Quick ROOT tutorial

Benjamin D. Suh

May 23, 2019

1 Introduction

This is a quick tutorial on how to use ROOT to view waveforms, create spectra, and calibrate that spectra. I am by no means anything even close to resembling a coding expert, so there will likely be mistakes. The purpose of this tutorial is to give you a general starting point from which you can build your own scripts. Any comments, corrections, or suggestions are welcome. You can reach me at bdsuh@iu.edu, and I will try to get around to it.

2 Data Acquisition

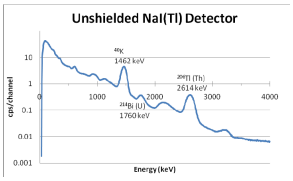

Figure 1: Sample NaI spectra. Source

The scripts that were used are updated to ROOT version 6.14/02. Data was taken using a NaI(Tl) crystal biased to 1300V. ADAQ was set to an offset of D050 and a trigger at 580. Run is 5 minutes long.

NaI(Tl) was chosen for its ease of set-up as well as the two prominent peaks from potassium and thallium that allows us to perform a two-point calibration. A sample spectra is shown in figure (1). NaI has a 250ns decay time, which would be visible if I converted the waveforms to a time scale.

Data has already been converted from .adaq.root files to .root files. I assume the reader has ROOT installed and is able to follow along with the text.

3 View Waveform

The associated macro for this section is *viewWaveform.cpp*.

Once we have our .root file, often the first thing we want to do is look at some waveforms to make sure our signal is healthy. To run a macro, first open a terminal and navigate to the folder where you have the macro saved. Run the command "root -l" to start ROOT. Load the macro using ".L viewWaveform.cpp". Run a program inside the macro using "viewWaveform(TString file, const Int_t waveformNumber)" where "TString file" is the path to where you saved the .root file, and "const Int_ waveformNumber" is the event number of the waveform you wish to view.

Raw waveforms do not look like this. Instead, they sit on some non-zero baseline value. The viewWaveform script looks for the baseline and subtracts it from our waveform so that our pulse sits at 0. It does this by looking at the first 50 samples and averaging their values. We would expect

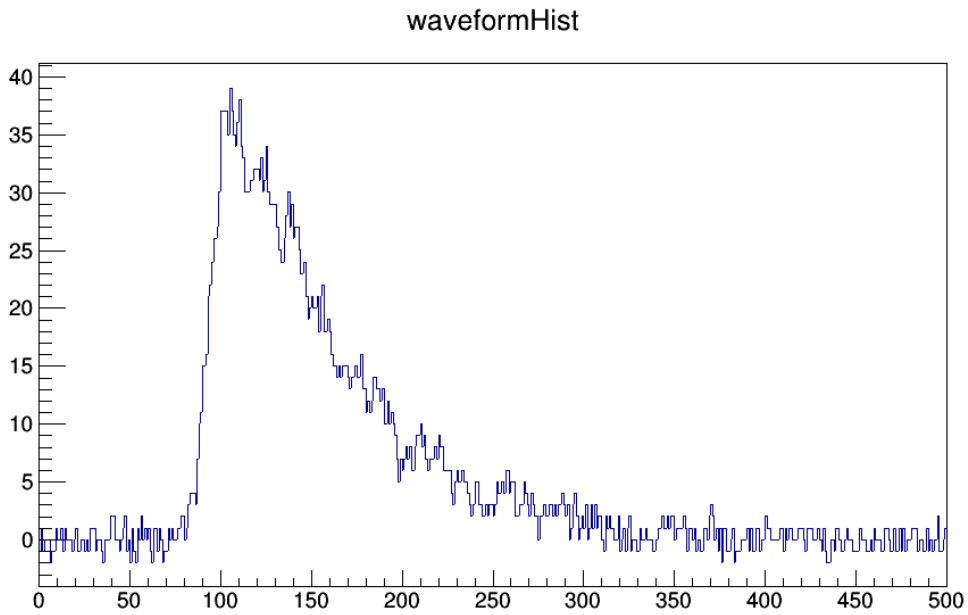

Figure 2: Example waveform

that if there were no pulse, the integral would be 0 since noise should be Gaussian distributed around 0. This method of baseline subtraction could fail if we choose too long a window such that it encompasses part of the signal or if there were a small signal below threshold just before our main pulse.

In this case, the raw waveforms are negative-going.We want to flip negative-going pulses so that when we integrate them, we get a positive value.

4 Spectra

The associated macro is *viewSpectra.cpp*.

Figure (3) shows the output of the macro. A spectra is a histogram of waveform energies. Each entry in the histogram is a single event. We can calculate the integral of that event, which is related to the energy of the photon which triggered it. If we histogram these events, we get a spectrum. Peaks will show up where something happens. For example, NaI has an internal source of ^{40}K, which translates to 1460keV gammas. An abundance of these gammas creates a peak compared to background. Even if there is no source, the detector will still trigger due to thermal noise. Heat will cause electrons on the photo-cathode to be knocked off and trigger the detector. These are low-energy events.

