Blinded by the Light: Commissioning of the Hall C SHMS Heavy Gas Cherenkov Detector

A Thesis

Submitted to the Faculty of Graduate Studies and Research

In Partial Fulfillment of the Requirements

For the Degree of

Master of Science

in

Physics

University of Regina

By

George Ryan Ambrose

Regina, Saskatchewan

October, 2018

UNIVERSITY OF REGINA

FACULTY OF GRADUATE STUDIES AND RESEARCH

SUPERVISORY AND EXAMINING COMMITTEE

George Ryan Ambrose, candidate for the degree of Master of Science in Physics, has presented a thesis titled, ***Blinded by the Light: Commissioning of the Hall C SHMS Heavy Gas Cherenkov Detector,*** in an oral examination held on October 31, 2018. The following committee members have found the thesis acceptable in form and content, and that the candidate demonstrated satisfactory knowledge of the subject material.

External Examiner:	Dr. Paul Laforge, Electronic Systems Engineering
Supervisor:	Dr. Garth Huber, Department of Physics
Committee Member:	Dr. Geoffrey-Fathom Grinyer, Department of Physics
Committee Member:	Dr. Aram Teymurazyan, Department of Physics
Committee Member:	Dr. Mauricio Barbi, Department of Physics
Chair of Defense:	Dr. Sandra Zilles, Department of Computer Science

Abstract

The University of Regina built a heavy gas Cherenkov (HGC) detector for use at the Thomas Jefferson National Accelerator Facility (JLab) as it enters a new era of 12 GeV experiments. This detector will play an important role in π/K separation between 3 - 11 GeV/c in experimental Hall C and is housed in the Super High Momentum Spectrometer (SHMS). Recently, the first experimental data have become available, allowing the performance of the HGC to be evaluated. The calibration procedure developed is presented, as well as the characterization of the Poisson nature of the photomultiplier tube photoelectron response. Calculation of the HGC efficiency is also performed and used as an independent confirmation of the index of refraction of the Cherenkov media. This thesis also outlines JLab's experimental Hall C, the data acquisition, the HGC simulation, and the particle identification capabilities of the HGC in an experimental setting.

Acknowledgements

First and foremost, I would like to thank my supervisor Garth M. Huber for the many hours of patient guidance he has provided. I also appreciate the spirit of camaraderie he builds in his research group, finding opportunities for us to socialize outside of a physics environment, although our conversations tend to return to physics regardless.

It was a pleasure to work with my fellow graduate students: Samip Basnet, Rory Evans, Ahmed Foda, Wenliang (Bill) Li, and Dilli Paudyal. Working in a windowless corridor was made more bearable by all of you.

I would also like to thank Derek Gervais for his assistance in the construction of the acrylic rings.

The work I have done would not have been possible without the assistance of the Hall C scientists: Mark Jones, Eric Pooser, and Brad Sawatzky. They provided invaluable insight into the software of Hall C and without hesitation answered any question I had, even if it was the same one several times.

Special thanks as well to the faculty members of the Department of Physics for being available for discussions and greatly assisting my education as a physicist.

I am also grateful for the funding provided by NSERC. Without it I, and many other graduate students, would be forced to eat exclusively ramen (instead I now choose to).

Lastly, I would like to thank my friends and family. Their support brought me to where I am today.

Contents

List of Figures

List of Tables

Listings

Abbreviations

ACD Aerogel Cherenkov Detector

ADC Analog to Digital Converter

ASCII American Standard Code for Information Interchange

BCM Beam Current Monitor

BPM Beam Position Monitor

CAEN Costruzioni Apparecchiature Elettroniche Nucleari

CEBA Continuous Electron Beam Accelerator

DAQ Data Aquisition

ECAL Electromagnetic Calorimeter

ECL Electrical Connections Ltd.

EPICS Experimental Physics and Industrial Control System

fADC Flash Analog to Digital Converter

FPGA Field Programmable Gate Array

GPD Generalized Parton Distribution

HGC Heavy Gas Cherenkov

HMS High Momentum Spectrometer

HRS High Resolution Spectrometer

JLab Thomas Jefferson National Accelerator Facility

KPP Key Performance Parameters

L-T Longitudinal - Transverse

Linac Linear Accelerator

NGC Noble Gas Cherenkov

NIM Nuclear Instrumentation Module

NPE Number of Photoelectrons

NSA Number of Samples After Threshold

NSAT Number of Samples Above Threshold

NSB Number of Samples Before Threshold

P-T Pressure - Temperature

PE Photoelectron

PL Programmable Latency

PMT Photomultiplier Tube

pQCD Perturbative Quantum Chromodynamics

PTW Programmable Timing Window

QCD Quantum Chromodynamics

REF Reference

RF Radio Frequency

RG Radio Guide

RICH Ring-imaging Cherenkov

RTV Room Temperature Vulcanizing Silicone

SHMS Super High Momentum Spectrometer

SIDIS Semi-Inclusive Deep Inelastic Scattering

SPE Single Photoelectron

SRF Superconducting Radio Frequency

TDC Time to Digital Converter

UV Ultraviolet

VME Versa Module Europa bus

Chapter 1

Introduction

The objectives of this thesis are to: detail the calibration and efficiency evaluation procedures for the SHMS heavy gas Cherenkov (HGC) detector, report its current performance in an experimental setting, and describe future plans to improve detector performance.

The HGC was built at the University of Regina from 2010-2013 and plays an important role in the recently completed 12 GeV upgrade at the Thomas Jefferson National Accelerator Facility (JLab), specifically in experimental Hall C. This detector functions as a particle identification device, taking advantage of Cherenkov radiation produced by fast moving charged particles. It is capable of separating between $\pi^{\pm}$ and heavier charged particles ($K^{\pm}$, p, etc.) from 3 - 11 GeV/c. Using this information in conjunction with other particle identification detectors, for instance other Cherenkov detectors and calorimeters, specific particle species can be cleanly determined. Thus specific reactions can be isolated, leading to insights of the underlying physical processes.

The work done commissioning the HGC has a direct impact on the experimental capabilities of Hall C and plays a role in the research program of the lab. JLab as an institution has the goal of pursuing research into the strong force of nature, the interaction responsible for holding atomic nuclei together. This force is best described by quantum chromodynamics (QCD), however it is very difficult to apply in the low energy regime where most of the interactions in our world take place. Connecting the observed properties of strongly interacting particles, such as their mass and spin, to the fundamental theory of QCD is one of the major unresolved problems in modern physics. Hence the need for facilities like JLab which are able to probe this question experimentally.

Scientific achievements at JLab include insights into hadronic and mesonic form factors and structure functions, the transition to perturbative quantum chromodynamics, and searches for physics beyond the standard model (Montgomery, 2011) (Arrington et al., 2011) (Gilman et al., 2011) (Roche et al., 2011). These discoveries deepen our understanding of nature on a fundamental level. As well, it may be that one day these discoveries will have applications to the energy sector as we better understand the atomic nucleus. However, more exciting are the consequences we are not yet aware of, such as how radio antennas were an unforeseen invention from James Maxwell's improvement to Ampère's Law.

The structure of the thesis is as follows, in Chapter 2, Hall C instrumentation is explained, with emphasis placed on Cherenkov detectors. Chapter 3 gives the physical theory of Cherenkov radiation from classical electrodynamics. Chapter 4 describes the data acquisition system for the HGC and the particular parameters used to perform the calibration and efficiency measurements. Chapter 5 and Chapter 6 detail the calibration and efficiency evaluation procedures respectively. Chapter 7 presents the simulation work done on the HGC to characterize the detector. Lastly, Chapter 8 demonstrates the performance of the HGC in an experimental setting. Ending the thesis is a brief summary, Chapter 9, with the future outlook for the heavy gas Cherenkov detector.

Chapter 2

Thomas Jefferson National Accelerator Facility

JLab is renown for hosting the world's largest superconducting radio frequency (SRF) linear accelerator, named the Continuous Electron Beam Accelerator (CEBA). To

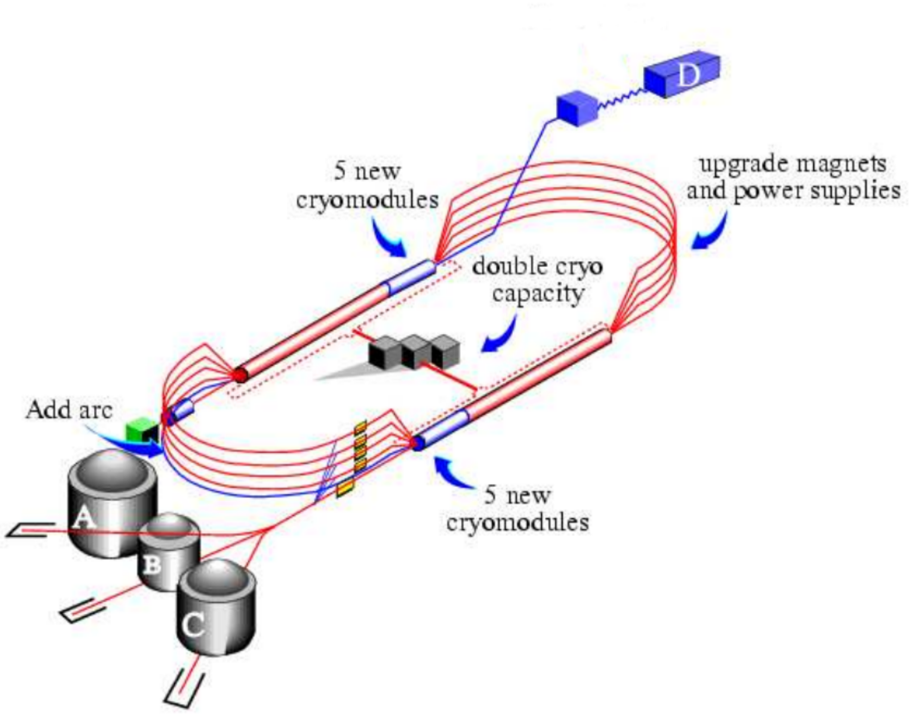


Figure 2.1: Schematic of accelerator facility at JLab (Jefferson Lab, 2018a).

achieve the energies required to perform experiments addressing the physics questions raised in Chapter 1, the electron beam is designed to enter a racetrack shaped accelerator and perform several passes. The accelerator uses novel technology, employing SRF cavities of specific geometric configuration, supporting fields allowing for continuous acceleration of the electron beam (Ciovati et al., 2016). These cavities are made of niobium, which become superconducting at or below 4.2 K (Ciovati et al., 2016), provided by the Central Helium Liquifier which produces Helium at 2 K. By using a particular radio frequency, electrons are grouped into microbunches and injected at an energy of 45 MeV with 0.667 ns separation (RF repetition rate of 1497 MHz). Altogether, the linear accelerators (linacs) host 50 cryomodules and are connected by recirculation arcs providing 2.2 GeV/pass to the electron beam. After 5 passes, the beam is diverted into the experimental hall, corresponding to a 11 GeV beam of up to 100 μA. The complete configuration of the accelerator facility is presented in Figure 2.1 (Jefferson Lab, 2018a).

After the electron beam exits the South Linac, it enters the beam switch yard, to be diverted between Halls A, B, and C by a series of magnets. Each hall has a specific experimental goal: Hall A is equipped with two high resolution spectrometers (HRS) which are optimized to study high precision nuclear structure, Hall B is designed to allow large solid angle acceptance (4π) and a broad momentum range to capture produced neutral and charged products in nuclear interactions, and Hall C is specialized to perform high luminosity experiments, allowing rare interactions to be studied at a high event rate (Gross, 2011). Lastly, past the North Linac lies Hall D, the newest experimental hall. The additional acceleration provides an electron beam energy maximum of 12 GeV, which is used to produce a beam of linearly-polarized photons. Hall D is configured to allow a large acceptance of these broad-band photons to study nuclear structure.

The experimental hall of concern in this thesis is Hall C, hosting two spectrometer arms, one of which contains the HGC. Once on the Hall C arc, the beam passes through a series of monitors summarized in Figure 2.2. The beam profile monitors, or Superharps, consist of a frame with three wires, two vertical and one horizontal. The profile is analyzed by passing this device through the beam at an angle of 45^o, obtaining vertical and horizontal position with a resolution of 45 μm. Naturally, this is a destructive measurement and is taken before or after data taking. Beam position monitors (BPM), beam current monitors (BCM), and an Unser Current Monitor measure beam parameters nondestructively. The position and angle of the beam are determined from the BPMs closest to the target. Lastly, the hall arc is outfitted with a pair of fast raster magnets to reduce localized heating, and therefore density reductions, in the liquid targets. This is achieved by steering the beam in a linear, uniform pattern giving a uniform distribution of the energy deposition (Yan et al., 2005). Once in the hall, the beam will interact with the cryogenic target, producing a variety of possible reactions. Targets include: liquid hydrogen, liquid deuterium,

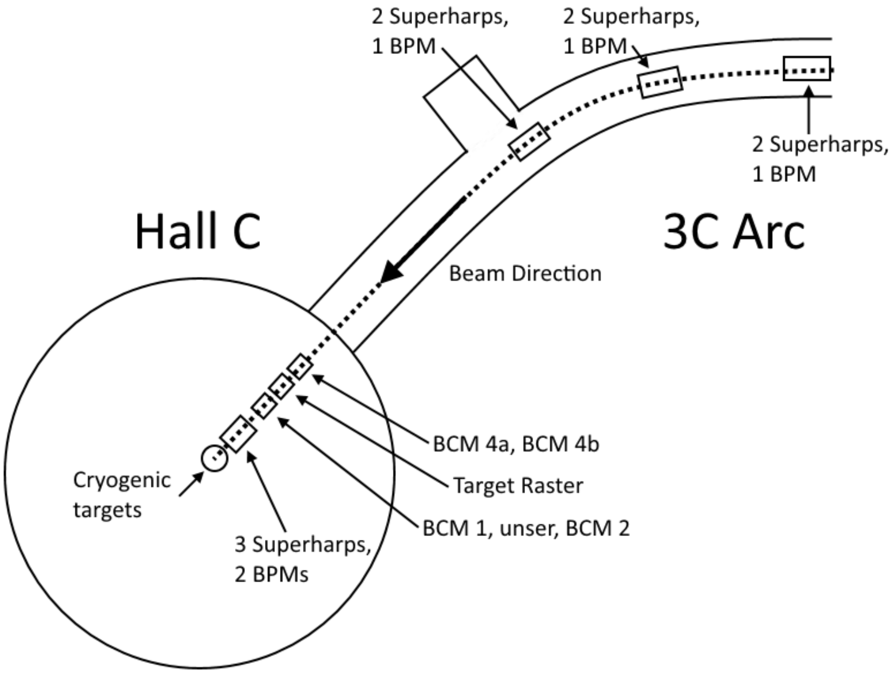


Figure 2.2: Schematic of beamline entering Hall C.

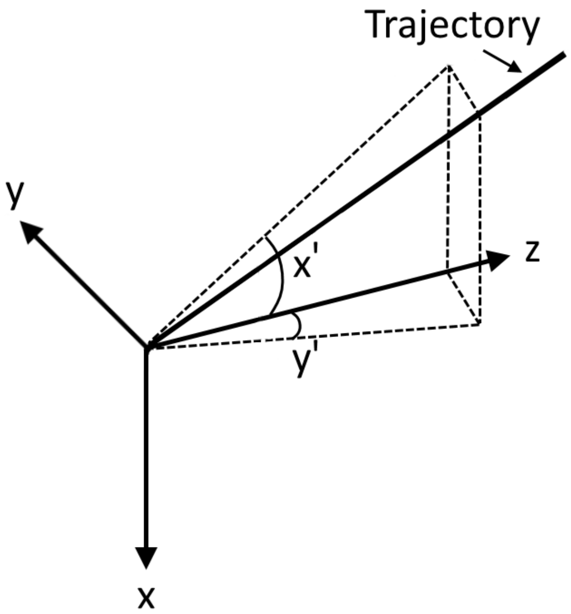


Figure 2.3: Coordinate axis for the spectrometer. Unprimed coordinates define the axes for the spectrometer while primed coordinates define the Euler angles x' and y'. Definitions in the text.

aluminum (for subtraction measurements), and several samples of different densities of carbon and metallic alloys. A polarized ^{3}He target will also be installed in 2019. To detect the products of these reactions, Hall C has two spectrometer arms, the High Momentum Spectrometer (HMS) and the Super High Momentum Spectrometer (SHMS).

The coordinate axis for both spectrometer arms is displayed in Figure 2.3. It is the convention used in charged particle transport in dispersive magnetic systems. In this case, the z axis is the direction of charged particle travel through the spectrometer (the optical axis). The x axis is the direction of increasing particle momenta, dictated by the dispersion of the dipole magnet, and from the Hall's frame points downward. The y axis is deduced by $z \times x$, and therefore points left (Li, 2012). These axes then define the Euler angles used in the tracking discussion, indicated by primed coordinates.

The SHMS is an 18.4^o vertical bend spectrometer with a maximum central momentum of 11 GeV/c. Products from the target ensemble enter the vacuum-coupled series of superconducting focusing magnets. The first of which is a horizontal bender, shifting most produced charged particles by 3^o allowing for an exceptionally small scattering angle to be detected. The next three are octagonal quadrupole magnets designed to focus the particles into the dipole aperture, improving the solid angle acceptance of the spectrometer. Lastly, the particles are separated by charge and momentum by a dipole magnet which bends the central ray, corresponding to the trajectory along the z axis, by 18.4^o. This ensemble of focusing magnets allows for a horizontal scattering angle minimum of 5.5^o with an acceptance of $\pm 1.3^o$, i.e. the smallest detectable angle is 4.2^o from the electron beam path with the spectrometer centred at 5.5^o. As well, the magnets can be tuned to select particles of a particular central momentum or charge polarity. After the magnet ensemble, the stream of particles is directed into the detector package to perform the actual physics measurement. The package is housed within the detector hut, which is built from a

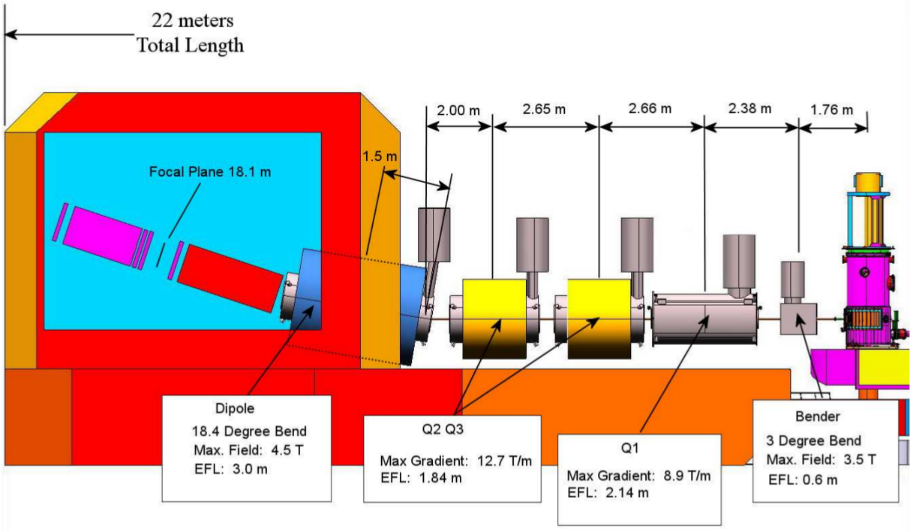


Figure 2.4: Schematic of magnet configuration for SHMS (Jefferson Lab, 2018b). From the right is the cryogenic target, followed by the horizontal bender and three quadrupole magnets. The large blue square is the dipole magnet, which is partially inside the detector hut. The red and pink squares represent the detector stack, showing how far downstream the SHMS focal plane is.

Quantity	Specification	
	HMS	SHMS
Dipole Bend Angle	25^o	18.4^o
Maximum Central Momentum	7.4 GeV/c	11 GeV/c
Focal Length	26.0 m	18.1 m
Scattering Angular Range	10.5^o to 85^o	5.5^o to 40^o
Momentum Acceptance	±10%	-10% ; +22%
Momentum Resolution	<0.1%	0.03% - 0.08%
Solid Angle Acceptance	6.7 msr	4.0 msr
Horizontal Acceptance	±27.5 mrad	±24 mrad
Vertical Acceptance	±70 mrad	±40 mrad
Horizontal Resolution	0.8 mrad	0.5 - 1.2 mrad
Vertical Resolution	0.9 mrad	0.3 - 1.1 mrad
Target vertex Length	±7 cm	±15 cm
Target Vertex Reconstruction Accuracy	1 mm	0.1 - 0.3 mm

Table 2.1: Table of SHMS and HMS Specifications (Jefferson Lab, 2018b).

combination of concrete and a boron/lead mixture to absorb stray charged particles, photons, and neutrons. A picture of the SHMS configuration is presented in Figure 2.4 (Jefferson Lab, 2018b).

