SHMS Noble Gas Cerenkov Calibration

Cameron Cotton (University of Virginia)

February 2023

1 Introduction

This document provides a description of the SHMS Noble Gas Cerenkov (NGCER) calibration code written by Cameron Cotton. The calibration code was written and tested using ROOT version 6.22.06.

2 Overview

The purpose of the SHMS NGCER calibration is to determine the "gain" or conversion factor between the charge output by each of the NGCER's PMTs (in pC) and the number of photoelectrons (NPE) extracted from the photocathode of that PMT. The SHMS NGCER calibration method used in this script can be broken into three primary steps. First, selecting an appropriate data set to be used for this calibration. Second, applying cuts on the data set to get clean electron samples for each of the PMTs. Finally, using the clean electron samples to fit the pulse integral distribution for each of the PMTs with a function to determine the calibration constant. We will go into more detail on each of these steps now.

3 Selecting an Appropriate Data Set

In order to get the best results from this calibration, one needs to first determine a good data set to feed into the calibration code. Timing window and reference time cuts should already be set before performing this calibration. Below are some key features of a good data set:

1. The event distribution covers all 4 of the NGCER PMTs.
    - At some kinematics, only a subset of the PMTs will get reasonable event rates compared to the others. You should generally avoid using data taken at these kinematics, otherwise you won't be able to calibrate the low-statistics PMTs well.
    - This can be checked by looking at the P.ngcer.xAtCer and P.ngcer.yAtCer plots generated by the code. Specifically, look to see if the plots for one or more of the PMTs has significantly less events than the others. You should at least have on the order of $1E5$ events for each PMT.

2. The data set is comprised primarily of electron events.

    - For the NGCER calibration, we need a clean sample of electron events. Although it acts as the primary PID (Particle IDentification) detector, the Cerenkov itself cannot be used to make PID cuts on the data used for its own calibration. We then must solely rely on cuts on the Calorimeter spectrum to separate electrons from pions and other particles to obtain a clean electron sample.
    - Therefore, to minimize pion contamination and maximize the fraction of good electron events, it is recommended to start with a data set that is already primarily composed of electron events if possible.

3. The data set contains on the order of millions of events or more.

    - To get a good number of statistics for each of the PMTs, millions of events are recommended to be used, either from a single data set, or by chaining together multiple data sets at the same setting.
    - Because this code uses the RDataFrame object to work with the data, the calibration completes in mere seconds even when millions of events are used. Processing time should not be a limiting factor.

Once appropriate runs for the calibration have been determined, you may execute the calibration code and enter the run numbers for the runs you have selected (space separated) when you are prompted. The script defaults to look for the replay files using the path and file name pattern below. Feel free to change this if your directory layout or file naming pattern is different.

../../ROOTfiles/SHMS/CALIBRATION/shms_replay_cer_%i_-1.root

4 Applying Cuts for a Clean Electron Sample

Now that we have input our selected data set(s) to the calibration code, several cuts are applied to get a clean electron distribution for each PMT:

1. SHMS nominal momentum acceptance cut.

    - This is a cut on momentum relative to the SHMS central momentum setting (dp), stored in the branch P.gtr.dp. For the SHMS, this cut is $-10 < dp < 22$, removing events outside of the nominal acceptance.

2. Calorimeter PID Cut (Fig 1)

    - This cut separates electron events from pion background.
    - If you selected a data set following advice from the previous section, the pion contamination should be low and this cut alone should give a clean enough electron sample (for the purpose of this calibration).

