Anode analysis routine

Nicolás Sánchez

November 25, 2025

1 Introduction

In this document, we explore the periodicity of detected signals in the anodes after the coincidence analysis. The goal is to identify any underlying patterns or periodic behaviour in the data that allow us to better understand the nature of these signals.

2 First Routine

The analysis is performed using a C++ routine that processes the data from the ROOT files. The routine works as follows:

- It reads the data from the ROOT files corresponding to different run numbers.
- It applies a threshold to only retain signals with an amplitude above a certain value ($abs(amp) > 400$ and $tof > -900$ in our case). We will take into account negative amplitudes in order to see if they correspond to real signals or just noise.
- We create a structure to store the relevant information of each signal, namely the run number, BunchNumber, time, psTime, PSPulse, pulse intensity, detector number, time-of-flight and amplitude.
- We will also load the PKUP signals, which are stored in another tree in the same ROOT file. We will use these signals to correct the time-of-flight of the anode signals, since they vary because of the different discrimination levels of the DAQs.
- The routine then applies a time-of-flight correction to each anode signal based on the corresponding PKUP signal. This is done by finding the PKUP signal with the same beam-related parameters as the anode signal and subtracting its time from the anode signal's time-of-flight.
- A sorting algorithm is applied to arrange the signals in a chronological order based on their time stamps. In addition, we also sort the signals with

respect to the other variables (except for amplitude and detector number) to facilitate the identification of coincidences, since these value must match between the different signals in order to establish a coincidence.

- The routine then identifies coincidences by grouping signals that occur within a specified time window (time_for_coincidence = 10 ns in our case). This can be done just by iterating through the sorted list of signals and checking the time difference between one signal and the consecutives. When the time difference is less than the specified window, the signals are in coincidence.

- The routine also handles ”lonely” signals that do not have any coincidences. This is needed since our algorithm depends on the choice of the first signal to start the coincidence search. To arrange this, we will check every unused signal after the coincidence search and see if it can be merged with any of the already found coincidences. If the time difference between the lonely signal and any signal in a coincidence is less than the specified window, we merge them.

- Finally, the routine search for ”merged” coincidences, which are groups of signals that correspond to the same event but were not initially identified as such. This is done by checking if any signal in a certain configuration can be in coincidence with any signal in another configuration. If so, we merge the two configurations.

- The final configurations of coincidences are then stored in a new ROOT file for further analysis. These output file contains a tree in which each entry corresponds to a coincidences configuration, with branches for the multiplicity of signals (number of hits) in each detector and the amplitude and time-of-flight for each of them. We also store the beam-related information (RunNumber, BunchNumber, etc.). In addition, we also store the total multiplicity (sum of the number of hits in all detectors) in a separate branch.

3 Second analysis routine: γ flash selection.

The second analysis routine focuses on selecting the γ flash events from the previously obtained coincidence data. The steps involved in this routine are as follows:

- The routine reads the coincidence data from the ROOT files generated by the first routine.

- It applies selection criteria to identify γ flash events. This involves checking for correlated signals across all detectors within a very short time window, indicative of a γ flash event. These events come from the prompt radiation produced when the proton beam hits the spallation target. This radiation can arrive at the detectors, since these gamma rays are not deflected by the magnetic field used to separate charged particles. With this radiation, pair production can occur in the tube before the experimental area, thus relativistic particles are produced that can reach the detector area before the neutrons. These events can thus be used to calibrate the time-of-flight of the detectors by adjusting the distribution. Since our resolution of the acquisition system is around 1 ns, we have to take into account that the γ flash events will not arrive at exactly the same time in all detectors, because the time spent by light traveling from the first to the last detector is bigger than the resolution, thus corrections to the distance to the spallation target are needed.

- The routine then stores the selected γ flash events in a new ROOT file for further analysis. This output file contains a tree with branches for the multiplicity of signals in each detector, the amplitude and time-of-flight for each of them, as well as the beam-related information.
- Then, for each detector, we will create histograms of the time-of-flight distribution of the γ flash events. These histograms allow us to fit the peak corresponding to the γ flash and determine its centroid. This centroid will be used to calibrate the time-of-flight for each detector.
- In addition, we will apply the time-of-flight calibration to the original coincidence data. This involves adjusting the time-of-flight values for each detector based on the determined centroids from the γ flash histograms.

