Pulse-shape discrimination for gamma-neutron separation

Second School on Dark Matter and Neutrino Detection

July 10, 2024

Plastic or liquid scintillators are very powerful fast neutron detectors and have been extensively employed in nuclear and particle physics applications. For example, in experiments such as those designed to measure the response of liquefied noble gases (like Liquid Argon LAr) to low-energy nuclear recoils induced by neutrons, they are used as the main neutron detectors. Even in the presence of ubiquitous gamma radiation, scintillators show excellent photon-neutron separation power through the Pulse Shape Discrimination (PSD) technique. In this laboratory activity, students will study the time response of a scintillator detector to a gamma source and learn to construct such pulse-shape parameters using an experimental setup that includes nanosecond time-resolution electronics. Finally, neutron and gamma population data will be analyzed to demonstrate separation power.

1 Experimental Setup

In the Figure 1 its is show the instrumentation and their connection as used in this experiment. They are:

- EJ-276 PSD Plastic Scintillator [1].
- PMT Hamamatsu R7724 negative HV. Max operation 2000 V [2].
- High Voltage Power supply ORTEC 556 (up to 2900V ± 3 V) [3].
- Cables RG-62 A/U to power PMT and to send pulses from PMT anode output into digitizer.
- Digitizer Caen 2740/2745 (64 channel 16 bits)(50 ohms coupling) [4].
- MIDAS software for data taking [5].
- red-midas code for data reconstruction.
- Radioactive source: Americium 241Am, mono energetic photons of 59.54 keV.

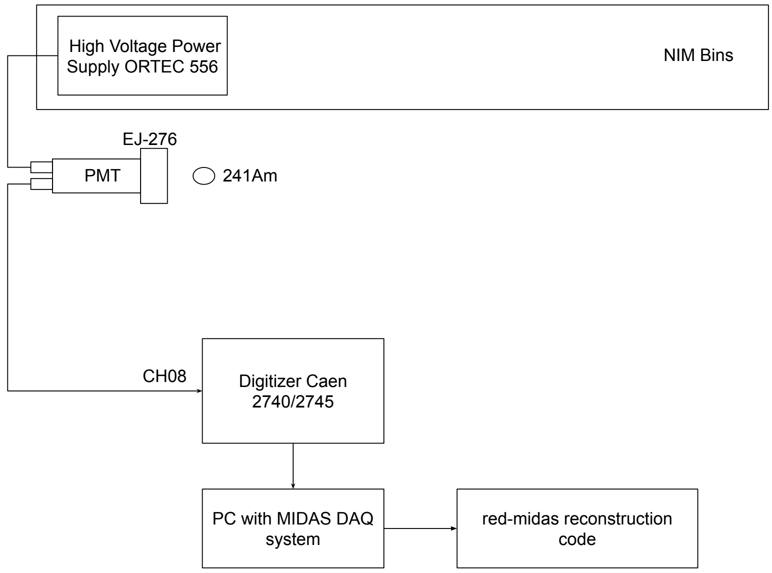

Figure 1: Instrumentation and their connections.

2 Experimental Proceeding

1. Place the ORTEC 556 into the NIM Bins [6] and feed its power source (appropriated 110 or 220V). Check if it was selected to apply negative voltage on PMT. You can check this by looking at the back of this module.

2. Connect cable in the output terminal from the back of the High Voltage Power Supply in PMT HV power input.

3. Connect the other cable from the PMT output anode to oscilloscope channel. Check the pulses displayed on the screen. Observe the pulse characteristics: rise time, peak height, fall time.