The HMS has a nearly identical setup as the SHMS. The focusing magnets are identical, except there is no horizontal bender to allow for such small scattering angles. This spectrometer has a 25^o vertical bend for the central ray and has a maximum central momentum of 7.3 GeV/c. The horizontal scattering acceptance is between 10.5^o and 85^o with $\pm 1.576^o$ horizontal acceptance. Therefore, the smallest measurable angle is 8.92^o. The specifications of the SHMS and HMS are listed in Table 2.1 (Jefferson Lab, 2018b).

The detector package for the SHMS is shown in Figure 2.5. It is composed of a series of Cherenkov detectors for particle identification, a calorimeter for energy measurement, drift chambers for particle trajectory, and hodoscopes for triggering and timing. The first of which is the Noble Gas Cherenkov Detector (NGC), which is used to separate electrons from heavier charged particles at momenta higher than 6 GeV/c (Day, 2017). This is critical, as the SHMS is capable of reaching kinematic

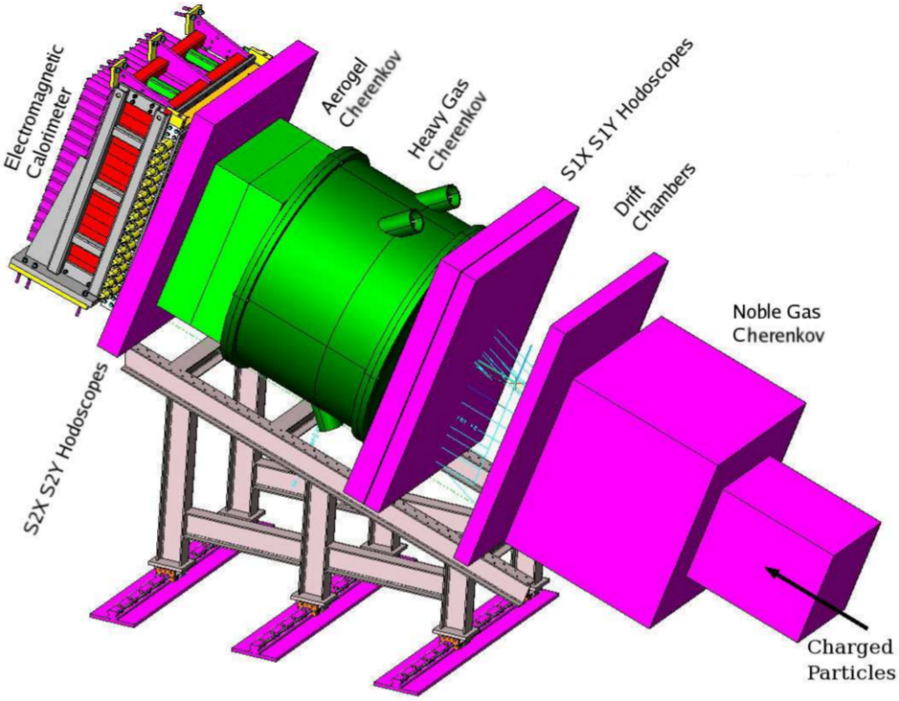


Figure 2.5: Schematic of detector package in SHMS (Metzger, 2012).

regions where the pion background rate dominates the electron scattering rate by more than 1000:1. To perform the separation, neon is the noble gas used, where Figure 2.6 shows the Cherenkov thresholds for various gas candidates. Due to spatial constraints, the NGC is placed before the focal plane, located between the drift chambers, and therefore may affect the reconstructed momentum resolution. For experiments with a central momentum less than 6 GeV/c, the NGC is replaced by a vacuum tank of identical dimension to eliminate possible multiple scattering. The construction of the NGC is similar to the HGC, it is a tank filled with Cherenkov media with mirrors to reflect produced light onto a set of four PMTs.

Next are the drift chambers, composed of sensing wires to determine the horizontal and vertical angles, and positions of charged particles before and after the focal plane (Christy et al., 2016). This is to determine the particle momentum and trajectory. Reconstruction is performed by obtaining data from various angles of sensing wires giving accurate information of particle path.

The hodoscopes are used to generate the trigger for the data acquisition system (Yero, 2018). Three hodoscopes are composed of plastic scintillators placed horizontally and vertically. The last hodoscope plane is made of quartz to provide additional particle identification, since the neutron background cannot produce Cherenkov radiation in this plane.

Two Cherenkov detectors are sandwiched between the hodoscopes, the heavy gas Cherenkov Detector (HGC) and aerogel Cherenkov detector (ACD). As with the NGC, these are used in particle identification, HGC for electron/pion separation at low momenta or pion/kaon at higher momenta (Li, 2012), ACD for pion/kaon separation at low momenta or kaon/proton separation at higher momenta (Horn, 2017). As the ACD features a solid Cherenkov media, changing the index of refraction for different momentum settings is more involved. The ACD has four separate aerogel trays with index of refraction: 1.03, 1.02, 1.015, and 1.011 (Mkrtchyan, 2016), which

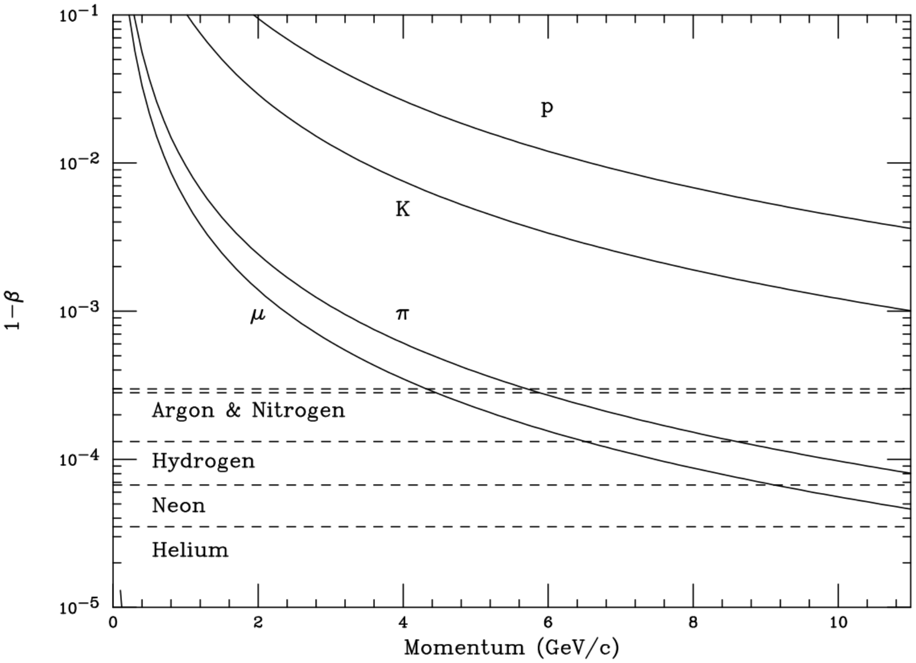


Figure 2.6: Visualization of using a noble gas for particle species identification. plotted is hadron velocity as (1 - β) against momentum. The dashed horizontal lines indicate the index of refraction as (n - 1) for several gases at 1 atm. The requirement for Cherenkov radiation is (1 - β) < (n - 1) (Day, 2017).

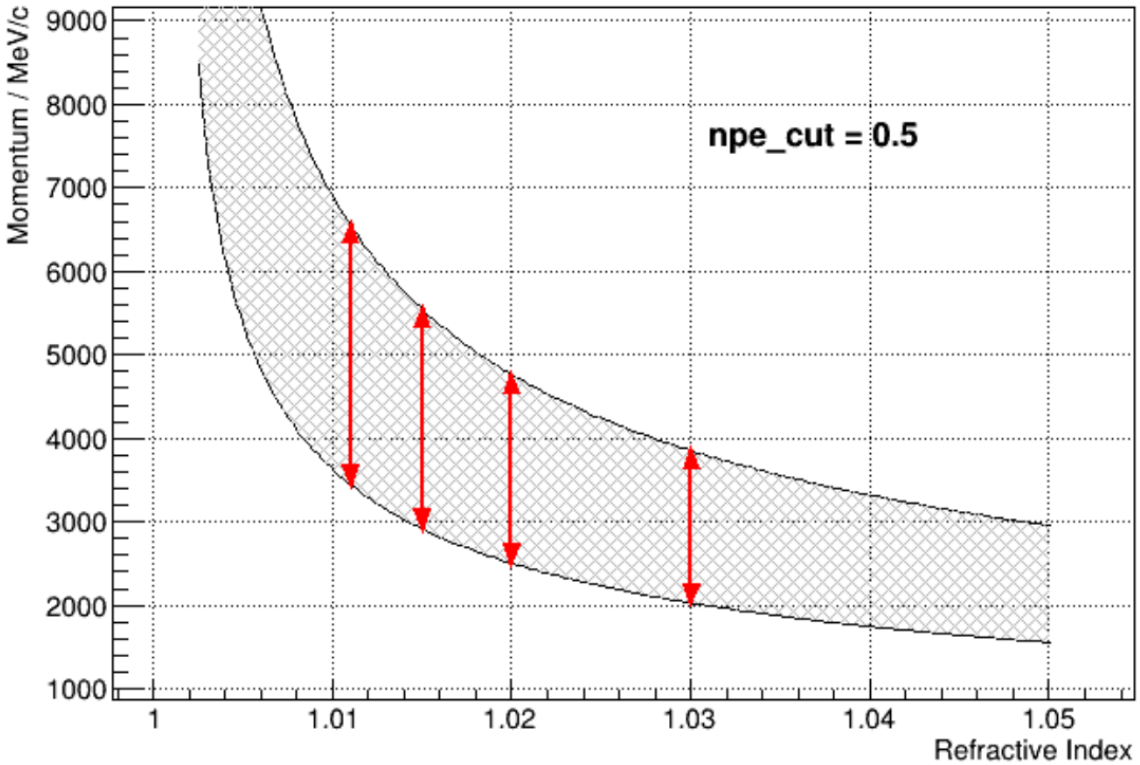


Figure 2.7: Visualization of the ranges for the various aerogel trays. The Aerogel detector is a diffusion box with four exchangeable aerogel trays of index of refraction: 1.03, 1.02, 1.015, and 1.011. The red arrows demonstrate the range of momenta for which the particular tray can perform kaon/proton separation at a threshold of 0.5 photoelectrons (Mkrtchyan, 2016).

must be manually switched out. The range of momentum appropriate for each tray to separate kaons and protons is shown in Figure 2.7. It is important to note that the various Cherenkovs and calorimeter must be used in tandem to achieve a clean sample of correctly identified particles.

Lastly, the Electromagnetic Calorimeter (ECAL) is used to confirm the electron-hadron identification (Mkrtchyan, 2012). It is composed of two lead-glass components: pre-shower and shower. The pre-shower is to detect the shallow penetrating particles with higher precision , while the shower measures the resultant spray of particles from deeply penetrating particles.

The physical structure of the HGC is displayed in Figure 2.8, where it is designed

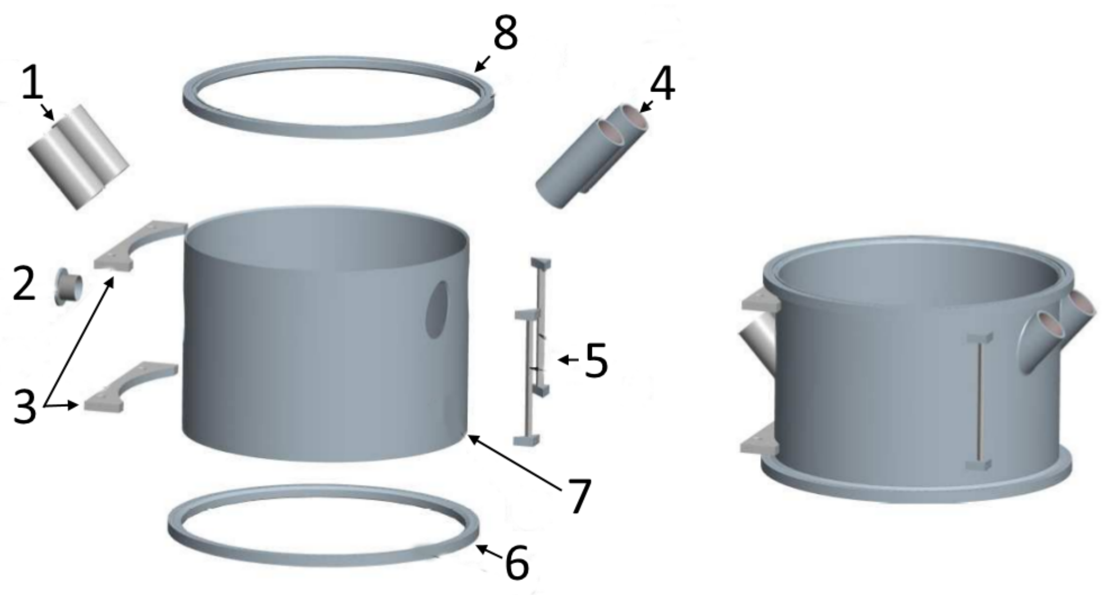


Figure 2.8: Visualization of the heavy gas Cherenkov detector. Components are 1. PMT sleeves; 2. Gas port; 3. Cradle brackets; 4. PMT sleeves; 5. Lifting lugs; 6. Window flanges; 7. Vessel cylinder; 8. Window flanges (Li, 2012).

to maximize Cherenkov radiation in the allowed space and tightly focus the emitted photons onto the PMT detectors. The HGC is the fourth component of the detector stack, its front is located 18.8 m from the target chamber along the optical axis and therefore 0.7 m downstream of the focal plane. It measures 1.829 m in diameter and 1.3 m in length to allow for sufficient length of Cherenkov radiator gas and efficient workings for the light collection optics. The components include: vessel cylinder, window flanges, lighting lugs, cradle brackets, gas port, and PMT sleeves. Four PMTs are used for light detection, the top two are tilted at 42^o and the bottom two at -42^o (Li, 2012).

The spread of the resultant Cherenkov envelope is 90 cm × 80 cm in the x-y plane at the mirror location and great effort was expended optimizing the mirror-PMT alignment (Li, 2012). To achieve this, four concave mirrors with four 5" PMTs were used. Each mirror is 60 cm × 55 cm with a radius of curvature of 110 cm and thickness 3 mm. However, the mirrors are not purely spherical. Instead, they underwent the slumping technique by Sinclair Glass (Sinclair Glass, 2018), whereby they were

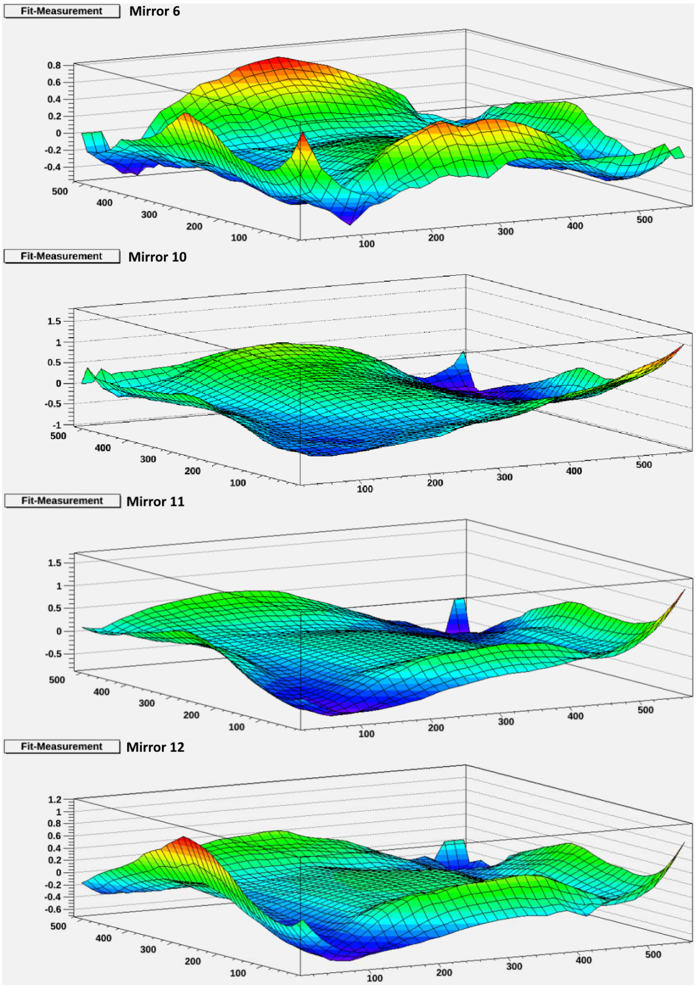


Figure 2.9: Profile of mirrors used in the HGC. The x and y axis refer to mirror position and the z axis refers to Δz, the difference between measured height compared to theoretical calculation of a perfect oblate shape (Li, 2012). The placement of the mirrors in the HGC are as follows: mirror 11 is quadrant one, mirror 10 is quadrant 2, mirror 12 is quadrant 3, and mirror 6 is quadrant 4 (Li, 2018a).

heated in a molding oven to obtain a certain curvature (Li, 2012). This inexpensive method results in an imperfect oblate ellipsoid shape, more details of the mirror surface can be seen in Figure 2.9. Of particular interest is the larger surface height deviation along the mirror edges, due to the slumping technique. Additionally, the mirrors are coated with an aluminum grain by Evaporated Coatings, Inc. (Evaporated Coatings, 2018) to enhance the reflection of deep UV photons (Li, 2012). As the acceptance for generated Cherenkov photon wavelength is between 150 to 600 nm (detailed discussion on the range of Cherenkov radiation is given in Chapter 3), it is important to improve performance at lower wavelength. An example of a reflectivity test for the aluminized HGC mirrors is shown in Figure 2.10 (Li, 2012), where one can see that below 300 nm wavelength photons the mirror reflectivity begins to worsen. Below 200 nm wavelength almost no light is reflected.

To ensure there are no gaps, the mirrors are interleaved such that the closest mirror to mirror approach is between 7 - 10 mm. The geometry of the mirrors is depicted in Figure 2.11. The order from lowest to highest z is: 4, 3, 2, 1. In other words, mirror 4 is furthest upstream and mirror 1 is furthest downstream. Mirror 4's bottom left corner marks the central point at (+0.9 cm, +0.5 cm) then extends to (+0.9 cm, -54 cm), (-56 cm, +0.5 cm), and (-56 cm, -54 cm). Mirror 3 lies behind, sharing the right edge with mirror 4 and covers to (+0.9 cm, +54 cm), and (-56 cm, +54 cm). Mirror 2 lies below mirror 4 giving coverage to (+57 cm, +0.5 cm), and (+57 cm, -54 cm). Lastly is mirror 1, completing the square at (+57 cm, +54

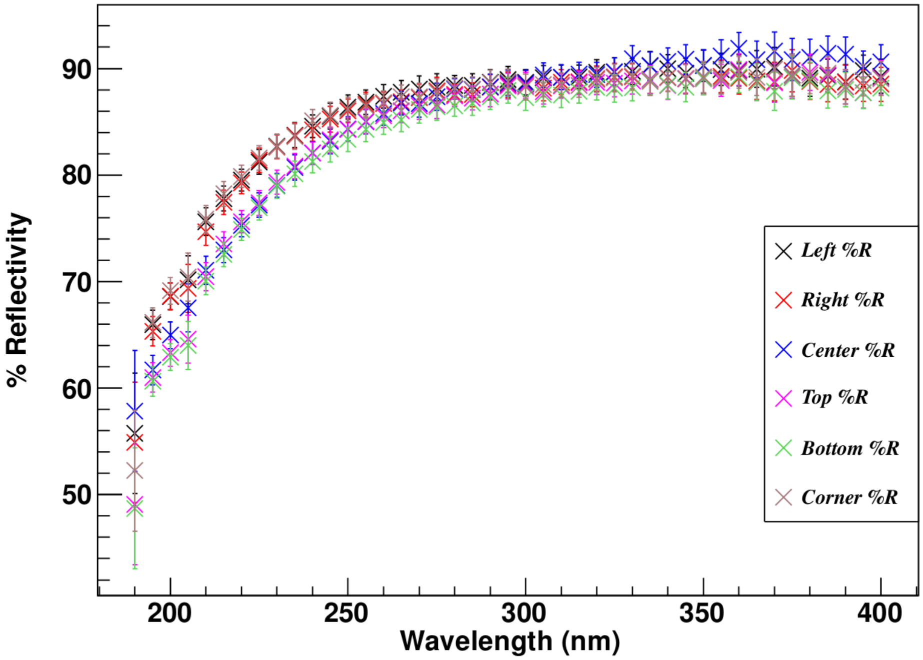


Figure 2.10: Example of mirror reflectivity test performed on the HGC (Li, 2012). The different locations refer to positions across the mirror surface to determine positional dependence on reflectivity.

