Charge method for pile up event removal in SHOE

Giacomo Ubaldi - 13/04/2022

First implementation on SHOE

The implementation of this method was merged from “newgeom_v1.0” to the branch “pileUpStudy” of **SHOE**.

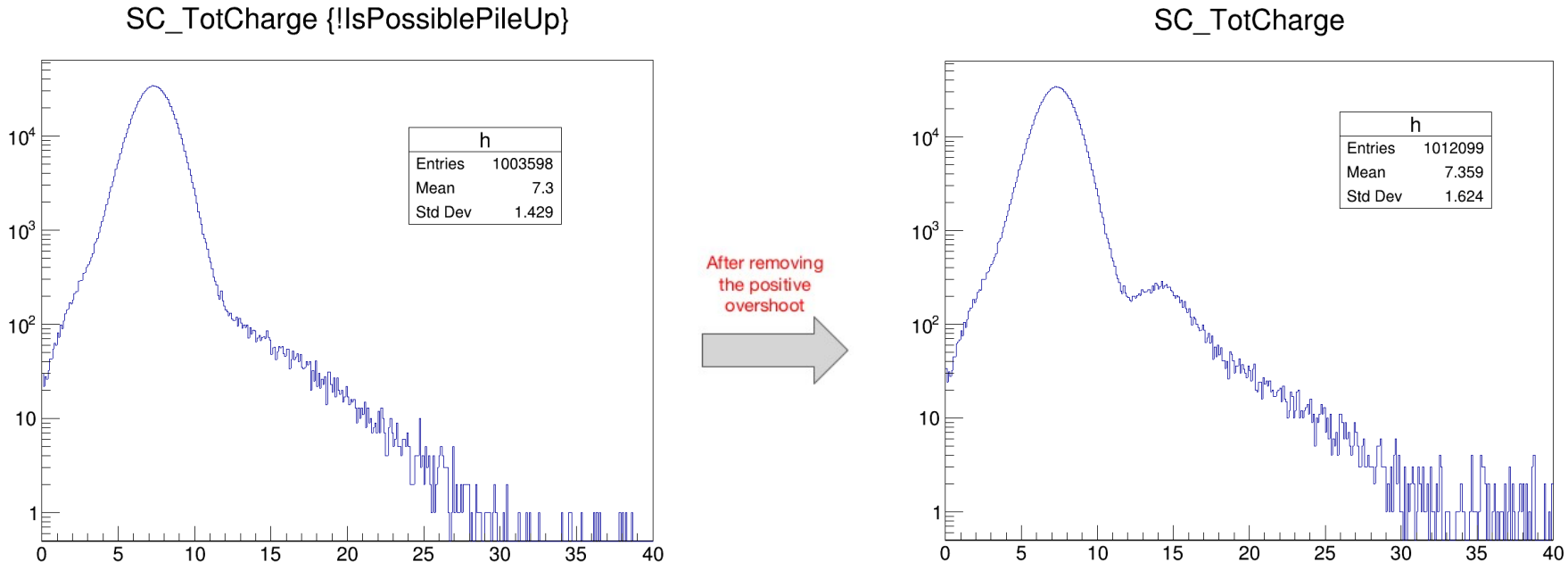
In particular, we focused on the modified function:
double TASTrawHit::*ComputeCharge*(TWaveformContainer *w, double thr)

Parallelly, we also studied the method via the function of the stand – alone **SLIPPER**:
Float_t SCWaveFormContainer::*GetSCTotalCharge*(std::vector<Int_t>* Channels)

The first attempt was to obtain the charge spectrum of events and so to check if,
applying the threshold on the integral of the signal,it is actually possible to distinguish different peaks of charge.

