Field response check

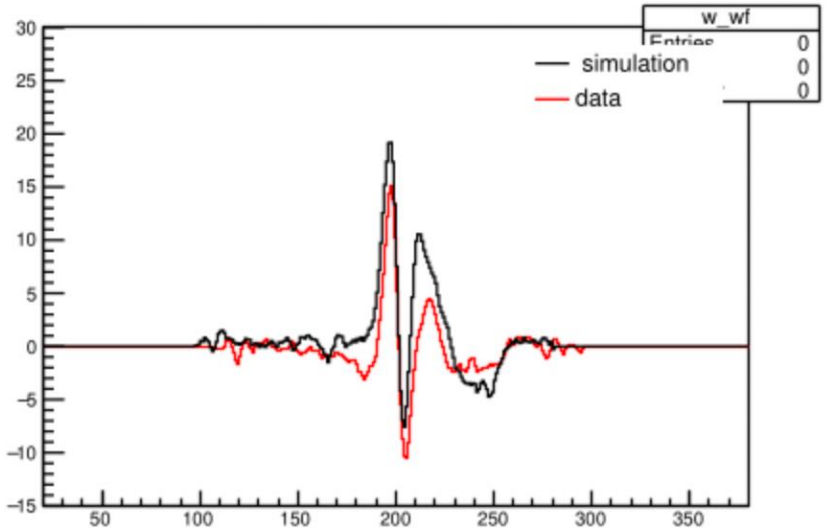
Xuyang Ning & Wenqiang Gu

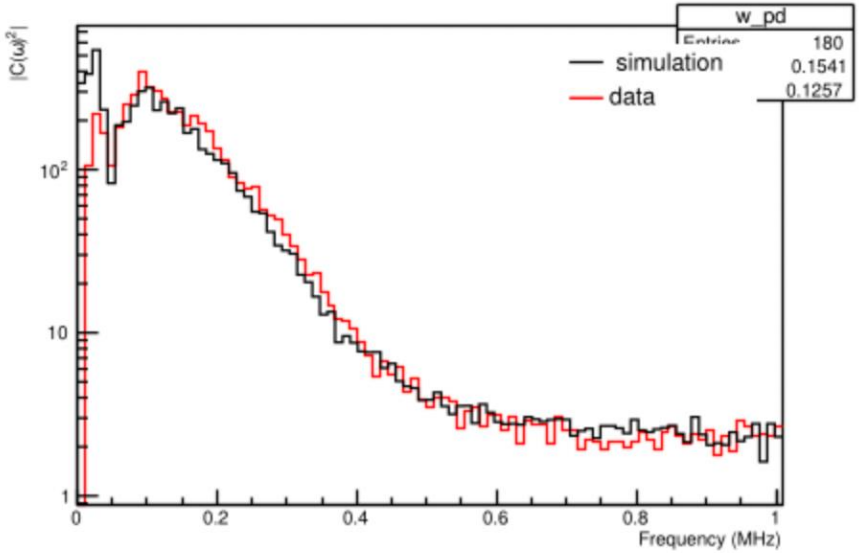
02/12/2025

Brookhaven National Laboratory

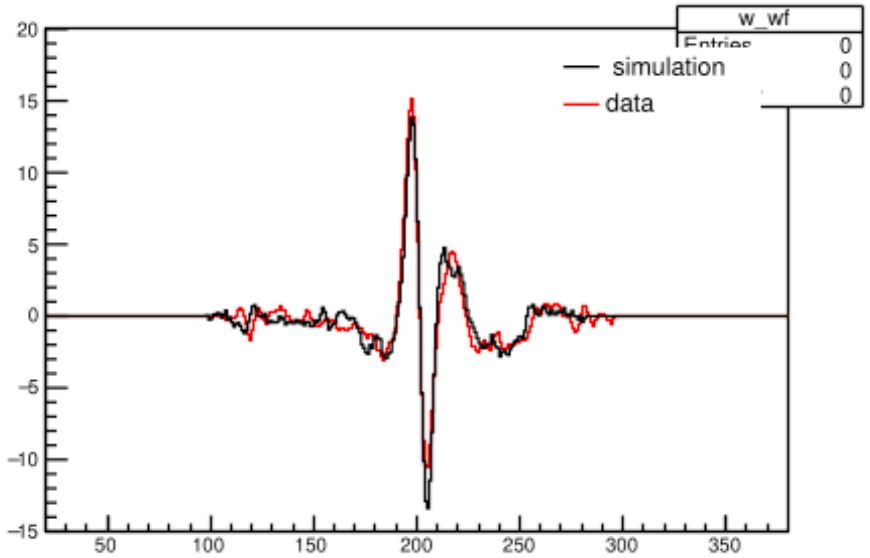
Simulation w/ and w/o CNR

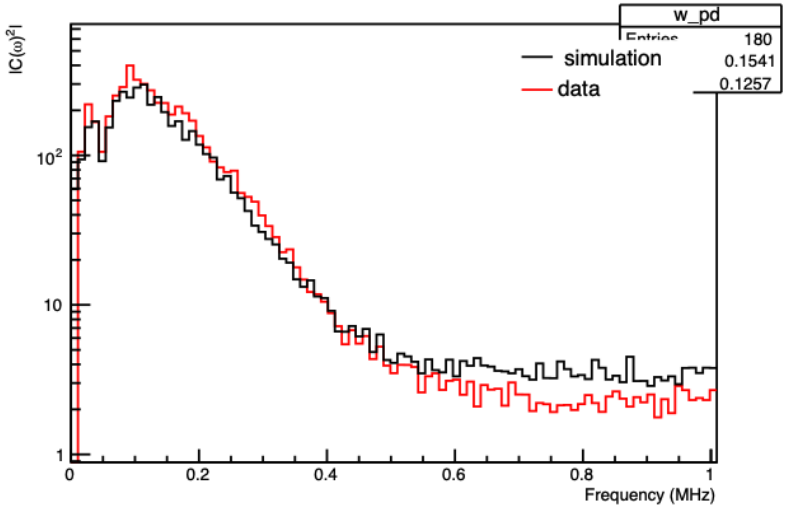
Coherent noise removal not applied





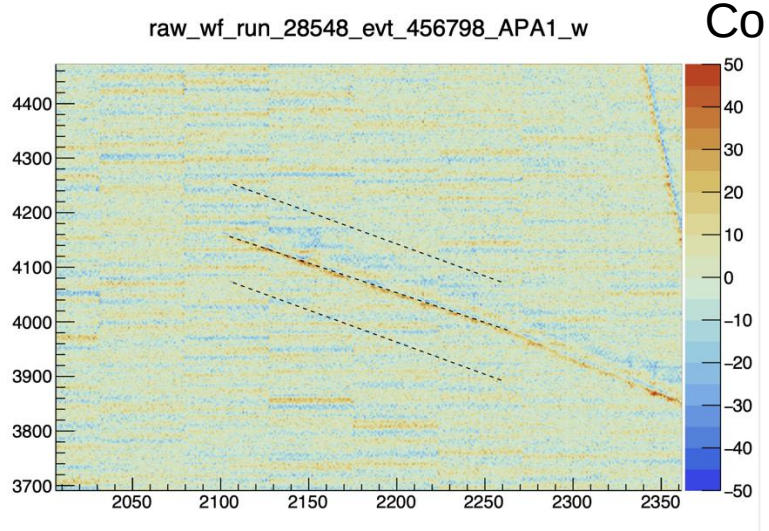
Coherent noise removal applied

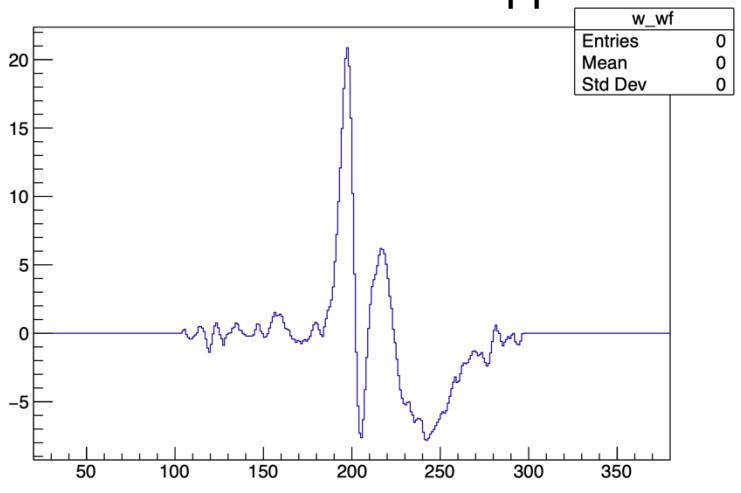


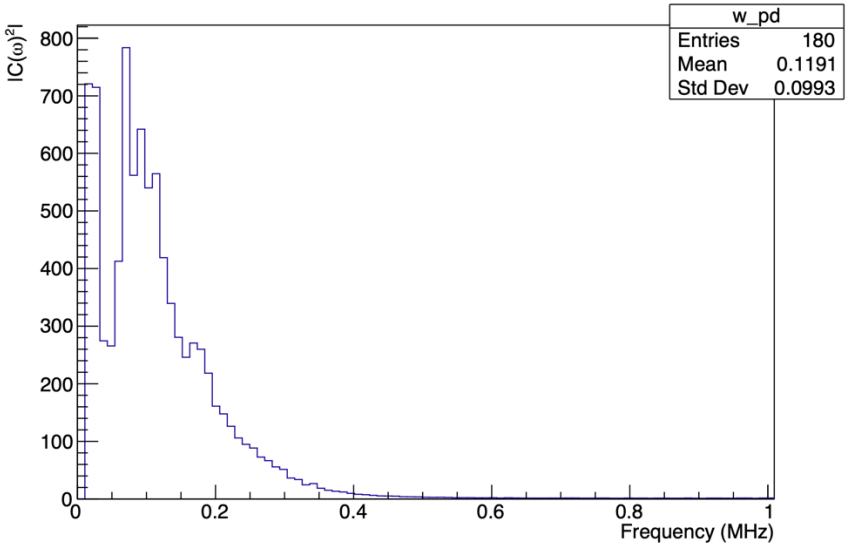


Data w/ and w/o CNR

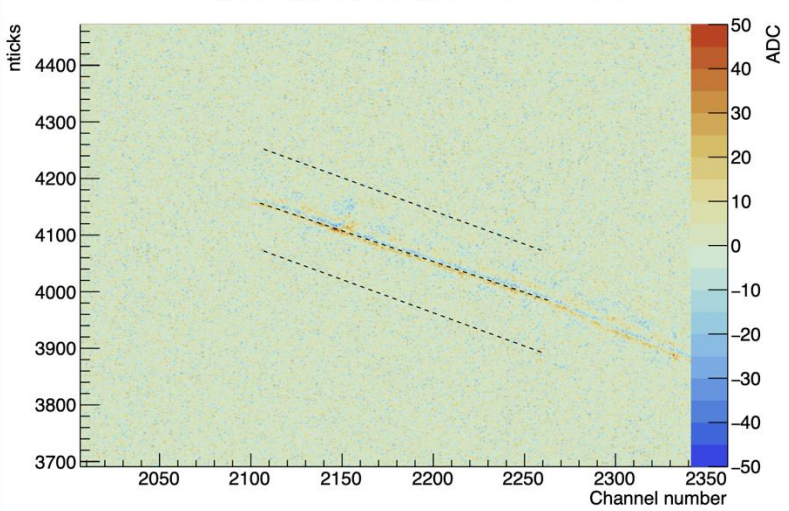
Coherent noise removal not applied

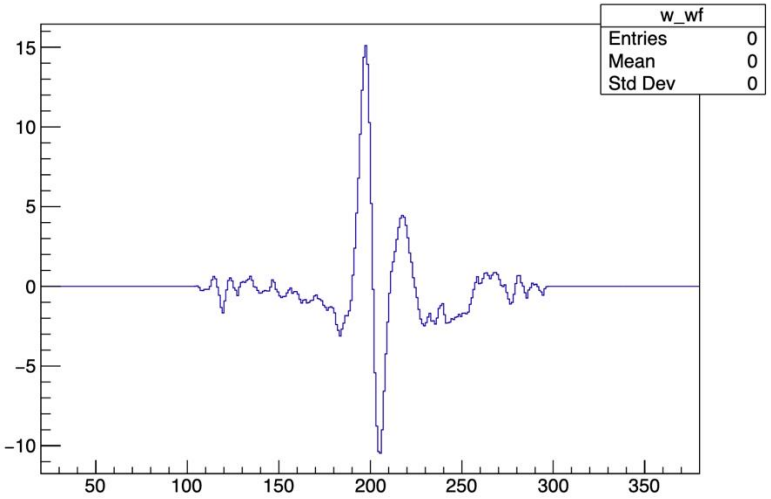


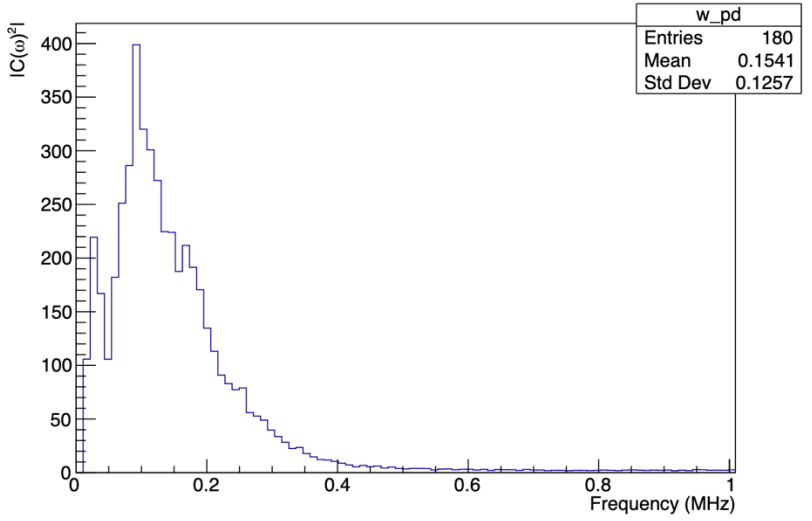




Coherent noise removal applied

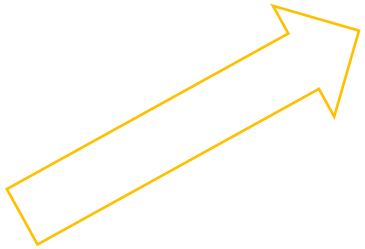






Redo everything with no CNR

- 5 Parameters
  - Different renormalizing factors to 0.4 pitch (f_5) and 0.5 pitch (f_6)
  - One starting point for the 2 electron paths
  - Two time stretch factors for 2 electron paths



- A python script to control everything:
  - Minimize Chi2
  - MCMC

Generate renormalized field response from wire-cell-python

Simulation using a wire-cell-tookit

Compare data with simulation in a root script:

Compute X^2_{total}

Install wire-cell-python on gpvm

- For wire-cell-python:
- Previously I couldn’t install it on gpvm.