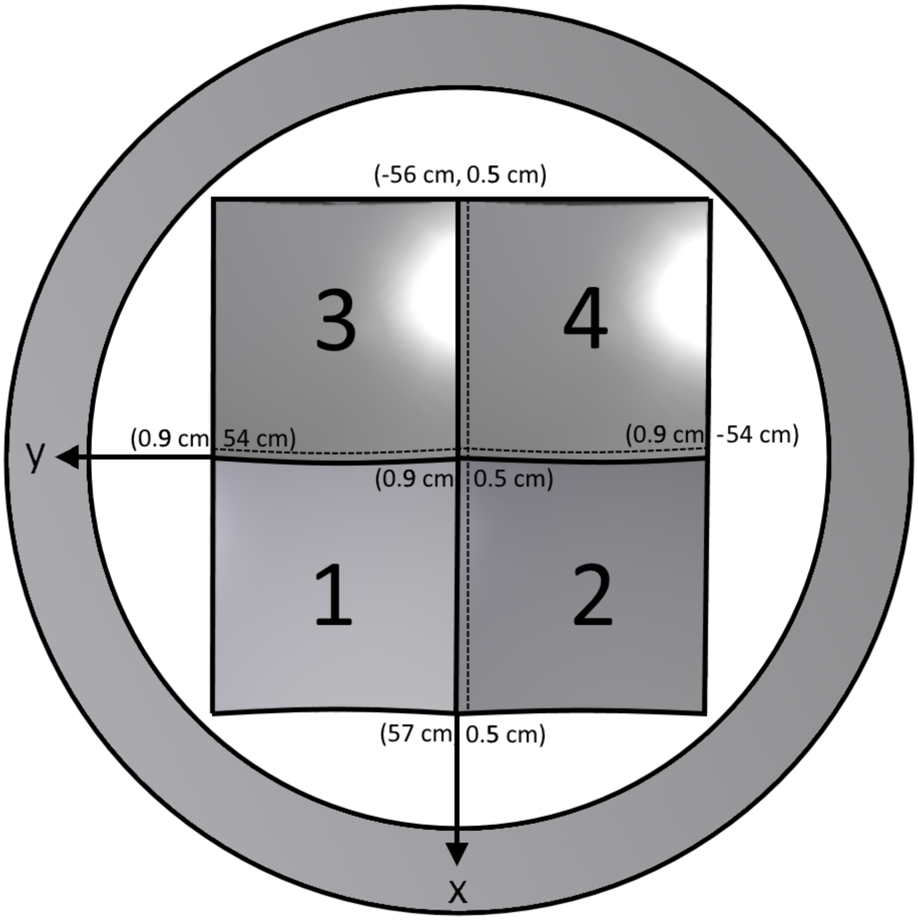


Figure 2.11: Dimension and orientation of mirrors in HGC. Dotted lines reflect the interleaving of the mirrors. Note that the x axis points down and y axis to the left, where future plots of the mirror plane will have the x axis pointing up and y axis to the right.

cm). The positions of the mirrors are taken from the initial specifications, and are implemented in the detector simulations. However, the mirror alignment has been reoptimized, resulting in the numbers presented being only approximate. Therefore, one should expect an error of approximately ±1 cm with the configuration presented here.

In anticipation of the calibration strategy in Chapter 5, a brief description of how the light is reflected in the HGC tank will be given here with a visual presented in Figure 2.12. Each mirror quadrant of Figure 2.11 is specifically shaped to focus light onto a single PMT, with very little dispersion except around the very edge. However,

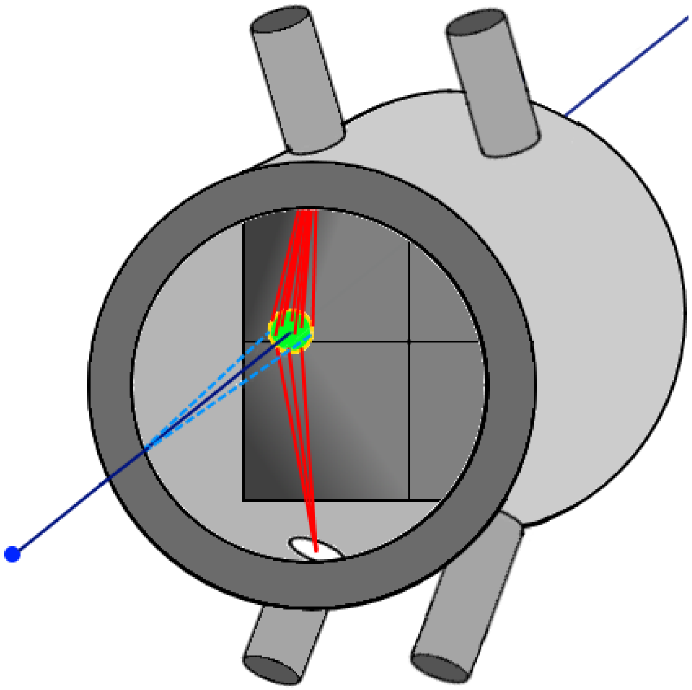


Figure 2.12: Visualization of the Cherenkov cone illuminating multiple mirror quadrants. The particle is in blue, the dark blue line indicating its trajectory. The light blue dashed lines indicate the radius of the Cherenkov cone, typically 4 cm. The green circle gives the projection onto the mirror plane. The red lines represent the trajectory of the reflected photons. Note that although the trajectory is in mirror quadrant three, the extent of the cone allows light to reach mirror quadrant one.

the mirror quadrants are interleaved, leaving no gap between the various quadrants. As explored in Chapter 3, a particle which produces Cherenkov radiation will do so in a cone, the base of which has a radius of 4 cm to 7 cm in the mirror plane (assuming $\beta \sim 1$, CO_2 with $n = 1.00046$ gives a 4 cm cone, and C_4F_8O with $n = 1.00139$ gives a 7 cm cone). Thus, a particle which passes through, for example, mirror quadrant three may produce a Cherenkov cone that overlaps with mirror quadrant one. As only a small portion of the light will reach these "off-quadrant" mirrors, these events are ideal to isolate the single photoelectron which leads to an accurate calibration. This is the driving logic behind the quadrant calibration strategy of Chapter 5.

The last HGC specific component to consider is the coupling of the photomultiplier tubes (PMTs) to the Cherenkov media. As one can see in Figure 2.8, the PMTs

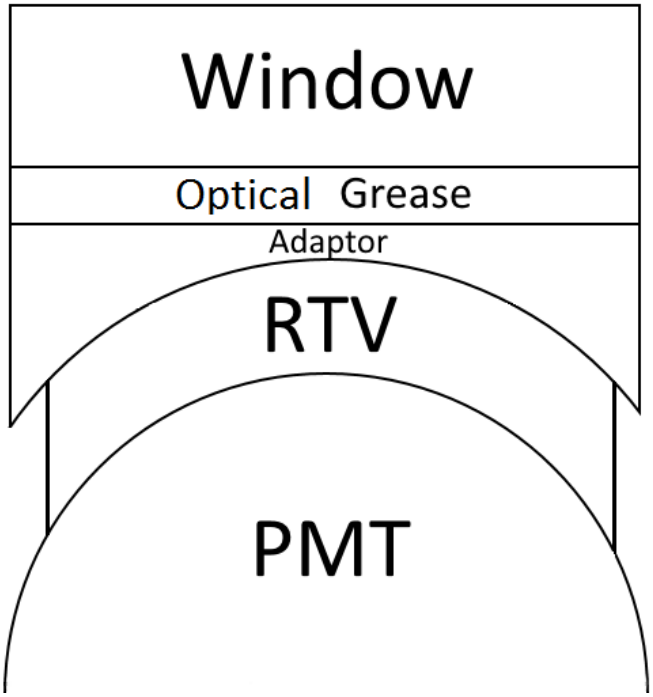


Figure 2.13: Cartoon of the PMT coupling assembly to the HGC window. Dimensions are not to scale.

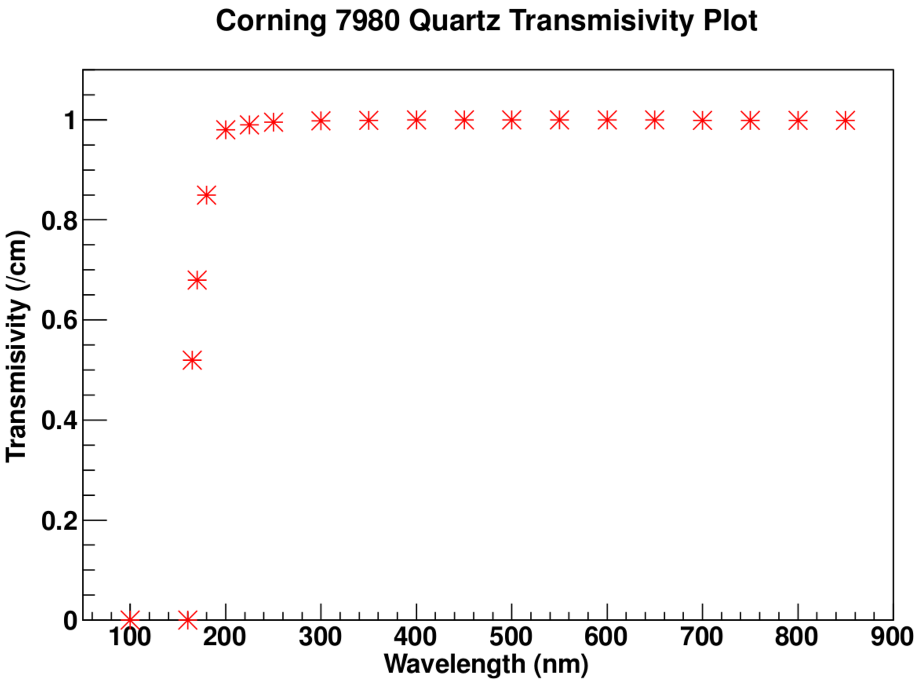


Figure 2.14: Relationship between the transmissivity and wavelength for Corning 7980 quartz, the material used in the HGC window and adaptor (Corning, 2018).

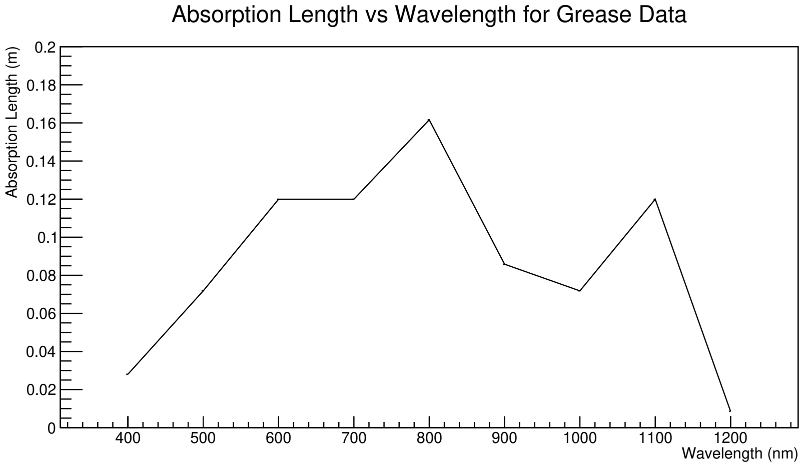


Figure 2.15: Absorption length versus wavelength for Dow Corning Q2-3067 optical couplant (Dow Corning, 1998). Note that the definition of absorption length refers to distance travelled until a proportion is absorbed. Therefore, lower absorption length results in a lower transmission.

are designed to be accessible from outside the tank to facilitate maintenance and to avoid mechanical strain due to variable gas pressure (Li, 2012). Since the tank must be a closed vessel, the PMT surface is coupled to quartz windows in the HGC tank, with care taken to avoid diffraction, dispersion, etc. This is done by matching the index of refraction of the various optical components. A cartoon of the optical configuration is given in Figure 2.13, where it is understood that dimensions are not accurate. Light reflected from the mirrors first passes through a 1 cm thick Corning 7980 quartz window (Corning, 2018) which has an index of refraction $n = 1.45840$, and high transmission of deep UV wavelength light (Abrisa, 2010). The transmission dependence on wavelength is given in Figure 2.14. The window is then coupled to the flat side of a Corning 7980 quartz adaptor (identical optical properties as the quartz window) with Dow Corning Q2-3067 optical couplant (Dow Corning, 1998). This layer of optical grease has sub-millimeter thickness, where its optical properties are shown in Figure 2.15 and has an index of refraction $n = 1.4658$. Although this material has strong absorption for UV photons, the layer is so thin the effect is expected to be minimal. The quartz adaptor is designed to fit both the curved PMT surface and

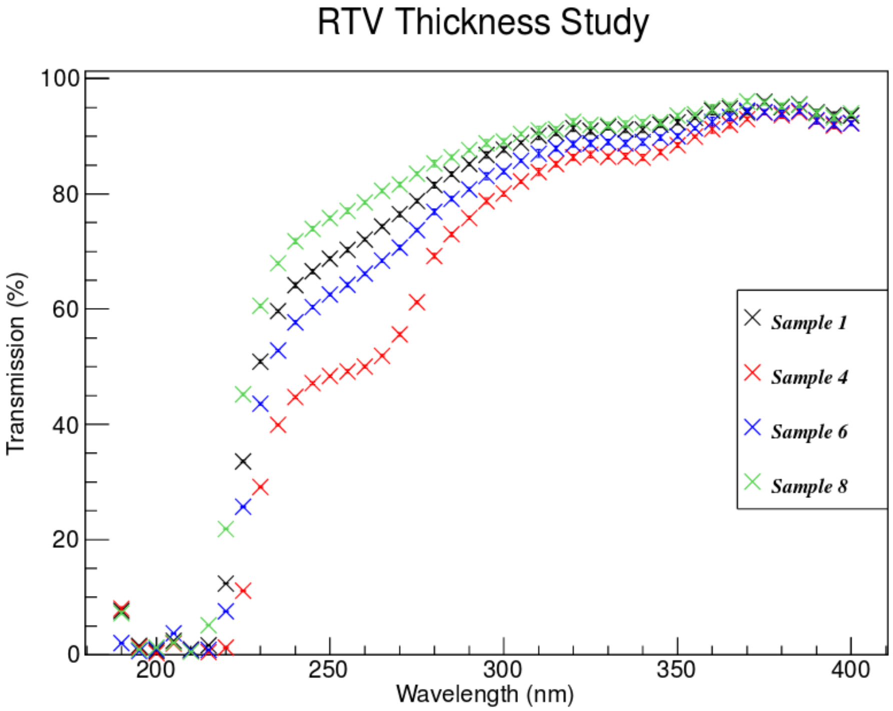


Figure 2.16: Transmission versus wavelength for RTV 615 as measured by the University of Regina group (Li and Huber, 2014). The various samples have differing thickness: sample 1 is 1.90 mm, sample 4 is 3.35 mm, sample 6 is 2.30 mm, and sample 8 is 1.37 mm. In the HGC, the RTV thickness is $\sim 150\ \mu$m Li (2018b).

the flat HGC window, hence the machined shape (Hardin Optical Company, 2018). The adaptor is 2.0 cm in length with a radius of curvature 13.20 cm, giving a width of 0.35 cm at the center. Next, the light passes through a thin room temperature vulcanizing (RTV) 615 silicone cookie designed to fill the air gap between the adaptor and PMT. The RTV cookie was produced by Momentive (Momentive, 2018) and the optical properties were measured separately by two Jefferson Lab groups (Wilson, 2014) (Li and Huber, 2014). The RTV is approximately 150 μm (Li, 2018b) and the transmission for various wavelengths and thicknesses are given in Figures 2.16 and 2.17 for the University of Regina and North Carolina groups respectively. The index of refraction for RTV 615 is $n = 1.406$.

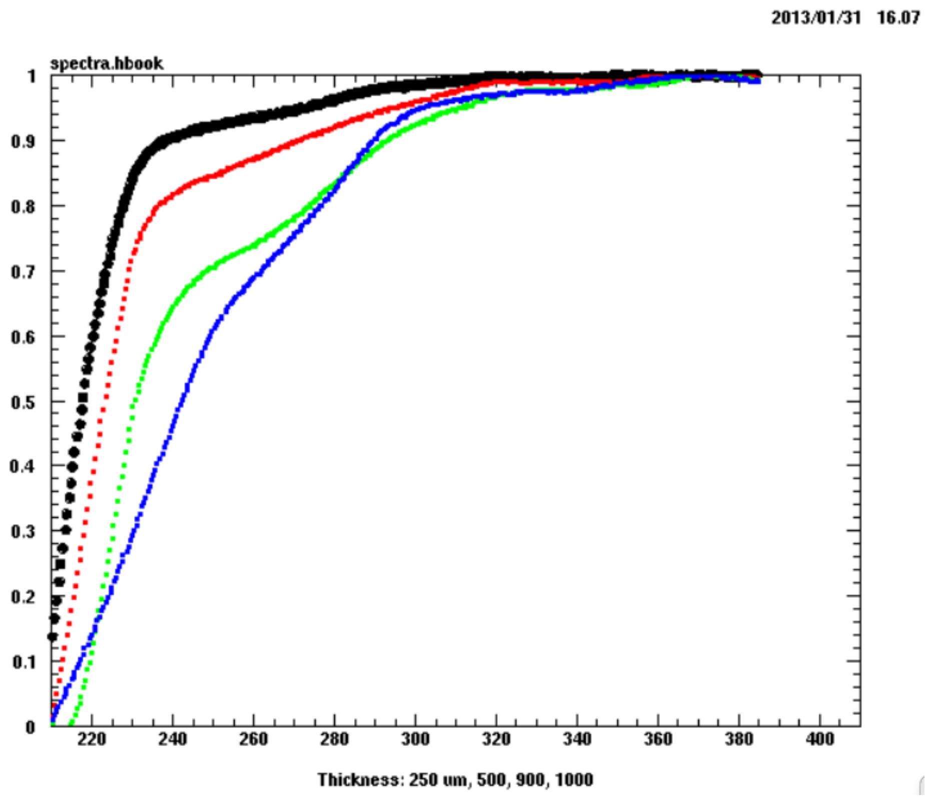


Figure 2.17: Transmission versus wavelength for RTV 615 as measured by the North Carolina group (Wilson, 2014). The various colours have differing thickness: black is 250 μm, red is 500 μm, green is 900 μm, and blue is 1000 μm.

2.1 Cherenkov Detectors

As mentioned in Chapter 2, the purpose of Cherenkov detectors is particle identification based on velocity. The specific physics involved are discussed in detail in Chapter 3, but of interest currently is that heavier particles will produce less light in a Cherenkov detector. Thus, to determine which particle is observed one must collect the light produced. For this purpose, the HGC, ACD, and NGC are outfitted with photomultiplier tubes (PMTs), which collect light and convert it into an electrical signal. Once this has been done, each PMT must be gain matched to have the same response to incident light, and then the signal must be converted to the physically meaningful number of photoelectrons (Chapter 5).