- The method was tested on real **data taken at GSI** of beam of **016** of 400 Mev/n against a target of C
  (DataGSI2021/data_test.00004310.physics_foot.daq.RAW._lb0000._FOOT-RCD._0001.data)

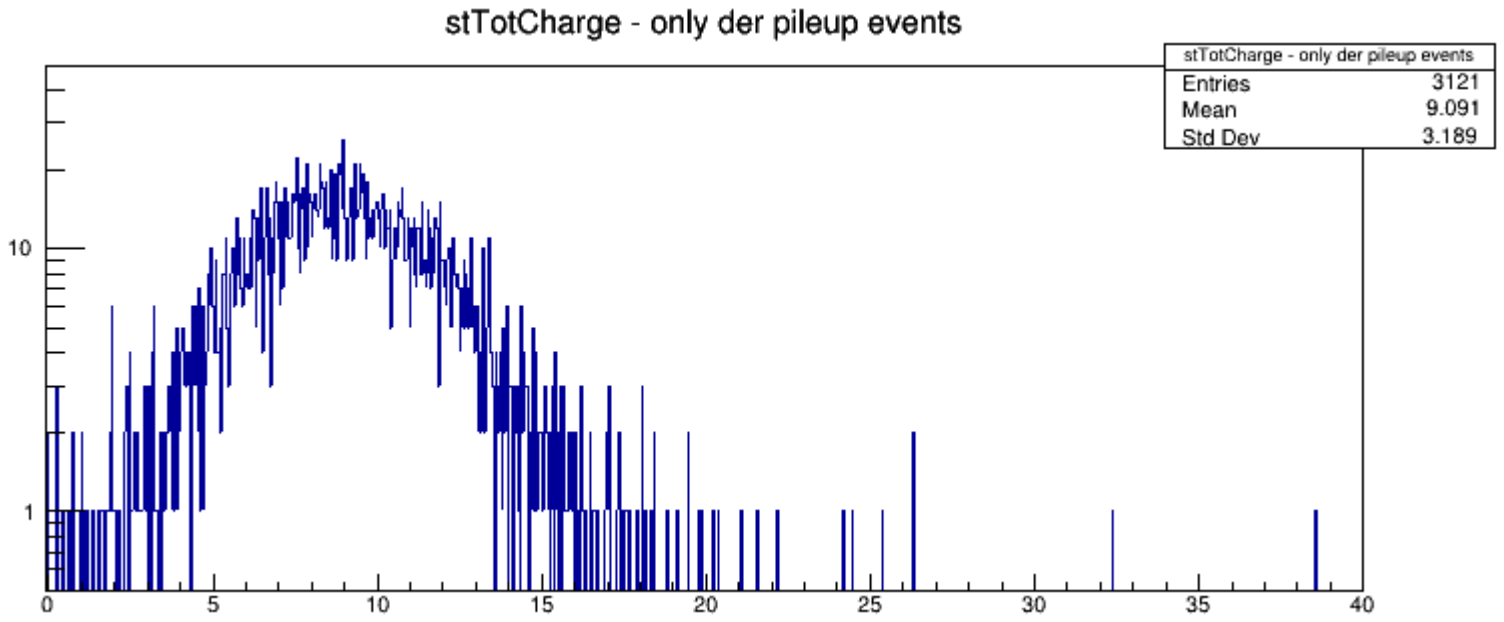
Results on slipper



With **SLIPPER,** a variation of the spectra is found. The first peak became less smooth and a second one is found around a valure of 12 V*ns (arbitrary unit for charge).