4. Take the PMT anode cable exit out from the oscilloscope and put it directly into digitizer Channel 8.

5. From DAQ computer: open MIDAS data acquisition system that communicates with the digitizer.

6. Go to section ODB and load the following configuration file: SchoolExp07.json.

7. Click on start run. Check rating of X events/s.

8. After accumulate some statistics, stop this run.

9. Now we are ready to reconstruct our raw data. But before that, lets define Pulse Shape Discrimination.

3 Pulse Shape Analysis

Pulse Shape Discrimination (PSD)[7] is a tool used to distinguish particle that interact within a given detector. Consider gammas and neutrons interacting with a scintillator. On average, the pulse of gammas decay faster then neutrons, so a shape analysis determine a PSD characteristic for each particle population. In Figure 2 we represent an average pulse for gamma and neutron particles. The difference in shape is prominent when pulse integration.

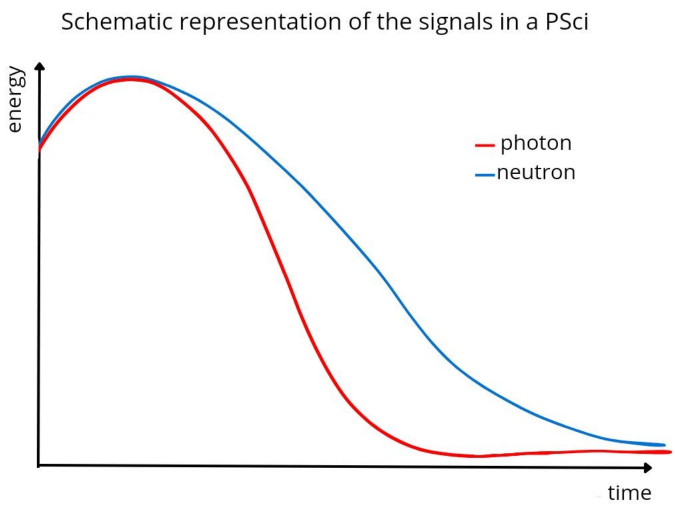

Figure 2: Average Pulse formed in time for gammas and neutrons. Gammas decay faster then gammas.

The pulse shape discrimination is defined as:

$$PSD = 1 - \frac{\int_{\Delta t_s} Sig(t)dt}{\int_{t_i}^{t_f} Sig(t)dt} = 1 - \frac{Q_s}{Q_l} \quad (1)$$

where $t_i < \Delta t_s < t_f$, and $Sig(t)$ is given in ADC (or in keV) by the digitizer. We call the whole interval from t_i to t_f the longer time window of the pulse (the integral is Q_l), while the shorter time window is Δt_s (the integral is Q_s). This shorter time window may be defined at the beginning the pulse rise time. If detector is calibrated (this is the case for our experiment), we can plot the PSD in function of particle energy (which is related to pulse height).

We can also work with the $fprompt$ parameter, given by:

$$fprompt = 1 - PSD = \frac{Q_s}{Q_l}, \quad (2)$$

where this just invert the PSD relation given previously.

3.1 Data Reconstruction

Now after data taking, we should reconstruct data with red-midas code. This opens raw data where the pulses were recorded in the disc and then it starts to calculate baseline for each pulse and then determine pulse height and signal integration for the PSD.

- cd to red-midas directory. The reconstruction command is: bash rec.sh 294 PSci (change 294 to your run number).
- check pulse traces with the following command: ./RedLevel1 -s 1 -r 294 -p.
- open root and plot PSD with the following commands: .L analysis/ExpSchool01.cpp and then PlotPDS(294).
- Time Window integration configurations are stored in cfg/detector.cfg. Inside this file, find the detector label PSci0, go to seventh column and change the window sample (1 sample equals to 8 ns). The longer window is located at fifth column and it is set to 100 samples.
- Reconstruct the same event again but with the following command: ./RedLevel1 -s 1 -r 294 -o results.red/run_294_1.root

Congratulations, now you know how to calculate PSD for photons.

3.2 Neutron and Gamma Separation Analysis

Since we do not have a neutron source in our experiment 7, we provide a root file with data taken from radioactive source Californium 252 (252Cf). This is a source of gammas and neutrons with energy ranging from tens keV to few GeV. In the following we describe the experimental setup used for this data and then instructions to run scripts that will open it and plot data for PSD analysis.