2.1.1 Photomultiplier Tubes

PMTs have similar schematics consisting of: a photocathode, an electron multiplication system, and an anode where the signal can be taken (Leo, 1987). A simple schematic is presented in Figure 2.18, where it is understood the specific geometry is approximate. The photocathode stage uses the photoelectric effect to convert incident Cherenkov radiation into a current of electrons. This photosensitive material is placed on the inside of the PMT window, which is typically made of glass or quartz to allow a wide range of photon wavelengths to easily enter the tube. The photoelectric effect is dictated by Einstein's well-known formula,

$$E = h\nu - \phi, \tag{2.1}$$

where E is the energy of the emitted electron, ν is the frequency of the Cherenkov light, and ϕ is the work function giving the threshold of wavelength required before the effect may occur. However, even above this threshold, the photoelectric effect has a strong probability dependence on frequency called the quantum efficiency (Leo, 1987)

$$\eta(\lambda) = \frac{\text{Number of photoelectrons released}}{\text{Number of incident photons on cathode}(\lambda)}, \tag{2.2}$$

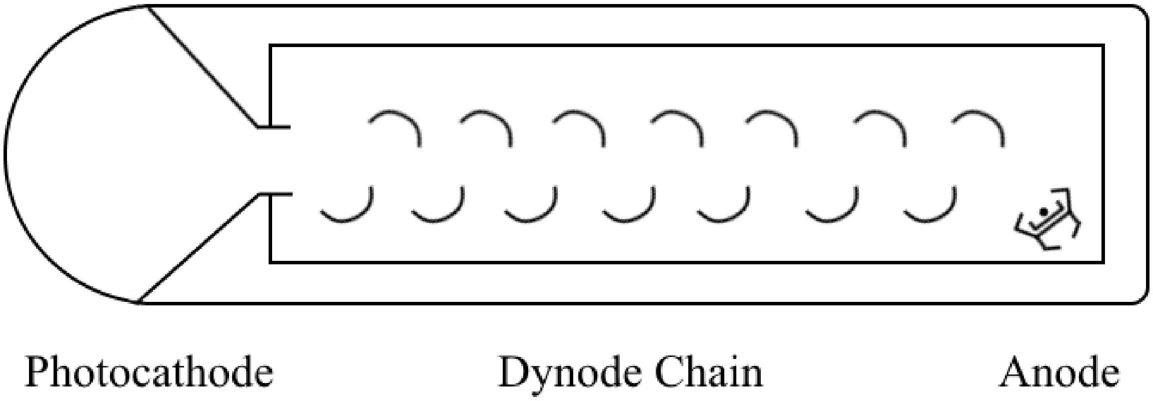


Figure 2.18: Schematic diagram of a photomultiplier tube (PMT). The general design has three components: a photocathode, a dynode chain, and an anode.

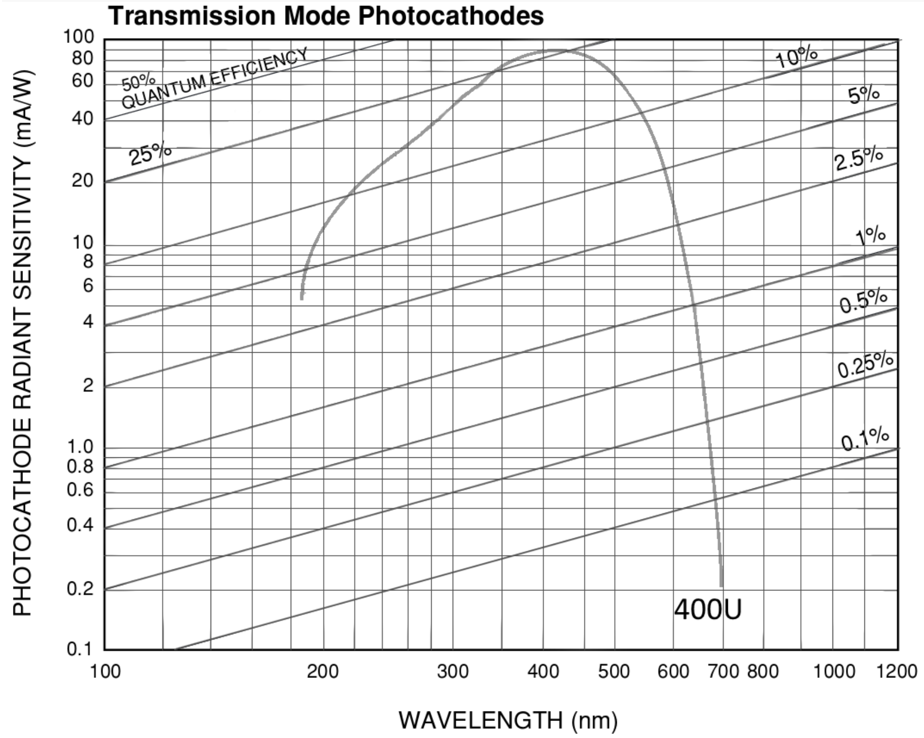


Figure 2.19: Typical spectral response (quantum efficiency indicated by diagonal lines and radiant cathode sensitivity by horizontal lines) for bialkali, the photocathode material for R1584 PMTs with an UV transparent window. Corresponds to Hamamatsu curve code 400U with spectral response range from 185 to 650 nm (Hamamatsu, 2006).

where this quantity has a strong dependence on both photon wavelength and the material's structure. For the HGC, the photoactive material used is bialkali, which has a quantum efficiency depicted in Figure 2.19 (Hamamatsu, 2006). Also shown is the radiant cathode sensitivity (or cathode radiant sensitivity), which is equivalent to the quantum efficiency and is given by (Leo, 1987)

$$E(\lambda) = \frac{\lambda \eta(\lambda)}{1240} \,[\text{Amps/Watt}]\,. \tag{2.3}$$

As previously mentioned, the window material is an important factor in the production of photoelectrons. A sample of material transmittance against wavelength is given in Figure 2.20 (Hamamatsu, 2006), where the HGC PMTs use UV transmitting glass. These wavelength dependence plots were used in the simulation studies of

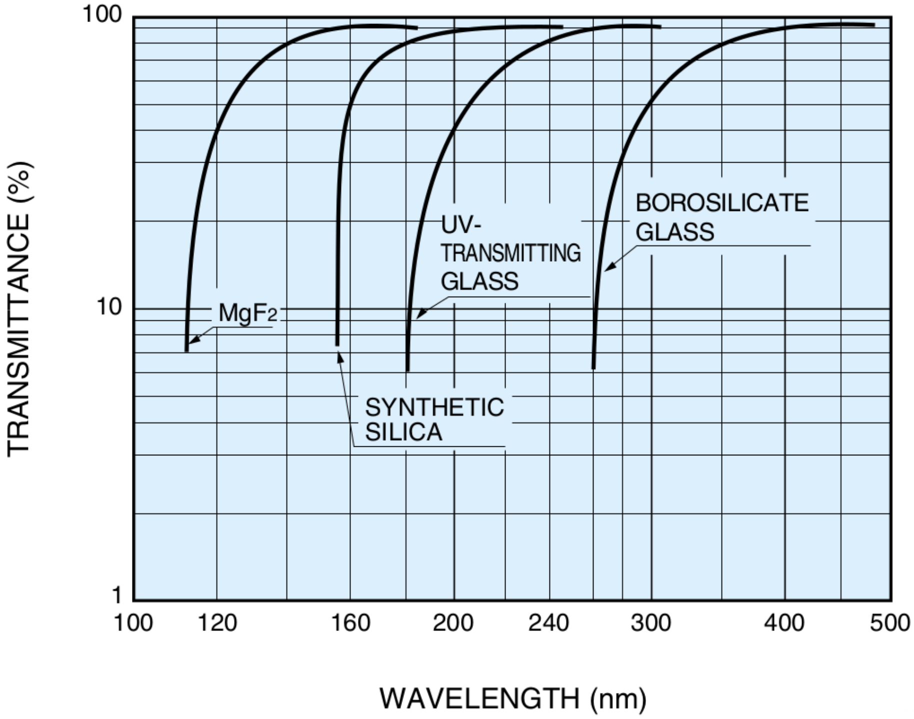


Figure 2.20: Typical transmittance of various window materials where R1584 PMTs have UV-transmitting glass (Hamamatsu, 2006).

Chapter 7.

The next stage of the PMT is the dynode chain, whose purpose is to amplify the electron signal from the photocathode into a measurable signal at the anode. The specific geometry of this portion is PMT-dependent, the SHMS HGC uses R1584 PMTs, which have a linear-focused configuration with 14 stages (Hamamatsu, 2006). The low photoelectron current from the photocathode is focused by an electric field onto the first stage of the dynode chain. The energy of the accelerating photoelectron is transferred to the electrons in the dynode material, allowing a large number of secondary electrons to be emitted (Leo, 1987). This current increases as it flows through the chain, the amount of which is called the gain. The specific gain at each electrode is known as the secondary emission factor δ (Leo, 1987) and is given by (Hamamatsu, 2006)

$$\delta = AE^{\alpha}, \tag{2.4}$$

where A is constant, E is the interstage voltage, and α is a dynode-specific coefficient (typically between 0.7 to 0.8) (Hamamatsu, 2006). For an ideal PMT, each stage has an identical secondary emission factor and the total gain is (Hamamatsu, 2006)

$$G = \delta^{n}, \tag{2.5}$$

where n is the number of dynode stages. Placing Equation 2.4 into Equation 2.5 yields

$$G = (AE^{\alpha})^{n} = \left(A \cdot \left(\frac{V}{n+1}\right)^{\alpha}\right)^{n} = KV^{\alpha n}, \tag{2.6}$$

where a voltage V is applied between the cathode and anode of the PMT (K is a constant). Considering the R1584 PMTs used in the HGC, $\alpha n \approx 10$, and so a fluctuation in applied voltage will cause a ten-fold change in the measured gain. For reference, the typical gain for the R1584 is 1.4×10^{7} at 2000 V (Hamamatsu, 2006). Lastly, the pulse is collected from the anode to be processed, the specifics of which are described in Chapter 4.

2.1.2 Previous Work

Cherenkov detectors are a common feature in particle physics experiments and thus consulting other successful calibrations provides background on possible hurdles, benchmarks, etc. In particular, the Hall A aerogel Cherenkov detector (Alexa et al., 1995) (Brash et al., 2002) and an ultra low signal (Degtiarenko, 2017c) calibration and performance are used as references.

The Burle 8854 PMTs used in the Hall A aerogel detector had good single photoelectron (SPE) resolution (Brash et al., 2002), which makes calibration easier since a

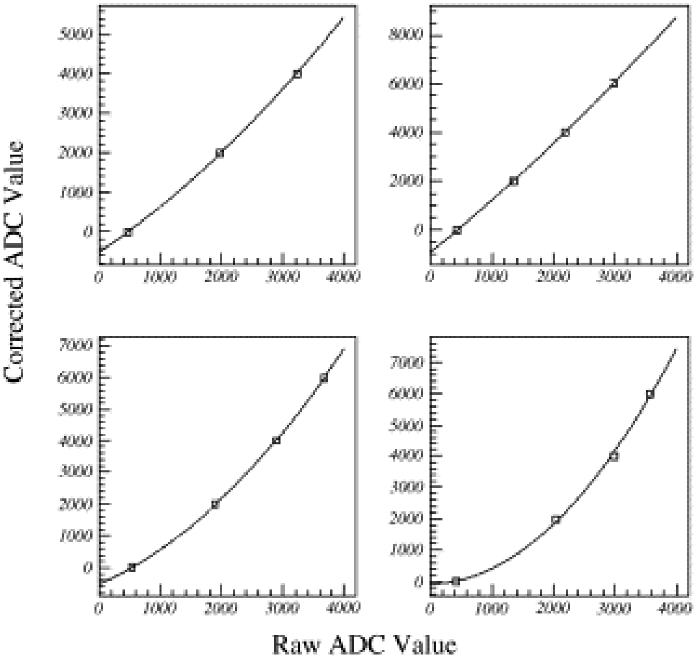


Figure 2.21: Correction to ADC values used in the Hall A aerogel calibration (Brash et al., 2002). A deviation from linearity implies non-linear effects in the gain.

direct translation from number of photoelectrons (NPE) to ADC channel can be obtained. However, the PE were not evenly spaced, implying non-linear gain responses were present and can be observed in Figure 2.21 (Brash et al., 2002).

The source of this non-linearity was attributed to two factors: an impedance-matching resistor added to the resistive dynode chain, and the near-maximum PMT voltage causing space-charge effects towards the end of the dynode chain (Brash et al., 2002). Although changes were made to the HGC base, this resistor was not included (Wenliang Li, 2013). To ensure that space-charge effects, or other non-linearities, are not present in the HGC, the spacing of the PE peaks is considered in Chapter 5.

A thorough consideration of the SPE peak, and resultant spectrum, is provided by Pavel Degtiarenko (Degtiarenko, 2017c). In his report, he addresses the non-linear, complex behaviour in ADC spectra in terms of non-uniformity in the amplification between the various stages in the dynode chain, with emphasis placed on the first stage. This non-uniform response may arise from irregularities in the dynode surface, edge effects, “almost elastic” collisions between the photoelectron and dynode, and other physics. These effects cause the PE peaks to deviate from the expected Poisson or Gaussian shape, increasing the difficulty in obtaining an accurate calibration. Considering only the SPE peak, these processes cause the photoelectron to not multiply properly at the first dynode, causing events to be spread between the noise and peak (Degtiarenko, 2017c). Thus, the true SPE peak is shifted to a lower ADC channel than observed, which can be seen in Figure 2.22 (Degtiarenko, 2017a). This phenomena will be considered when analyzing HGC spectra, where this effect may reduce the SPE peak by 30% (Degtiarenko, 2017b). However, the context for the ultra low signals measured in Degtiarenko’s article is for the RICH detector development at JLab, in which more than 27 thousand channels had to be analyzed. Furthermore, multianode photomultiplier tubes were used, where differing gain effects are expected to be observed due to spatial dependence of the amplification process. Only single anode PMTs are used for the HGC, and so this effect may be minimized.

Once the calibration is completed, there are several metrics to determine the accuracy of the work and to gauge the performance of the detector. The first to consider is the efficiency of the Cherenkov detector, defined as the probability the detector will correctly identify an incident particle and is expressed as (Alexa et al., 1995) (Brash et al., 2002)

$$\eta_{\text{HGC}} = \frac{[\text{Particle ID with HGC}]}{[\text{Particle ID without HGC}]} = 1 - e^{-N_{\text{avg}}}, \tag{2.7}$$

where η_{HGC} is the Cherenkov efficiency and N_{avg} is the average number of PE in the detector. The ratio in Equation 2.7 has as a denominator the combination of cuts on calorimeter, tracking, other Cherenkov’s, etc. to select a particular particle, and as a numerator the identical cuts with an additional cut on the HGC. The details

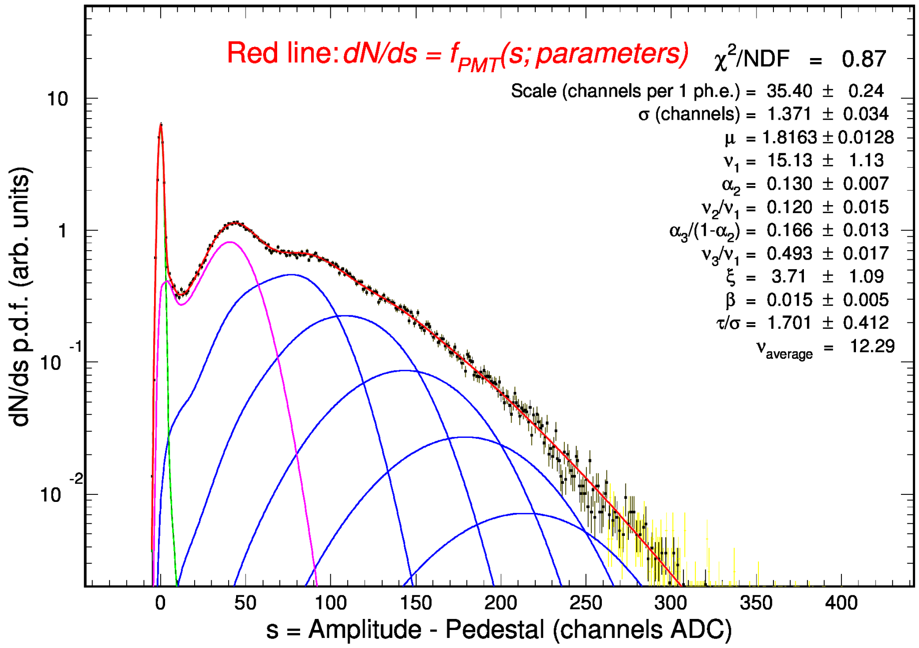


Figure 2.22: Analysis of a low signal ADC spectra by Pavel Degtiarenko (Degtiarenko, 2017a). The green line is the pedestal, pink the SPE spectrum, blue the other PE spectra, and red shows the total fitting function. Of particular note is that the peak of the SPE is at a lower ADC channel than one would assume by fitting the SPE peak with a Gaussian.

on the efficiency of the HGC are described in Chapter 6. Another gauge for the accuracy of the calibration is how well the calibrated spectrum, i.e. the spectrum of produced photoelectrons, is characterized by the expected Poisson statistics. If the collected photons are governed solely by statistical factors, with all other physical effects held constant, the spectrum of produced photoelectrons should be characterized by a distribution of the form (Alexa et al., 1995)

$$y = \frac{\mu^x e^{-\mu}}{\Gamma[x+1]}, \tag{2.8}$$

where y is the number of counts/channel, x is the calibrated ADC value, μ is the mean of the Poisson distribution, and Γ is the usual gamma function given by (Alexa et al., 1995)

$$\Gamma[x] = \int_0^\infty t^{x-1} e^{-t} \mathrm{d}t. \tag{2.9}$$

Examples of calibrated ADC spectra fit using this method are shown in Figure 2.23 (Alexa et al., 1995). Clearly illustrated is the distribution sensitivity on the choice of μ, showing this is an excellent method to verify the accuracy of the calibration. Another test of the calibration is an independent confirmation of the index of refraction of the Cherenkov media. This is based on the concept of a threshold momentum for particles to produce Cherenkov radiation in a media, a topic explored in Chapter 3. Given a particular index of refraction, one can calculate the expected threshold velocity β and momentum $p_{\text{threshold}}$ of a particular particle species. By determining the efficiency of particle identification over a momentum range, one can fit the distribution with a function of the form (Brash et al., 2002)

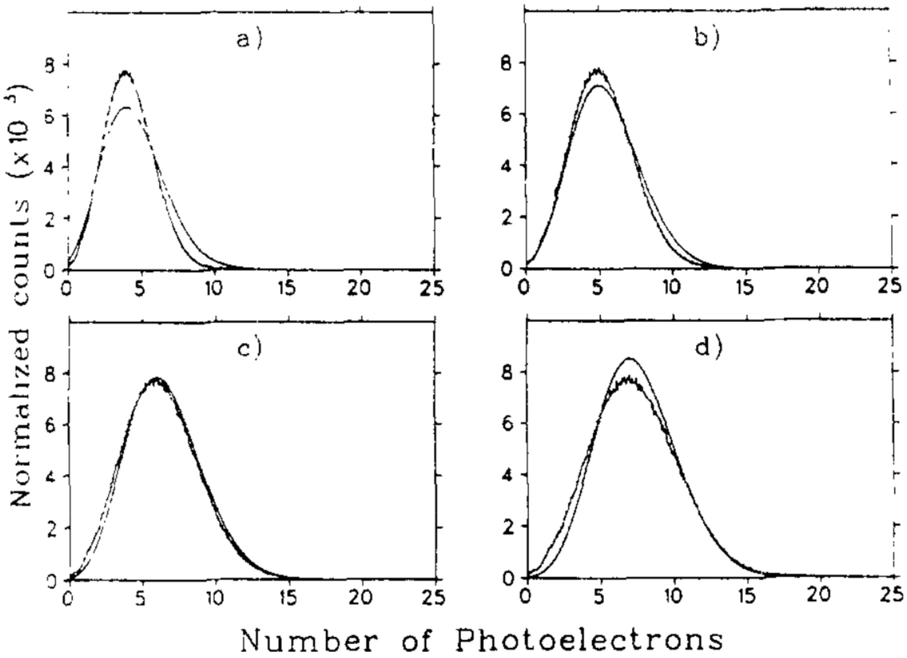


Figure 2.23: Calibrated ADC spectrum from the Hall A aerogel (Alexa et al., 1995) with Poisson distributions assuming different μ and average NPE: a) $\mu = 4.5$, b) $\mu = 5.5$, c) $\mu = 6.5$, d) $\mu = 7.5$. Both the data and fit are shown on the above panels.

$$\eta_{\text{HGC}} = 1 - e^{-(p-p_o)/\Gamma}, \tag{2.10}$$

where p_o and Γ are free parameters. Once one defines the effective momentum thresh-

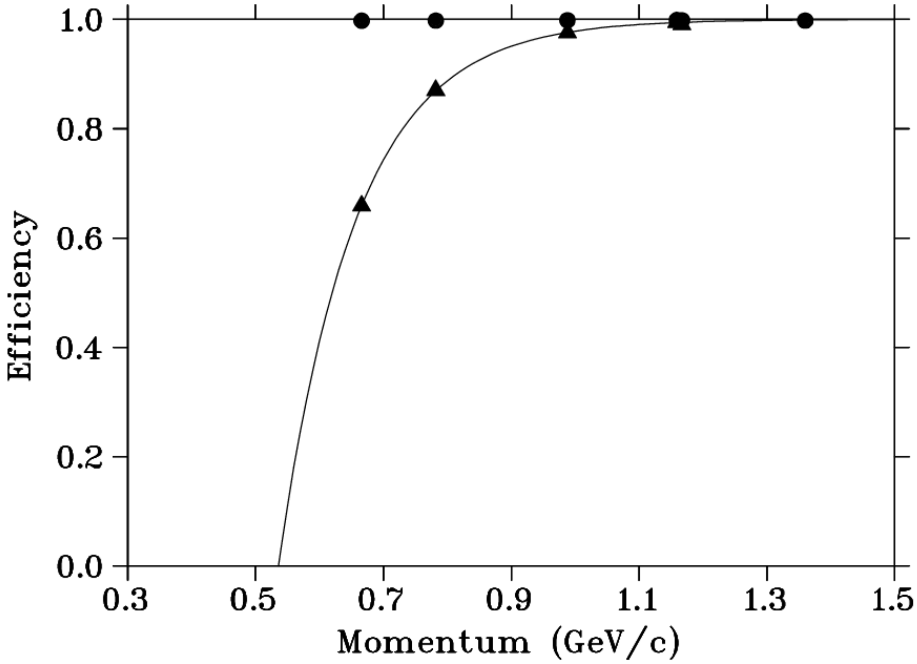


Figure 2.24: Determining the index of refraction of the electron experimentally for the Hall A aerogel (Brash et al., 2002). The triangles refer to pion data, the circles electrons. By fitting pion data with Equation 2.10, one obtains the threshold momentum for the particle to Cherenkov which yields the index of refraction.

old (typically chosen to be $\eta_{\text{HGC}} = 0.50$) (Brash et al., 2002) the value of p can be solved for, which corresponds to $p_{\text{threshold}}$. Thus, the index of refraction can be calculated experimentally to compare to the theoretical value. A visualization of this process for the Hall A aerogel detector is shown in Figure 2.24.