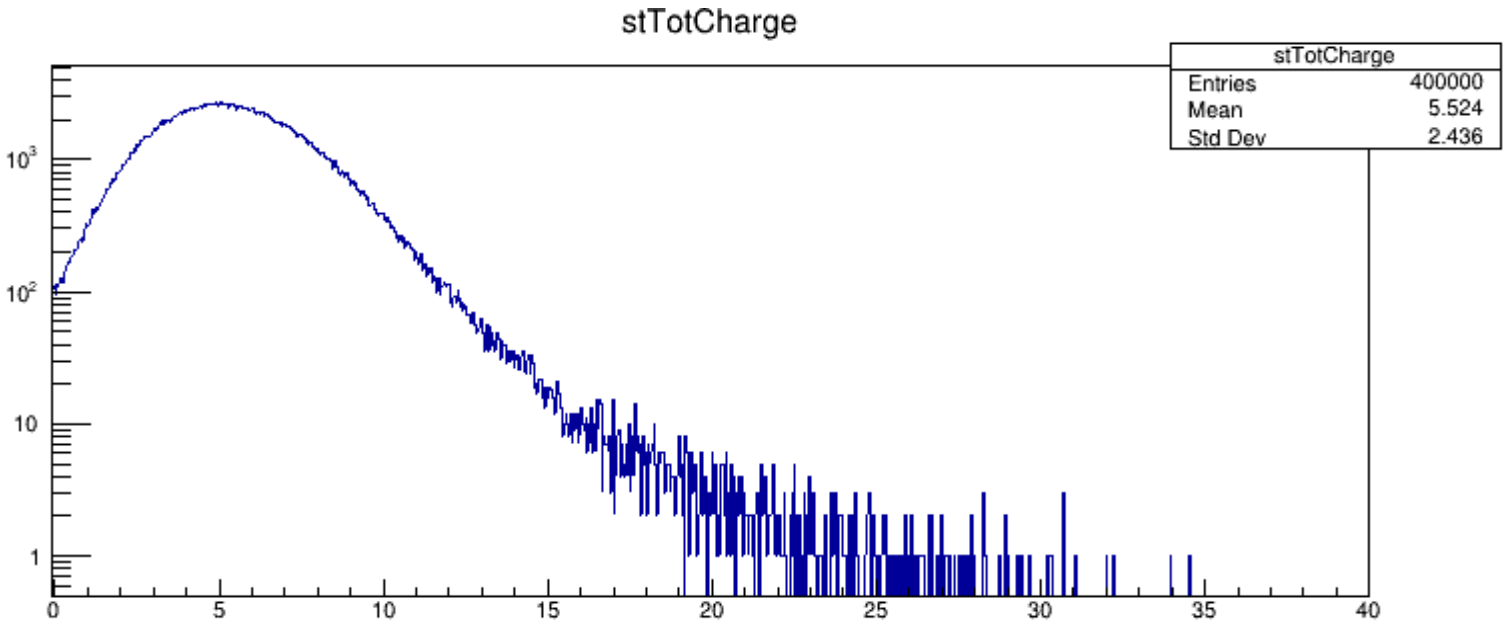
Results on SHOE

Applying the **derivative method in SHOE**, the spectra of pile up events charge would be exactly the one peaked around 12 V*ns found in the **SLIPPER** charge method.



So it seems that the derivative method and the charge method give similar results, but the problem is in the implementation on SHOE.

