LA-UR-16-26603

Title: Gain Shift Corrections at Chi-Nu

Author(s): Brown, Tristan Brooks
Devlin, Matthew James

Intended for: Report

Issued: 2016-08-30

Gain Shift Corrections at Chi-Nu

Tristan Brown[a,b], Matt Devlin[a]

[a] P-27, Los Alamos National Laboratory, Los Alamos, NM 87544, United States

[b] Department of Physics and Applied Physics, University of Massachusetts Lowell, Lowell, MA 01854, United States

Abstract

Ambient conditions have the potential to cause changes in liquid scintillator detector gain that vary with time and temperature. These gain shifts can lead to poor resolution in both energy as well as pulse shape discrimination. In order to correct for these shifts in the Chi-Nu high energy array, a laser system has been developed for calibration of the pulse height signals.

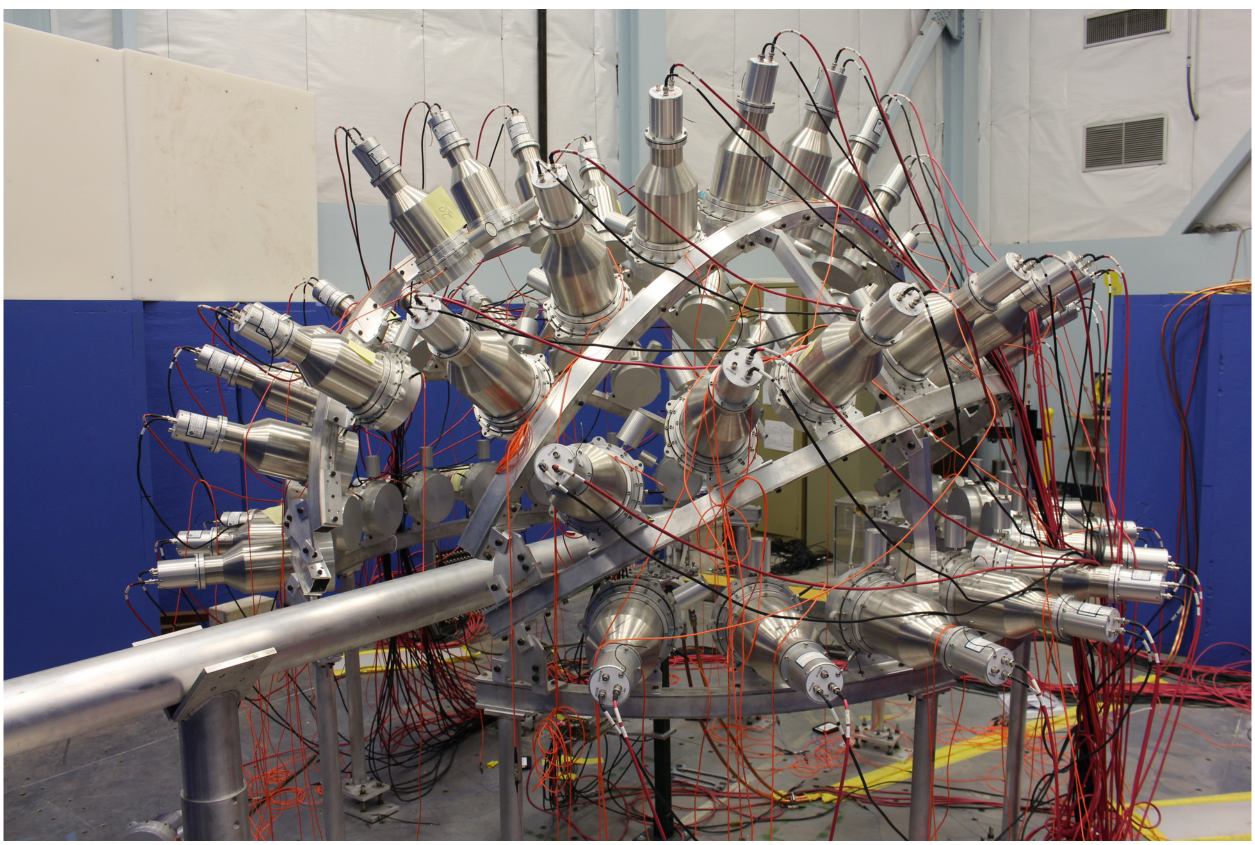

Figure 1: The Chi-Nu High Energy Array, with 54 Eljen 309[1] liquid scintillators.