3.2.1 ReD Experiment

The Recoil Directionality Experiment (ReD) is a sub group of DarkSide collaboration and it has a three-fold goal: check if a dual phase Liquid Argon (LAr) Time Projection Chamber (TPC) has sensitivity to the direction of Ar recoil, characterize the response of the LAr TPC to low-energy recoils (few keV) and act as a test bench of the technical solutions for DarkSide-20k TPC.

In Figure 3 it is shown a scheme of ReD experimental setup. The radioactive Californium ^{252}Cf is source of gammas and neutrons through spontaneous fission with a probability of 3% where for each fission generates on average 18 photons and 4 neutrons as well as fission fragments.

Neutrons generates a low energy nuclear recoils inside the TPC and by the geometrical configuration, these neutrons are detected by the Plastic Scintillator PSci as shown in the Figure 3. **We use the PSD technique to select neutrons in the PSci that were scattered in LAr TPC. So this is the main role of PSD analysis in ReD experiment.**

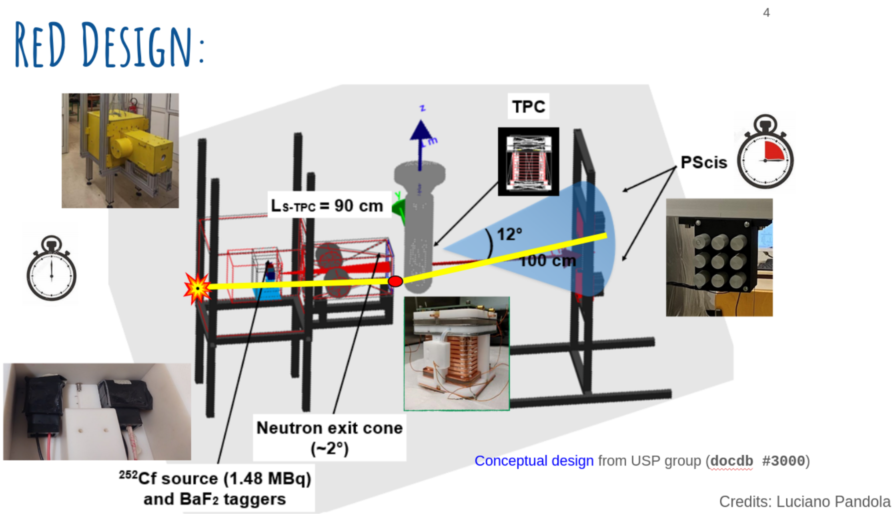

Figure 3: ReD experiment configuration to take data from 252Cf whick is source of gamma and neutrons. In this analysis we are interested in the PSD from the neutron spectrometer PSci.

In the following we give a detailed description on how ReD works: Two Barium Fluoride detectors (BaF_2) are located near to the 252Cf source so they detect the photons in the fission actin like taggers. After this, the neutrons are collimated by a lead apparatus in a 2° cone and then reach the TPC, located at 90 cm distance from source and the BaF_2s. The neutrons then interact with the Liquid Argon (LAr) nucleus inside the TPC, generating a low energy nuclear recoil that generates two signals inside the TPC, the S1 due to scintillation and the S2 due to electroluminescence. The neutrons than reach the PSci array, located at 100 cm distance of the TPC, dislocated of the center of the beam by an average of 12°. We have two PSci array, each one with 9 PSci, one above and the other below the beam axis. This geometry constrain neutrons that are scattered in LAr by a low angle, leaving an energy of a few keV in the TPC. The time difference between the BaF_2 tagger and the PSci is used to eliminate background interactions.

So, to acquire the data we used the coincidence between the BaF_2 and the PSci to select events. When a signal is detected by a PSci, the signal of all the 18 of them is also registered. In this setup, BaF_2 is used as a tagger and we have all PSci used as triggers and slaves. Our electronic setup permits us to have a time resolution of 2 ns, so each sample beam is equivalent to 2 ns in our data.