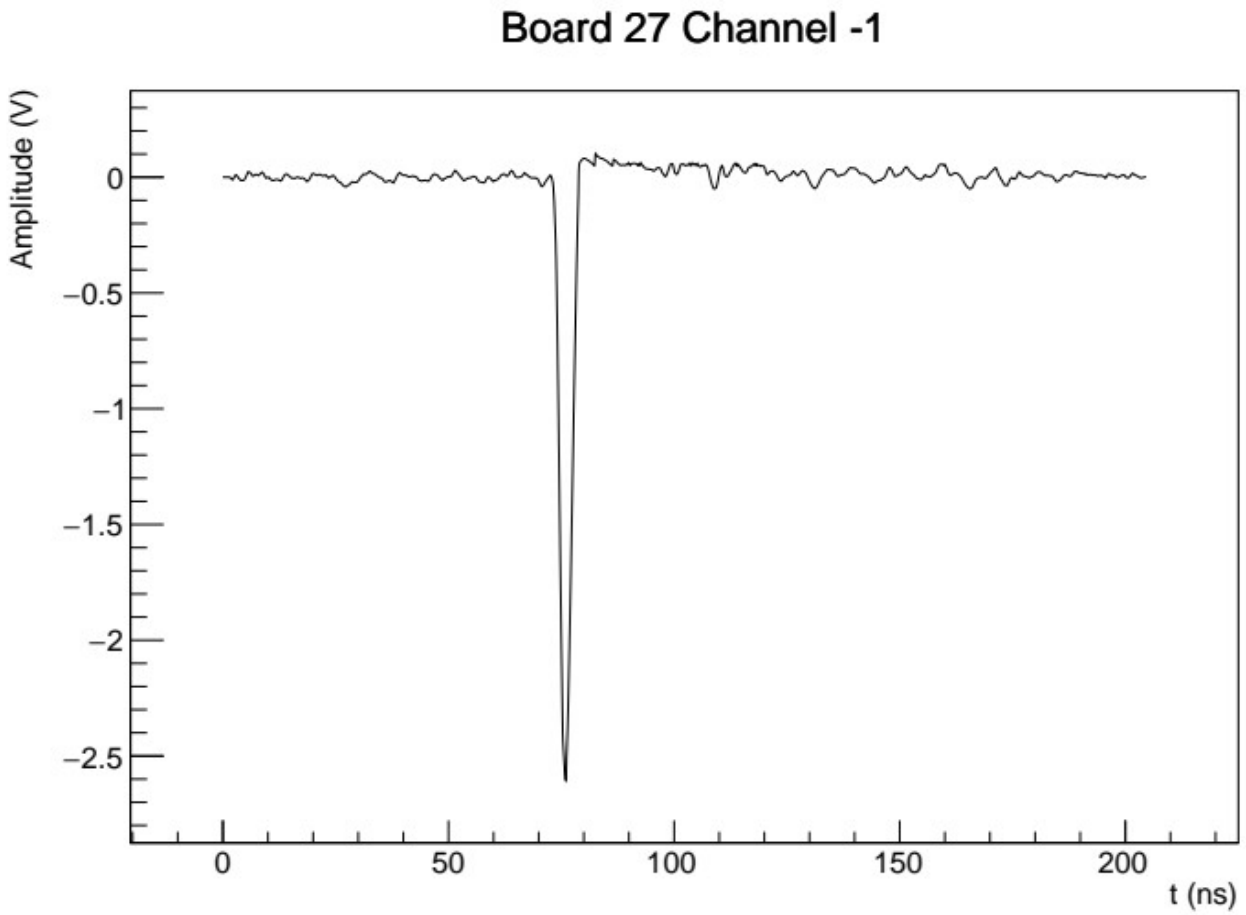
Results on SHOE



After removing overshoot with Thr = 0.008

In SHOE, no variation in the spectra was found applying the new charge method. It's like the energy Resolution is too wide to appreciate the pile up peak (which should be around 12 V*ns)