Introduction

Chi-Nu is a project to measure the prompt fission neutron spectrum from neutron-induced fission in actinides. Chi-nu uses an array of 54 Eljen 309[1] liquid scintillators coupled to photomultiplier tubes to detect fast neutrons, as depicted in Figure 1. Experiments with the Chi-Nu liquid scintillator array found variance in gain as ambient conditions varied. These changes come from external variables that cannot be controlled well enough to completely remove their influence, such as temperature. The gain drift becomes a problem once analysis begins, as across either long runs or many runs, drifts lead to significant degradation of resolution in energy as well as pulse shape discrimination.

In order to solve this gain drift problem, steps were taken to create a calibration system. In order to achieve correction, the calibration system must have a constant pulse shape and pulse height that goes into the photomultipliers through the scintillation medium, as well as being distinct in pulse shape from experimental data signals.

Hardware

A PicoQuant FSL (Fast Switched Laser) 500 laser driver[2] and PicoQuant LDH-S-C-405 laser head[3] were chosen as the light source for the calibration system, capable of producing square waves of light with customizable pulse widths and triggering internally, externally, or using a combination of the two. For this system, an external Stanford Research Systems Model DG535 pulser with better timing capabilities is used along side the laser driver. The laser head has a peltier-based temperature stabilizer to better than 1 Kelvin for typical room temperatures between 15 and 30 degrees Celsius, allowing its light output to be constant and independent of the ambient conditions.

This laser is fed into a fiber optic cable leading to a fiber optic Tree Coupler, which branches out into eight other Tree Couplers. These Tree Couplers connect to other fiber optic cables with each of their eight branches and are fed into the fiber optic ports of the liquid scintillators.