- Decouple everything about gojsonnet
- Modify to be suitable for Python version 3.9 (With the help of ChatGPT)

- Successfully install it on gpvm and inside the container.

wire-cell-python/wirecell/sigproc/response/schema.py

2 Modify Code for Python 3.9 Compatibility

If you cannot upgrade Python, you need to **edit** `schema.py`:

1. Open the file:

bash — Copy

```
nano /exp/dune/data/users/xning/git/wire-cell-python/wirecell/sigproc/response/schema.py
```

2. Find the problematic line:

python — Copy

```
paths: List[PathResponse] | None = None
```

3. Change it to:

python — Copy

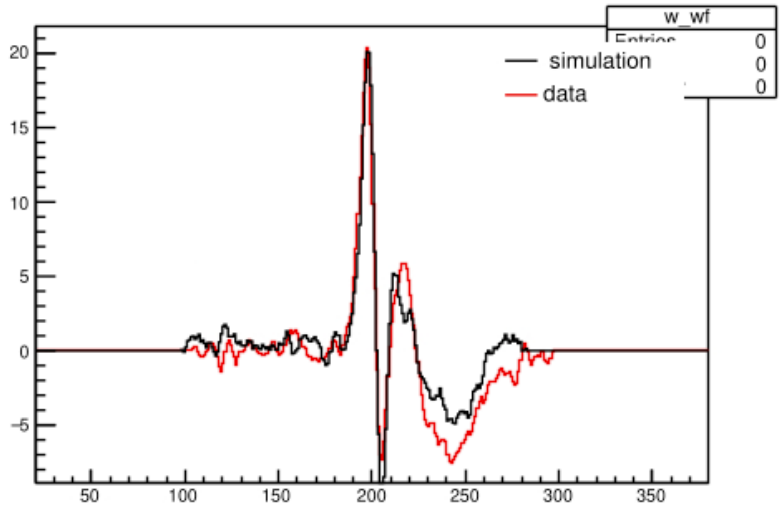
```
from typing import List, Optional
paths: Optional[List[PathResponse]] = None
```

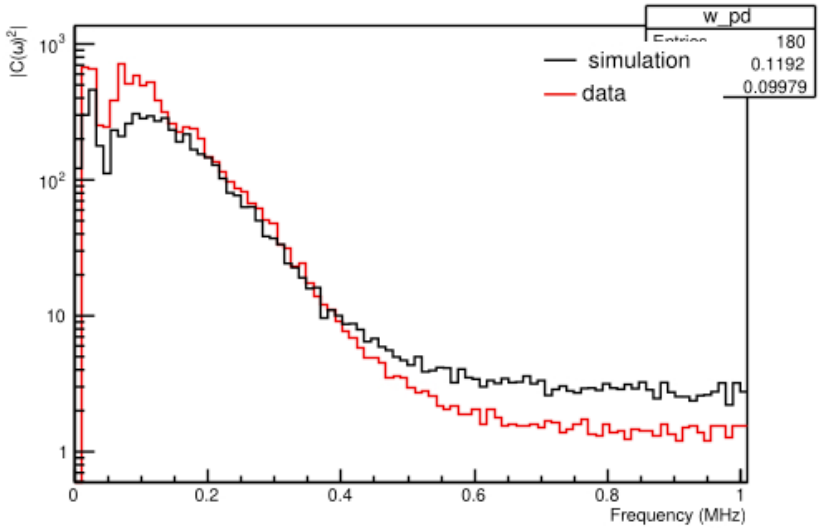
Do fitting using MCMC

- Define X^2_{total} :
  - For w plane, compare waveforms and power density spectra
  - For each plot:

$$X^2 = \sum_{i=1}^{nbins} \frac{(x_i - m_i)^2}{|m_i|} \; /nbins$$

  - X^2_{total}: For different plots in different tracks, multiply all the X^2
    - Only use 7 tracks, because the last one is strange





Do fitting using MCMC

- A very simple MCMC with the help of **OpenAI 4o**:
  - Since the chi2 is dependent on simulation, the same parameters cannot produce same result all the time.
  - MCMC only relies on the result of the last step.
  - It is easy to deal with lots of parameters.

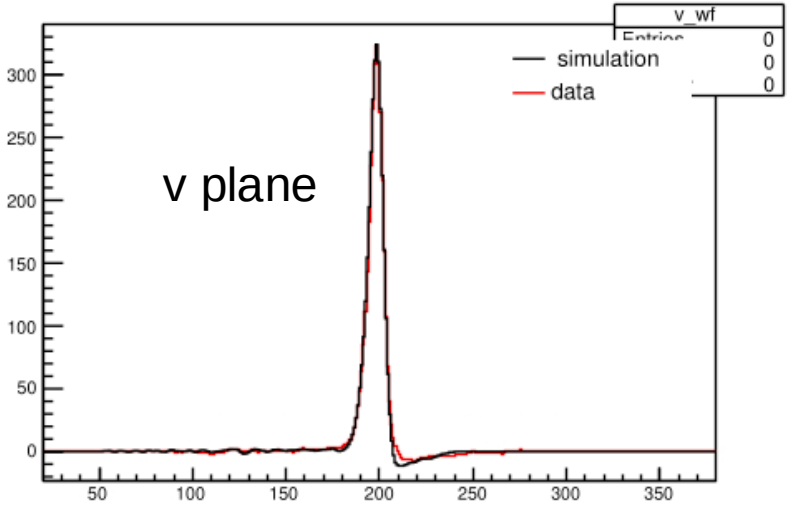
- Random walk: a new parameter set is randomly generated according to the current parameters.
  - The step size is fixed.

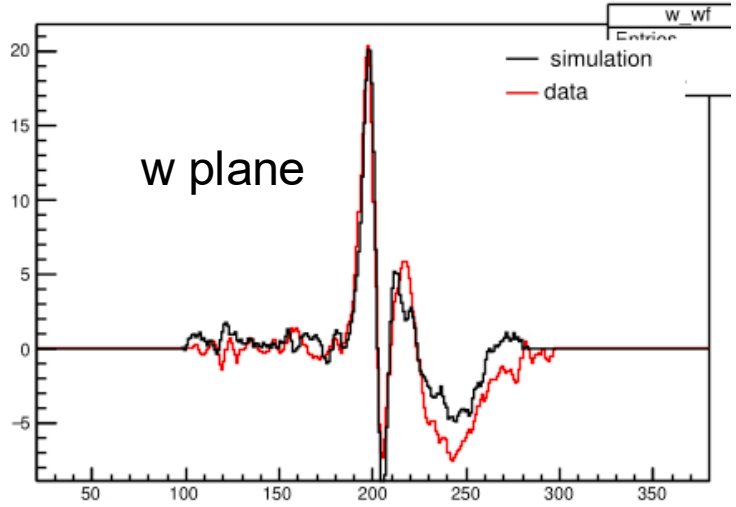
- If $X^2_{total}(current\ parameters) > X^2_{total}(new\ parameters)$
  - Accept new parameters;
- Else
  - $new\ parameters$ still has some chance to be accepted.
    - Metropolis-Hastings acceptance rule.