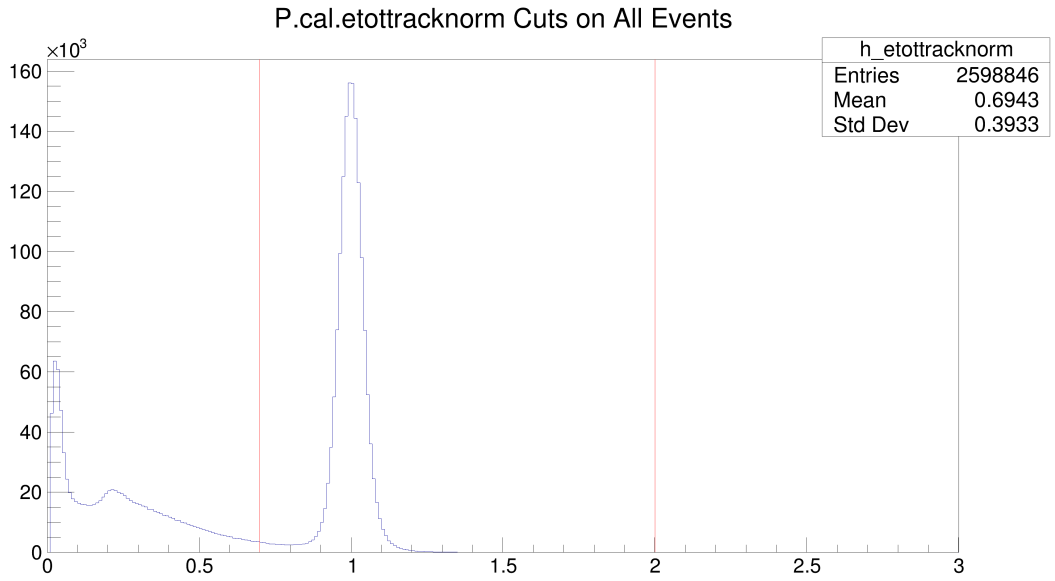

Figure 1: Sample run with a large number of electrons relative to background. Very loose cuts shown in red. Events to the left of P.cal.etottracknorm = 0.7 are assumed to be pions and removed.

3. Cerenkov Multiplicity and Position Cuts (Fig 2)

   - These cuts allow us to generate a data set for each PMT where we can reasonably assume that most of the Cerenkov radiation from the event was deposited into a single PMT and not spread over several PMTs. First, we apply a multiplicity cut to remove events that were detected by more than one PMT, or events that had multiple hits within the timing window (timing windows and reference times should already be set before this calibration). Then, for each of the PMTs, we remove events whose position is not in the same quadrant as the PMT that detected it, only passing events where we can reasonably assume that the detected signal came from Cerenkov radiation that was reflected squarely from the mirror corresponding to that PMT.

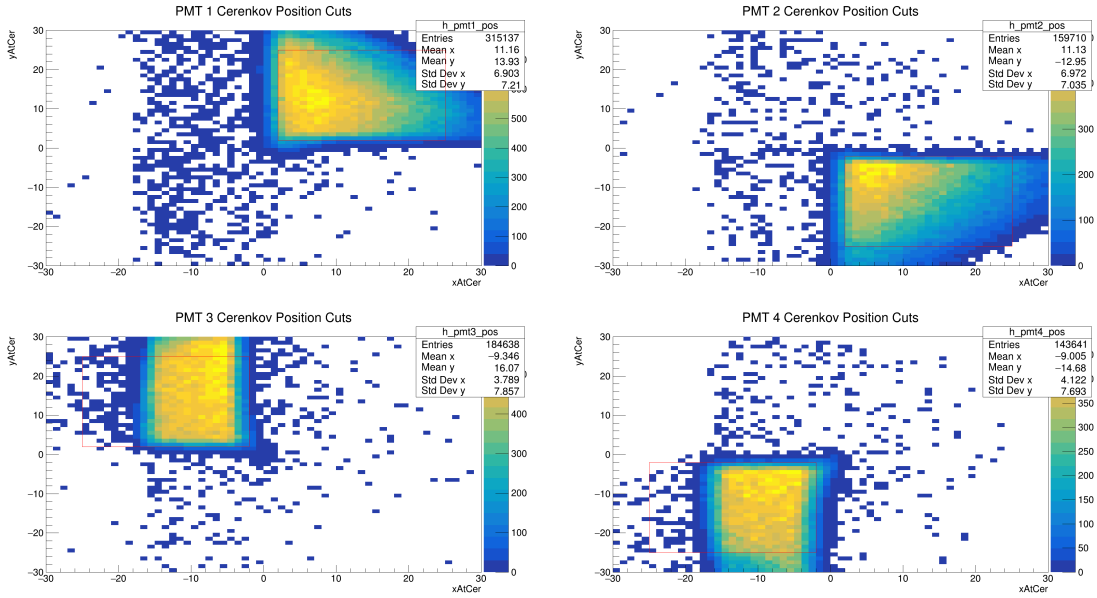

Figure 2: Sample run with a good distribution of events across all 4 PMTs after electron and multiplicity cuts. Cerenkov position cuts shown in red.

5 Determining the Calibration Constant

Now that we have data set for each PMT where we can reasonably assume nearly all remaining events:

- are electrons
- deposited all Cerenkov radiation to a single PMT

we can plot and fit the pulse integral distributions for each of the PMTs and pull the calibration constants from those fits.