Note that the spectra produced here are uncalibrated. The x-axis is in units of ADC samples*bits. The associated file contains two ways to calculate the spectra. We could either loop through each waveform and calculate the integral by summing up each value. Alternatively, we

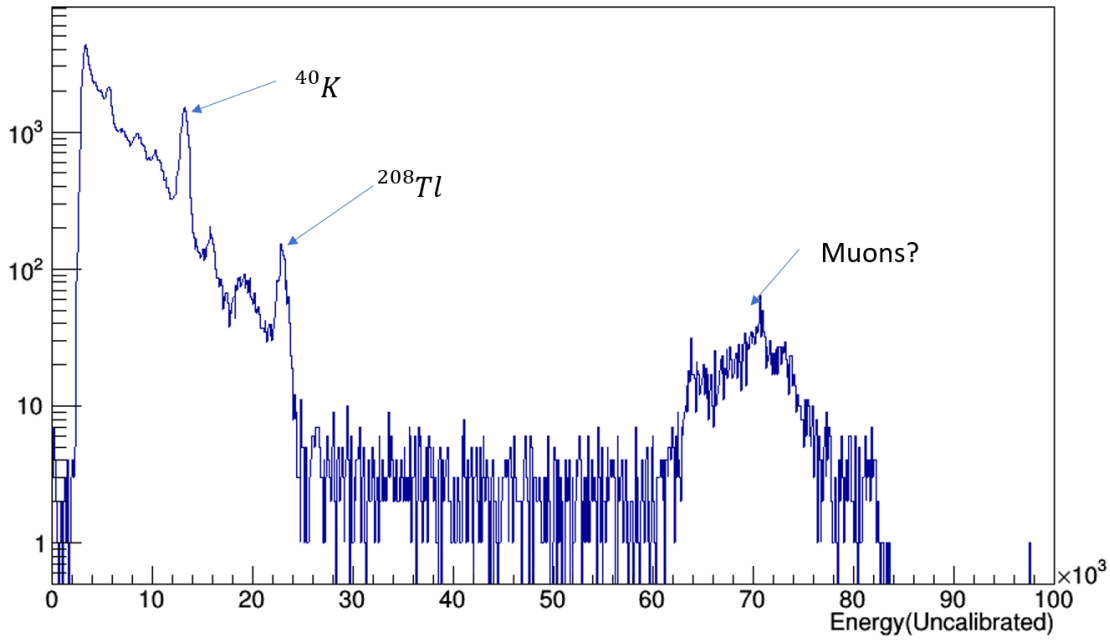

Figure 3: Produced sepctra

could read the values from a TTree since the integral values are already calculated when converting the files to .root. The reason why you wouldn't want to do the second method is if you are looking at a different region of integration than I did. I looked at the region between 51 and 500 ADC samples, but if you want to look at a tighter region, you would need to redo the waveform analysis.

5 Calibration

Once we have our uncalibrated signal, we then want to calibrate them. As stated, NaI is good for this because it has two prominent peaks that we can use for a two-point calibration. We have a 1460keV peak from ^{40}K and a 2615keV peak from ^{208}Tl. We use a TFit to find a position for these peaks rather than eyeballing it. We then match these peaks to their associated energy and use the linear response of the digitizer to place every other point. Figure (4) shows a calibrated spectra. It should look the same as figure (3) with a different x-axis. We could then perform more calibrations to normalize the y-axis.

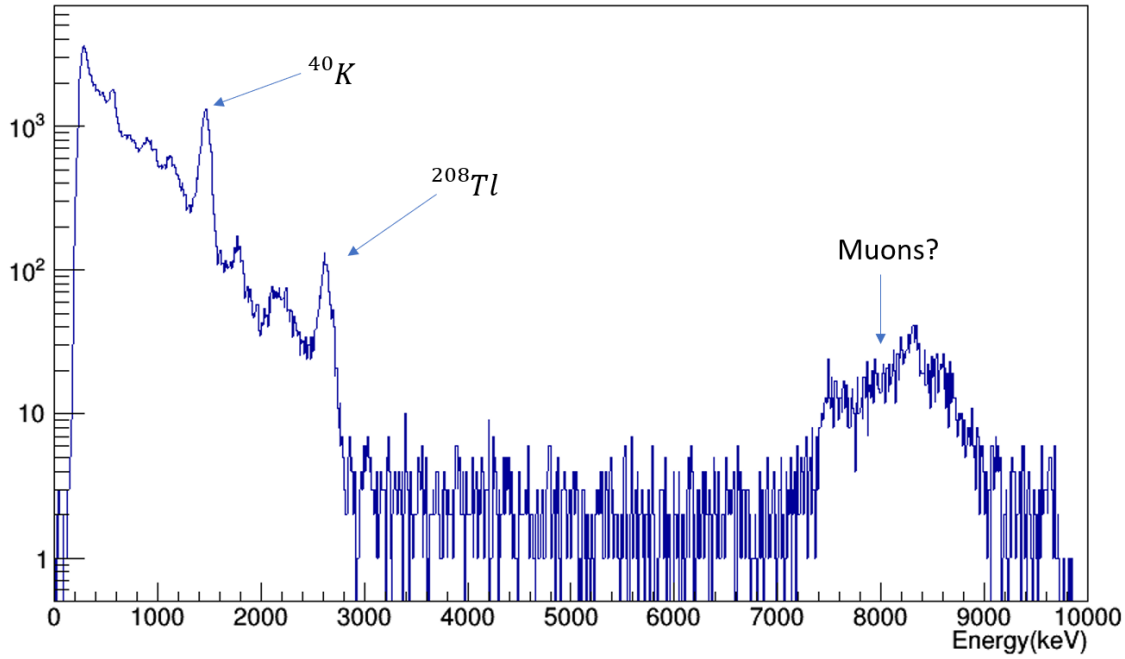

Figure 4: Calibrated Spectra