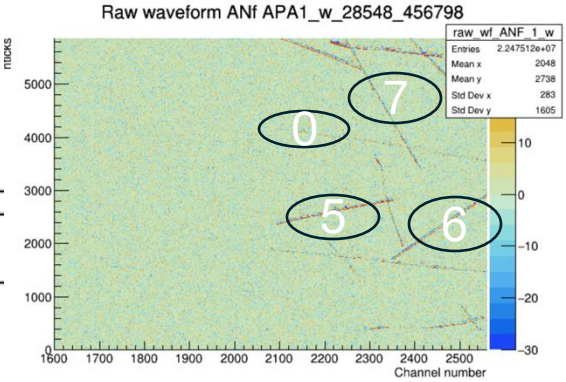
Best one this time

Track0; beam

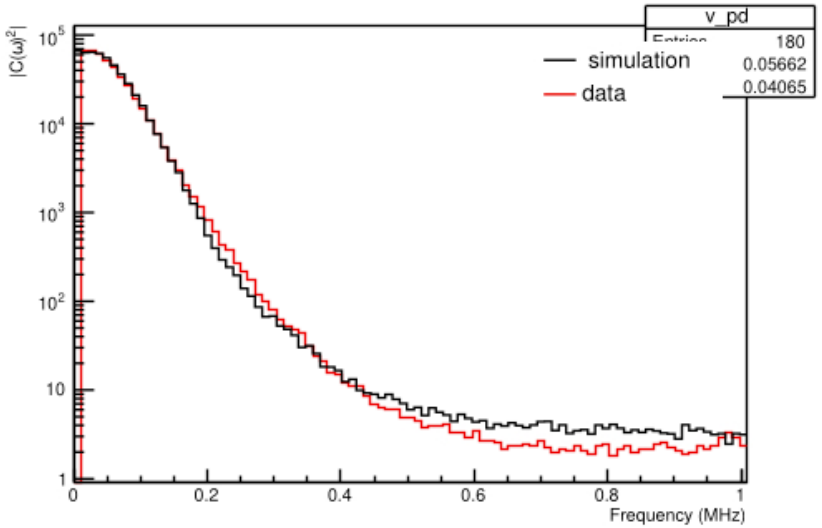
waveform

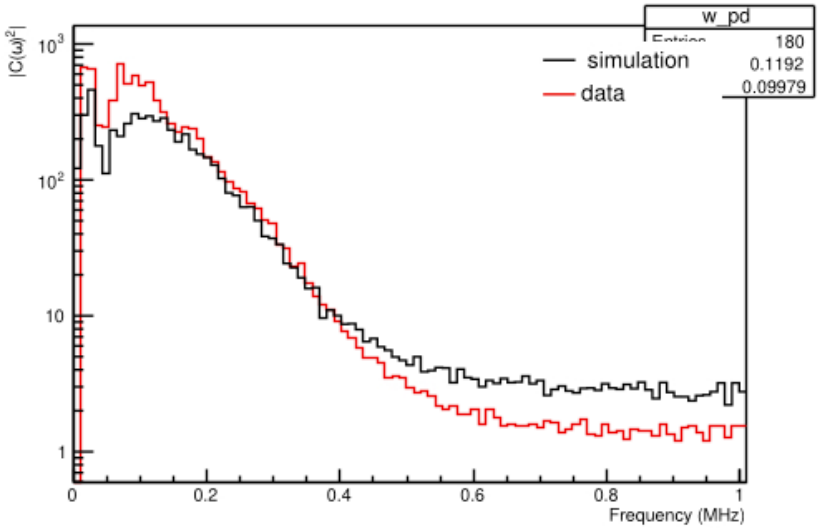






Power spectra

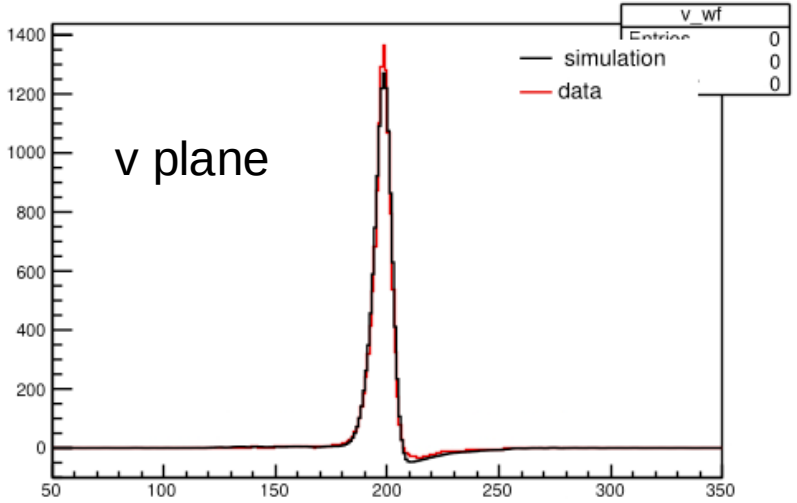


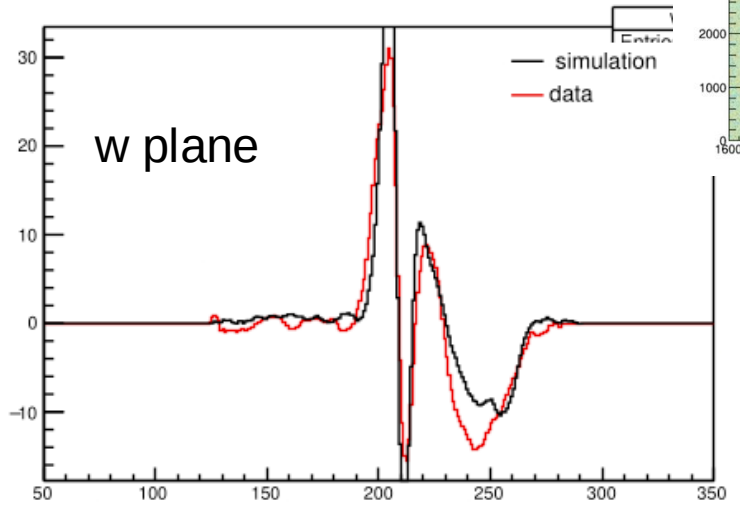


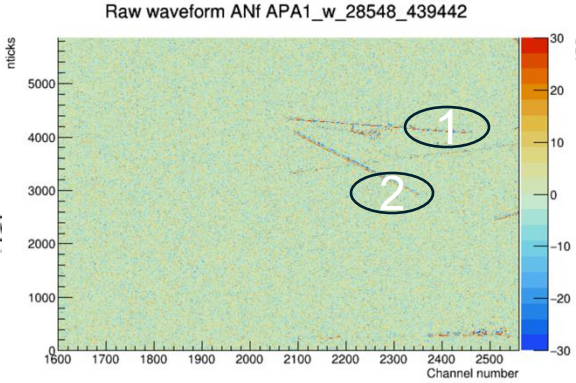
-0.04_2.10_1.00_2.50_650

Best one this time

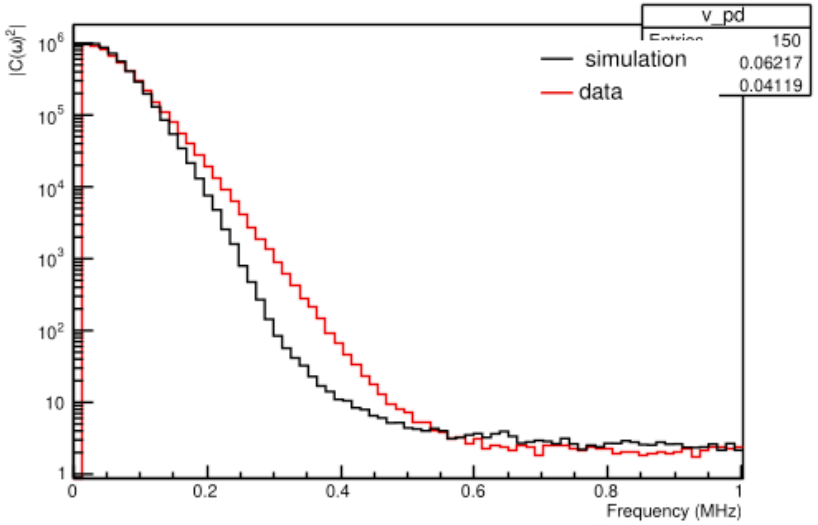
waveform

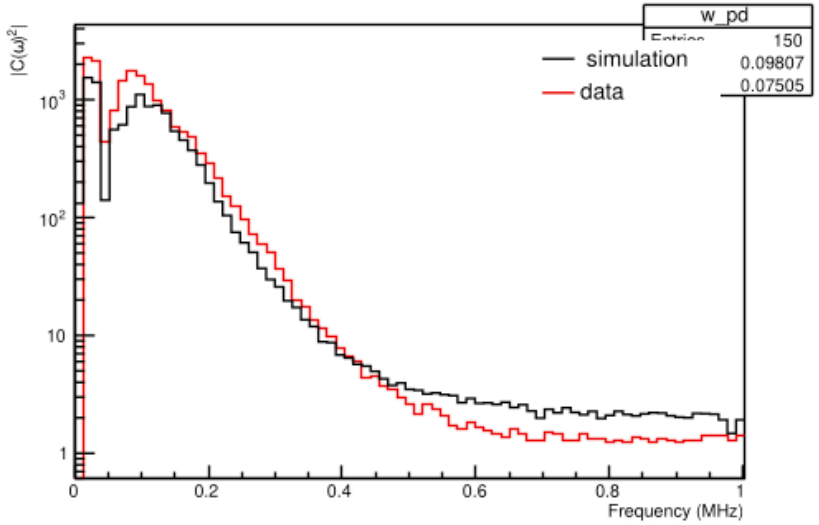






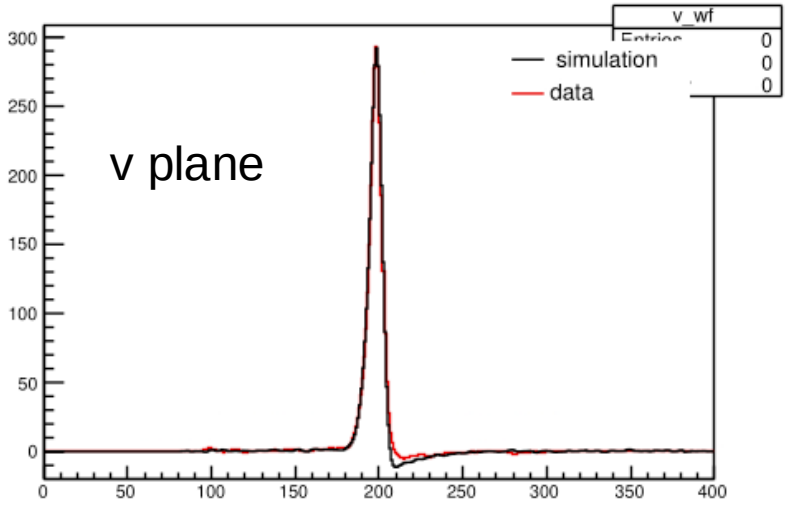
Power spectra

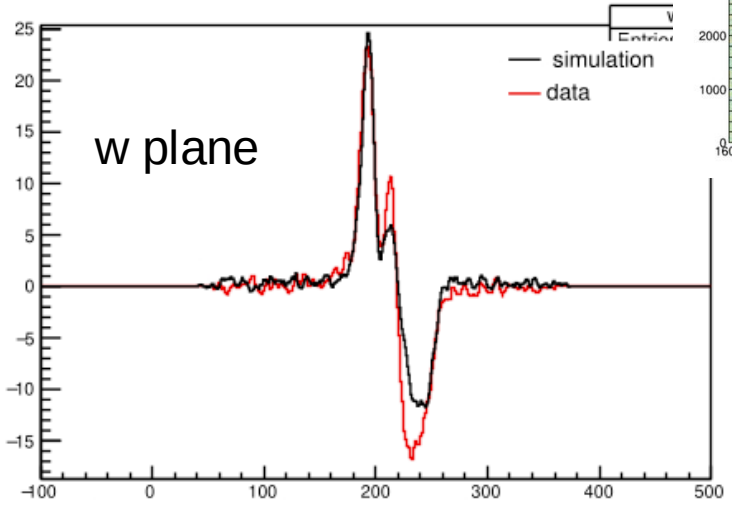


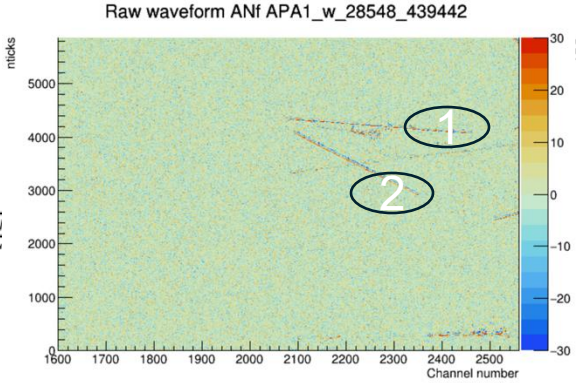


Best one this time