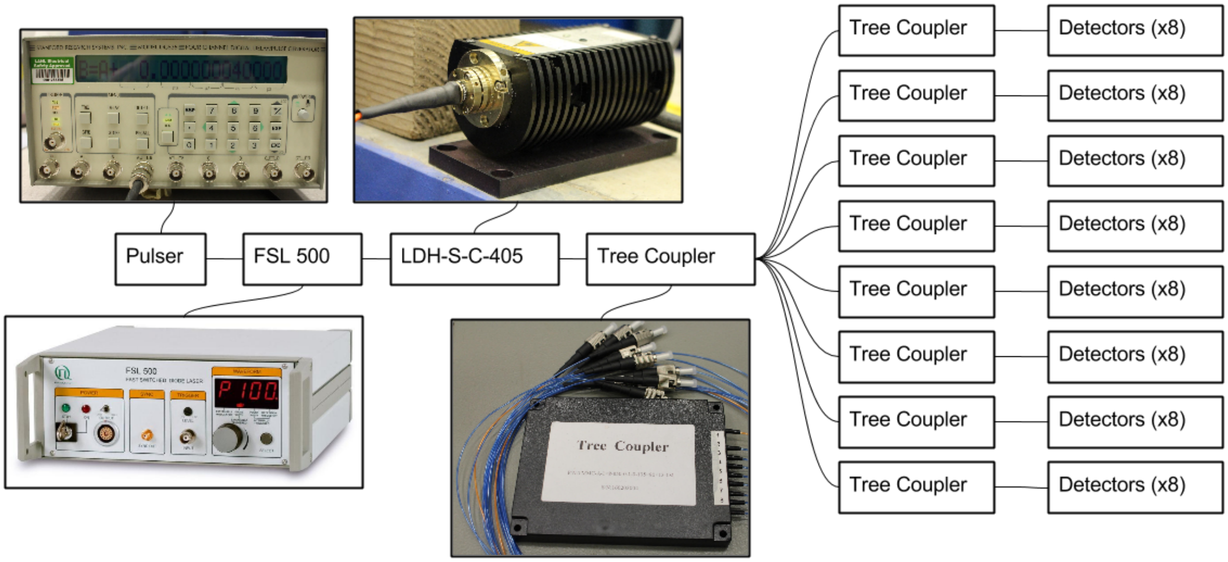

Figure 2: Block diagram of laser system hardware

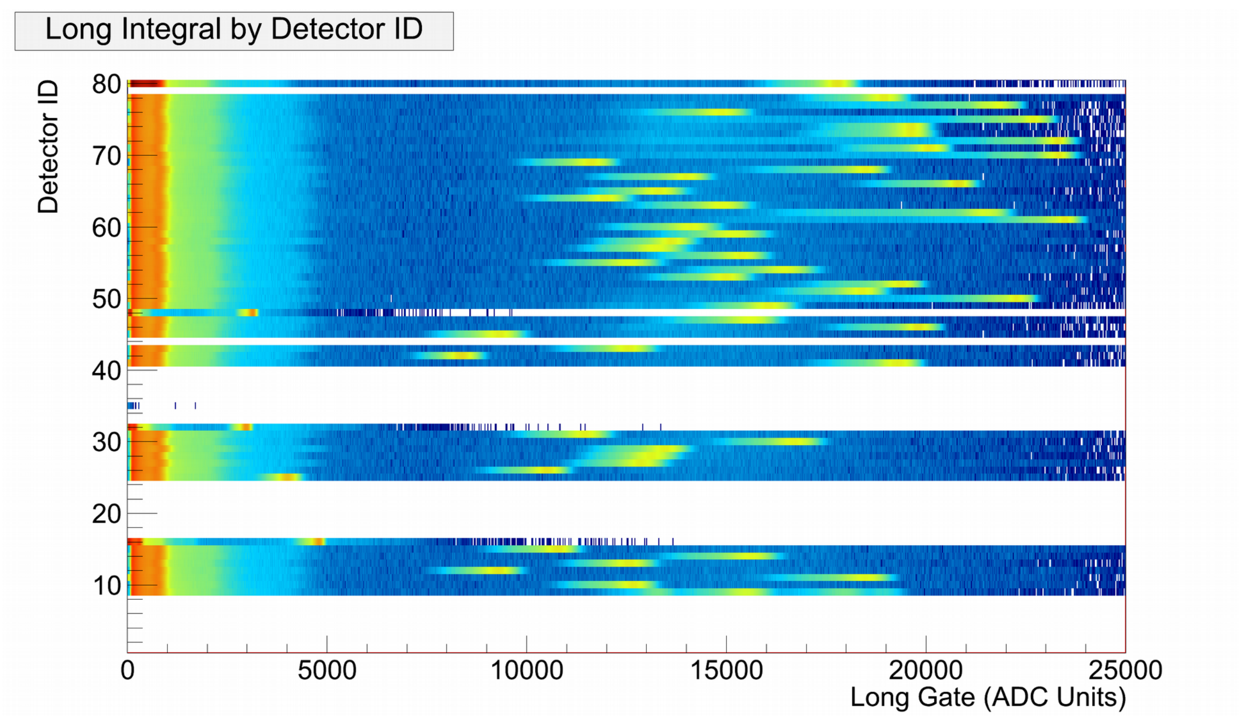

Figure 3: 2D Histogram of detector ID vs long integral. Gamma pulses lie in the red band to the left, while laser pulses are spots to the right. Such a plot is a useful way of finding reference gains.

Setup

The laser driver is set to its 35% power setting to avoid saturating the ADC. The trigger logic is set to start the pulse with an external signal but determine the pulse width internally. Pulse rate should be chosen to be low enough to not cause pileup, but a higher rate will tend to lead to faster stabilizing corrections or more confident corrections with higher sample size. Pulse width will influence the pulse shape discrimination value for separation as well as having the potential to increase sensitivity to gain shifts as more samples will be gain shifted. Due to the internal constant fraction discriminator timing logic internal to the CAEN V1730 digitizer boards, longer pulses will have incorrect timestamps and can cause mistriggering and/or retriggering off one pulse, limiting how wide a pulse can be while still being uesful. The recommended standard setting would be 100Hz at 37ns wide.

The laser head has only one setting, the stabilization temperature. This should be set to be as close as possible to the ambient temperature from its three options. As with any change to the system, the laser reference gain must be measured and set again for each detector.

To measure the reference gain for each detector, place a gamma calibration source at the center of the array. Take data with the laser active. Use the gamma ray from the test source to confirm that the array is gain matched. Then, record the centroid of the peaks in energy corresponding to the laser for each detector. This should be done when any change to the system occurs that can change the light output of the lasers. Gain shifts can also cause interactions with the threshold voltage that may influence efficiency at low energies. Correcting gain shifts may allow for quantification of this error. Future expansion on this work may include actively changing the high voltage to counteract gain changes as the occur.

Algorithm

Each detector has a queue dedicated to it. The queue will be used to calculate a moving average of the long integral. When a pulse is processed, it is determined whether or not it is a laser using the integration gates for pulse shape discrimination.

If the pulse is deemed to be a laser signal, the queue size is checked. If the queue is at its maximum size, the front is popped. Then, the pulse is added to the queue. Laser pulses can either be thrown out, or corrected as a normal pulse would be.

If the pulse is to be corrected, the average of the queue for the pulse's detector is retrieved and compared to the laser reference gain for that detector. A correction factor is created from this by finding the ratio, and this number is multiplied into the integrals of the pulse to correct its gain.

$$Average = \sum_{i=1}^{N} \frac{a_i}{N} \quad Correction = \frac{Reference}{Average}$$

Sample Results

In order to fabricate gain shifts and demonstrate the efficacy of this algorithm, the high voltage to a photomultiplier tube was changed to create dramatic artificial gain shifts. Figures 4-7 demonstrate the results of applying this correction.

