QDRT

Benjamin D. Suh

July 30, 2019

1 Introduction

This is a quick tutorial on how to use ROOT to view waveforms and create spectra. I'll be going over the basics including how to open a .root file, view and manipulate a TTree, and how to write a new TTree and output it to a file. 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.

I will assume that you've already installed ROOT. All scripts used here are updated to ROOT version 6.14/02. As always, in general, if you have questions on how to do something, google is your friend. The ROOT documentation is always quite useful.

Tutorial scripts can be found on my github here. A sample .root file to play with can be found here.

2 Data Acquisition

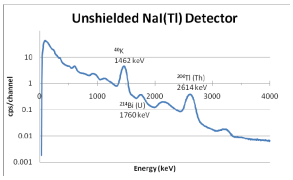

Figure 1: Sample NaI spectra. Source

This section serves both to explain the data as well as remind myself the settings used. Some of this may not be pertinent to your particular experiment.

The data provided was taken using a NaI(Tl) crystal biased to 1300V. The digitizer used was a CAEN DT5720 desktop digitizer. DC offset was set to D050 and trigger threshold was set to 580. Run was 300 seconds long. Conversion script updated as of 7/25/2019.

NaI(Tl) was chosen for its ease of set-up as well as the two prominent peaks from potassium (1460keV) and thallium (2615keV) that allow us to perform a two-point calibration. A sample spectra is shown in figure (1). Waveforms from a NaI crystal have a 250ns decay time.

3 View Waveform

Once we've taken some data, the first thing we want to do is look at some waveforms to make sure our signal is healthy. First things first, what is a waveform? We'll start by briefly describing how an oscilloscope works. An oscilloscope waits for some signal to pass a certain threshold. Once that oscilloscope is triggered, it takes a snapshot and shows what's on

the screen. For data taking, we use digitizers, which can be thought of as better oscilloscopes that can store data. A waveform can then be thought of as a single snapshot. Refer to figure (2) for how an oscilloscope works.

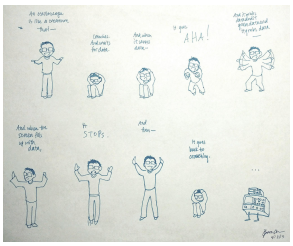

Figure 2: Oscilloscope comic. Courtesy Grace Chen

As a quick reminder, to run a script, first open a terminal. If you want to see what is in the current directory, type **ls -lrt**. To move to a different directory, use **cd**. To move one directory up, use **cd ../**. If we want to run a script, we must first navigate to the folder containing said script or navigate to the folder containing the data file. As we can see in figure (3), conveniently, these two folders are the same. We then start ROOT by typing **root -l**. The -l modifier means that we suppress some opening animation. We then load the macro using **.L scriptName**. In general, I try to have only one macro loaded at a time; however, this may take extra time if you're running remotely. Try not to load the same macro twice as ROOT gets mad at you if you do that. We then want to run a particular program. If you're not sure about the inputs, type the program name and a left parenthesis and then hit the tab key. Tab completion is a useful tool in terminal. I often use it to make sure I'm typing a file name correctly. In this case, tab completion will tell you the inputs (assuming whoever wrote the scripts gave the inputs descriptive enough names). Once we've typed the program to run as well as the inputs, just hit enter, and as long as there are no bugs, we should see a waveform.

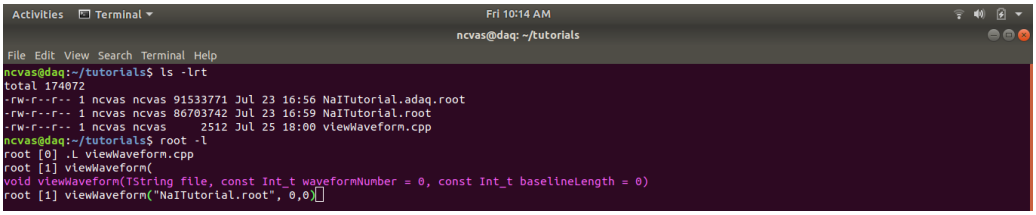

Figure 3: Running scripts

Figure (4) shows a raw waveform with no baseline subtraction. Note that the signal is offset from some amount (by about 620 ADC). In addition, the waveform is upside down. This is what is known as a negative going pulse, and so we will have to remember to invert the signal when we draw it.