Chapter 3

Cherenkov Radiation

The proper operation of the experimental hall, and therefore the capacity to perform good physics measurements, demands high quality detectors. This requires that every aspect of their signal alone, and in comparison to the rest of the detector package, has to be understood. To begin, one must understand what is capable of being detected. As previously stated, the interaction of the electron beam with a hydrogen target will produce a shower of particles: electrons, pions, kaons, and protons (among others). With the 12 GeV upgrade to the CEBA accelerator, these charged particles will have speeds approaching the speed of light, such that their time of flight will be essentially identical given the spectrometers resolving time. To accurately perform particle identification, Cherenkov detectors with different refractive indices are used. This is based on the principle of Cherenkov radiation, in which the speed of a particle is larger than the phase velocity of electromagnetic fields at a particular frequency

$$v > \frac{c}{\sqrt{\epsilon_r(\omega)\mu_r(\omega)}} = \frac{c}{n}, \tag{3.1}$$

resulting in a cone of radiation being produced. In the above equation, v is the particle velocity, c is the speed of light in vacuum, ω is the frequency of the electromagnetic wave, $\epsilon_r(\omega)$ is the relative permittivity of the media, $\mu_r(\omega)$ is the relative permeability of the media, taken to be one, and n is the index of refraction of the media. This phenomena can be found by considering classical electrodynamics of a relativistic particle. The energy loss per unit distance with impact parameter (b) larger than some electromagnetic source dimension (a) is given by (Jackson, 1999)

$$\left(\frac{dE}{dx}\right)_{b>a} = -ca \ \mathrm{Re} \int_0^\infty B_3^*(\omega) E_1(\omega) \ d\omega, \tag{3.2}$$

where the subscripts represent different components of the electromagnetic field with respect to the velocity (E_1 is the component parallel to $\vec{v}$, B_3 is the component perpendicular to $\vec{v}$). Equation 3.2 represents the energy lost by the particle into regions of space a distance greater than $b = a$ away from its path. The components of electric field and magnetic induction are given by (Jackson, 1999)

$$E_1(\omega) = -\frac{ize\omega}{v^2}\left(\frac{2}{\pi}\right)^{1/2}\left[\frac{1}{\epsilon_r(\omega)} - \beta^2\right] K_o(\lambda b), \tag{3.3}$$

$$B_3(\omega) = \frac{ze\beta\lambda}{v}\left(\frac{2}{\pi}\right)^{1/2} K_1(\lambda b), \tag{3.4}$$

where K_i is the modified Bessel function of order i, ϵ_r is the relative permittivity of the media, and z is the atomic number of the media atom. If we take $|\lambda a| \gg 1$, this corresponds to having energy deposited throughout the medium, far from the source. This results in a large b as well, meaning the Bessel functions will approach their asymptotic forms yielding

$$E_1(\omega, b) = \frac{ize\omega}{c^2}\left[1 - \frac{1}{\beta^2\epsilon(\omega)}\right]\frac{e^{-\lambda b}}{\sqrt{\lambda b}}, \tag{3.5}$$

$$B_3(\omega, b) = \frac{ze\beta}{v}\sqrt{\frac{\lambda}{b}} e^{-\lambda b}. \tag{3.6}$$

In this limit, Equation 3.2 becomes

$$\left(\frac{dE}{dx}\right)_{b>a} = \frac{z^2e^2}{c^2}\,\mathrm{Re}\int_0^\infty \left(-i\sqrt{\frac{\lambda^*}{\lambda}}\right)\omega\left[1-\frac{1}{\beta^2\epsilon(\omega)}\right]e^{-(\lambda+\lambda^*)a}\,d\omega. \tag{3.7}$$

This equation gives the energy deposited far from the path of the particle in terms of the Fourier components of the source. Looking at general behaviour, if the real component of $\lambda > 0$, the exponential will cause this expression to vanish rapidly with increasing a. This implies all the energy is deposited near the path of the particle. If λ is purely imaginary, the real component of the exponential is unity and the expression will be independent of a. Therefore, some energy will propagate to infinity as radiation. Given the general expression for wavelength (Jackson, 1999)

$$\lambda^2 = \frac{\omega^2}{v^2}\left[1-\beta^2\epsilon(\omega)\right], \tag{3.8}$$

it is clear that λ is imaginary if $1-\beta^2\epsilon(\omega) < 0$ (assuming $\epsilon(\omega)$ is real, so there is no absorption. This is a common assumption, but in general mild absorption can be allowed) which immediately leads to the original condition in Equation 3.1.

Equation 3.7 also implies a wavelength dependence on the energy deposition, and therefore effects the Cherenkov threshold as well. This can be seen by expressing the index of refraction as (Jackson, 1999)

$$n = \sqrt{\epsilon_r(\omega)\mu_r}\ , \tag{3.9}$$

where ϵ_r is the relative permittivity and μ_r is the relative permeability of a media. For most natural materials in the optical frequency range, one can assume $\mu_r = 1$ (Griffiths, 2013). Thus, one needs only consider the behaviour of relative permittivity, which can be expressed as (Jackson, 1999)

$$\epsilon_r(\omega) \simeq 1 + \frac{4\pi N e^2}{m} \sum_j \frac{f_j}{\omega_j^2 - \omega^2 - i\gamma_j \omega}, \tag{3.10}$$

where N is the number of molecules per unit volume, ω_j is the natural resonant frequency of the molecule, γ_j is the damping ratio of electromagnetic fields in each molecule, and f_j is the number of electrons that have the resonant frequency ω_j and damping ratio γ_j. There are assumptions folded into Equation 3.10 which need to be addressed: it is assumed the substance has relatively low density, the amplitude of oscillation is sufficiently small to permit evaluation of the electric field at the average position of the particle, and that the electromagnetic fields can be expressed as harmonic plane waves. Since the Cherenkov media for the HGC is gaseous, the energy

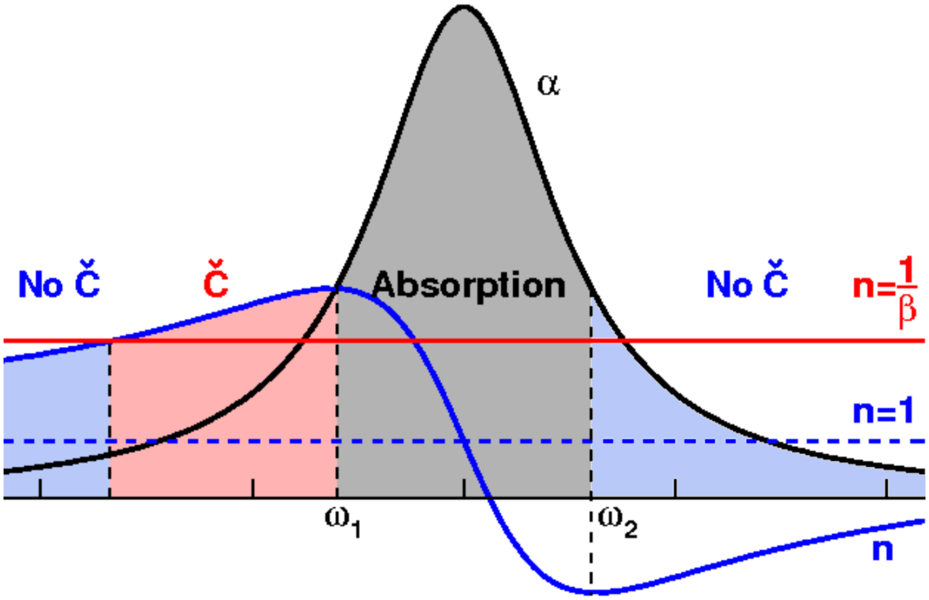


Figure 3.1: Relationship between index of refraction n and absorption α versus frequency ω. Index of refraction is the blue curve, absorption the black curve. The solid red line indicates the Cherenkov threshold condition, $n = \frac{1}{\beta}$, below which radiation cannot be produced. The dotted blue line indicates the index of refraction $n = 1$. The coloured bands refer to competing Cherenkov effects: blue band does not satisfy $n = \frac{1}{\beta}$, red band allows Cherenkov radiation, and the gray band is the absorption region where Cherenkov radiation will not propagate. Note that ω_1 and ω_2 represent the boundaries of anomalous dispersion, where $\frac{\mathrm{d}n}{\mathrm{d}\lambda} > 0$ (Li, 2012).

deposition per unit length is sufficiently small, and the permittivity is considered on a molecular scale, allowing a plane wave description, Equation 3.10 should be valid. By inserting Equation 3.10 into Equation 3.9 and applying the Taylor series expansion one obtains

$$n \simeq 1 + \frac{2\pi N e^2}{m} \sum_j \frac{f_j \left(\omega_j^2 - \omega^2\right)}{\left(\omega_j^2 - \omega^2\right)^2 + \gamma_j^2 \omega^2} + \cdots . \tag{3.11}$$

Figure 3.1 shows the relation Equation 3.11 versus ω. Also depicted is the absorption coefficient α of the dielectric media (Griffiths, 2013)

$$\alpha \simeq \frac{4\pi N e^2 \omega^2}{mc} \sum_j \frac{f_j \gamma_j}{\left(\omega_j^2 - \omega^2\right)^2 + \gamma_j^2 \omega^2}, \tag{3.12}$$

where this has a clear frequency dependence as well. Thus, there exist two mechanisms restricting the emission frequency bands for Cherenkov radiation, a varying index of refraction and absorption. Of particular note is the correlation between the dispersive regime and the absorption. The regime of anomalous dispersion $\left(\frac{\mathrm{d}n}{\mathrm{d}\lambda} > 0\right)$ is bounded by ω_1 and ω_2 on the lower and higher ends respectively, coinciding with the maximum absorption region. Outside this region, normal dispersion $\left(\frac{\mathrm{d}n}{\mathrm{d}\lambda} < 0\right)$ occurs. The specific wavelengths (and therefore frequencies) at which the media used in the HGC fully absorbs Cherenkov radiation are shown in Figure 3.2. Thus, it is clear why so much effort was spent in the HGC construction (end of Chapter 2 and Chapter 2.1.1) to facilitate the collection of ultra-violet light ($\lambda \approx 200$ nm). Additionally, Equation 3.7 shows that

$$\left(\frac{dE}{dx}\right)_{b>a} \propto \frac{1}{\lambda^2}, \tag{3.13}$$

which implies that (Fernow, 1986)

$$\left(\frac{dN_e}{dx}\right)_{b>a} \propto \frac{1}{\lambda}, \tag{3.14}$$

where N_e refers to number of photoelectrons. Thus, less light will be produced at

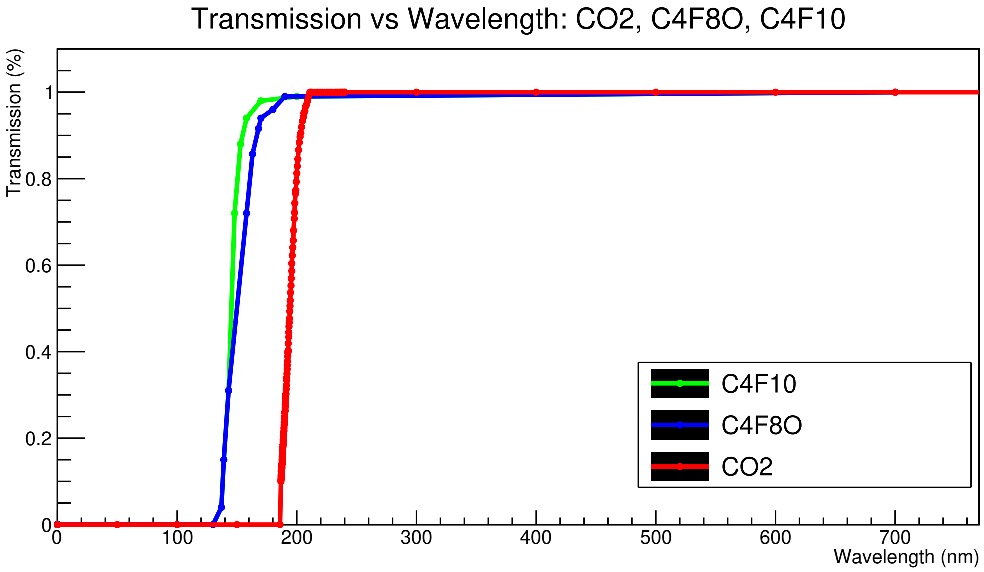


Figure 3.2: Wavelength dependence on the transmission of light through the heavy gas media CO_2, C_4F_8O, and C_4F_{10} (Huber, 2018a). Particular attention should be placed at the limit where the media fully absorbs the light, corresponding to the value of ω_1 on Figure 3.1.

larger photon wavelengths, and emphasizing the need to optimize collection of UV light.

Another useful consequence of Cherenkov radiation is the characteristic emission angle of the emergent cone of radiation shown in Figure 3.3. This angle is given by the ratio of parallel and perpendicular electric fields, which reduces in the far fields approximation as

$$\cos(\theta_C) = \frac{1}{\beta\sqrt{\epsilon(\omega)}}. \tag{3.15}$$

This angle can also be obtained from basic trigonometry arguments, after consulting Figure 3.3. Therefore, knowing the index of refraction and measuring the Cherenkov angle can yield the velocity of the radiating particle. However, the HGC is not designed to measure Cherenkov cone angle and, due to the nearly identical speed of the particle species, it would not yield much information. Despite this, estimating the dimension of the Cherenkov cone in the HGC is straightforward with Equation 3.15. Assuming the HGC is filled with CO_2, at 1 atm and 20^o C the relative permittivity is 1.00092 (Bergström and Sunner, 1976) which yields a Cherenkov angle $\theta_c = 1.73^o$. The HGC is approximately 130 cm long, thus the Cherenkov cone base will have a radius of approximately 4 cm. An identical calculation can be done for C_4F_8O using $n = 1.00139$. This yields a Cherenkov cone radius of 7 cm.

The HGC in the SHMS is designed to provide good $\pi^{\pm}$ identification over a momentum range of 3-11 GeV/c. This is achieved by varying the index of refraction of the heavy gas, which will cause only particular particles to emit radiation. From Equation 3.1, it is clear that Cherenkov radiation will occur if

$$n > \frac{1}{\beta}, \tag{3.16}$$

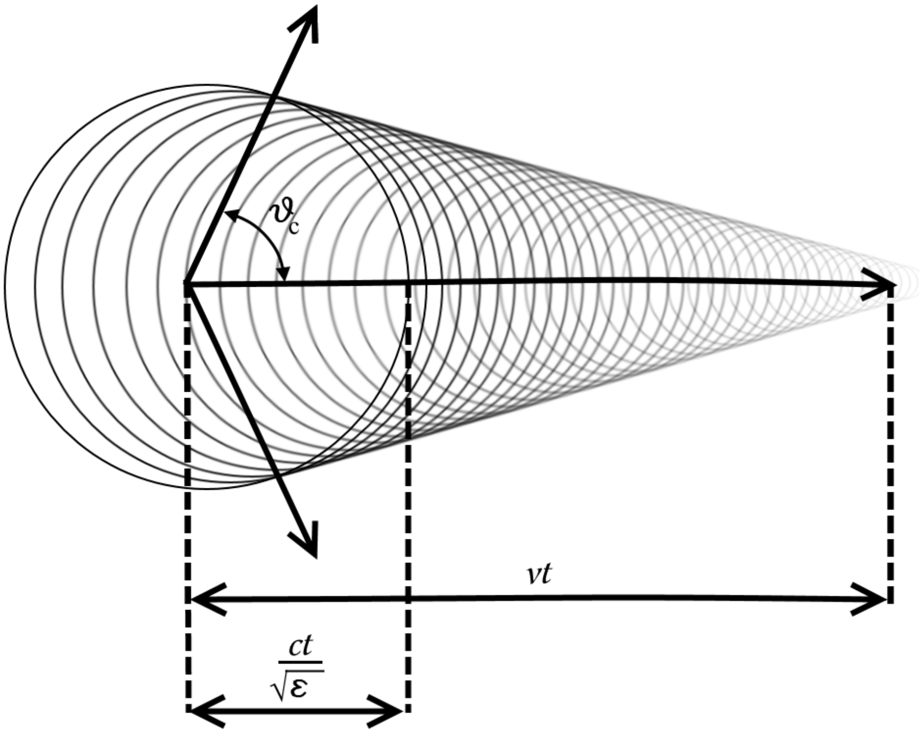


Figure 3.3: Visualization of the emergence of a characteristic angle from Cherenkov radiation given by Equation 3.15

and from special relativity,

$$p = \gamma m \beta c, \tag{3.17}$$

$$E = \gamma m c^2, \tag{3.18}$$

$$\frac{p}{E} = \frac{\gamma m \beta c}{\gamma m c^2}, \tag{3.19}$$

$$\therefore \beta = \frac{pc}{E}. \tag{3.20}$$

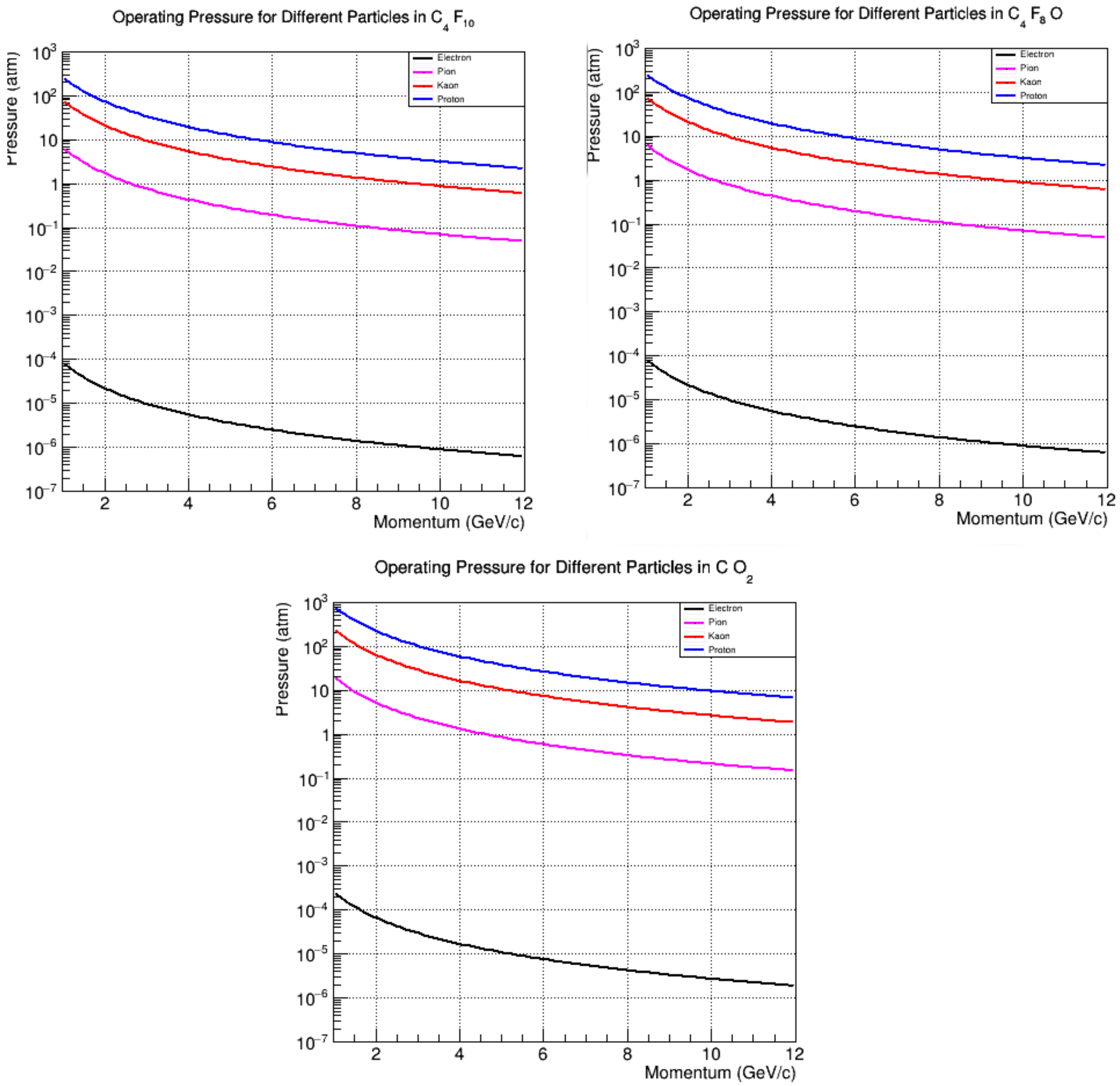
Combining the Cherenkov condition, the expression for β, and relativistic energy, one obtains

$$n > \frac{\sqrt{p^2c^2 + m^2c^4}}{pc}, \tag{3.21}$$

clearly demonstrating that for a particular index of refraction and momentum, particles can be selected for depending on their mass. A summary of the necessary indices of refraction for several momenta ranges for the four most common particles is presented in Table 3.1. For example, if one desires to separate electrons and pions at 3 GeV/c, the index of refraction for the gas needs to be $1.00108 < n_{\text{detector}} < 1.01345$. For calibration purposes, the HGC was initially filled with CO_2, an inexpensive gas with an index of refraction $n = 1.0005$ at 1 atm. When physics data was collected, C_4F_8O was used instead as it can better separate $\pi^{\pm}$ and $K^{\pm}$. The explicit relation-

Momentum	n_e	n_π	n_K	n_p
GeV/c^2	0.5 MeV/c^2	139.57 MeV/c^2	493.67 MeV/c^2	938.27 MeV/c^2
3	1.00108	1.01345	1.01435	1.04778
5	1.00039	1.00486	1.00519	1.01745
7	1.00020	1.00248	1.00265	1.00894
9	1.00012	1.00150	1.00160	1.00542
11	1.00008	1.00101	1.00107	1.00363

Table 3.1: Table of the index of refraction requirements to generate Cherenkov radiation (Li, 2012).