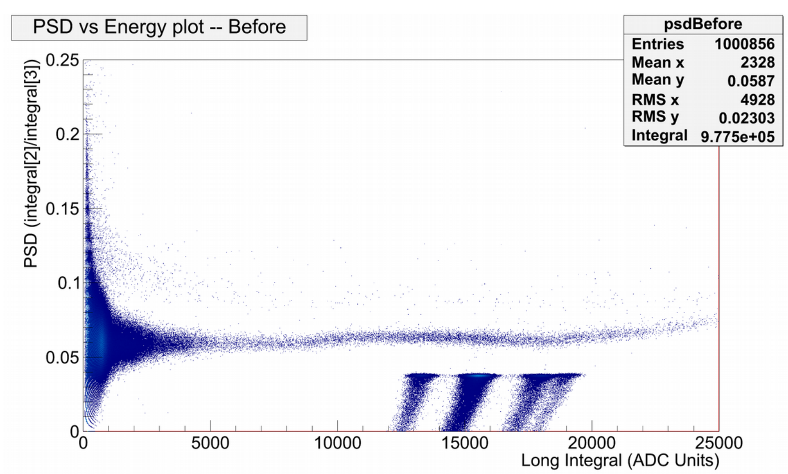

Figure 4: PSD vs Long Integral before correction. The laser spot for different gains can be seen in the lower right, with four long integral values corresponding to the four high voltage settings used during data collection.

In Figure 4, a 2D plot comparing the Pulse Shape Discrimination (PSD) value and the long integral value. The band in the middle represents gamma rays, and below that, the laser pulses can be seen. Their unique shape gives them a different and distinguishable PSD, and the position along the long integral axis will be proportional to the current gain of the detector. Four distinct spots can be seen in this laser pulse area, corresponding to four different voltage settings that were used during data collection. The goal of this correction is to merge these four spots in post processing, knowing that in reality there is one true light output value for the laser.

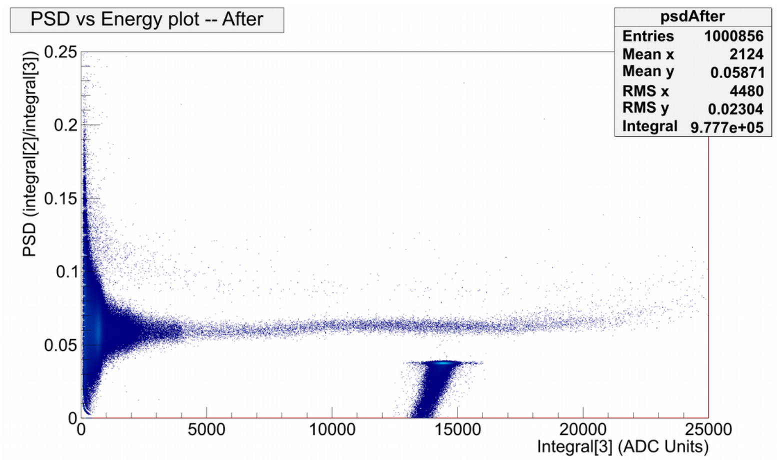

Figure 5: PSD vs Long Integral after correction. The laser spot in the lower right has merged with the gain corrections, and the gamma rays have been shifted as well, sharpening their resolution.

Figure 5 shows the results of applying the correction algorithm to the data in Figure 4. The four separate spots have been combined into one taller, sharper spot. The gains of the gamma rays are adjusted as well, sharpening the resolution of this data.

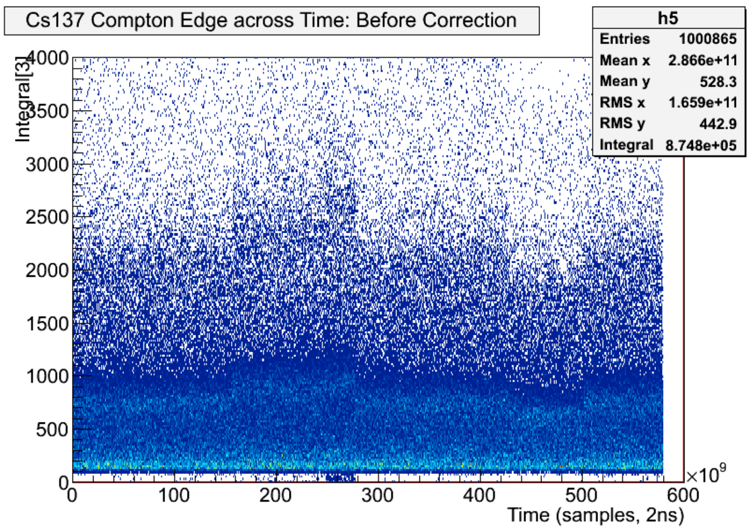

Figure 6: Long integral across time with ^{137}Cs, before corrections. Points in time where the high voltage applied to the detector was changed can be readily observed, causing clear shifts in the long integral value of the ^{137}Cs calibration source.