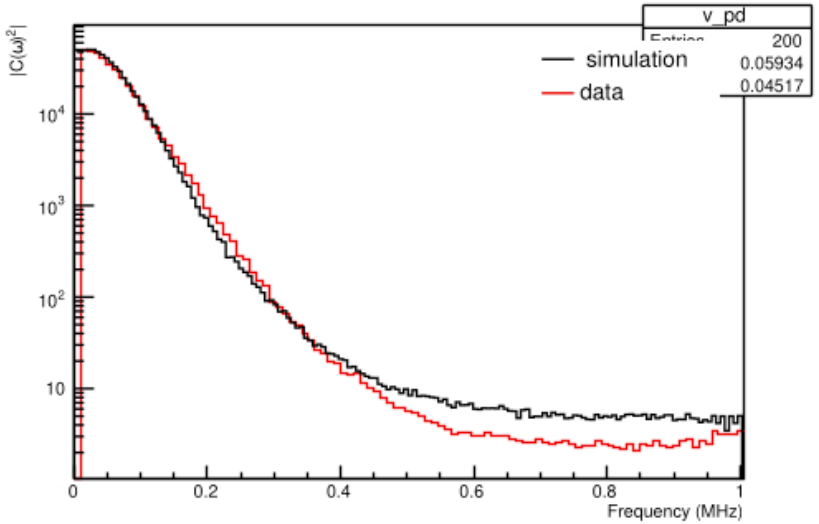
waveform

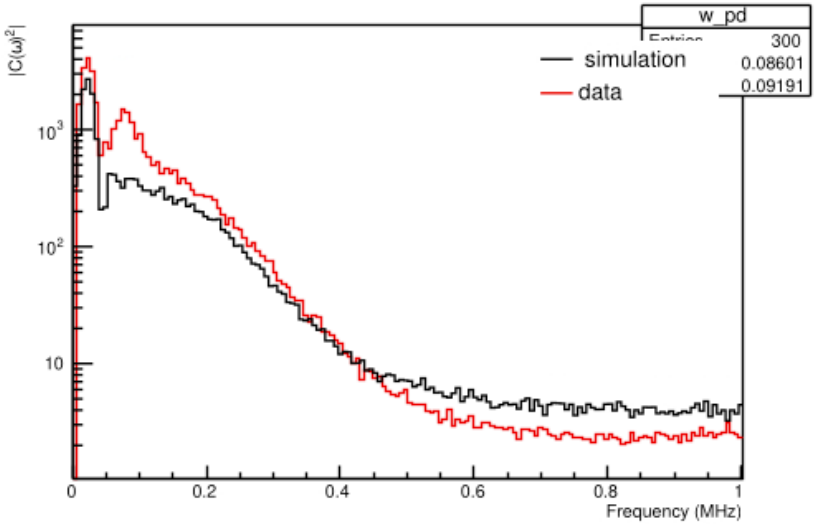






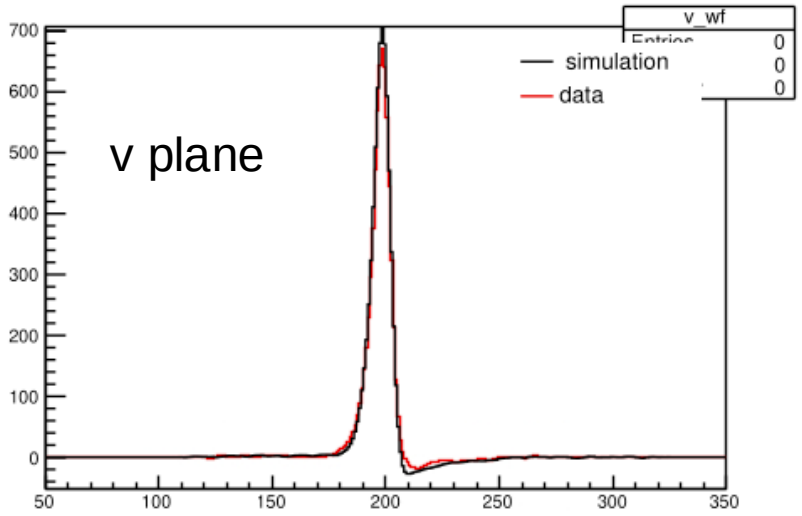
Power spectra

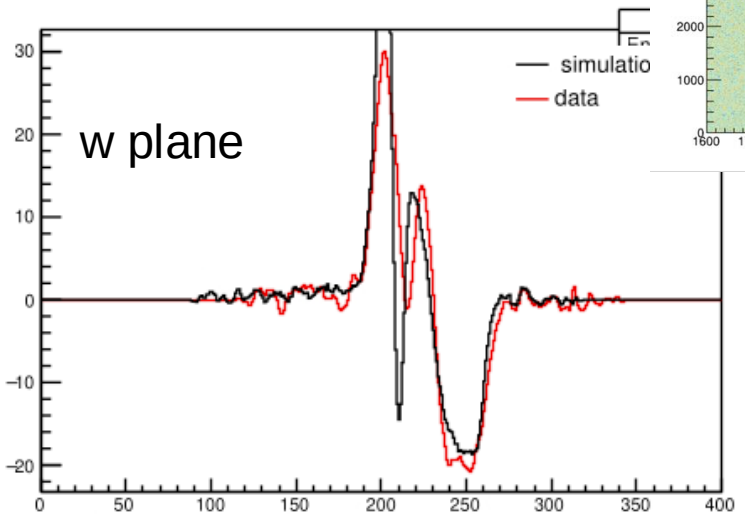


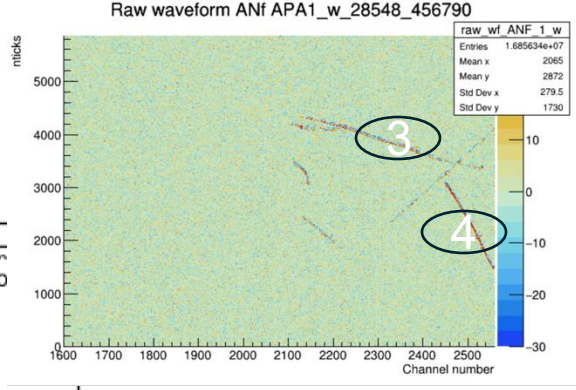


Best one this time

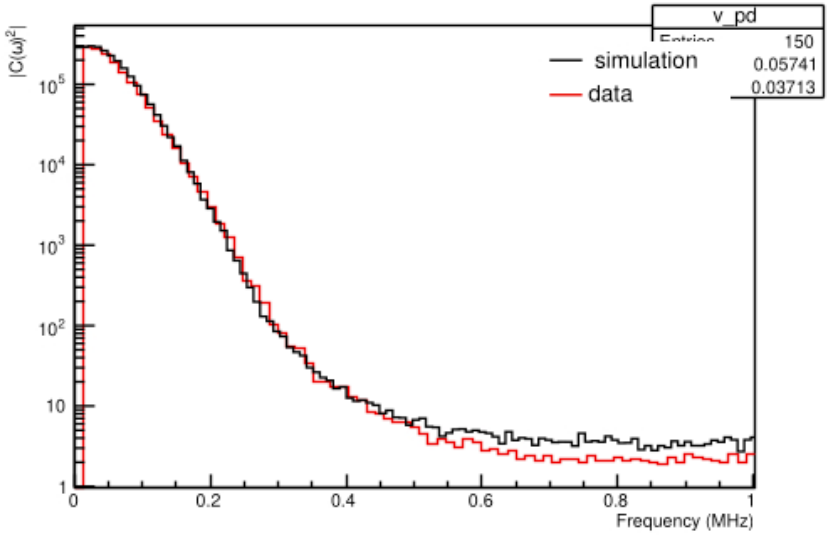
waveform

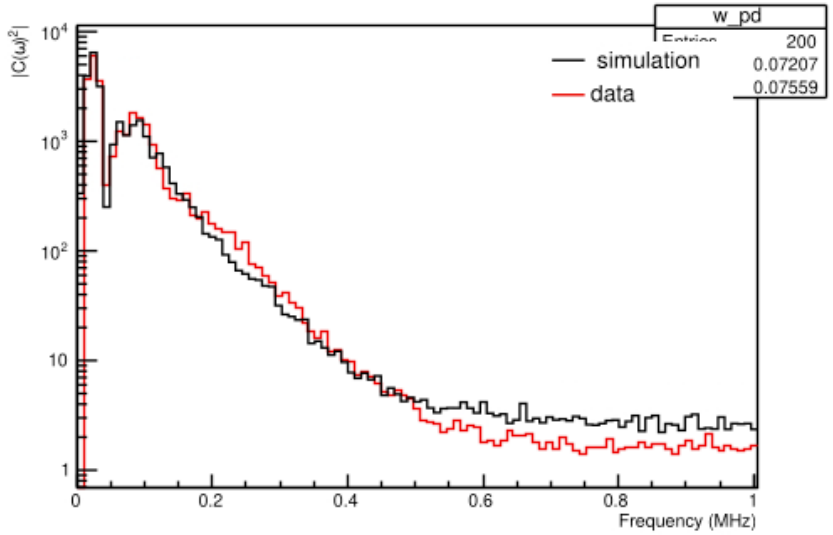






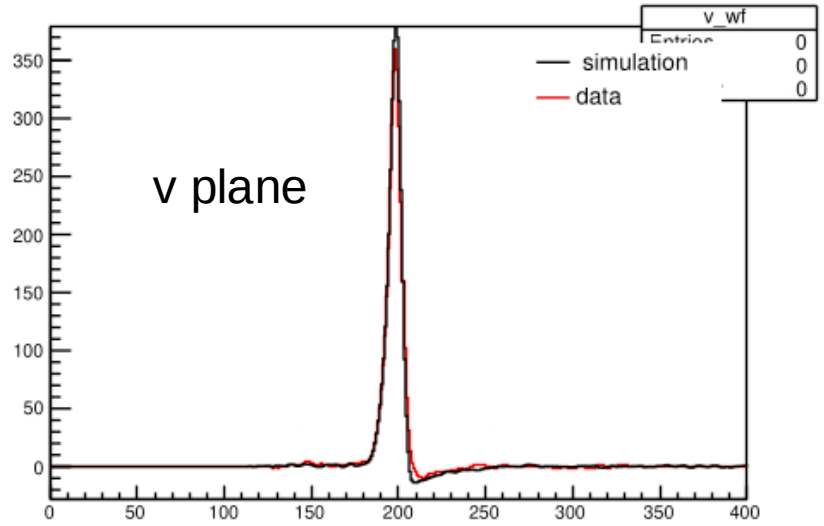
Power spectra

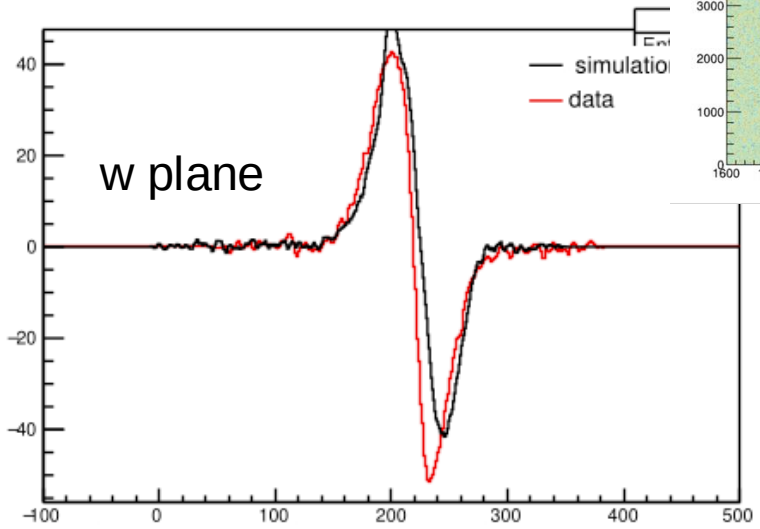


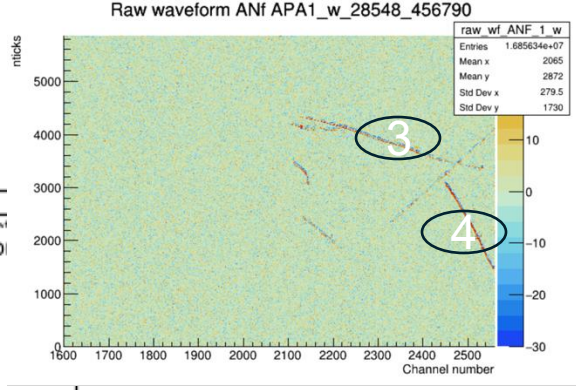


Best one this time

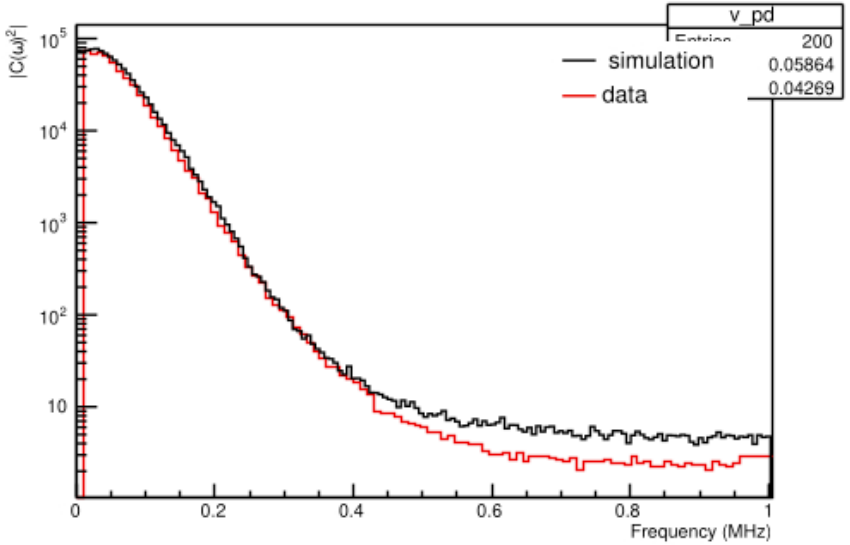
waveform

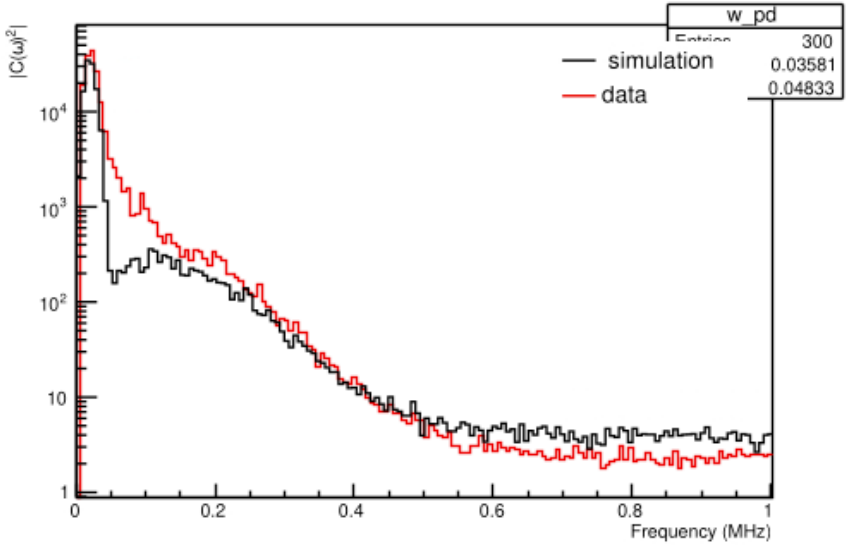