Operating Pressure for Different Particles in C_4F_{10}

Operating Pressure for Different Particles in C_4F_8O

Operating Pressure for Different Particles in CO_2

Figure 3.4: Threshold for Cherenkov radiation in C_4F_{10}, C_4F_8O, and CO_2.

ship between index of refraction and gas pressure is given by (Li, 2012):

$$P = \frac{(n-1)}{(n_{1\text{atm}}-1)}, \tag{3.22}$$

which is valid up to the vapour pressure of the gas. For heavy gases the vapour pressure is approximately 2 atm, at which point it condenses into a liquid. Example plots of Cherenkov threshold pressure vs particle momentum for CO_2, C_4F_8O, and C_4F_{10} are presented in Figure 3.4. Note that, as seen in Equation 3.22, pressure and index of refraction have a linear relationship. Thus, Figure 3.4 will have identical trends as index of refraction vs momentum. When determining the proper pressure to select for particles, one has to keep in mind the SHMS has a momentum acceptance of $-10\% < \delta < 22\%$, so a central momentum of 8 GeV/c will allow any particle from 6.96 GeV/c to 9.76 GeV/c. Of particular interest is that to cleanly separate pions and kaons with C_4F_8O, the gas pressure must be reduced above a momentum setting of 7 GeV/c. This will be done in a linear manner for ease of operations. At a momentum of 11 GeV/c, the HGC filled with C_4F_8O will have a pressure of 0.35 atm (Li, 2012). A last consideration for the Cherenkov pressure is the effect of temperature. As the index of refraction and pressure have a linear relationship, the effect of temperature on pressure will be of similar form to index of refraction. A P-T graph for C_4F_{10} is given in Figure 3.5, where it can be seen the dramatic effect of temperature on pressure. Hall C and the detector hut are temperature controlled to avoid any temperature effects.

A differing level of Cherenkov threshold is not the only quantifiable difference between the three HGC gases: CO_2, C_4F_8O, and C_4F_{10}. The amount of Cherenkov light produced is also gas dependent, as seen in Equation 3.7. In particular, the energy deposition is inversely proportional to the permittivity of the media, and proportional to the square of the atomic number. Additionally, Equation 3.9 shows that the relative permittivity is proportional to the square of the index of refraction of the media. A summary of the properties for the gases used in the HGC are presented

Gas	Atomic Weight	Index of Refraction	UV Absorption Wavelength (nm)
CO_2	22	1.00045	182
C_4F_8O	104	1.00139	125
C_4F_{10}	114	1.00175	125

Table 3.2: Table of the densities and atomic number of CO_2 (Chem Spider, 2018a), C_4F_8O (Chem Spider, 2018c) (Artuso et al., 2006), and C_4F_{10} (Chem Spider, 2018b) (RuggeroTurra, 2009). Measurements are taken at 1 atm and 22^oC for C_4F_8O and 0^oC for CO_2 and C_4F_{10}. Index of refraction is for $\approx$ 200 nm light.

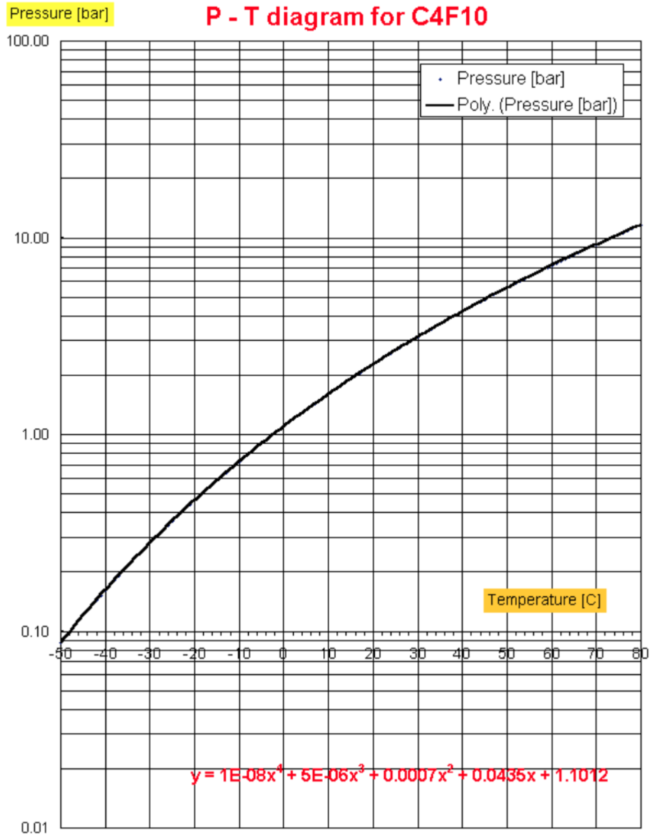


Figure 3.5: Relationship between pressure and temperature for a heavy gas, in this case C_4F_{10} (Vacek, 1999). When considering this with the threshold pressure requirements of Figure 3.4, it is clear that maintaining a stable temperature is important for clean particle identification.

in Table 3.2. As one can see, the index of refraction increases as the HGC is filled with a heavy gas, yielding a larger relative permittivity. However, this effect is offset by the dramatically larger atomic number. Thus, it is expected that the amount of Cherenkov light produced by a heavy gas media will be much more than for CO_2. Lastly, one may wonder why the Cherenkov media chosen for experimental use is C_4F_8O, rather than C_4F_{10}. The reason is due to cost, C_4F_{10} has become prohibitively expensive in the last few years, unlike C_4F_8O.

A major source of background for Cherenkov detectors are δ-rays, or "knock-on" electrons. This occurs when a primary particle, which is below the Cherenkov momentum threshold, ionizes an atom of the Cherenkov media causing a free electron to be produced. Consulting Figure 3.4, it is clear that essentially any electron will produce a Cherenkov signal, and thus the primary particle will be incorrectly identified as an electron (false positive). The kinetics of the δ-ray can be calculated from energy and momentum conservation as (Vovenko et al., 1964),

$$E' \simeq 2m_e \frac{p_\pi \cos(\theta)^3}{M_\pi^2 + p_\pi^2 \sin(\theta)^2}, \tag{3.23}$$

where it is assumed the primary particle was a pion and θ is the trajectory of the electron relative to the primary particle. In the limit $\theta \to \pi/2$, the δ-ray energy tends towards the electron rest mass, $E' \to m_e$. This is the case for pions scattering at very forward angles, where the Coulomb singularity drives the cross section to infinity (Day, 2017). Therefore, the majority of δ-rays will be produced at right angles to the primary particle trajectory. As the mirrors are aligned for tracks down the spectrometer axis, and the PMTs have a limited solid angle acceptance, the contribution from δ-rays is expected to be small. Nevertheless, this phenomena will cause some light to be detected by particles below the Cherenkov threshold. This is the primary motivation behind having the HGC function as a threshold Cherenkov detector. Only events that generate an acceptable number of photoelectrons are accepted as a real particle, events that produce a number of photoelectrons below this threshold are considered δ-rays and are rejected.

Chapter 4

Data Acquisition

Once the electrical signal from the PMTs is emitted (Chapter 2.1), it must be collected and processed before any physical analysis can be performed. The process for collecting the electrical signal is summarized in Figure 4.1. The electrical signal from each PMT is transported from the experimental hall to the counting house using a series of RG-58 and RG-8 cables (Yero, 2017) (impedance of 50 Ω) connected to patch panels with BNC connectors. This covers a distance of 137.5 m introducing a delay of 561.83 ns (Yero, 2017) to the signal. This may affect the timing response measured in the analysis phase, as different lengths of cables will introduce different delays. The possibility of this effect is explored in Chapter 5.

In the counting house, the signal is output from the patch panel as 4 RG-174/U (50 Ω impedance) cables with LEMO connectors (hereafter simply called LEMO cables), one for each PMT. These signals enter a passive splitter, dividing them between ribbon cable and another set of LEMO cables. The ribbon cable is then converted back into four independent PMT signals, each on their own RG-58 wire with LEMO connector and fed into a 16 channel 250 MHz VME64x flash analog-to-digital converter (fADC). The other splitter output is transported to a linear fan in/fan out module where they are summed into HGC Sum, then again split into two signals. One signal is read by

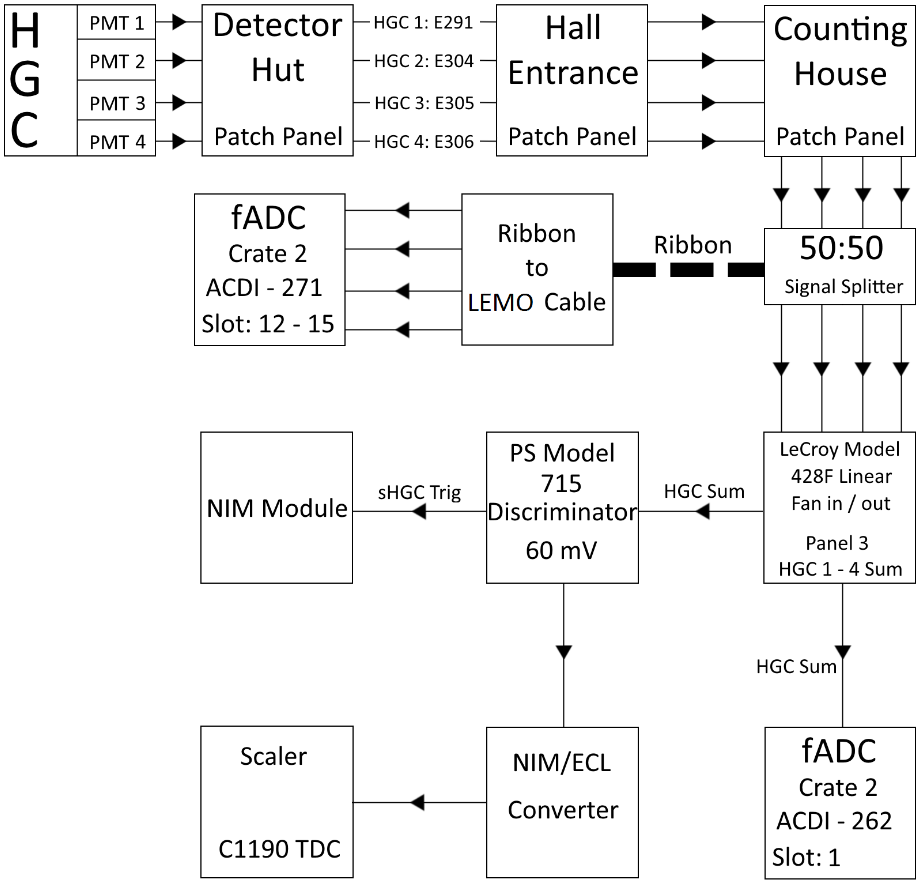


Figure 4.1: Schematic of the procedure to collect the electrical signal from the HGC PMTs.

another fADC, the other is input to a discriminator with a threshold and gate width of -50 mV and 30 ns (Yero, 2017). Two signals are output from here: one is a trigger (sHGC Trig) and read by a NIM module, the other is converted from a LEMO cable to an ECL cable (twisted pair) and read by the scaler and a CAEN 1190 TDC (time to digital converter) module. Thus, the signal from the HGC is used in two settings: the pre-trigger system, and directly output to be processed.

4.1 Trigger System

The trigger system for the 12 GeV era of Hall C is very similar to the previous 6 GeV era, and is shown in Figure 4.2. Note that in this figure the logic units do not represent AND gates, but rather coincident modules. The ratio represents the number of signals that must be received in time, over the total number of input signals. There are 6 possible pre-triggers: STOF, EL_LO_LO, EL_LO, EL_HI, EL_REAL, and EL_CLEAN, where the trigger is chosen according to the requirements of the experiment (Pooser, 2018b). The scaler information for each trigger is read out regardless of pre-trigger selection and is available in the resulting data file. Additionally, the scaler information for each hodoscope plane, S1X, S1Y, S2X, S2Y, PR-LO, PR-HI, SH-LO, 3/4, and CER are available. Recalling that the HGC signal is discriminated if above a threshold of -50 mV, it can be seen that a Cherenkov signal is used as a condition for EL_LO and therefore EL_REAL and EL_CLEAN as well. A similar logic is used for the preshower and calorimeter signals, where the thresholds used for PR_LO, PR_HI, and SH_LO are -40 mV, 60 mV, and -45 mV respectively (Yero,

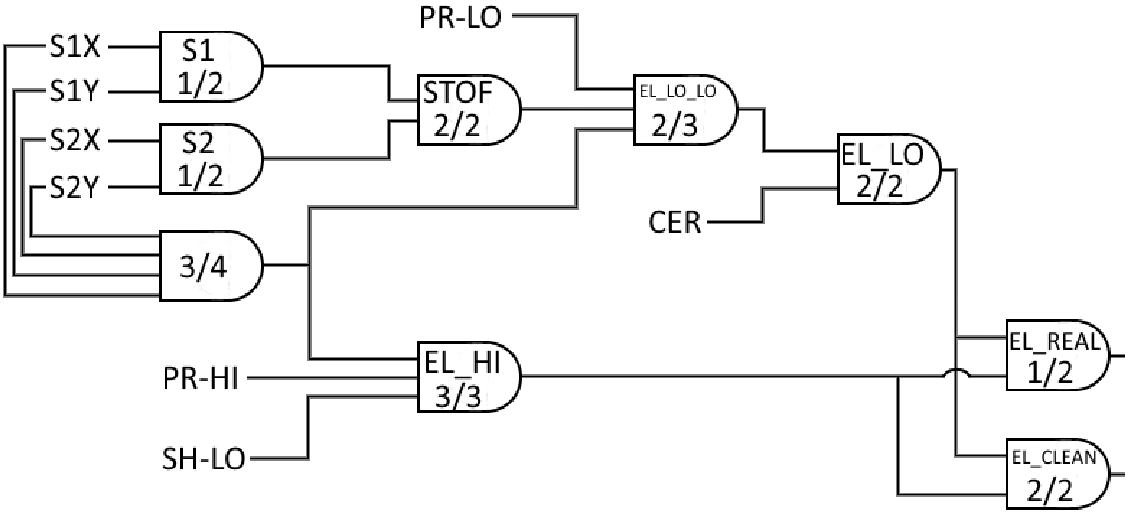


Figure 4.2: Pre-trigger circuit for a either the HMS or SHMS spectrometer in the 12 GeV era. The fraction in each logic units indicates the required number of coincident inputs, out of total inputs, required to generate a logic TRUE pulse.

2017). Lastly, the hodoscope signals S1X, S1Y, and S2X are formed by: a 13/14-fold OR between each panel's left/top PMT (13 on the left for X hodoscopes, 14 on the top for Y hodoscopes) AND a 13/14-fold OR between each panel's right/bottom PMT. The final S2Y plane is more complicated, as it is the quartz neutron veto. A signal is accepted if panels 1 - 16 OR 17 - 21 on top AND panels 1 - 16 OR 17 - 21 on the bottom registered a hit.

Once the pre-trigger has been formed, it is sent to a Trigger Master to be processed as an accepted trigger or, in the case of two particle coincidence experiments between both spectrometer arms, to a Trigger Supervisor. The circuit diagram for a single spectrometer arm trigger is shown in Figure 4.3. The pre-trigger (STOF, EL_LO_LO, etc.) is formed by the combination of detector information fed into a P/S Model 755 NIM Logic unit (Yero, 2017) to form an appropriate coincidence. From this unit, two logical signals are output. One is transported via ribbon cable to a scaler counter and a CAEN 1190 TDC. The other is copied thrice, two of which are sent to CAEN 1190 TDC modules and one as input to the Trigger Master. The TDC input serves as a reference time (REF time on Figure 4.3) to act as a common stop

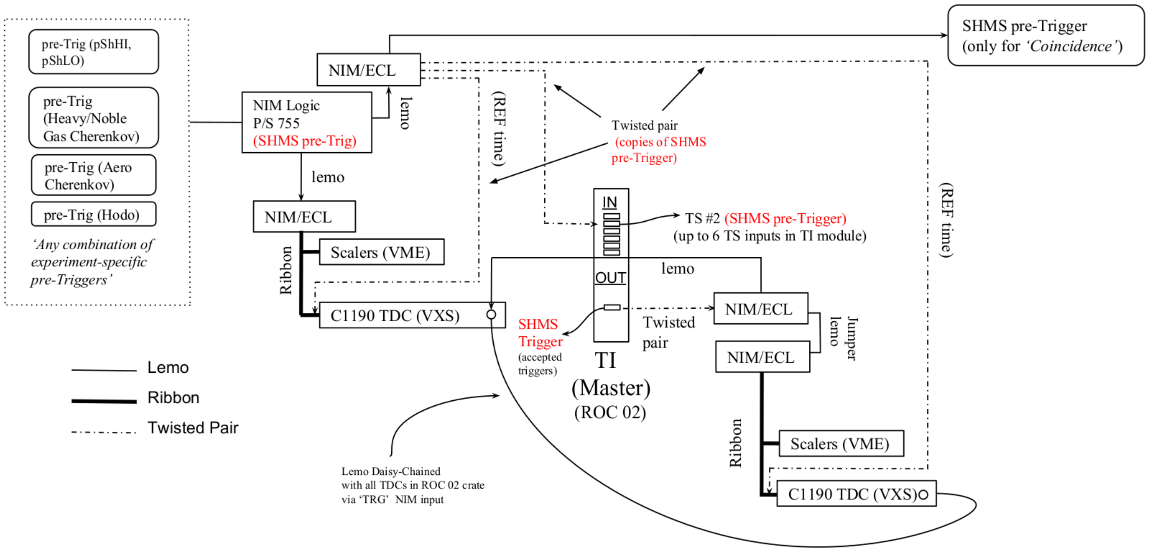

pre-Trig (pShHI, pShLO)
pre-Trig (Heavy/Noble Gas Cherenkov)
pre-Trig (Aero Cherenkov)
pre-Trig (Hodo)
'Any combination of experiment-specific pre-Triggers'
NIM/ECL
NIM Logic P/S 755 (SHMS pre-Trig)
lemo
lemo
(REF time)
Twisted pair (copies of SHMS pre-Trigger)
SHMS pre-Trigger (only for 'Coincidence')
NIM/ECL
Scalers (VME)
Ribbon
C1190 TDC (VXS)
IN
OUT
TS #2 (SHMS pre-Trigger) (up to 6 TS inputs in TI module)
lemo
(REF time)
SHMS Trigger (accepted triggers)
Twisted pair
NIM/ECL
NIM/ECL
Jumper lemo
TI (Master) (ROC 02)
Ribbon
Scalers (VME)
C1190 TDC (VXS)
Lemo
Ribbon
Twisted Pair
Lemo Daisy-Chained with all TDCs in ROC 02 crate via 'TRG' NIM input

Figure 4.3: SHMS Single Arm trigger diagram (Yero, 2017).