Figure 6 tracks the long integral value of a ^{137}Cs calibration source over time for the duration of the run. The points in time at which the high voltage for the detector was changed are apparent, causing clear shifts in the calibration source's ADC value.

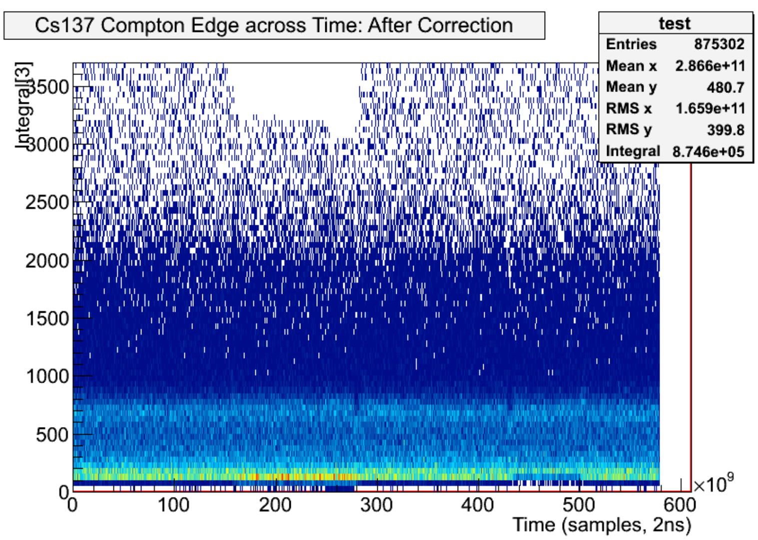

Figure 7: Long integral across time with ^{137}Cs, after corrections. Having applied the correction algorithm, the long integral value of the ^{137}Cs data has returned to its reference calibration value for the duration of the run.

Figure 7 shows the same data as figure 6 after applying the gain correction algorithm. The long integral value can be observed to have stabilized as the gain corrections shift it back to its calibrated reference value.

Software & Use

A C++ class has been written to perform gain corrections and should be able to be implemented in any C++ based analysis code.

Corrections rely on a few pieces of information. First is an array containing reference gains of what the laser pulse integral was at a reference time for each detector. Second is how many detectors there are. Third is the queue size to use. A large queue will have more data points, but a smaller queue will reach equilibrium more quickly. A higher laser trigger rate can make the moving average reach equilibrium more quickly as well. Fourth and fifth are the low and high psd bounds to apply, and sixth is the lower energy bound to exclude low energy physics events that have lower pulse shape discrimination values.

With 37ns pulses, a window of $0.029 < PSD < 0.045$ and integral[3] > 4000 will work. The standard queue size is 100.

In the UAC, this class has been implemented as a module with ODB parameters that can be set for everything that needs to be set.

References

[1]: Eljen 309 scintillator data :
http://www.eljentechnology.com/images/products/data_sheets/EJ-301_EJ-309.pdf
[2]: PicoQuant FSL 500 Datasheet:
https://www.picoquant.com/images/uploads/downloads/fsl500.pdf
FSL 500 image:
https://www.picoquant.com/images/uploads/product_images/16/fsl500_g.jpg
[3]: PicoQuant LDH Series:
https://www.picoquant.com/images/uploads/downloads/ldh_series.pdf

-Note that there is not a datasheet on the PicoQuant site for specifically the LDH-S series.

Code implementation example

```
#include "GainFix.h"

/*in your variable declarations and initializations*/
int referenceGains[number_of_detectors+1] = {/*your reference gains here*/} ;
//+1 to account for 1 based arrays
GainFix *Fixer = new GainFix(referenceGains, number_of_detectors,
max_queue_size, psd_hi, psd_lo, energy_cut_lo);
double gainCorrection;

/*In your event processing */
/*When you have access to the integrals and the detector id */

gainCorrection = Fixer->Process_Pulse(integral[2],integral[3],detector_id);
integral[1] *= gainCorrection;
integral[2] *= gainCorrection;
integral[3] *= gainCorrection;
```