There are two ways we could calculate the baseline. We could either take the average of the first x samples (here, we use the first 80 samples. Alternatively, you can set the baseline beforehand and use a constant value. For a signal from NaI where we expect to get only a single event per window, it is better to use the baseline-finding algorithm. However, if we have multiple signals in a window, it is better to use a constant baseline value.

One final thing we want to check is my claim that NaI has a decay rate of 250ns. The digitizer sampled at 250MHz, so each sample is 4ns. Once we have our waveform, we can go to the **View** tab and select **Event Statusbar**. This gives us the x and y coordinates under our cursor. The peak occurs around (132, 146). The signal drops by a factor of e at around 194 samples. This converts to 248ns, not bad.

One thing to watch out for is a saturated waveform (7). Here, we have a signal that is too large

for the digitizer to handle. We can tell this is what happens because the top of the waveform is flat. There are a couple ways to fix this. Either we lower the baseline (or because our signal is negative-going, raise it) or we could reduce the amplitude of the signal by reducing the voltage.

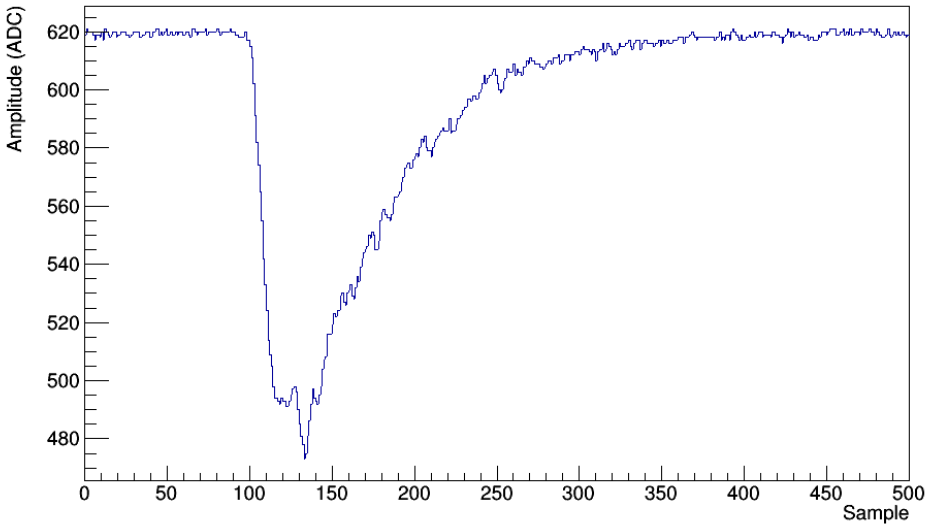

Figure 4: Waveform with no baseline subtraction

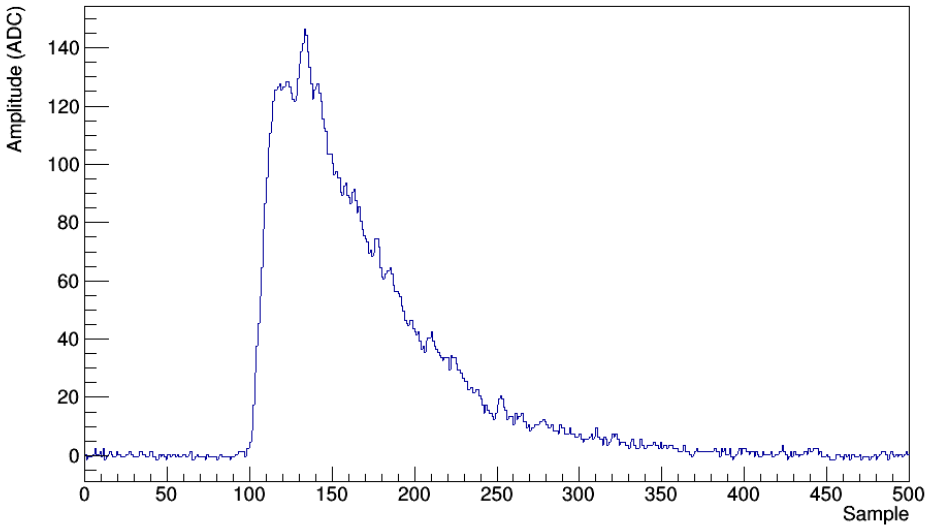

Figure 5: Waveform using baseline-finding algorithm

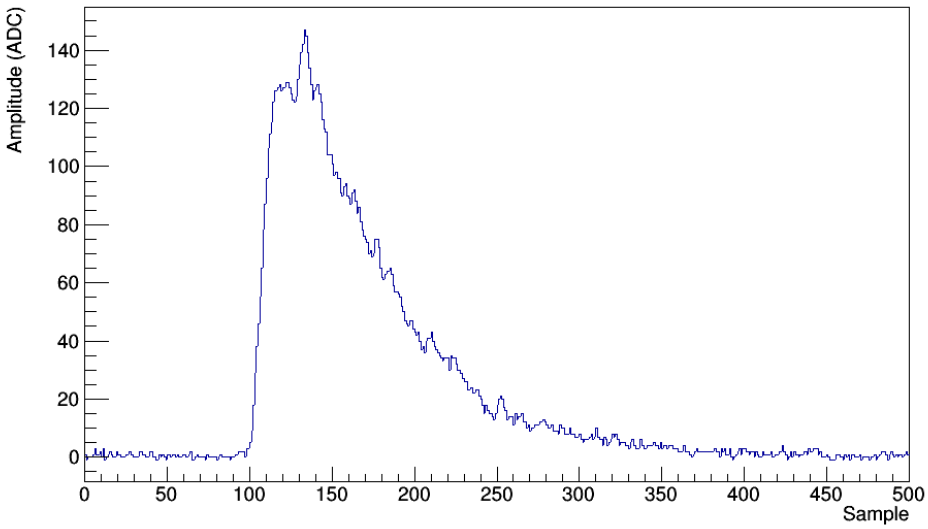

Figure 6: Waveform with set baseline

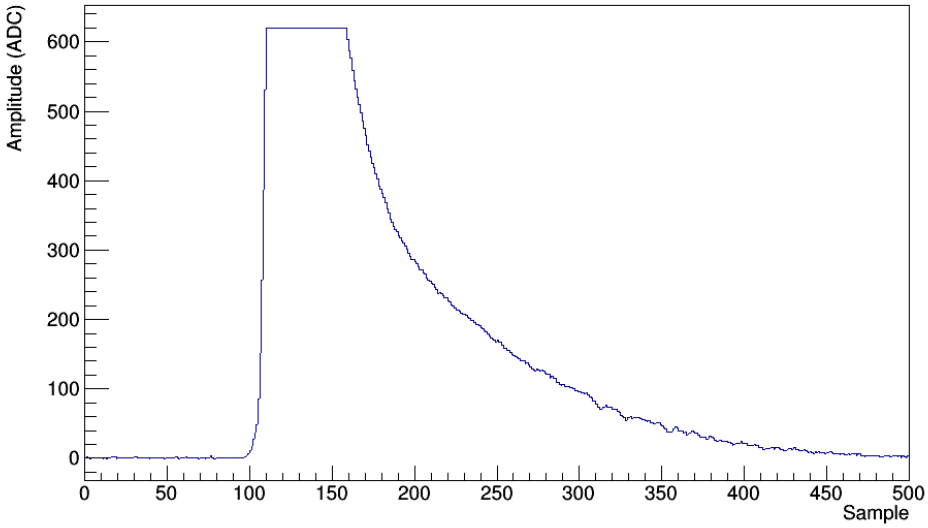

Figure 7: Saturated Waveform

4 TBrowser

One useful trick is to use TBrowser. Open ROOT and run the command **new TBrowser**. This should open a window. We can then inspect a file as shown in figure (8). We can view the branches as well as a histogram of their values. You may notice that one of the branches is named **waveform**. This is not the waveform from the previous section. Instead, what this shows is a histogram of all the values in every waveform. Viewing this branch will not show you a waveform.

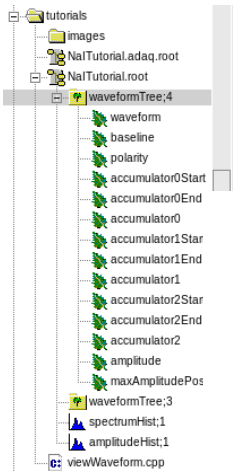

Figure 8: TBrowser

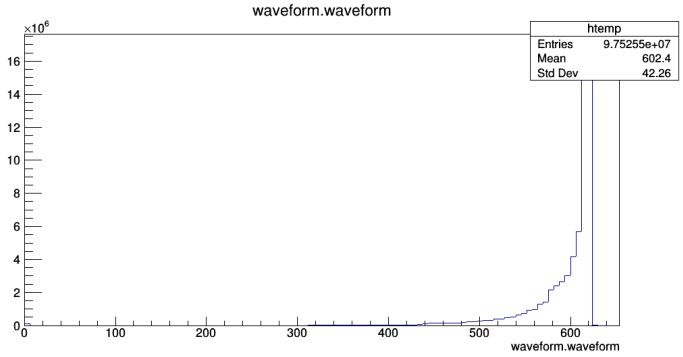

Figure 9: Not a waveform

5 View Spectra

Now we get to the real meat of this tutorial. Once we have a bunch of waveforms from a run, we want to know the aggregate information from that set of data. A spectra is a histogram of waveform energies. What we do is we take a waveform, find the integral in a certain region of interest. This integral is related to the energy of the photon that triggered it. Once we have our histogram, an example of which is shown in figure (10), we can look for peaks. These peaks correspond to interesting phenomena. For example, NaI has an internal source of ^{40}K, which produces a peak at 1460keV. 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.

Note that the spectra produced here are uncalibrated. The x-axis is in units of (ADC samples)*bits. If you want to calibrate, you would need to find two peaks, here ^{40}K (1460keV) and ^{208}Tl (2615keV). Once we have these two peaks, we can use these two points to form a linear relation between energy and integral. If we only had a single point, we could use an integral of 0 corresponding to zero energy.