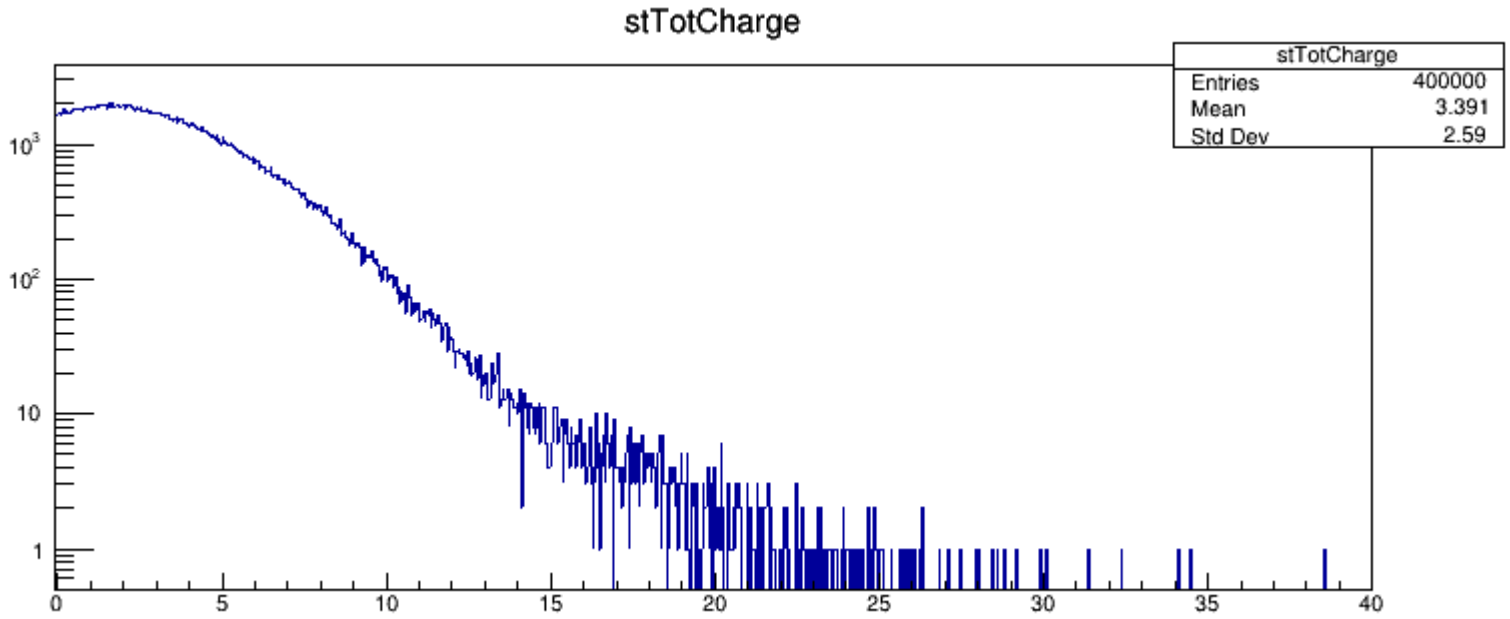
Results on SHOE



Observing the WF of the signal, we supposed that the std. Dev. Of **0.008** was not enough for our data because the integral would be inficiated by the noise of the signal.
so we put a value of **0.02**

Results on SHOE

But even with a higher value of the threshold, no peaks were found in the charge spectra.



After removing overshoot with Thr = 0.025

Furthermore, the energy resolution became also worsen.

SHOE vs SLIPPER

We tried to find the main differences in the two codes. In particular:

SHOE: double TAGbaseWD::ComputeBaseline(TWaveformContainer *w)

SLIPPER: void SCWaveFormContainer::SumSCWaveforms(std::vector<Int_t>* Channels)

- In SHOE the baseline is measured in the first 27 bins. In SLIPPER it is measured in the frist and **last** 27 bins.

SHOE: double TASTrawHit::*ComputeCharge*(TWaveformContainer *w, double thr)
SLIPPER: Float_t SCWaveFormContainer::*GetSCTotalCharge*(std::vector<Int_t>* Channels)

- In SHOE the thresold is a const, in SLIPPER it is measures the RMS dinamically for each event.

- The method of sum of the 8 channel in SuperHit is slightly different

Is it possible to converge to a unique method?

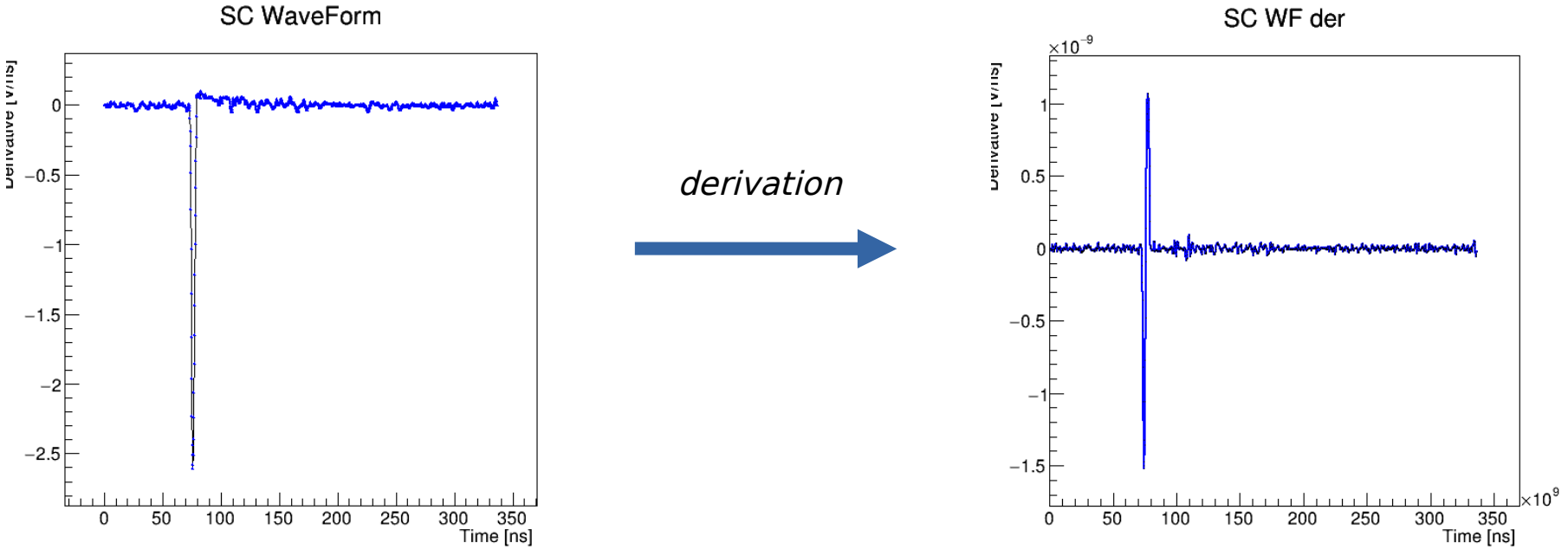
Back up slides

Derivative method, 1

At first, the derivative is made by an algorithm which iterates on every point of the SC signal.