4 Quick checks

4.1 Gamma flash selection

The first thing we can check is to obtain the gamma flash peaks for each detector.

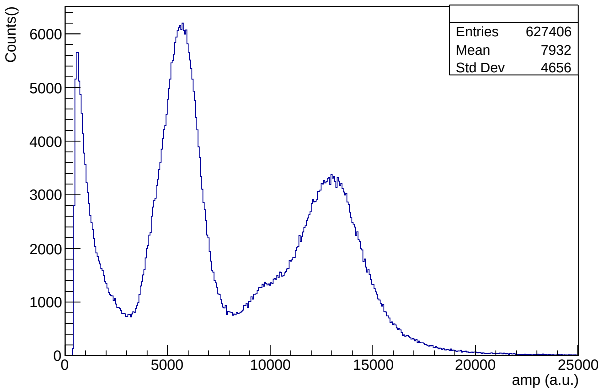


Figure 1: Gamma flash selection for detector 0. We can see 3 distinguishable peaks, which correspond, from left to right, to low amplitude noise, to the parasitic bunches and to the dedicated ones.

In fig. 1 we can see the gamma flash selection for the first detector. In order to check the quality of the selection, we can plot the time-of-flight correlation between two detectors, as shown in fig. 2. We can see a clear correlation between the two detectors, which indicates that the selection is working properly.

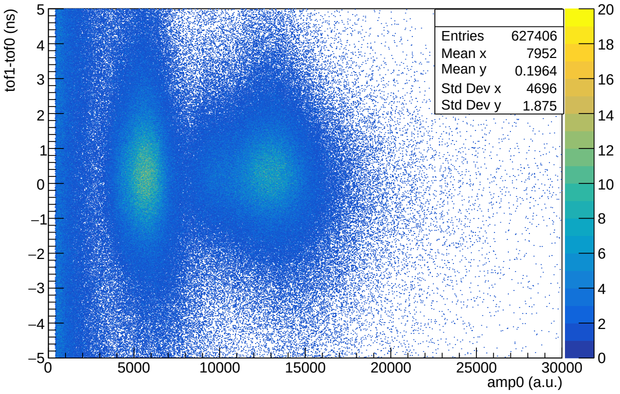


Figure 2: Time-of-flight correlation between detector 0 and detector 1 and the amplitude of the first detector. We can see that the first peak (at low amplitudes) does not show correlation with the times, whereas the other two are clearly centered around zero.

With this gamma flash selection, we can obtain a calibration in amplitudes. In order to do this, what we can do is to fit the gamma flash peaks for parasitic pulses for each detector. Once we have the centroid of each peak, we can align them. We will align them to the mean value of all centroids. We will get the ratio of amplitudes between the different detectors and the reference centroid and perform the calibration for all gamma flash events. Then, we will check for the dedicated peak and the low amplitude region to see if the alignment is correct.

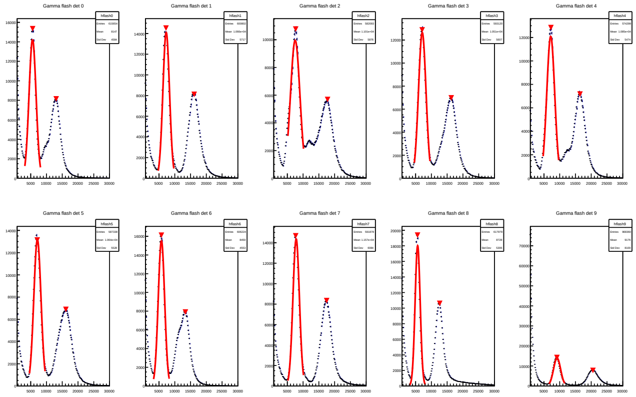


Figure 3: Calibrated gamma flash for all detectors (parasitic pulses).

As we can see in fig. 3, the fit and peak selection is satisfactory for all detectors. We can also check the dedicated pulses and the low amplitude region to see if they remain also aligned.