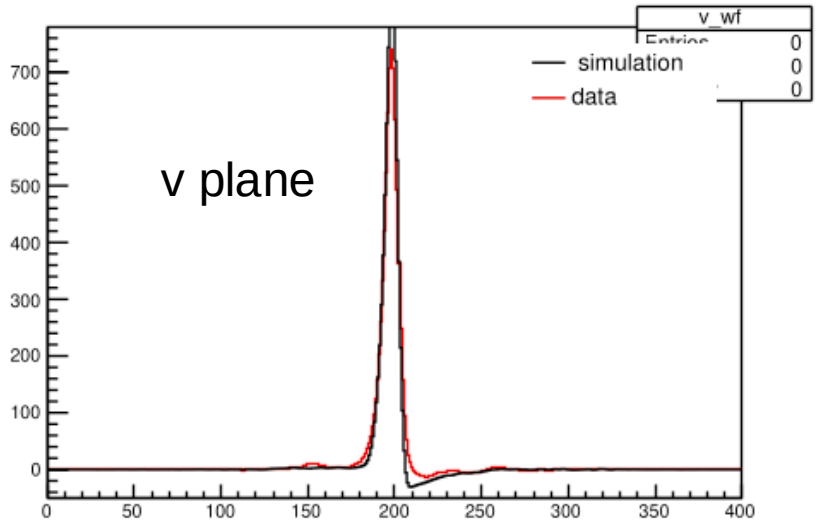
Power spectra

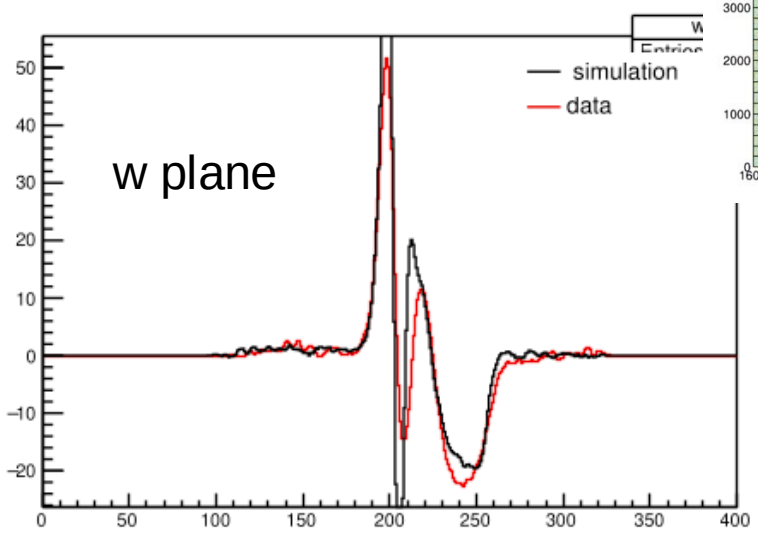


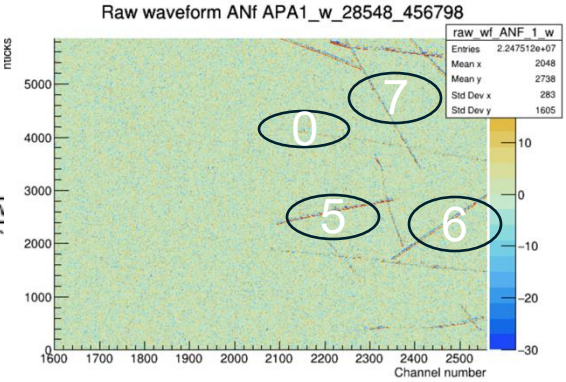


Best one this time

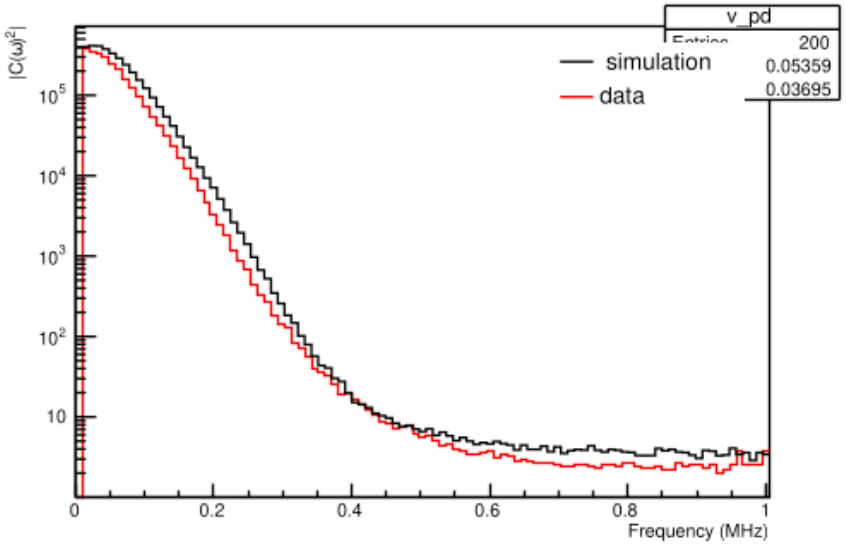
waveform

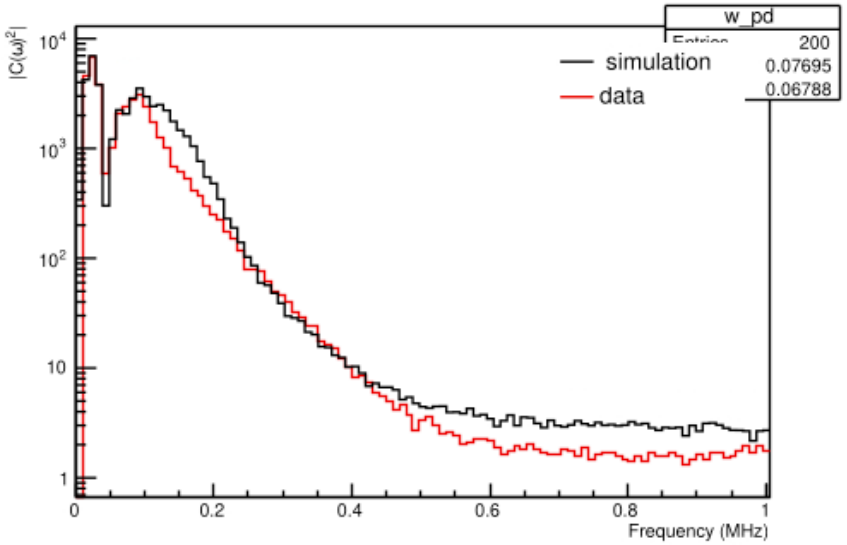






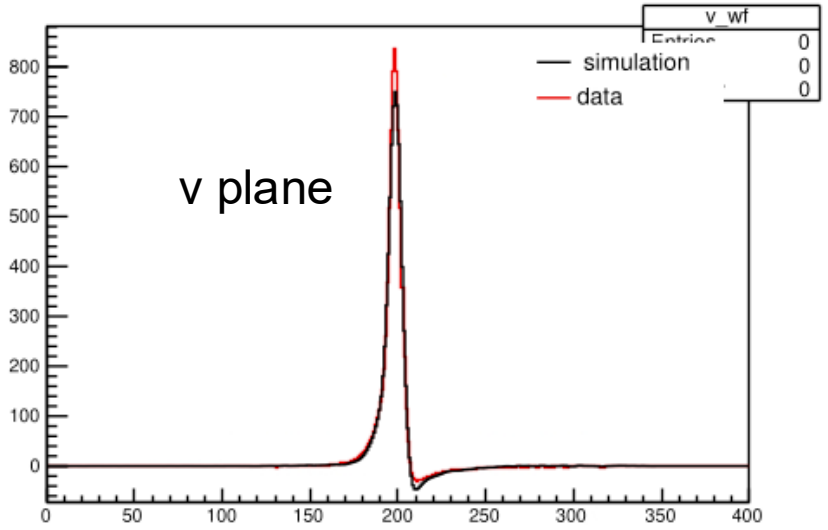
Power spectra

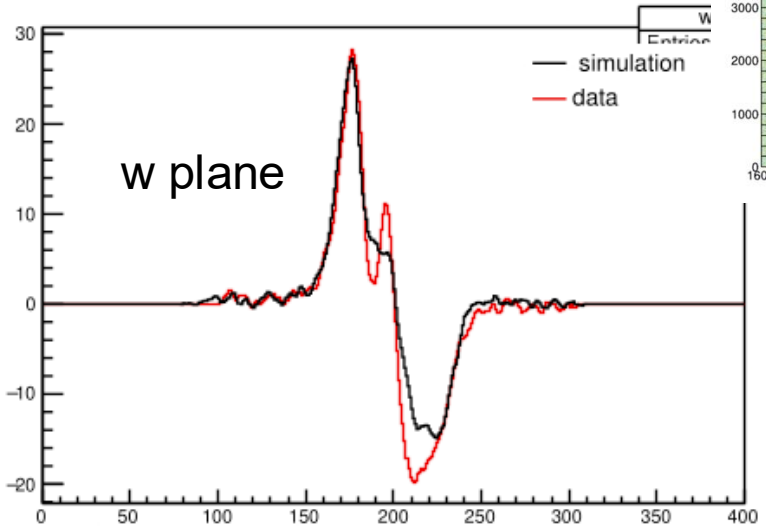


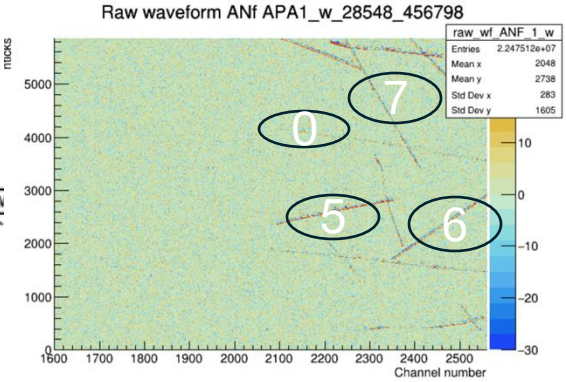


Best one this time

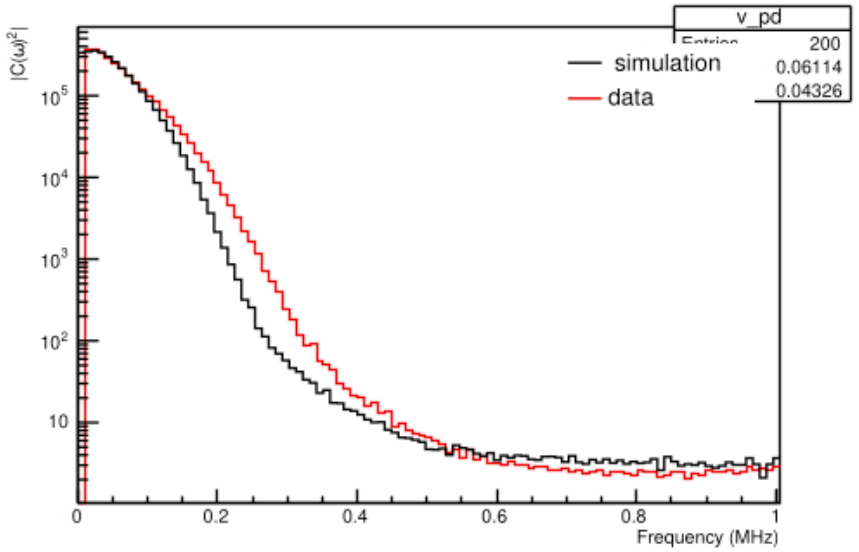
waveform

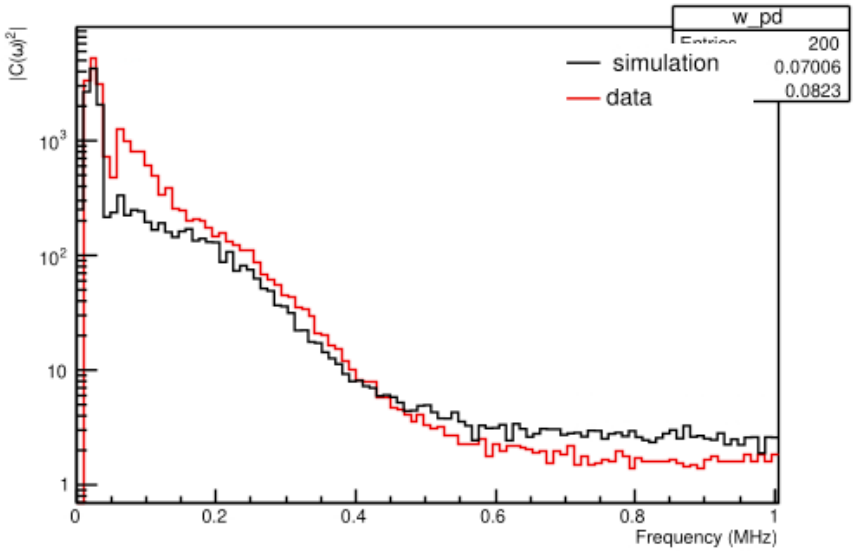






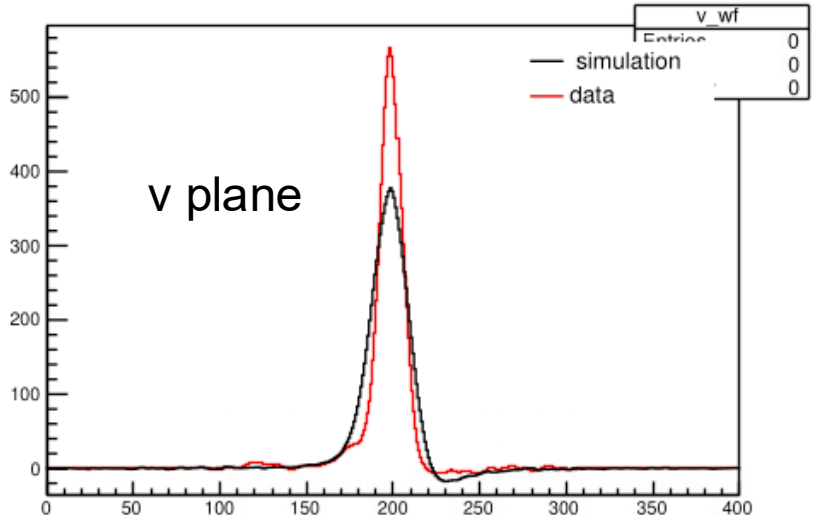
Power spectra

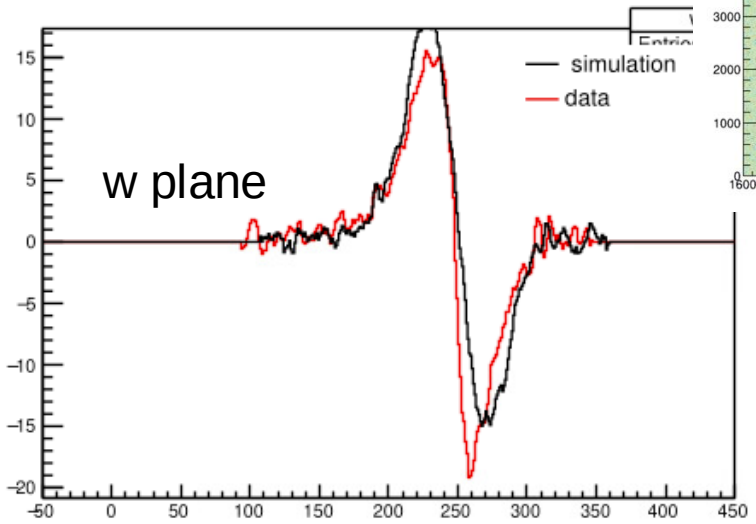


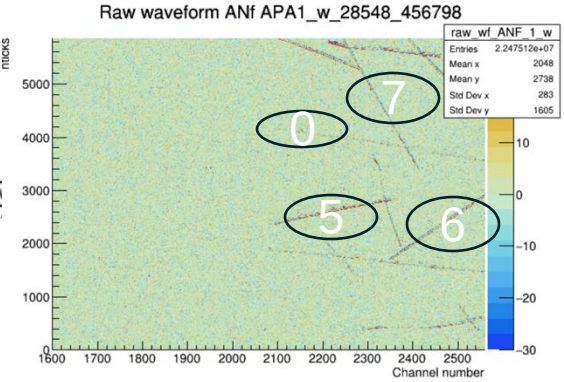