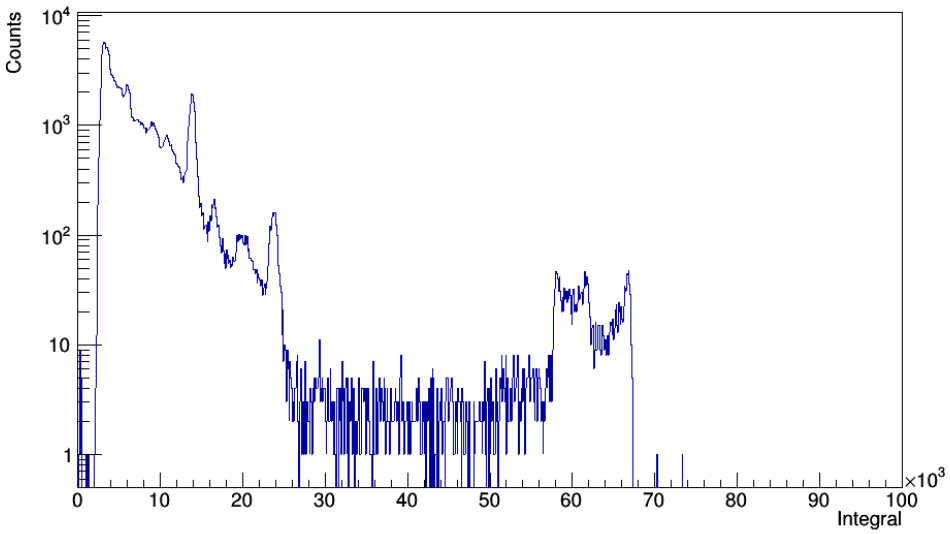

Figure 10: Spectra from accumulators

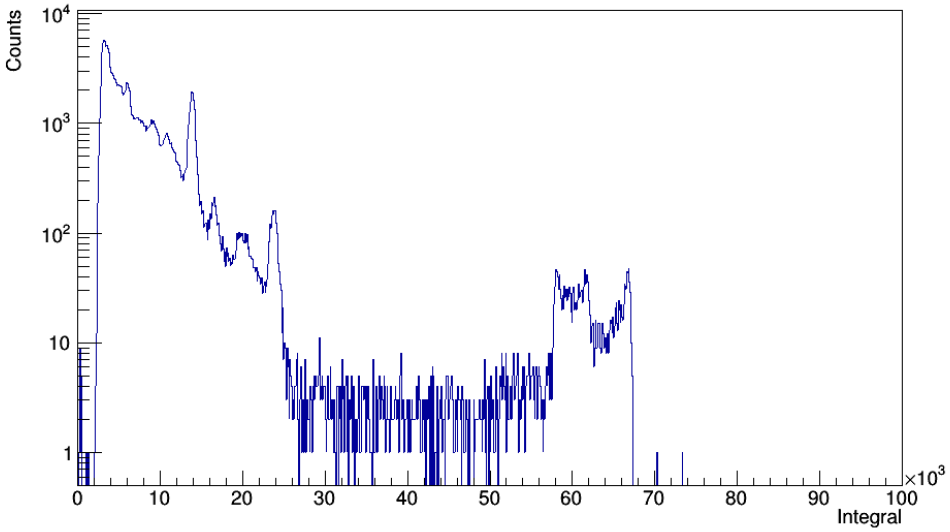

Figure 11: Spectra from integrating waveform over same region as accumulator

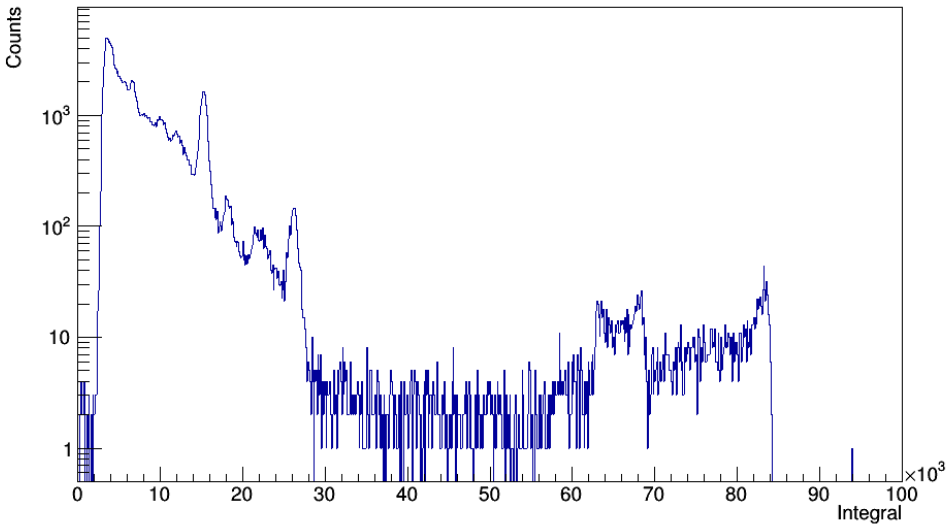

Figure 12: Spectra from integrating full waveform