In particular,

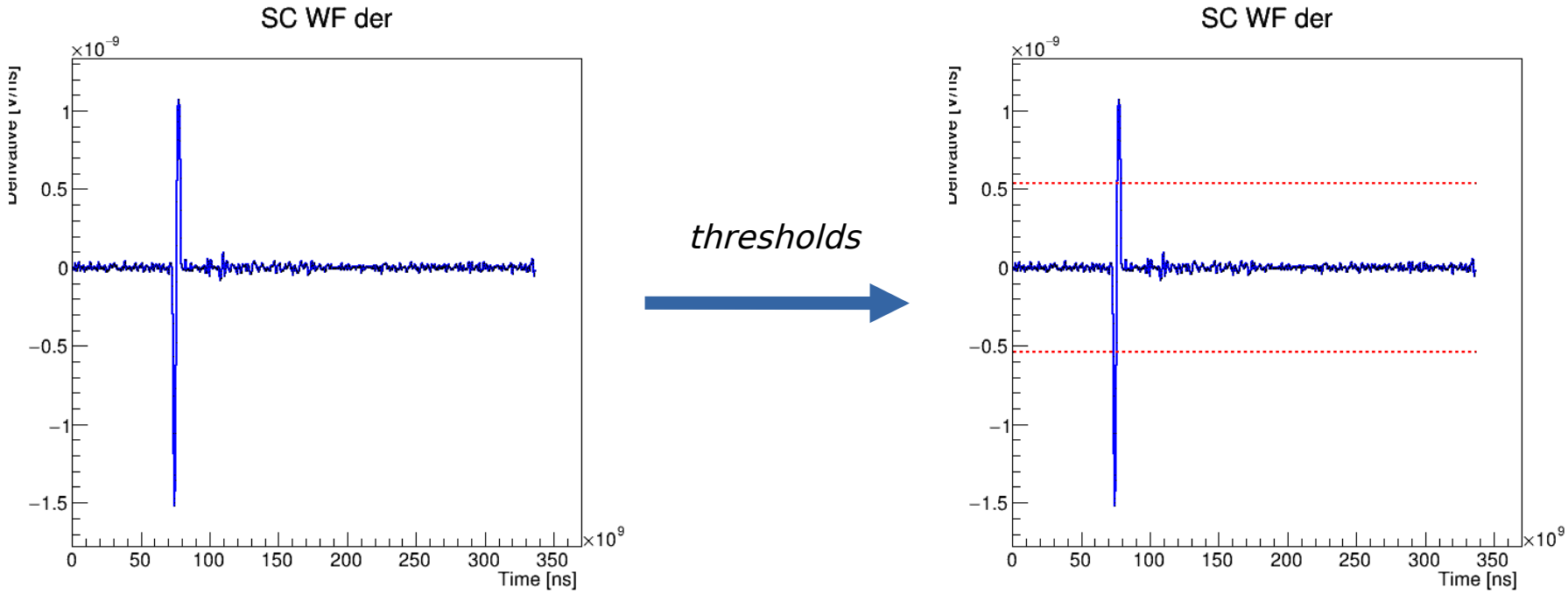
∀ point i: $D(i)=\frac{\Delta y}{\Delta x}=\frac{y[i+5]-y[i]}{x[i+5]-x[i]}$



- From every peak of the signal, we obtain **two** peaks in the derivative, whose height depends on the steepness of the primitive signal
- Baseline noise and overshoot are negligible with this method using 5 consecutive points in the elaboration of the derivative.

Derivative method, 2

Then a **threshold** is defined as the 50% of the positive peak height.