Best one this time

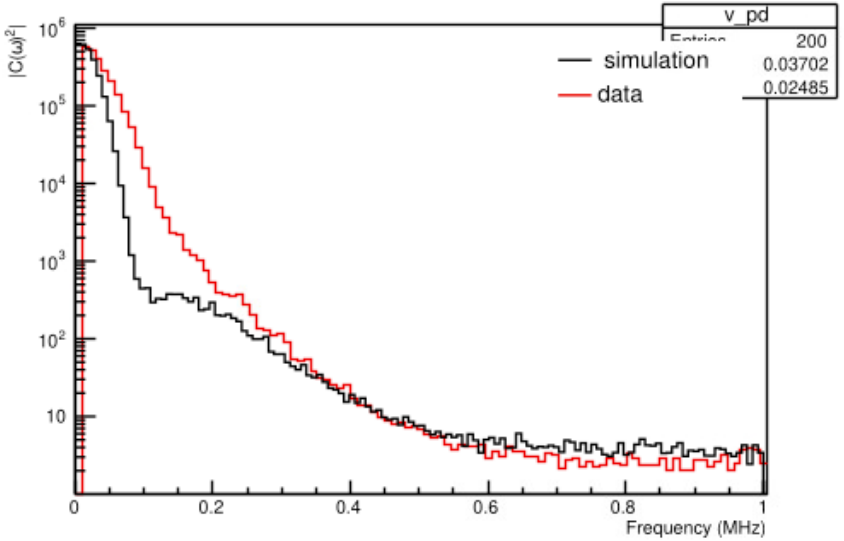
waveform

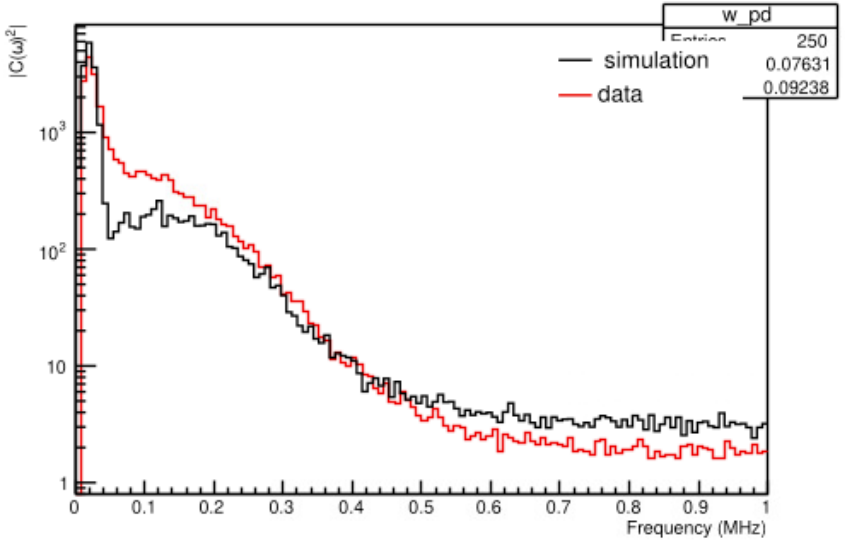




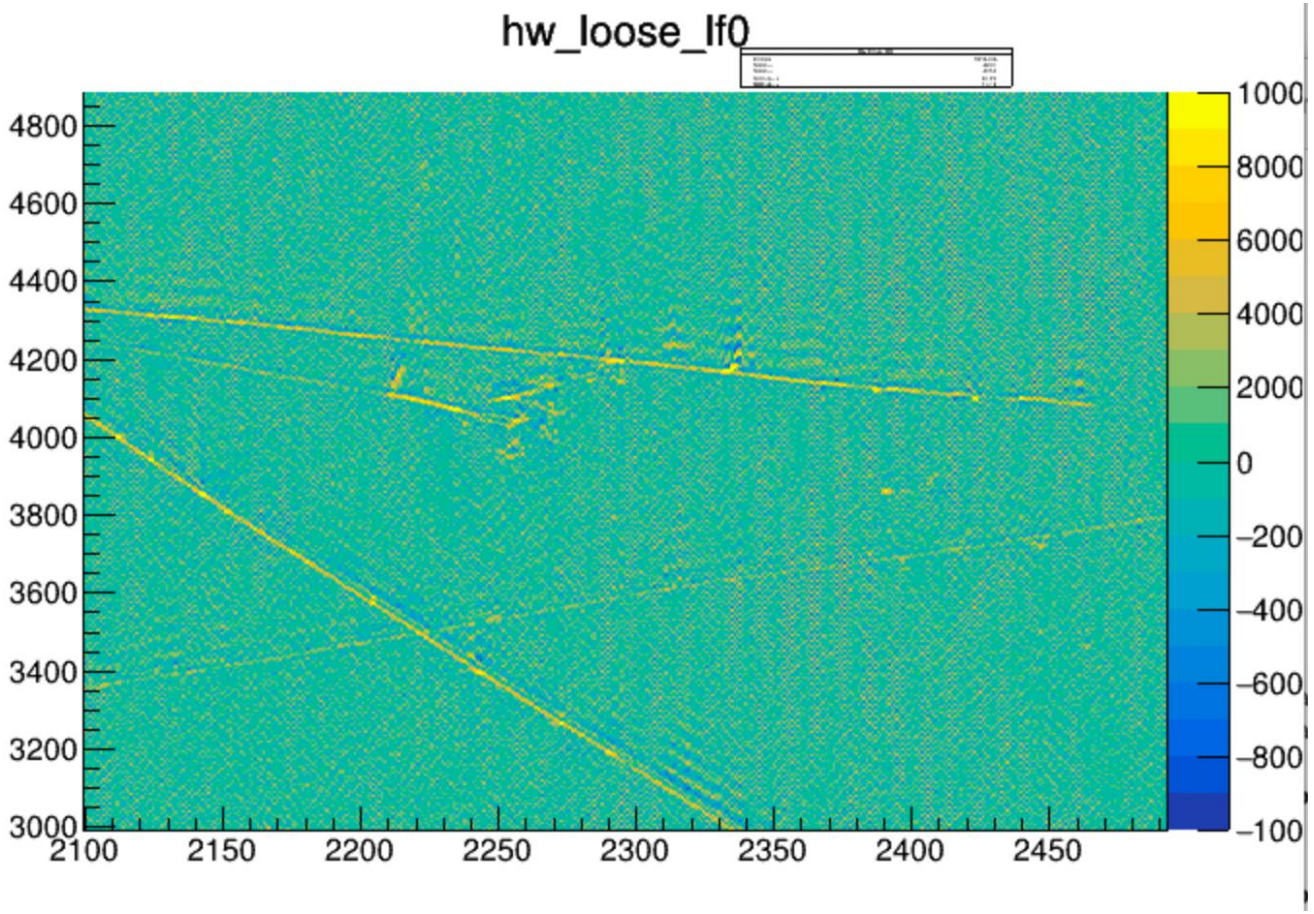


Power spectra

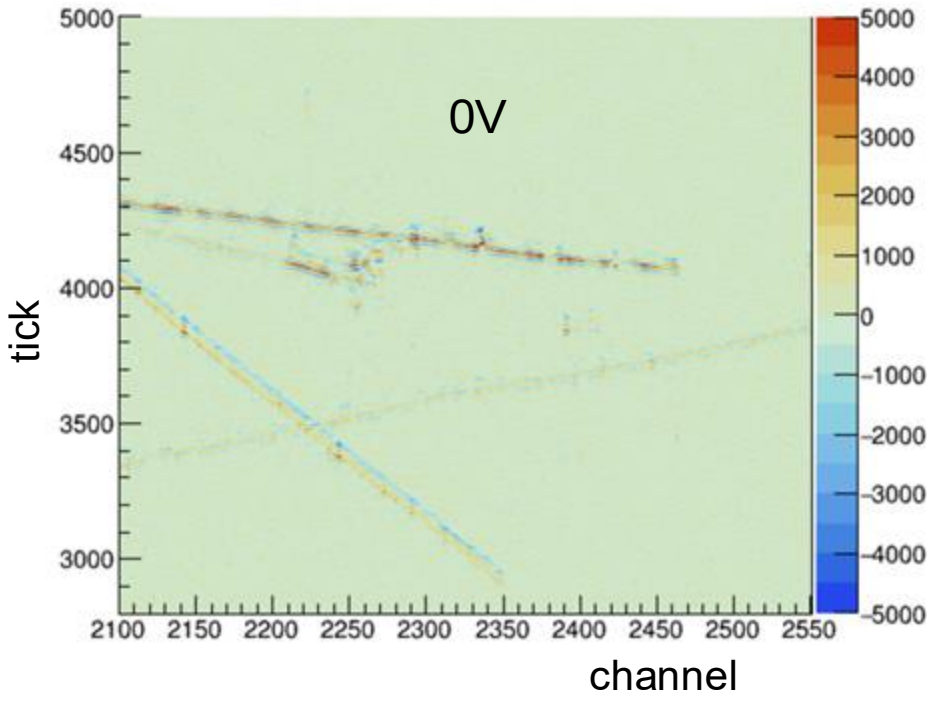


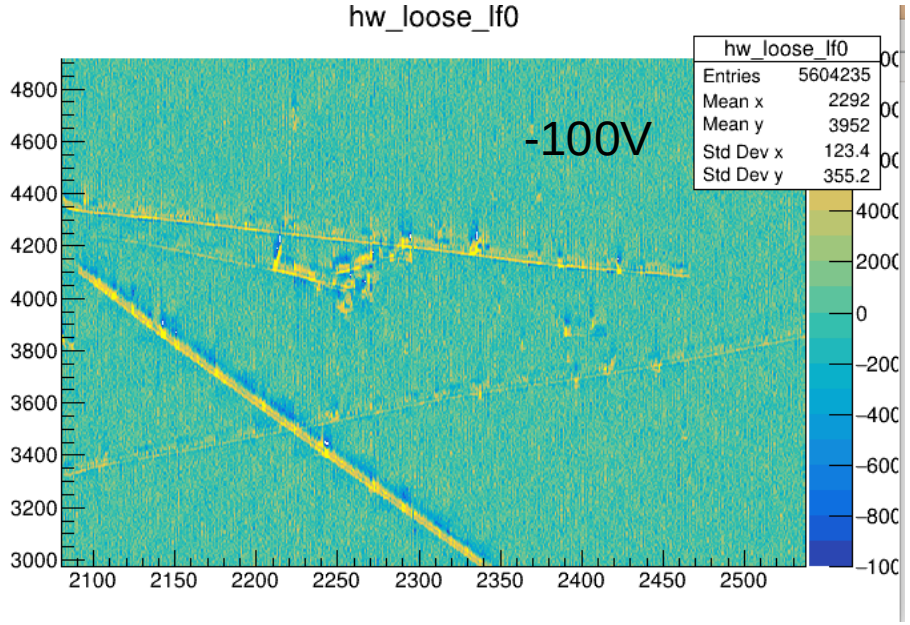


Check decon result

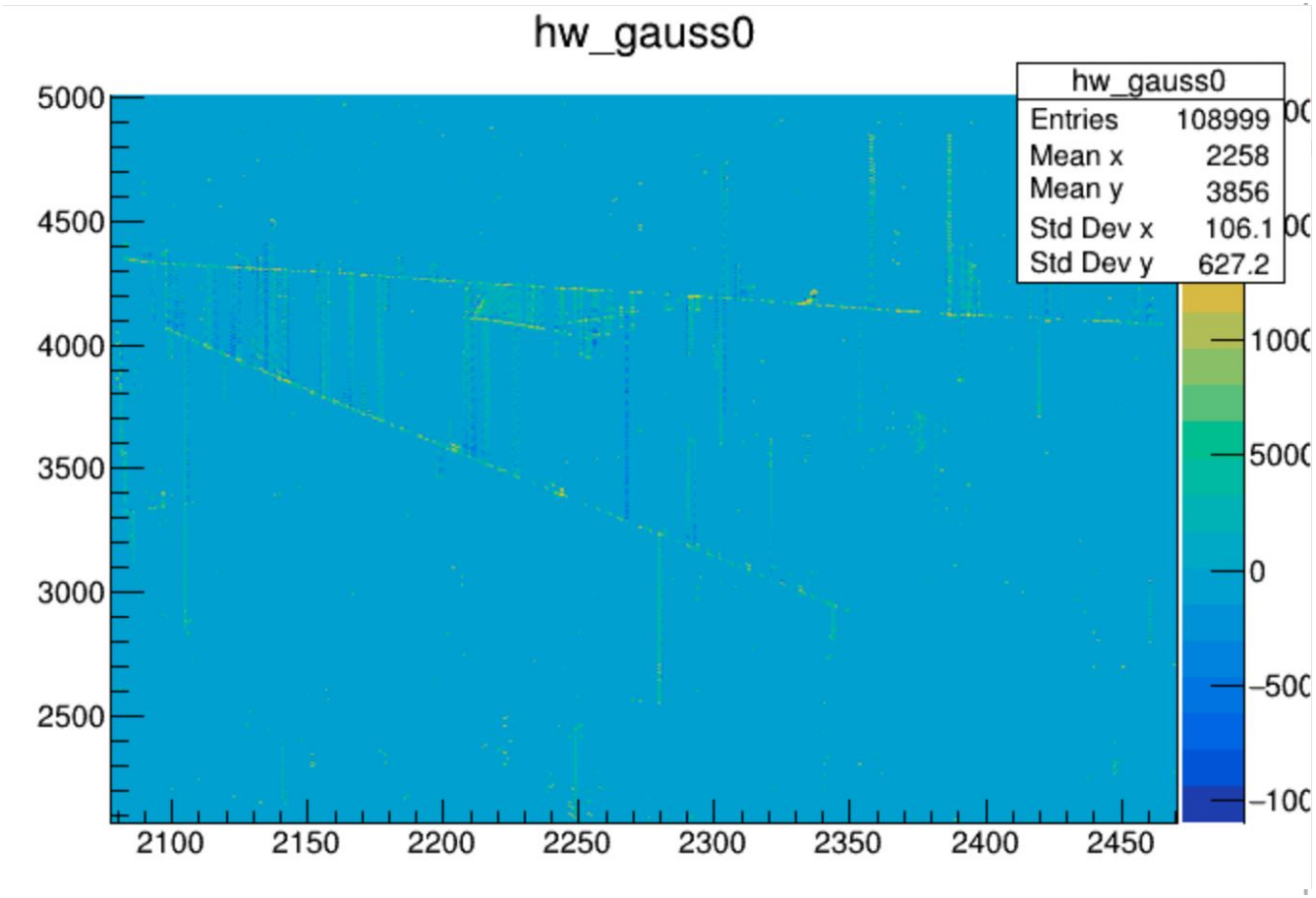


Newly fitted

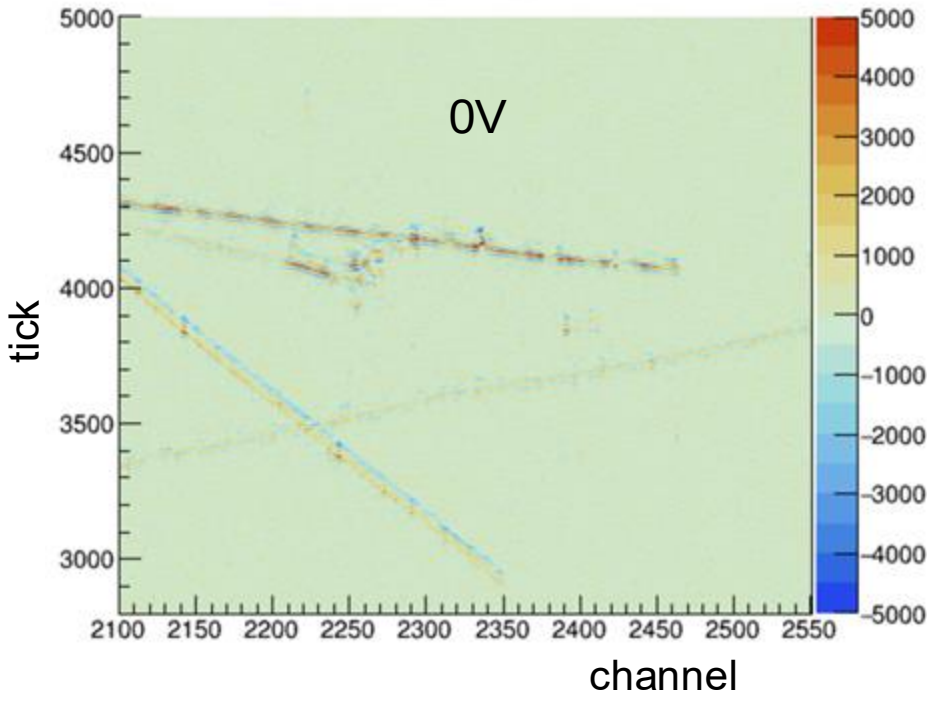


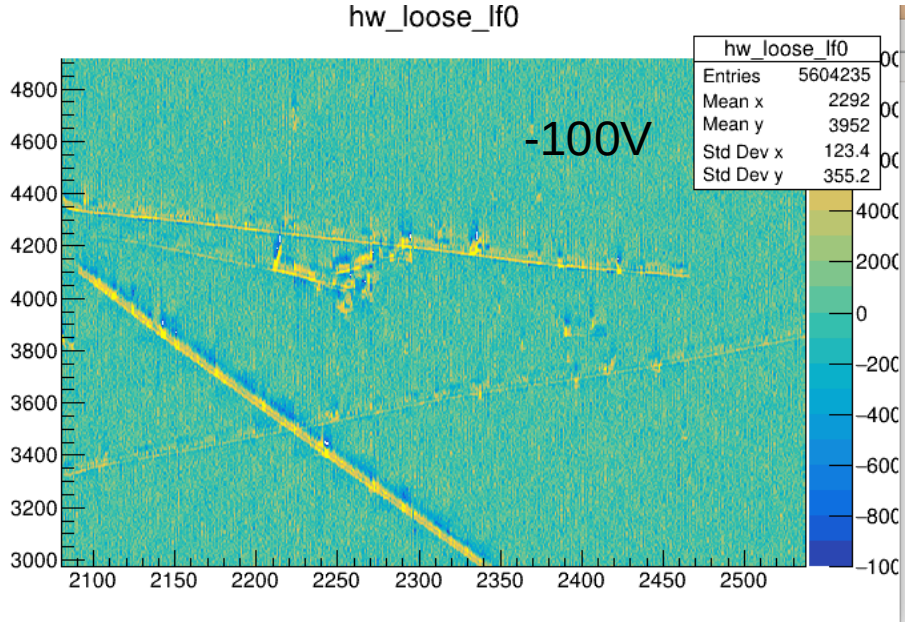


Check decon result



Newly fitted

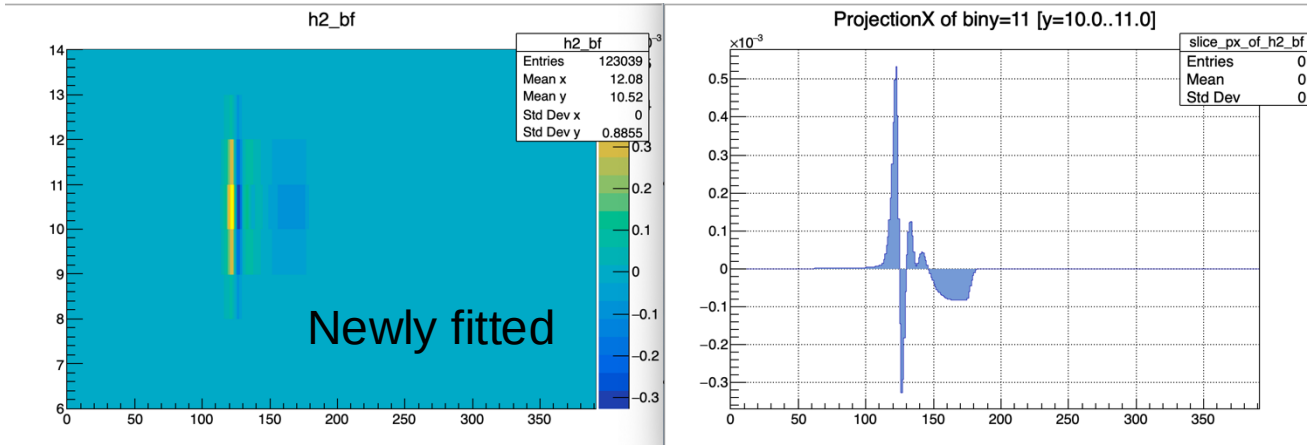


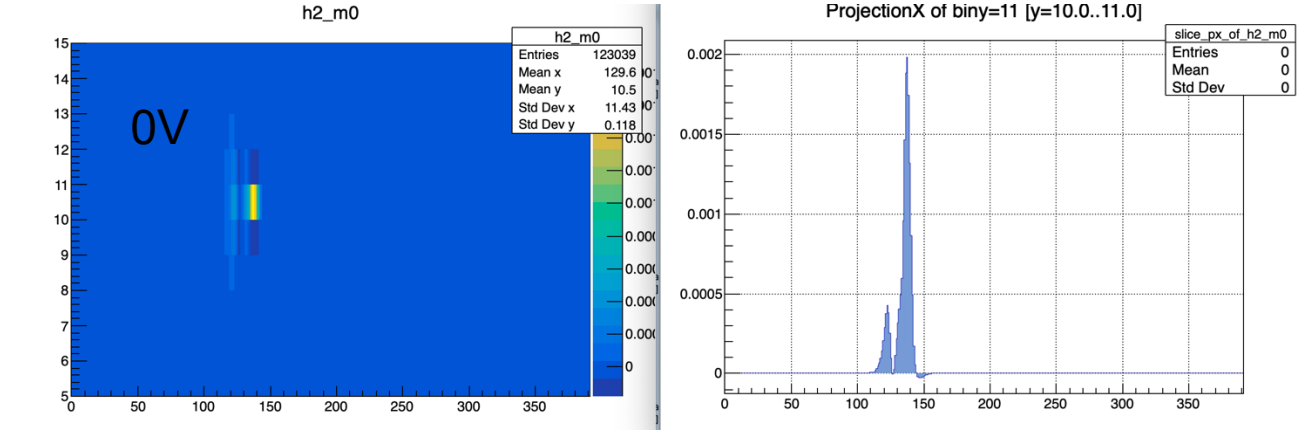


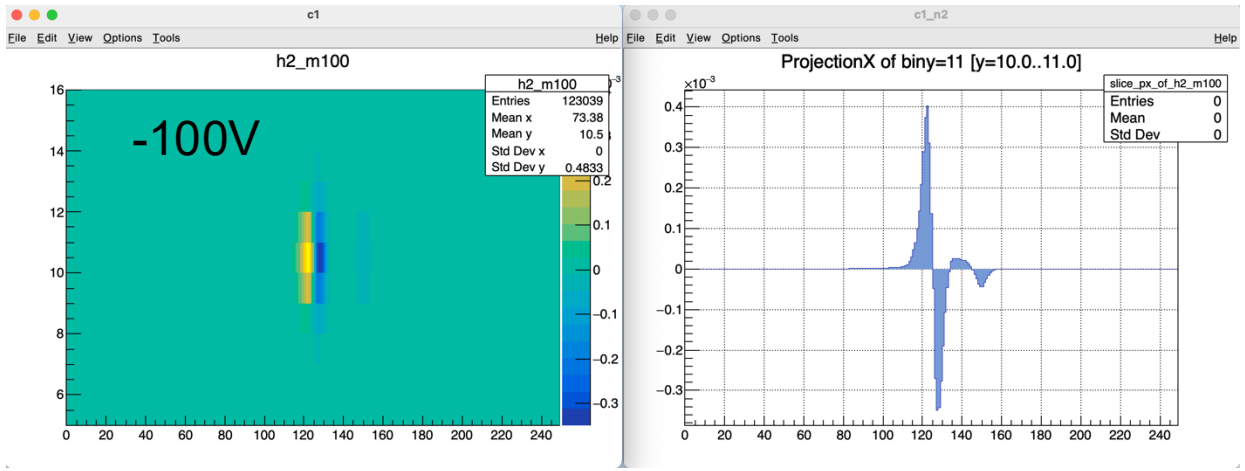
Add a filter to help ROI

- Still ongoing
- **Overall_resp[1]** should be the overall response that combine field response and electronic response.
- A 2D “filter” can be obtained by dividing these 2D response.
- Need more time to understand how to add it through config files.

```
for (size_t i = 0; i != overall_resp[plane].size(); i++) {
    for (int j = 0; j != m_fft_nticks; j++) {
        r_resp(i, j) = overall_resp[plane].at(i).at(j);
        // if(plane==1){
        // std::cout<<r_resp(i, j)<<"\t";
        // }
    }
    //     if(plane==1){
    // std::cout<<std::endl;
    //     }
}
//         if(plane==1){
// exit(0);
//         }
```