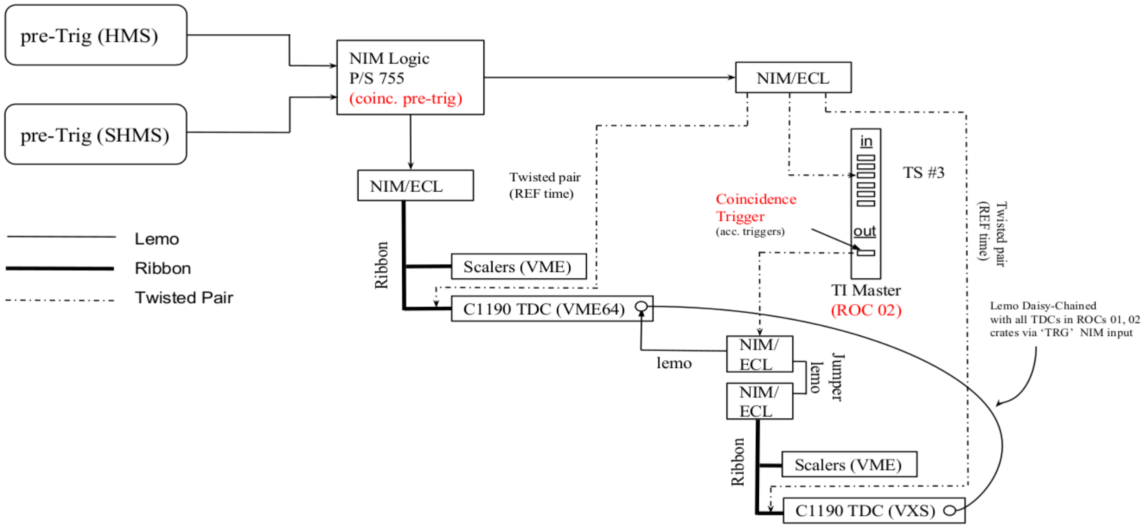


Figure 4.4: Coincidence trigger diagram (Yero, 2017).

for all detector signals being fed into the TDC. The signal sent to the Trigger Master is processed and accepted, then sent to all other modules. A copy of the accepted trigger signal is needed by all ADCs/TDCs in order to initiate readout of all their respective channels. The accepted trigger signal is also recorded by a scaler counter and a TDC. In the case of a coincidence trigger, a similar electronic configuration is used, as shown in Figure 4.4.

A copy of each spectrometer pre-trigger is sent to a NIM Logic unit, where the first trigger to arrive will open a coincidence time window. If the other pre-trigger arrives in time, the logic signal is split and follows a similar pattern as the single arm trigger. The only differences in coincidence mode are that: both spectrometer Trigger Masters are unused, only a Trigger Supervisor is operational, and several more accepted trigger signals are output since every TDC module (including both HMS/SHMS single arm TDCs) will require a coincidence reference time.

4.2 Signal Processing

In order to be analyzed, the Cherenkov signal must be digitized and output in a usable form. This process is performed by the VME64x fADC modules (see Figure 4.1) which measure a predefined number of raw samples at 4 ns intervals (Pooser, 2016) of each pulse within a time window. If any pulse exceeds the user defined threshold, the pulse will undergo an identification procedure by the fADC's field programmable gate array (FPGA). From this component, several useful quantities can be reported: pulse pedestal, integral, peak, and time.

4.2.1 Signal Acceptance

The fADC continuously stores incoming signals in a buffer until a trigger signal is received. The purpose is that if a trigger is registered by the Trigger Master (or Trigger Supervisor, see Chapter 4.1), then it is that signal which is of interest and

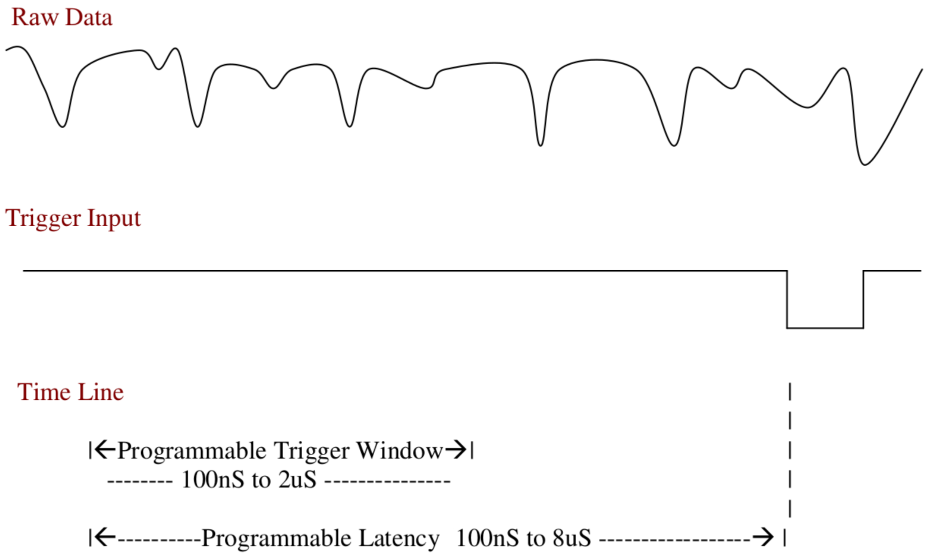

Raw Data
Trigger Input
Time Line
|←Programmable Trigger Window→|
-------- 100nS to 2uS ---------------
|←----------Programmable Latency 100nS to 8uS ------------------→ |

Figure 4.5: Visualization of how the FPGA returns a buffer of data to be analyzed after a trigger signal is received (the low pulse in Trigger Input). Once the trigger is received, all raw data in the programmable trigger window, a subset of the programmable latency, are returned. The programmable latency and trigger window are described in text (Jones, 2008).

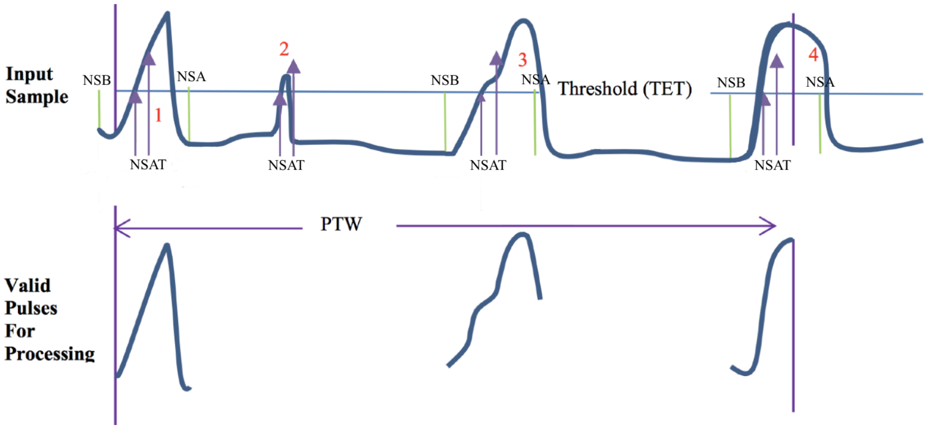


Figure 4.6: Visualization of acceptance/rejection criteria for incoming pulses to the FPGA (Pooser, 2017).

so a buffer of past information must be recalled. This process is shown in Figure 4.5 (Jones, 2008), where the programmable latency (PL) and programmable trigger window (PTW) can be manually set. Once a trigger is received (the low pulse in Figure 4.5), all data from the trigger to the programmable latency that falls within the trigger window are processed (Jones, 2008). While this buffer of information is processed, the fADC will continue storing incoming data with no loss of data (Jones, 2008). Once the appropriate buffer is selected, the pulses must pass a selection criteria to be further processed. This involves a set number of samples occur above threshold (NSAT) (Pooser, 2017), and that enough samples can be taken between pulses to determine the pedestal (Jones, 2008). If these criteria are met, a set number of samples before threshold (NSB) and after threshold (NSA) are returned. If any of these samples fall outside the trigger window (PTW), they will not be processed. This process is shown in Figure 4.6 (Pooser, 2017): signals 1 and 4 are truncated since they lie on the boundaries of the PTW, signal 2 is rejected since it does not satisfy the NSAT, and signal 3 is returned in full. Lastly, a buffer can return a maximum of 3 pulses (Pooser, 2016).

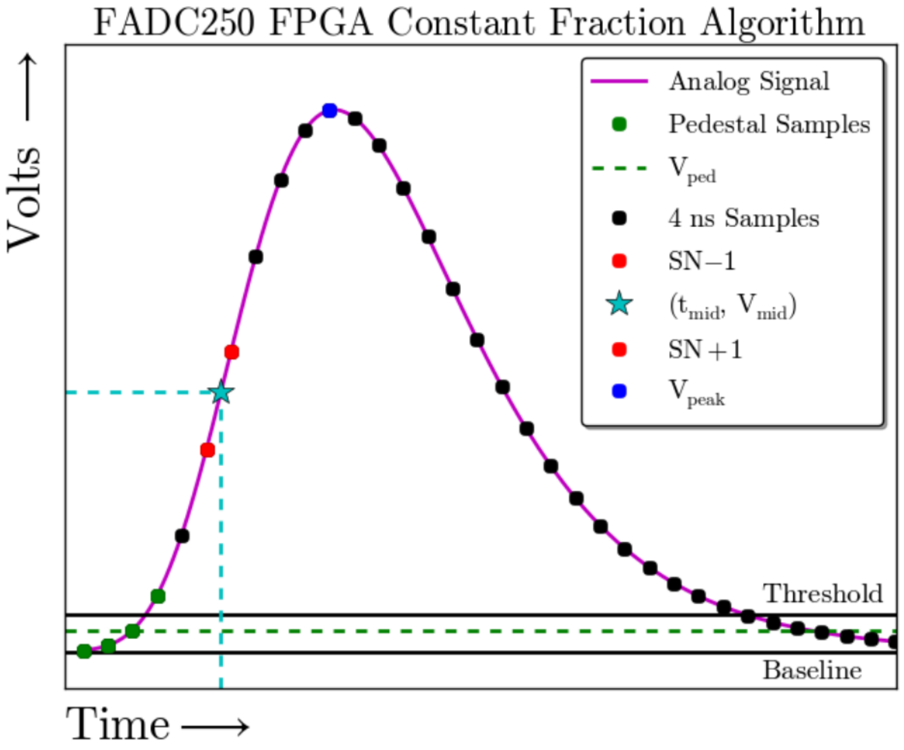


Figure 4.7: Visualization of how the fADC's FPGA digitizes an incoming signal by a constant sampling rate (Pooser, 2016).

4.2.2 FPGA Algorithm

The process by which the fADC's FPGA digitizes an incoming signal is shown in Figure 4.7 (Pooser, 2016). The input signal is sampled continuously every 4 ns for the entire trigger time window (typically 400 ns or 100 samples (Pooser, 2016)). If the input signal crosses threshold, the various algorithms programmed into the FPGA will report useful pulse information. The first quantity of interest is the pulse pedestal, giving the baseline from which the other pulse parameters are measured (Pooser, 2016). The pedestal is calculated by averaging the first four samples, provided at least one exceeded threshold. This is represented by the green points in Figure 4.7, the first point above threshold and the three points prior are averaged together to set the baseline indicated by the green dashed line. The pulse integral is also reported, which is simply the sum of each sample of the pulse subtracted from the pedestal.

Pulse peak is represented by the blue point in Figure 4.7 and is defined as the first instance of a sample value that has a lower value following it. The fADC has a restriction on the range of Volts it may process, and so an upper limit is placed at 2 V. If a pulse exceeds this maximum, a peak of 2 V will be reported instead. The final quantity of interest is the pulse time. This quantity requires additional attention, as the slope of the leading edge may introduce inconsistencies measuring the time. This phenomena is known as time-walk effects, and are corrected for (Pooser, 2016). This is achieved by taking the time it takes the pulse to reach half of its maximum, which is a better metric than time at which the pulse passes threshold (Pooser, 2016). The point at which the pulse reached half of its maximum is taken as the midway point between the pedestal and the peak

$$V_{\text{mid}} = \frac{V_{\text{peak}} + V_{\text{ped}}}{2}, \tag{4.1}$$

where this point is represented by the star in Figure 4.7. Since V_{mid} is known, the FPGA can numerically deduce the samples taken before and after this voltage ($SN-1$ and $SN+1$ respectively) represented by the red points in Figure 4.7 (Pooser, 2016). Thus, the appropriate time t_{mid} is found by performing a linear interpolation between these two points (Pooser, 2016)

$$t_{\text{mid}} - t(SN-1) = t(SN+1) - t(SN-1) \cdot \frac{V_{\text{mid}} - V(SN-1)}{V(SN+1) - V(SN-1)} \tag{4.2}$$

Timing information obtained for the HGC is taken in the fADC operating in high resolution timing mode, which introduces 64 sub-samples, separated by 62.5 ps, associated with each 4 ns sample (Pooser, 2016). This reduces Equation 4.2 to

$$t_{\text{fine}} = 64 \cdot \frac{V_{\text{mid}} - V(SN-1)}{V(SN+1) - V(SN-1)}, \tag{4.3}$$

since $t(SN+1) - t(SN-1) = 4$ ns $= 64$ sub-samples (Pooser, 2016) and $t_{\text{fine}} = t_{\text{mid}} - t(SN-1)$. Due to this calculation method, the timing spectra reported by the fADC is a convolution of Gaussians separated by 4 ns (Pooser, 2016). This is due to the location of V_{mid} occurring anywhere in the 4 ns timing window, giving rise to the Gaussian shape, and the timing window occurring along different points of the leading edge, giving rise to the 4 ns spacing. This jittering is removed by subtracting the fADC timing information from the TDC trigger information, as both have this inherent effect.

4.3 Hall C Analyzer: hcana

The data stream returned from the various fADC, TDC etc. are stored in a .dat file where the information is unwieldy. Instead, the file must be replayed by the Hall C analyzer framework, hcana, built from the Hall A analyzer podd (Hall-C, 2018). The replay process is very intricate: the raw data are decoded, run specific parameters are loaded, detector specific calibration, cuts, geometries, etc. are applied, fADC and TDC slots are mapped to the appropriate detector, global cuts are applied to the data, scaler, event, and EPICS data which are then all readout (Pooser, 2017). Due to the extent and sophistication of the analyzer, only the aspects pertaining to the HGC will be considered.

The hcana software is built from the CERN ROOT data analysis framework, itself constructed using the C++ programming language (ROOT, 2018). The ROOT framework provides many convenient features to store and manipulate data, many of which are used in this analysis. For the current discussion, only two aspects are relevant: the TClonesArray container, and the TTree object.

TClonesArray is a memory-efficient allocation of identical (clone) objects into an array. The advantage of using a TClonesArray over a conventional C/C++ array is that memory fragmentation is avoided, greatly improving program performance. When blocks of memory are reserved for C/C++ arrays, they must be deleted to free up this storage to avoid potential memory leaks. However, this decreases the efficiency of allocating new arrays later as the memory is now divided into blocks. This is called memory fragmentation and causes orders of magnitude loss in performance (ROOT, 2018). TClonesArray avoids this issue by reusing the same portion of memory each time it is created or deleted. Thus, a TClonesArray may be though of as a simple C/C++ array, with sophisticated memory management working behind the scenes.

In a similar vein, TTree is an optimized object designed to store large quantities of same-class objects (ROOT, 2018). The optimization lies in the compression algorithm of the TTree and how data are accessed. Since only same-class objects are stored in a TTree, the bit pattern representing the class is likely very similar facilitating compression. Data are then stored in TTree as a hierarchy of branches, allowing information to be accessed in a parallel manner. Additionally, information is stored across parallel branches as the same event, allowing all pulse information to be easily obtained. This branching structure of storing experiment information is the final output of hcana, and can be saved on disk as a .root file.

From the data file obtained experimentally, hcana will decode the raw hits into TClonesArray structures where the information output from the FPGA algorithms (Chapter 4.2.2) are stored into a unique container each. On the basic level, each TClonesArray is indexed by the number of pulses recorded for a trigger. That index is then filled one pulse at a time, each corresponding to a separate entry. For example, consider a trigger where the HGC PMTs registered the following hits: PMT 1, PMT 3, PMT 2, PMT 1, PMT 1, and PMT 4. The TClonesArray branch for storing PMT counter information will then start filling index five since six pulses were registered. Each PMT counter information will be added as a separate entry as (1, 3, 2, 1, 1, 4) (Pooser, 2018a). A consequence of this is that for raw information, the number of entries in a TClonesArray will very often be much larger than the number of triggers. The index filled for each trigger is stored in a separate branch, where the maximum value is twelve, as the four PMTs can return a maximum of three pulses each. At this raw stage, examining a branch of the TTree does not yield any PMT specific information, as each index of this array may correspond to a different PMT.

To do higher level analysis, one will instead use the “good” HGC branches, which are properly indexed according to PMT number. However, this also includes some cuts to the data as pulses must be selected from the raw TClonesArray. At this stage, only three cuts are applied: FPGA error flag, pulse time, and timing window. The first comes from the FPGA itself; if an error was encountered in applying the various algorithms, the pulse will acquire an error flag to facilitate its removal. The second is to ensure the prompt timing peak occurs in an acceptable range of the programmable trigger window (Chapter 4.2.1). The final cut is applicable if more than one pulse is accepted for a single PMT. If this occurs, then the last hit within the timing cut will be considered the “good” event. These events, due to their convenient ordering and minimum selections, are the quantities used in Chapter 5.

There exists higher-level processed information for the HGC as well, predominantly coming from the drift chamber tracking algorithm. These quantities include several cuts: $0.0 < \chi^2 < 25.0$, $0.0 < \beta < 1.2$, and $-20.0 < \delta < 25.0$ reconstruction, and a normalized energy from the calorimeter between 0.0 and 1.5. As well, only the first reconstructed track is considered. With these restrictions, the distribution of particles in the HGC mirror plane is obtained. A final set of useful quantities are the flags: NumTracksMatched and NumTracksFired. These flags are set if the event track reconstruction resulted in a trajectory intersecting a plane in the HGC (TracksMatched), where an additional minimum light cut, number of photoelectrons greater than 0.5, is applied (TracksFired). These flags are used in Chapter 5.4 to calibrate the heavy gas Cherenkov where the particle trajectory is a relevant quantity.

A summary of the cuts applied by hcana is provided in Table 4.1.

Quantity	Limits
FPGA Error Flag	False
Pulse Time	PMT 1: -20.5 ns $< t < +1.0$ns
	PMT 2: -22.0 ns $< t < -1.0$ns
	PMT 3: -22.5 ns $< t < -7.5$ns
	PMT 4: -22.0 ns $< t < -7.0$ns
Pulse Time Window	Last Hit
Track Number	First Entry
Tracking Accuracy	$0.0 < \chi^2 < 25.0$
Beta Cut	$0.0 < \beta < 1.2$
Normalized Energy	$0.0 < \frac{E}{p} < 1.5$
Momentum Cut	$-20.0 < \delta < 25.0$
NPE Threshold	0.5
Quadrant Centers	Quadrant 1: +27.5 cm, +25.0 cm, 0.0, 0.0
(x, y, θ, ϕ)	Quadrant 2: +27.5 cm, -25.0 cm, 0.0, 0.0
	Quadrant 3: -27.5 cm, +25.0 cm, 0.0, 0.0
	Quadrant 4: -27.5 cm, -25.0 cm, 0.0, 0.0
Quadrant Half Widths	Quadrant 1: +27.5 cm, +25.0 cm, 0.15, 0.1
(x, y, θ, ϕ)	Quadrant 2: +27.5 cm, +25.0 cm, 0.15, 0.1
	Quadrant 3: +27.5 cm, +25.0 cm, 0.15, 0.1
	Quadrant 4: +27.5 cm, +25.0 cm, 0.15, 0.1

Table 4.1: Table of cuts applied by replaying a run with hcana. Consult text for details of which quantities receive which cuts. (Hall-C, 2018).

