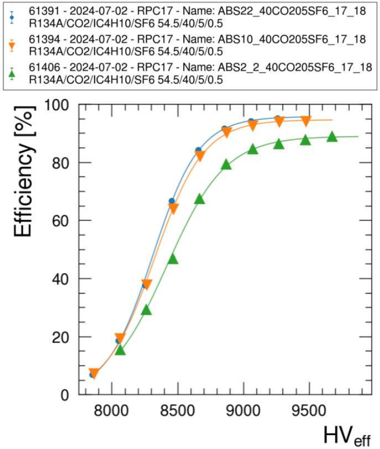
Decision: **Stick to Pandas with Numpy and upgrade it.**

Baseline Estimation

In the presence of background radiation, **noise signals appear in the baseline region**.



Underestimated features due to noise signals in the baseline region



Efficiency plateau problem
https://rpc-data-visualization.app.cern.ch/?ids=61394ids=61391ids=61406

Baseline Estimation

- Current implementation → Mean over a start-up window.
- Requirement → **Understand if the underestimation of the features is due to the Olefin framework or due to the physical nature**.
- Candidates → Full Mean, Savitzky-Golay filter, Mode, Median, Two Side Mean.

Method	Efficiency increase w.r.t mean	Time increase w.r.t mean	Problems
Mean baseline region			Affected by peaks in the baseline region
Mode baseline region	+0,18%	511x	Slow
Mean Full Signal	+1%	1.46x	More robust but takes much longer
Savitzky-Golay Filter	0%	54x	No performance increase
Two Side Mean	+1%	2x	Accurate only if peaks do not occur in the extremes

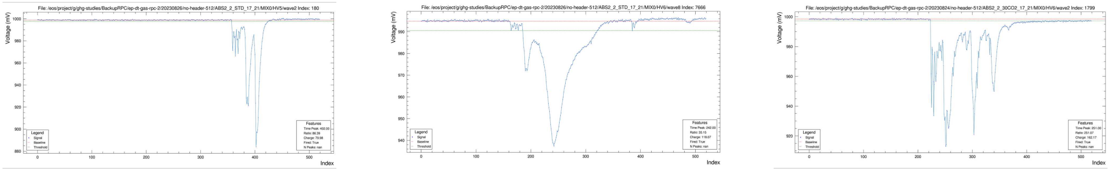
Comparison of different methods for baseline estimation

In conclusion, **no method provides an improvement** with similar performance.

Peak analysis

With the new ECO-friendly gas mixtures, the need to **study the presence of after pulses** was crucial.

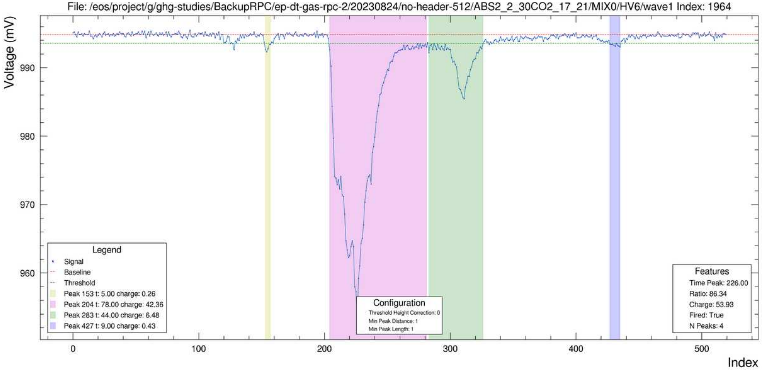
However, we wanted to do it in an efficient and robust manner.



Examples of different signals for peak identification

PEAK ANALYSIS: FIRST APPROACH

1. Calculate threshold.
2. Group samples under it.
3. Remove signals not interested.

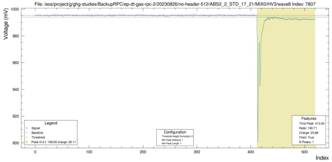


Example of peak analysis

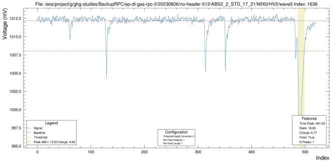
First Approach: Results

- Performance: 1.3x slower than normal analysis.
- Able to detect “simple peaks” (without the problems outlined below) with good accuracy.

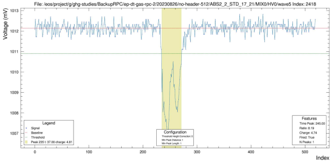
Problems



Signal not recovered



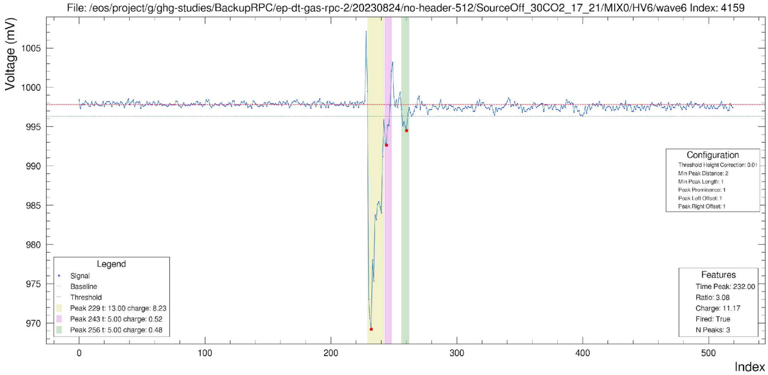
Single-point peaks not detected



Multiple peaks over the threshold

Peak analysis: Second Approach

1. Do the previous steps.
2. Detect peaks with scipy `find_peaks` function.
3. Clean and establish new peaks.
4. Calculate features of detected peaks.