Signal processing and imaging validation

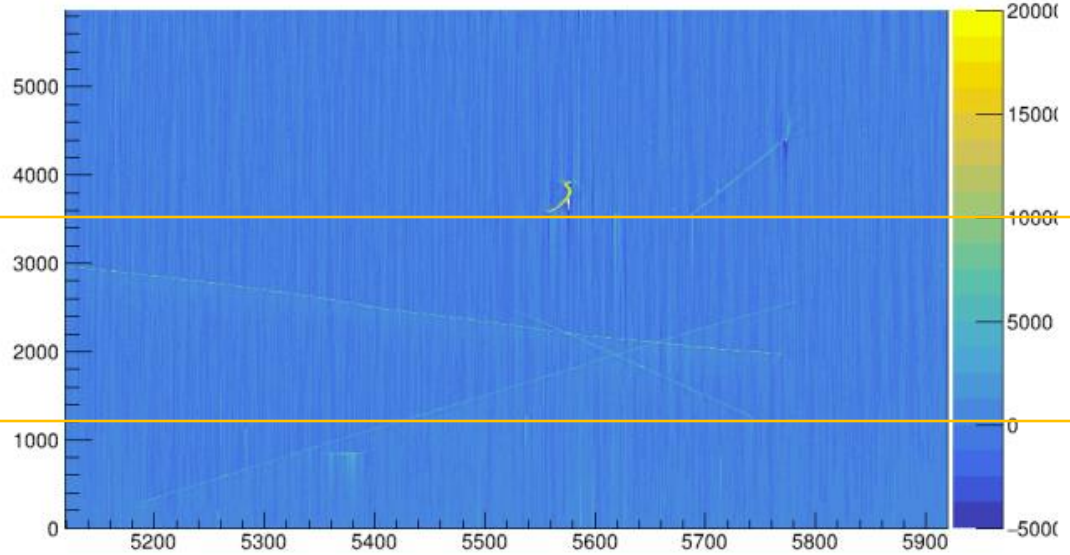
Xuyang Ning

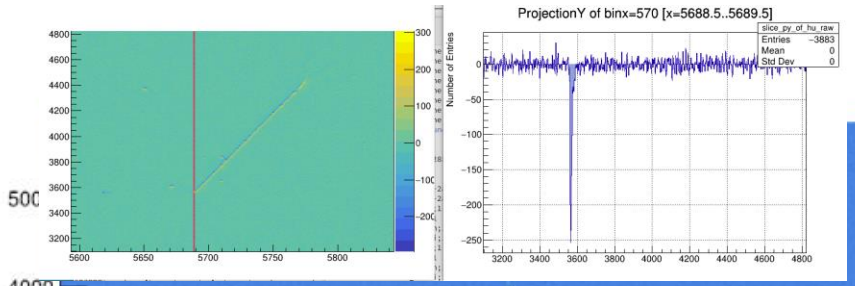
02/12/2025

Brookhaven National Laboratory

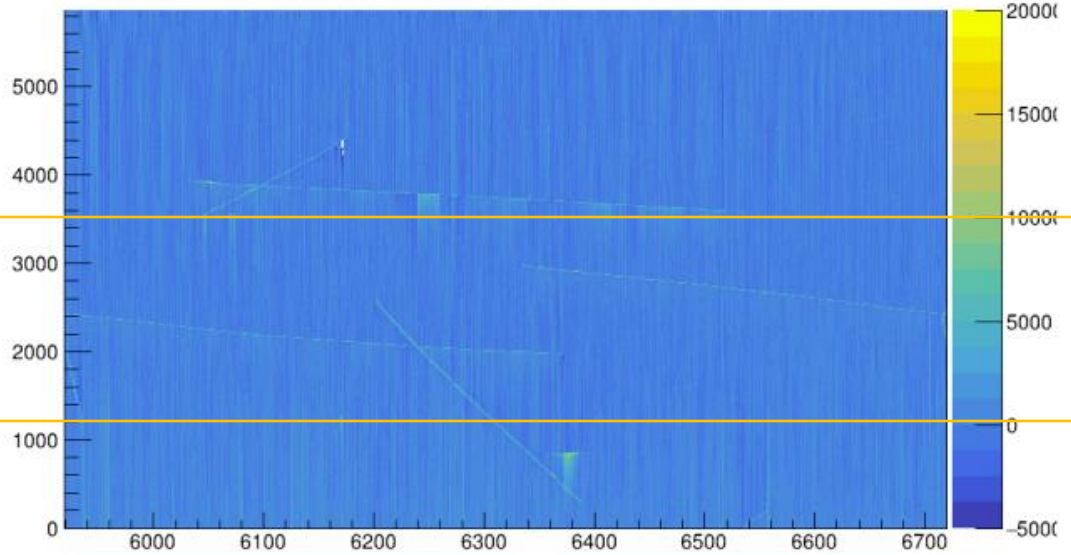
magnify-28548-456814-2.png

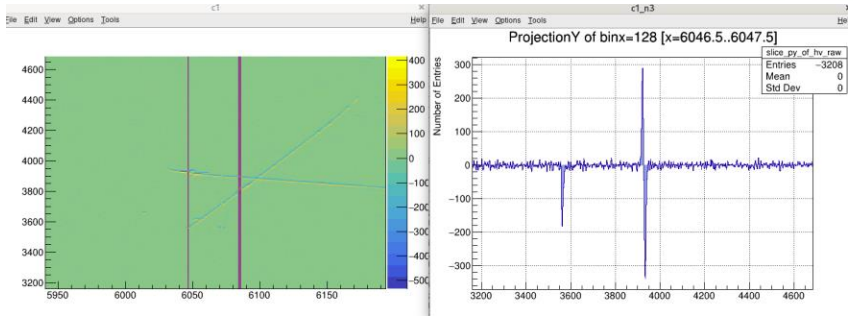
hu_decon_ori2

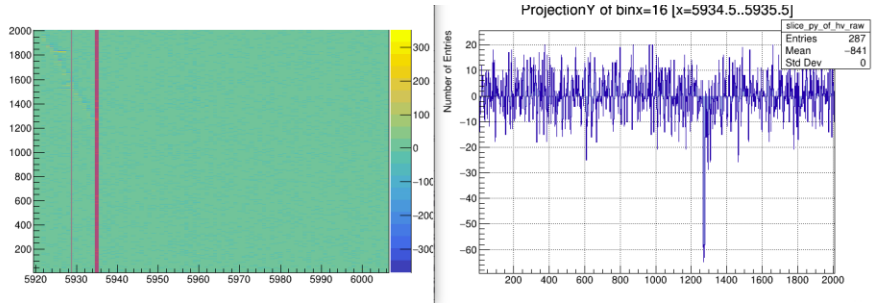




hv_decon_ori2

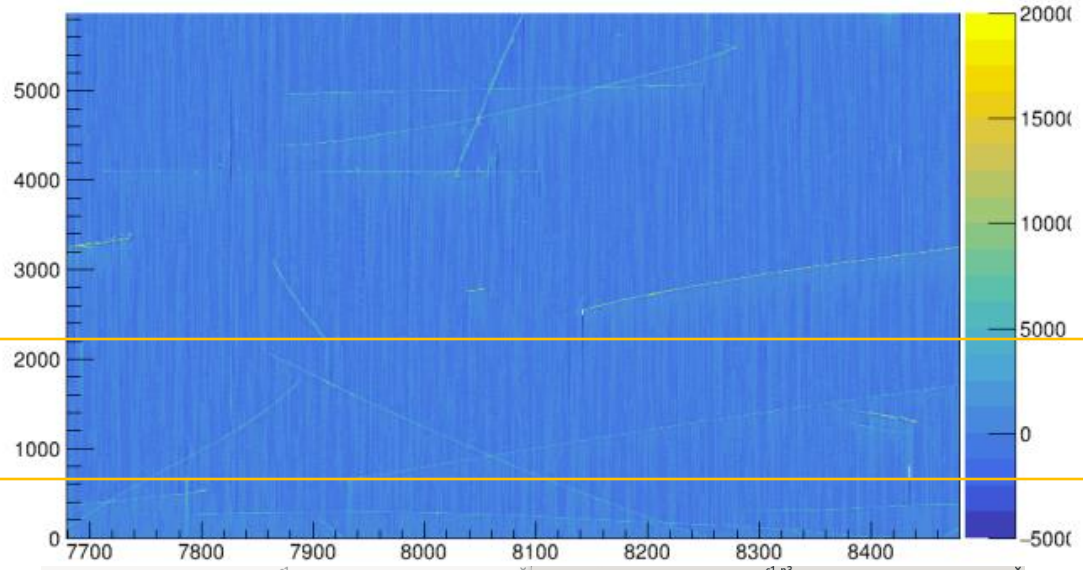


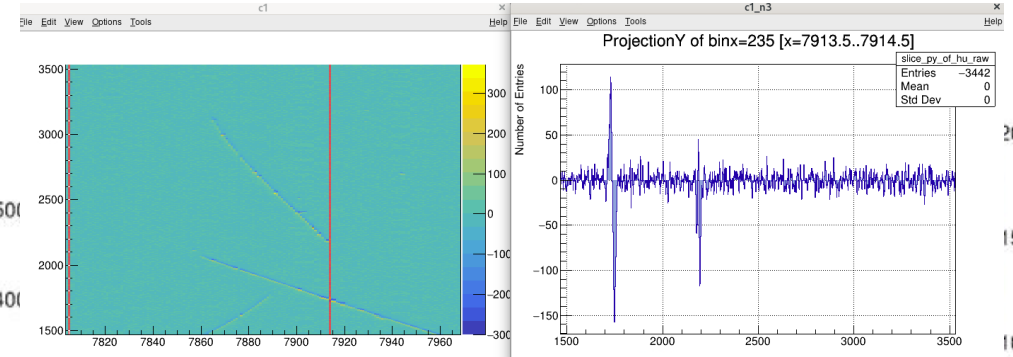


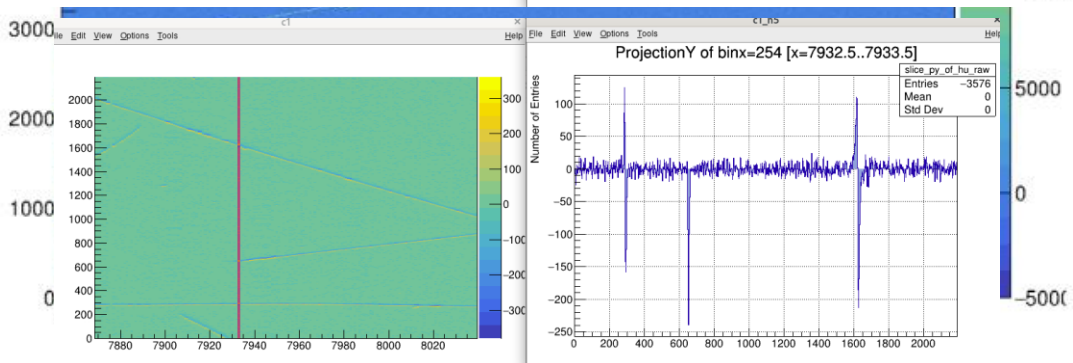


magnify-28548-456814-3.png

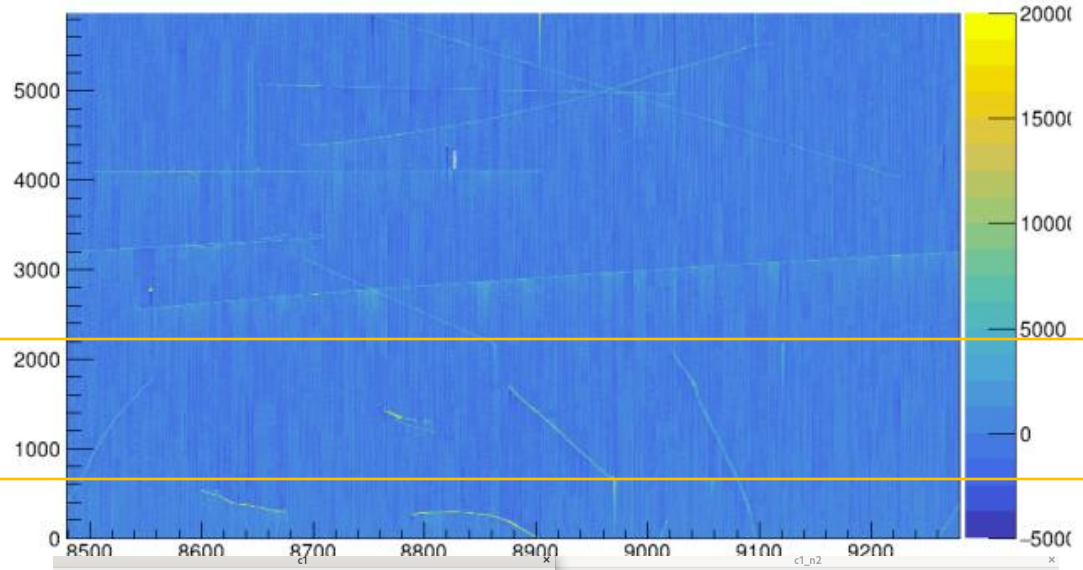
hu_decon_ori3

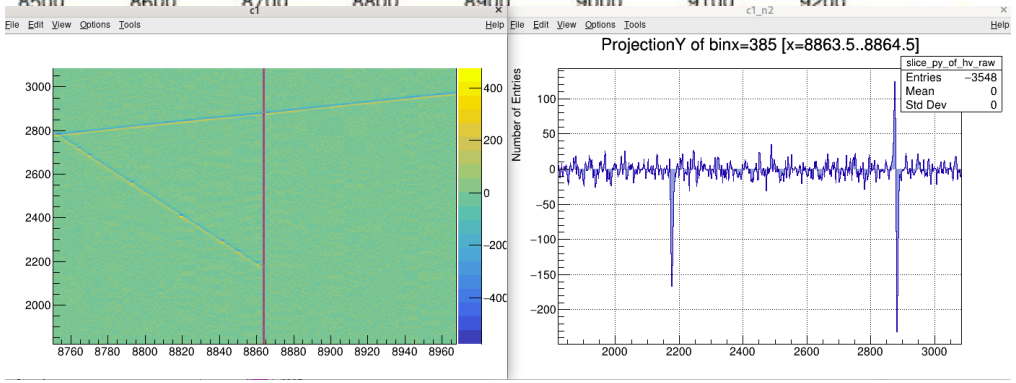


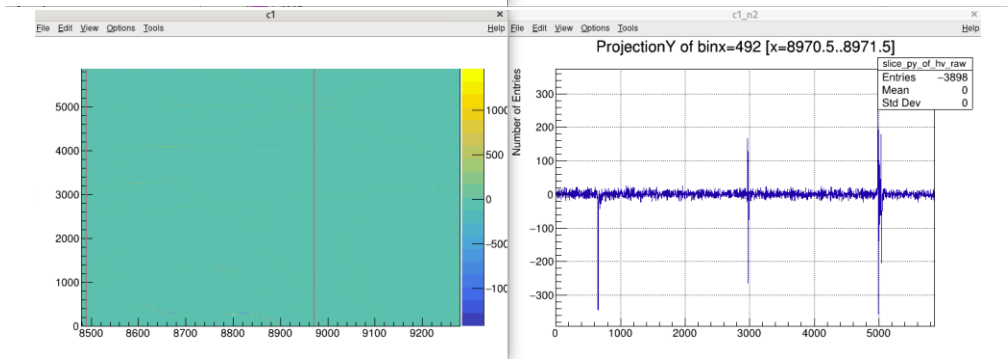




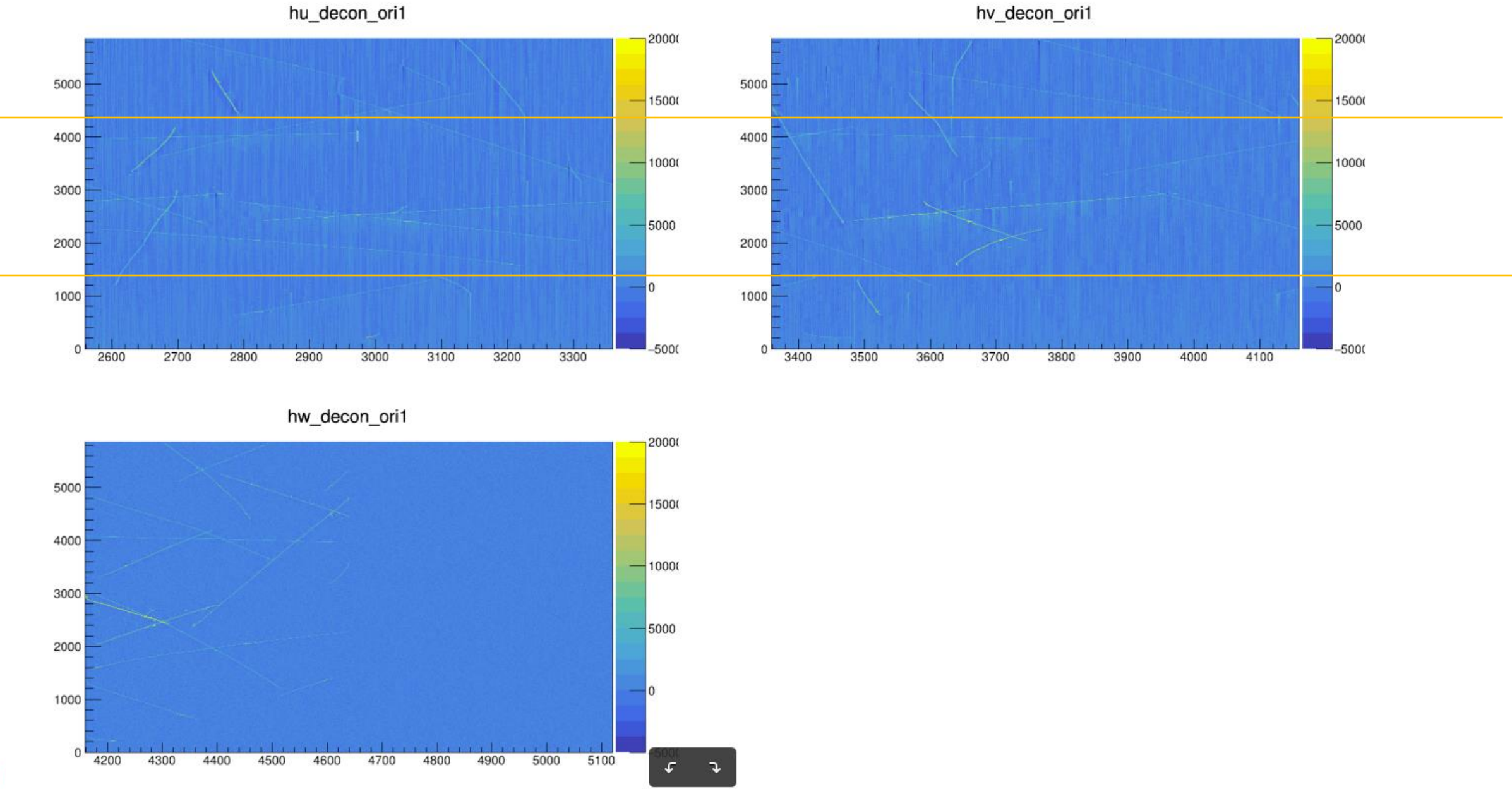
hv_decon_ori3







magnify-28548-456822-1.png



magnify-28548-456822-3.png