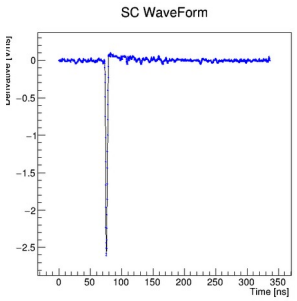


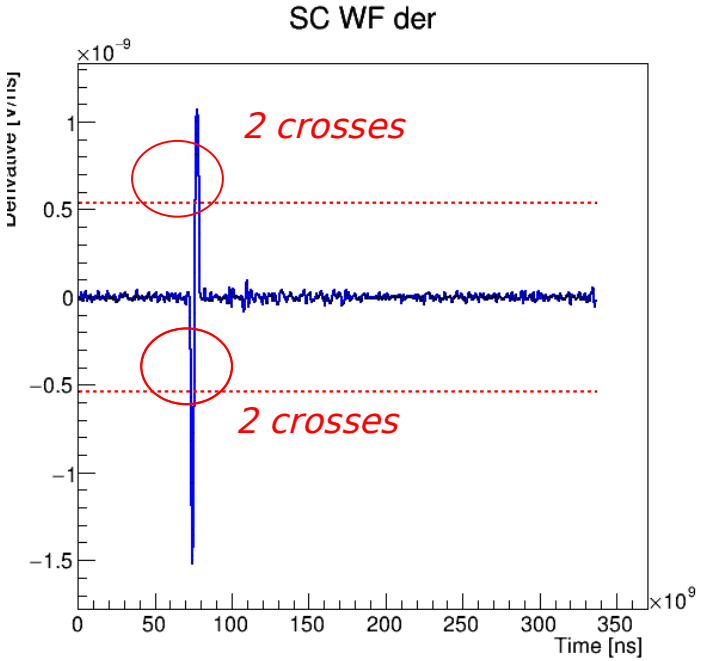
- The reverse value is used for the negative threshold.
- 50% is choosen to have a good distance between baseline and so for a good discrimination

Derivative method, 3

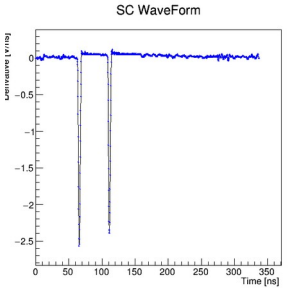
A single signal intersecates the thresold lines **4 times**: two for the positive and two for the negative ones.
For a **pileup** signal the intersections are **more than 4**.

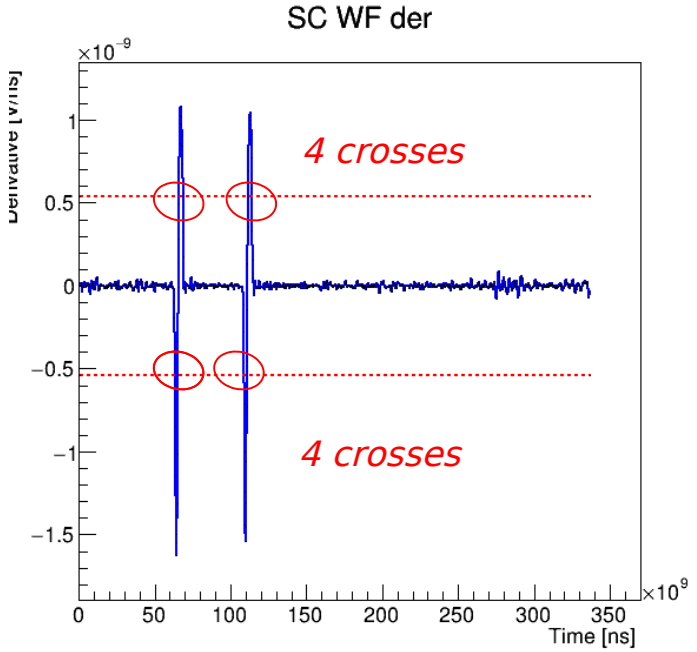
- This is at the basis of the **constant threshold discrimination method**.