3.2.2 Data structure

There is a root file for each run and inside this files one can find a tree named *treetraces* with the basic information of the trace saved in it. The four variables inside the tree are:

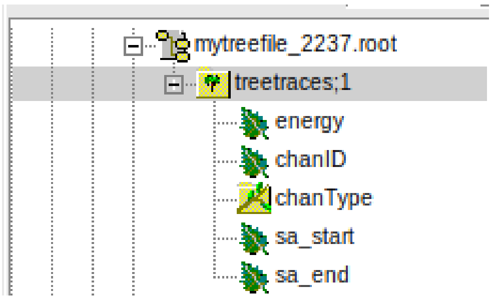

Figure 4: Structure of the tree with the traces information.

- energy: a **std::vector<float>** with the energy of the trace per sample in each bin;
- chanID: an **int** containing the channel number of the PSci which received the trace;
- chanType: a **std::string** containing the type of detector used (for this data set, we only used PSci's);
- sa_start: an **int** containing the absolute time the signal started;
- sa_stop: an **int** containing the absolute time the signal stopped;

3.2.3 How to use the code

First of all you need to put all the code files and the folder of root files in the same folder. In the material you received there are three files (trace_viewer, fprompt_analyzer, selection.runs) and one folder (threes_school) that contains all the root files mentioned above. The *selection.runs* is just a file containing a list of the run numbers used in this activity, used for simplicity.

The *trace_viewer* is a pulse visualizer, in which you can choose the run number and event number to visualize the trace detected in the trigger PSci in that event. The trace is shown in keV vs sample (2 ns). To use this code you need to open root inside the folder containing the code and load the file into root. Then, call the function TraceViewer, with a run and an event number. See Figure 5. If you choose an event number above the number of events in that run, the code will return you an error show showing the total number of events in that run, then you can choose a valid event number.

The *fprompt-analyzer* is the main code, used to analyze each event and calculate the fprompt for each trace, using the time windows of your choice. The

```
root [0] .L trace-viewer.C
root [1] TraceViewer(2237,20)
```

Figure 5: Loading script for trace viewer.

code returns a 2D histogram of fprompt vs. energy, where you can see all the particles plotted, and a 1D histogram of fprompt, where you can see the projection in fprompt of the previous 2D histogram. To use this code you need to open root inside the directory containing the code, and load the code into root. After that you can call the function FpromptCalc, which receive two variables, W_{short} and W_{long} to calculate the fprompt of the particles. See Figure 6. It's important to note that the choice of time windows can change considerably how good the particle populations are distinguishable from each other, so you can test different choices of windows to see how it changes. A good reasonable choice to start with a $W_{short} = 20sa$ and $W_{long} = 200sa$. It's important to choose a W_{long} considerably bigger than W_{short}, otherwise you won't see any population separation at all.

```
root [0] .L fprompt-analizer.C
root [1] FpromptCalc(20,200)
```

Figure 6: Commands to be entered in root to use the *fprompt-analyzer* code.

Congratulations, now you know how separate gammas from neutrons using PSD. Also there is a way to tune the best signal integration gates to better distinguish each particle population. This is done through the so called Merit Factor.

4 References

References

[1] https://eljentechnology.com/products/plastic-scintillators/ej-276

[2] https://www.hamamatsu.com/us/en/product/optical-sensors/pmt/pmt_tube-alone/head-on-type/R7724.html

[3] High Voltage Power Supply Ortec 556.

[4] Caen DT2745 digitizer.(https://www.caen.it/products/dt2745/)

[5] MIDAS (acronym for Maximum Integrated Data Acquisition System) is a modern data acquisition system developed at Paul Scherrer Institute - PSI and TRIUMF Canada's particle accelerator center. Installation webpage: MidasWiki Quickstart Linux (https://daq00.triumf.ca/MidasWiki).

[6] ORTEC NIM bin 4001A and 4001C

[7] G.F. Knoll, Radiation Detection and Measurement, 4th edition, John Wiley & Sons, Hoboken (2010).

[8] https://root.cern/