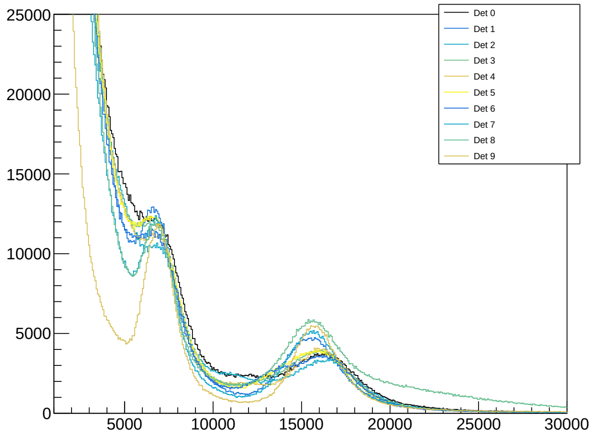


Figure 4: Calibrated gamma flash for all detectors (dedicated and parasitic pulses).

In fig. 4 we can see the calibrated gamma flash for all detectors, including both dedicated and parasitic pulses. The alignment is satisfactory, and the dedicated pulses are well-defined. Regarding the low amplitude region, we can see in fig. 5 that the alignment is also good for most detectors, although there is some spread due to noise.

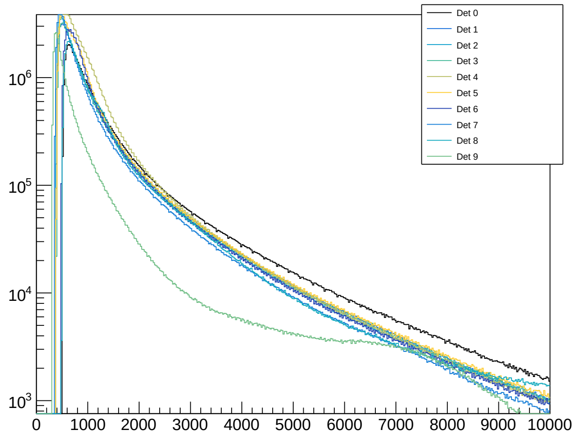


Figure 5: Calibrated gamma flash for all detectors (low amplitude region).

5 Fission event selection

Once we have calibrated the time-of-flight using the gamma flash events, we can proceed to select fission events. Fission fragments are accelerated by the Coulomb repulsion between the two nascent fragments, and emitted in opposite directions at high velocities. Therefore, we expect to see a signature of fission events in our detectors if two adjacent anodes detect signals almost simultaneously. To identify these events, we will look for coincidences between adjacent detectors within a very short time window (about 10 ns). Since fragments are of comparable mass and charge, we expect their amplitudes to be similar. We can also take advantage of the cathode signals to further refine our selectio, as the relation between anode and cathode time and amplitude signals can help us distinguish between real fission events and other types of reactions. First, we can use the fact that the time propagation of signals along the delay line is constant. Thus, we can impose the conditions

$$(tof_l - tof_{A_i}) + (tof_r - tof_A) = t_p, \tag{1}$$

and similarly for top and bottom signals, where t_p is the propagation time along the delay line. We must note here that the anode signal must serve as a reference for the cathodes, so we are neglecting the drift between anode and cathode signals (which is very small compared to the propagation along the delay line). In addition, we can also impose conditions on the amplitudes of the cathode signals. In theory, the ration of left-right and top-bottom amplitudes should follow a linear distribution with respect to the time differences, which is a consequence of the attenuation of the delay line. Thus, we can impose the conditions

$$\frac{A_l}{A_r} \in (0.5, 2.0), \tag{2}$$

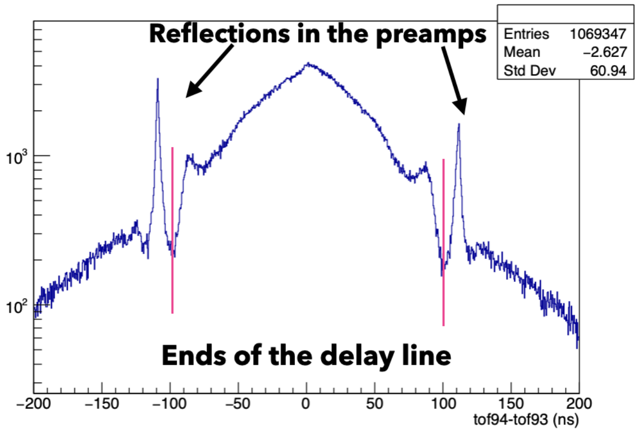
and the same applies to the top-bottom ratio. // Finally, we can also impose conditions on the time difference between left-right and top-bottom signals, which basically states that the signal must fall within the physical limits of the detector. Thus, we can impose the conditions