Single event: 4 crosses





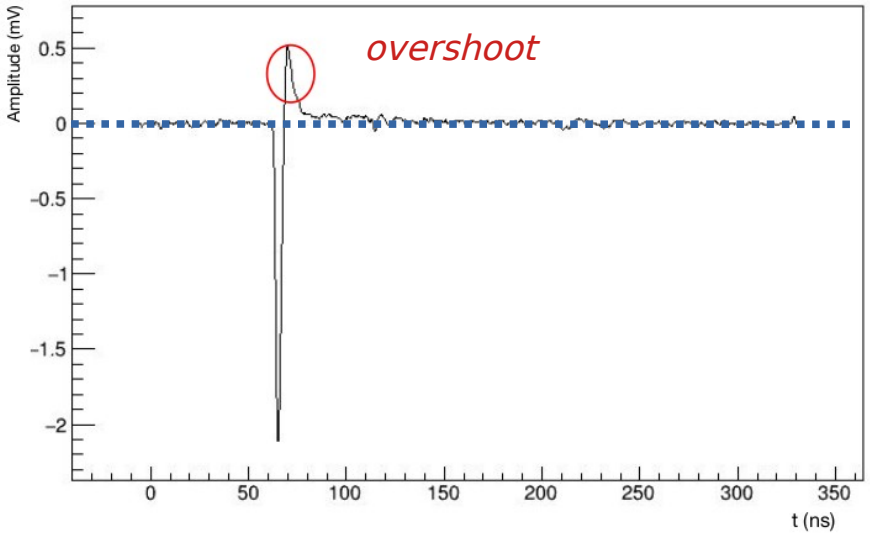
pileup event: >4 crosses

Charge method, 1

Another method to discriminate pile up is the **integral method**.

- Integrating the signal of the SC, it is possible to obtain the value of the **charge** of the crossing particle, from which discriminating single and pile up events.

- The signal integral is computed excluding the samples with ***amplitude > baseline + 2 st.dev***

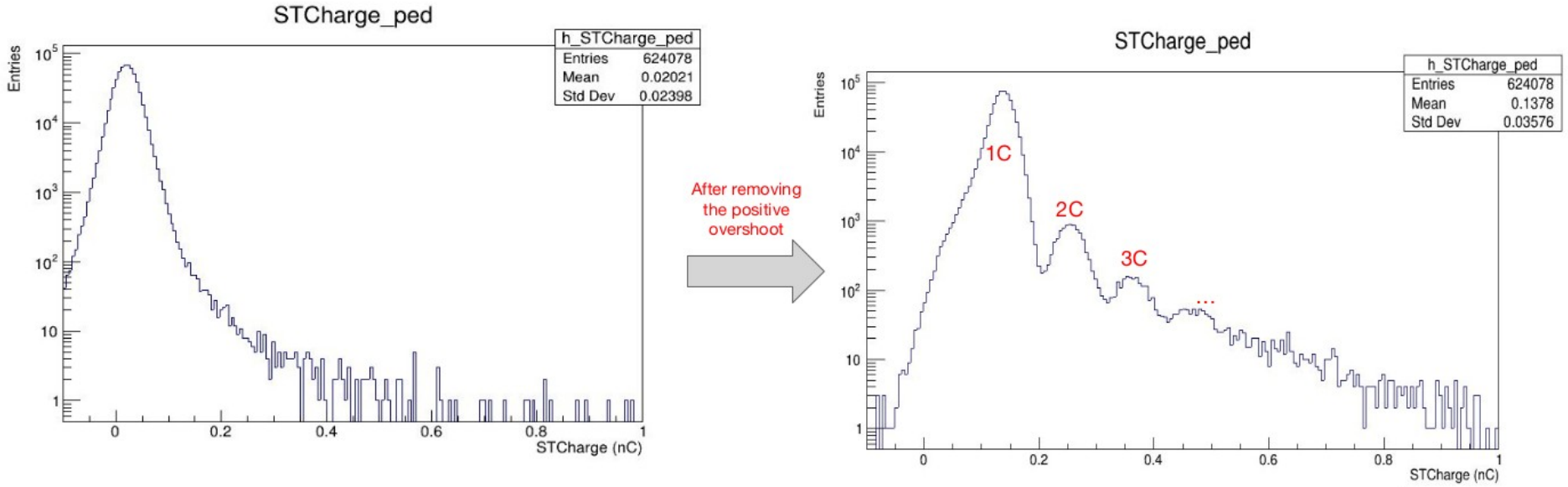


Credits to Marco Bon

- In this way we remove the positive **overshoot**, a positive peak of the signal due to electronics which is different from event to event.

Charge method, 2

Removing overshooting, it is possible to distinguish charges of single events and pileup ones as separate peaks in the histogram of charges.



Credits to Marco Bon

To filter out pileup events, a cut can be introduced in the value of charge. In this case as c < 0.2 nC.

This method is developed by Giacomo Traini et al.

And under implementation on Shoe