First, we must determine what function to use to fit the pulse integral distributions. The pulse integral response of the PMT is proportional to the number of photoelectrons (NPE) extracted from the photocathode of the PMT by the passage of Cerenkov radiation. The NPE distribution should follow Poisson statistics because only whole numbers of electrons can be extracted from the photocathode, and because the probability that a given photon extracts an electron from the photocathode is nearly independent of whether another photon extracted an electron or not (this is a very good approximation as long as the photocathode is not close to saturation).

Starting with the PMF of the Poisson distribution:

$$P(k) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{1}$$

We first extend this distribution it to all positive real numbers x using the Gamma function, giving the following PDF:

$$f(x) = \frac{\lambda^x e^{-\lambda}}{\Gamma(x+1)} \tag{2}$$

We then add an overall normalization factor to our fitting function A to account for the fact that we are interested in fitting a distribution of millions of events, not the probability distribution of a single event:

$$f(x) = A\frac{\lambda^x e^{-\lambda}}{\Gamma(x+1)} \tag{3}$$

Finally, the "discrete" increments of our signal distribution are not unity like the photoelectron distribution, but are equal to the nominal signal output by the PMT per photoelectron, precisely the calibration constant μ that we are interested in. To stretch the domain of our distribution we divide our x variable and the distribution mean λ by μ to get our final fit function:

$$f(x) = A\frac{\frac{\lambda}{\mu}^{\frac{x}{\mu}} e^{-\frac{\lambda}{\mu}}}{\Gamma(\frac{x}{\mu}+1)} \tag{4}$$

Using the above function, we can now plot and fit our pulse integral distributions (Fig 3). The calibration code will then print the calibration constants for each of the PMTs to the terminal. The fit range is currently hard-coded, so you may need to manually adjust the fit range in the code to get a good fit of the peak.

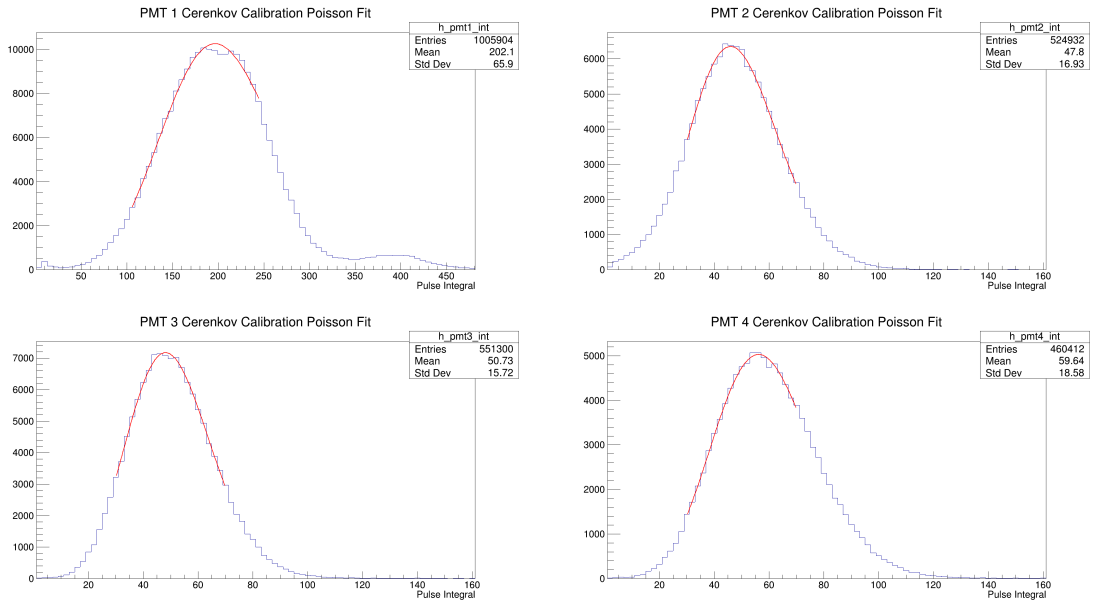

Figure 3: Sample pulse integral distributions and fits.

Once you are confident that you have a good fit, you can go ahead and complete the calibration procedure by inserting the calibration constants at the appropriate location in hallc_replay. Generally, you will change the variable *pngcer_adc_to_npe* at the path ./PARAM/SHMS/NGCER/pngcer_calib.param.