$$|tof_l - tof_r| < \frac{L}{v} \tag{3}$$

and similarly for top-bottom signals, where L is the length of the delay line and v is the propagation velocity of the signal along the delay line. We can compute the speed along the delay line by plotting the time difference and looking at the sharp peaks at +-110 ns, which stem from the reflections in the preamp. We can obtain the speed by setting the propagation time to be the mean of the two peaks value (in absolute value) and dividing the detector length by this time.
Finally, we can use the observable of the amplitude ratio between the sum of amplitudes of the two preamps of each cathode and the amplitude of the anode signal. This observables are clearly correlated, and allow us to distinguish between real fission events and other types of reactions. We can impose the condition

$$\frac{A_l + A_r}{A_a} \in (0.2, 1.8). \tag{4}$$

□



□

Figure 6: Computation of the speed along the delay line.

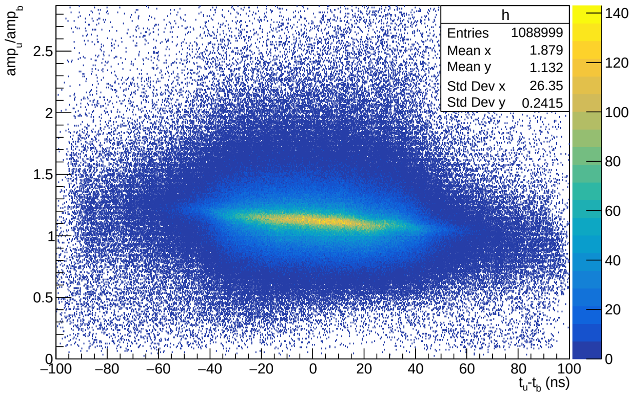


Figure 7: Attenuation of the delay line for the uranium target detector (PPAC 8). Linear behaviour as expected.

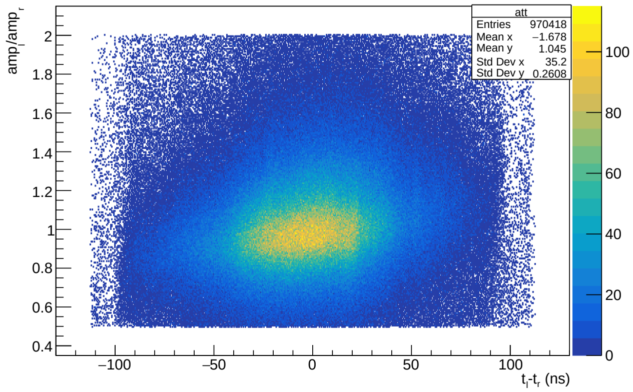


Figure 8: Attenuation of the delay line for the cerium target detector (PPAC 0). We can see that the selection is way worse for this detector. Why? Probably has something to do with the kinetic energy of the fragments being lower for cerium... ASK!

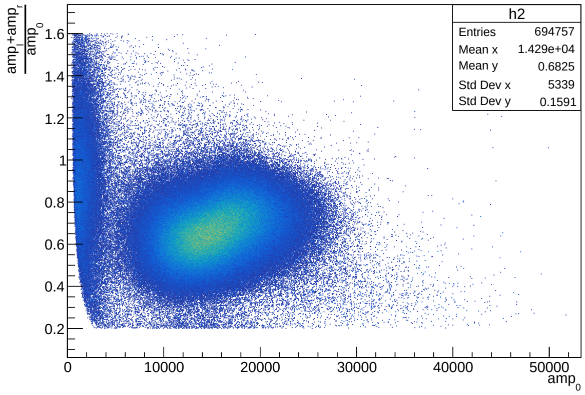


Figure 9: Correlation between the amplitude of the anode and cathode signals for uranium target detector. We can clearly distinguish between fission and non-fission events.

Once we have justified our selection criteria using uranium, we can proceed to apply them to the data and obtain the fission events also for cerium.