Example of peak analysis with the second approach

Second Approach: Results

EP-DT
Detector Technologies

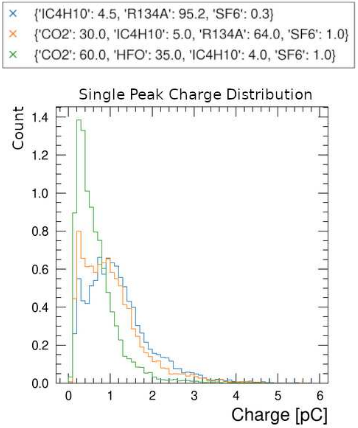
- **New insights** $\rightarrow$ further studies must be perform to understand the new ECO-friendly gas mixtures.
- Performance $\rightarrow$ **1.77x slower** than normal analysis (acceptable from 40s to 71s).
- **Parameters highly influence the results** $\rightarrow$ need to be tuned.
- Added as an extension to the `Olefin` library.

Insights: Single Peak Charge Distribution

EP-DT
Detector Technologies

The distribution of single peak charges for different amounts of CO_2.

For high amounts of CO_2 (60%), the single peak charge is smaller.

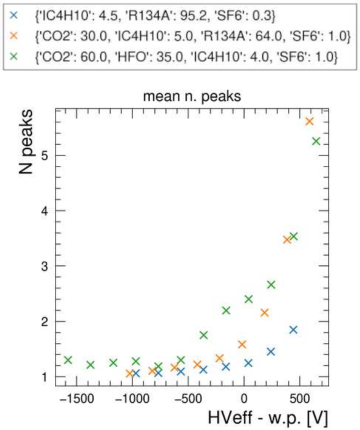


Single peak charge distribution of different levels of CO_2 gas mixtures

Insights: Mean Number of Peaks

The distribution of the mean number of peaks for different amounts of CO_2.

Around the working point, for **high amounts of CO_2 (60%), there average number of peaks increases.**

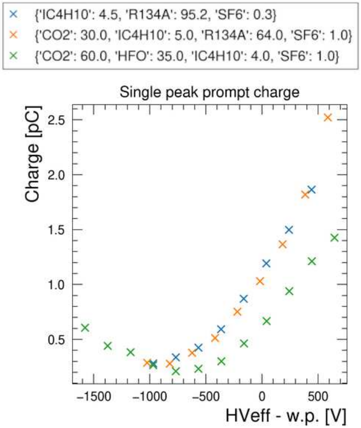


Mean number of peaks of different levels of CO_2 gas mixtures

Insights: Single Peak Prompt Charge

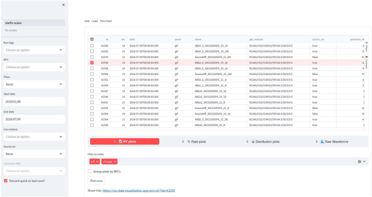
The single peak prompt charge for different amounts of CO_2.

Around the working point, for **high amounts of CO_2 (60%), the charge of the single peaks diminish.**

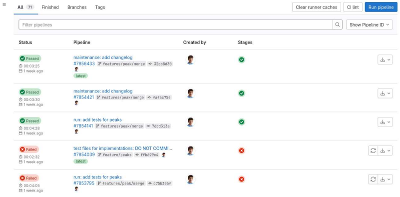


Single peak prompt charge of different levels of CO_2 gas mixtures

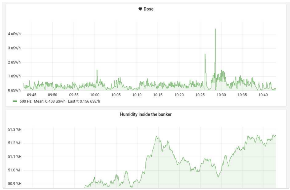
Other projects



Visualization tools for RPC data



CI/CD pipelines for Olefin



Grafana dashboard for gas system monitoring

Augmented the current visualization tool for RPC data.

https://rpc-data-visualization.app.cern.ch/

Implemented CI/CD pipelines for the Olefin project.

https://gitlab.cern.ch/gasteam/olefin

Software for continuous monitoring of dosimeters.

https://gitlab.cern.ch/gasteam/dosimeter-data-retreival

1. About Me & EP-DT-FS Gas Team

2. Motivation

3. Project

4. Work

5. **Conclusions & Future Work**

Conclusions

EP-DT
Detector Technologies

- **Backend Analysis:** **The use of Pandas with Numpy remains the most practical choice**, balancing performance and feature support. ROOT and Arrow are promising but require significant changes and have steeper learning curves.
- **Baseline Estimation:** **No alternative method surpasses the mean baseline** in both performance and accuracy.
- **Peak Analysis:** New implementation with **peak detection and characterization**.

Future Work

- Optimize peak analysis parameters:
  - **Calibrate the parameters** of the peak analysis methods to improve accuracy and efficiency.
  - Explore **adaptive parameter tuning techniques** based on different signal characteristics.

- Case study on recent test beam data:
  - Apply the peak analysis approach to the latest test beam data to **evaluate the performance in a real-world scenario**.

- Enhance visualization tools:
  - **Expand the capabilities of the RPC data visualization tools** to enhance the study of long-term tests.

Acknowledgements

Thanks to my supervisors, **Gianluca Rigoletti** and **Pieter Vanslambrouck**, and everyone in the **EP-DT-FS Gas Team** for all the help and support.

Thank you!

Andrés Navarro Pedregal