Chapter 5

Calibration

5.1 Background

This chapter details the methodology and resulting calibration of the SHMS heavy gas Cherenkov for data collected in the winter run of 2017/2018. The specific run analyzed is run 1583, corresponding to log entry 3501980 (hclog, 2017a). The running conditions were a central SHMS momentum of -3.0 GeV/c, carbon 0.5% radiation length target, HGC filled with CO_2 at 1 atm. Note that the polarity of the central SHMS momentum indicates the polarity of the accepted particles. The calibration script makes use of a processed run using hcana. It is important to realize this introduces inherent cuts/selection rules to all subsequent analysis. The details of these cuts can be reviewed in Table 4.1.

The calibration script is available on GitHub (Hall-C, 2018) under the directory CALIBRATION/shms_hgcer_calib/calibration.C, or in Appendix A as Listing A.4, and is formatted according to the TSelector framework for use in batch or in parallel modes using TPROOF (ROOT, 2018). The calibration script was developed by the author, for use by the wider collaboration. The script is broken into five components: Begin, SlaveBegin, Process, SlaveTerminate, and Terminate. Briefly: Begin outputs general information; SlaveBegin initializes histograms and logic conditions; Process acts as a sieve, sorting replayed data into histograms; SlaveTerminate currently has no function; and Terminate does the actual fitting and calibration from the histograms. The procedure for the use of these scripts is detailed in the README.md in the aforementioned directory.

5.2 Overview

Calibration is performed using two similar, but not equivalent, procedures. The first (hereafter called "quadrant" strategy) manually imposes tracking cuts on the data to analyze what each PMT detected from charged particles traversing each mirror quadrant. The second (known as "tracksfired" strategy) takes advantage of the hcana flag NumTracksFired and performs cuts based on whether this quantity was incremented or not. Additionally, both strategies invoke identical cuts: both have a loose timing cut of ±2.5 ns about the prompt timing peak, require there is only one valid reconstructed track, a $\beta = \frac{v}{c}$ cut of $|\beta$ - $1.0| < 0.2$, and allow for particle ID selection rules.

The goal of the calibration is to generate a reliable translation from the raw FADC channel (or charge in pC) to the physically meaningful number of photoelectrons (NPE). This is achieved by isolating the single photoelectron (SPE) peak to generate a calibration, then verify the calibration by calculating the linearity of the PE and their Poisson-like population distribution. Before embarking on the specifics of each strategy, a brief digression into the common cuts on particle ID, timing, and β are given. The particle ID is determined by plotting the energy of the shower calorimeter against the pre-shower. The resulting spectra in Figure 5.1 show two distinct regions, the larger being electrons and the smaller pions according to the expected energy deposition of these particles. An elliptical cut is performed to select the appropriate

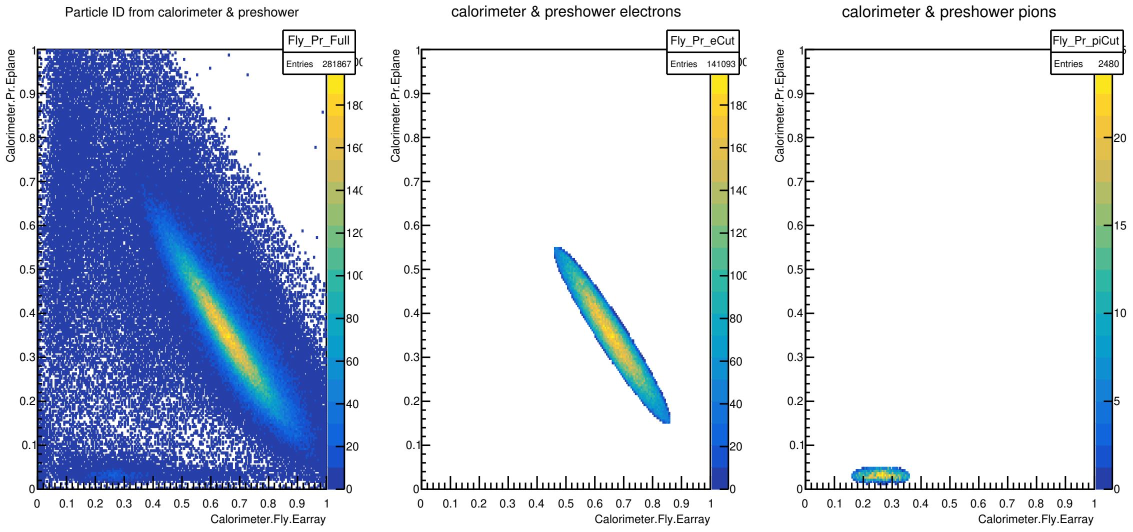


Figure 5.1: Shower and pre-shower response are plotted on the x and y axes respectively and are used to separate electrons and pions. Left: Spectrum before any cuts; Center: Region of selected electrons; Right: Region of selected pions (π^-).

regions according to the relation

$$\frac{\left[(E_{\text{fly}} - x_{\text{center}})\cos(\theta) + (E_{\text{pre}} - y_{\text{center}})\sin(\theta)\right]^2}{a^2} + \frac{\left[(E_{\text{fly}} - x_{\text{center}})\sin(\theta) - (E_{\text{pre}} - y_{\text{center}})\cos(\theta)\right]^2}{b^2} < 1, \tag{5.1}$$

where a and b refer to the semimajor and semiminor axes, respectively, and θ is the angle of rotation about the horizontal. The particle ID cuts are only used in the calibration to analyze the Poisson statistics of the PMTs, since they are only expected to become apparent when electrons are selected for. In Chapter 6, the particle ID is used to determine the electron efficiency and the fraction of false positives generated by pions in the electron sample (pion contamination).

Timing information of the HGC is collected using the “high resolution” pulse time setting in the FADC250’s FPGA, corresponding to reporting the time it takes a pulse to reach a certain threshold. Details of this data collection procedure are described in Chapter 4. An example spectrum is shown in Fig. 5.2, where the distinct timing

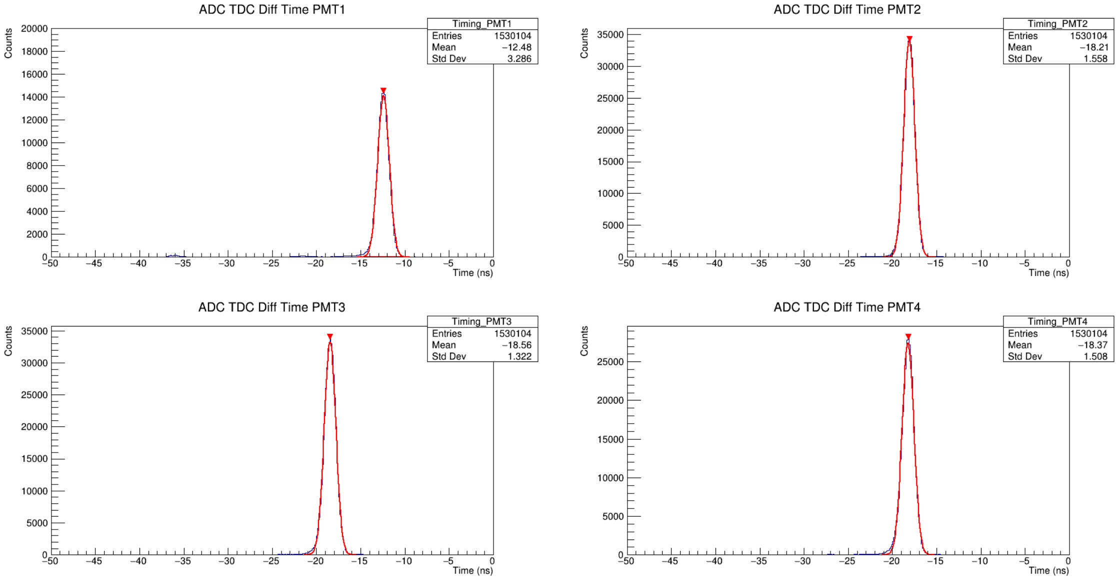


Figure 5.2: Timing information for Cherenkov calibration. Plotted on the x axis is the timing information in ns from the leaf P.hgcer.goodAdcTdcDiffTime. The y axis shows the counts per channel. Each plot corresponds to a different PMT according to each title.

peaks can be observed. Also visible is an unsynchronized timing peak from PMT 1 by approximately 5 ns. The cause of this may be due to two factors: the cable length for PMT 1 may be shorter than the others, resulting in a slightly earlier signal, or the reference time for PMT 1 may be out of phase with the other three. However, this is not concerning, since timing peak cuts are applied on a PMT-by-PMT basis. For future runs, a simple 5 ns delay will be introduced to the PMT 1 signal.

The last common cut criterion is on the reconstructed β, the velocity of the particle. This spectrum forms a very sharp peak, as all the particles are ultra-relativistic, and so the selection is easily made. Similar to the others, Fig. 5.3 displays the β information and cut used. Note that a β greater than one is due to uncertainty in reconstruction, and a β of zero corresponds to incorrect reconstruction.

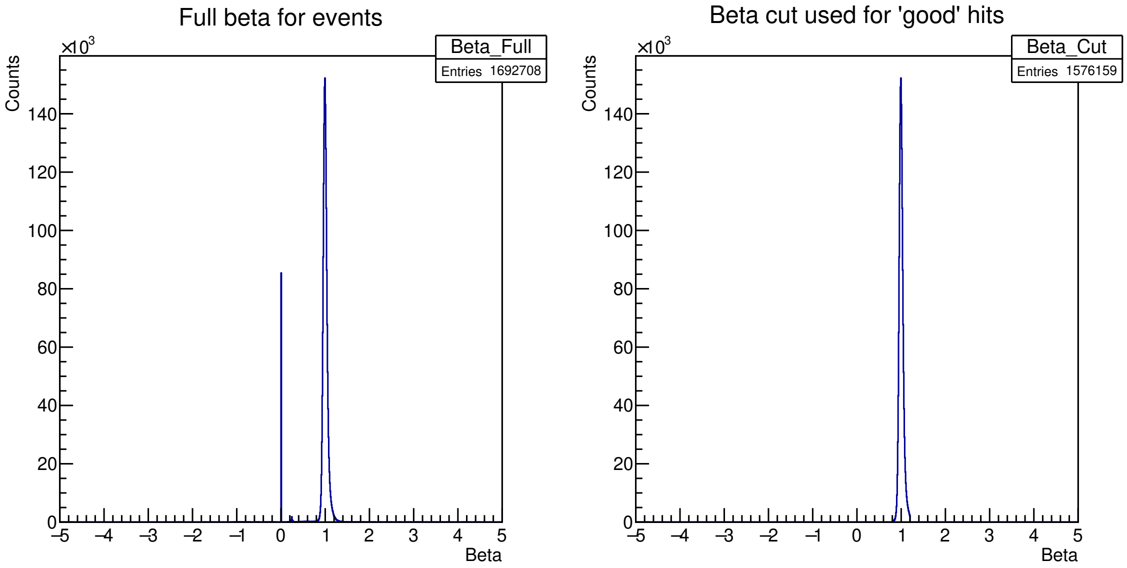


Figure 5.3: Reconstructed β information for Cherenkov calibration. The x axis shows the value for β while the y axis shows counts per channel. Values above 1.0 are due to uncertainty in the reconstruction and values at 0.0 are due to failed reconstructions. Left: Spectrum before any cuts; Right: Region of selected events.

5.3 Quadrant Strategy

As mentioned in Chapter 5.2, this calibration strategy manually imposes the track matching logic post-replay and then locates the SPE. The track matching occurs in the Process stage and follows the simple equation,

$$x_{\text{HGC}} = x_{\text{Focal Plane}} + x'_{\text{Focal Plane}} * z_{\text{HGC}}, \tag{5.2}$$

$$z_{\text{HGC}} = 156.27 \text{ cm}, \tag{5.3}$$

with an identical form for the y dimension. This procedure yields 16 histograms, four PMTs receiving signals from four mirror quadrants. Since only the SPE is desired, the histogram corresponding to a PMT looking at its own mirror is ignored, as too much light drowns the SPE (for example, the signal PMT 1 sees from mirror 1 is too strong to isolate cleanly the SPE). For a visual of how a particle track though a particular mirror quadrant produces signal in a different quadrant, consult Figure 2.12. Briefly,

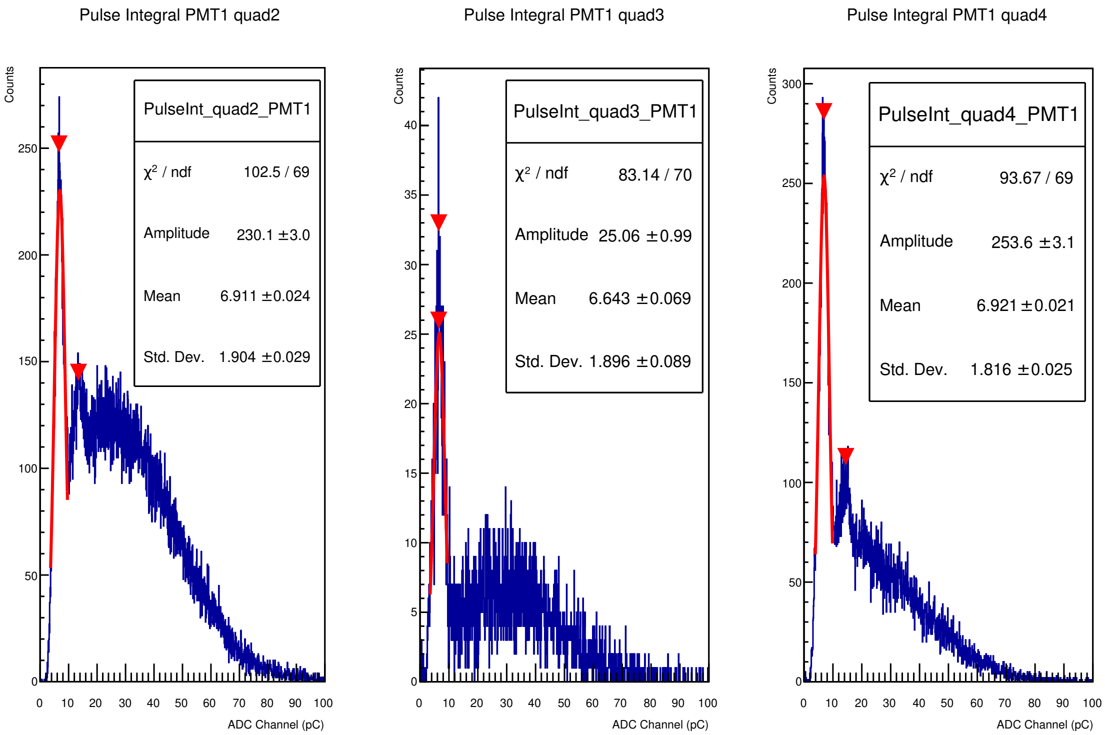


Figure 5.4: Illustration of locating the SPE in the HGC for PMT 1 FADC. Each plot shows the PMT pulse integral ADC for incident electrons in quadrants 2 - 4. The SPE is fit with a Gaussian and the value of this mean is compared between quadrants for consistency. The red arrows are drawn by the peak finding algorithm, and typically return the first and second photoelectron peaks.

the Cherenkov radiation creates a cone of radius 4 cm to 7 cm, depending on the media (consult Chapter 3 for detailed discussion). This finite spread allows photons to pass to “off-quadrant” mirrors.

The three histograms are then fit with Gaussian distributions about the SPE, and the mean of the three fits is taken to be the calibration. An illustration of the process is in Figure 5.4. In the case of low-statistics runs, conditions exist to ensure an accurate calibration is determined. First, it is required that the Gaussian fit of the peak converged. Second, the fitted Gaussian is checked to ensure the peak bin has a minimum of 90 events, otherwise it is neglected (in Figure 5.4 the quadrant 3 fit is ignored). This value of 90 events was chosen, as below this value the shot noise inherent in the Poisson random distribution of events caused unreliable SPE peaks. In case all three quadrants fail this statistics cut, the four quadrants are summed together and the procedure is repeated. Due to the SHMS particle envelope and focusing of the mirrors (discussed further in Chapter 6, consult Figure 6.3) this happens most often with PMT 4. When all the quadrants are combined, the SPE is clear, resulting in an accurate calibration.

To locate the SPE for these histograms, the ROOT TSpectrum package is used, in particular the Search function (CERN, 2018). This function locates peaks based on a deconvolution method: first a background is removed, then the spectrum is smoothed using a fine Markov chain algorithm, and the response function is generated using an input sigma and the deconvolution (Mariscotti, 1967) (Morháč et al., 2000) (Silagadze, 1996). The resultant peaks and their locations are returned in an array. The Search method takes as input four parameters: input histogram, sigma, options, and threshold. The sigma is used in various contexts to establish resolution, for example the size of the response vector and the number of iterations for the deconvolution. The threshold restricts the size of returned peaks, entries less than $threshold \times highest\_peak$ are discarded (CERN, 2018). For the calibration procedure, a sigma of 2.0 and a threshold of 0.9 yielded good results. Lastly, the option "nobackground" is used, as background subtraction is unnecessary.

With the SPE isolated, the peak is fit with a Gaussian distribution, that is, a function of the form

$$y = Ae^{-(x-\mu)^2/2\sigma^2}, \tag{5.4}$$

where A is a constant, μ is the mean, and σ is the standard deviation. Each mirror quadrant passing the statistical requirements is fit with Equation 5.4, where the value of μ is used to determine the calibration as it has units of pC/PE. The calibration for each PMT is taken as the average μ from each contributing mirror quadrant.

With the calibration estimate obtained, new histograms are created to store the calibrated spectra of all four quadrants combined per PMT. This is achieved by

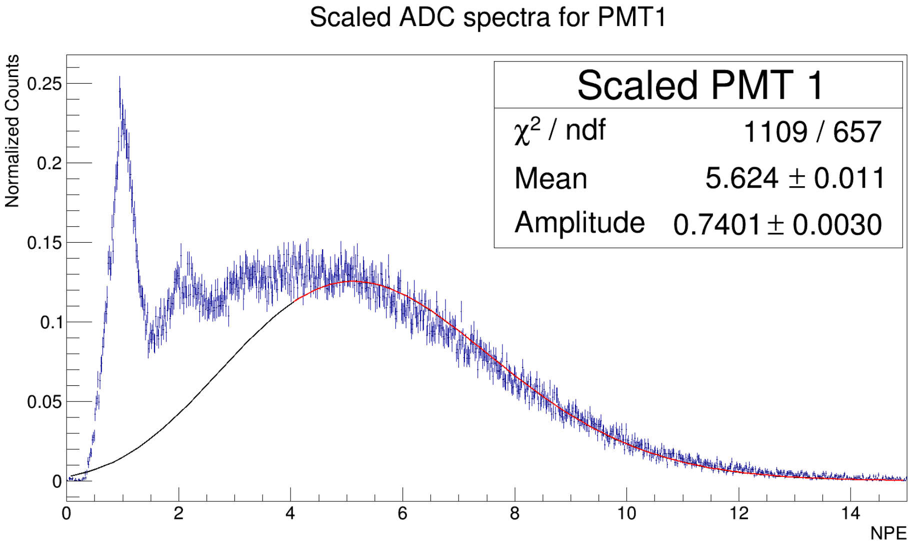


Figure 5.5: Result of calibrating a single PMT based on a Gaussian fit of the SPE. Note the peak chosen as the SPE can be seen to correspond to 1 PE. To facilitate the fitting of a Poisson distribution, given in Equation 2.8, the spectrum is normalized by its integral. The red curve is a Poisson distribution fit to the right side of the distribution to remove larger PE so the spacing can be determined. The black curve shows the full Poisson distribution.

filling this new histogram bin-by-bin with the previous histogram data scaled by the obtained calibration value. This is illustrated by the equation

$$b'_i = \frac{b_i}{\bar{\mu}}, \tag{5.5}$$

where b'_i and b_i are the i^{th} bins of the calibrated and uncalibrated spectra respectively, and $\bar{\mu}$ is the obtained calibration, in units of pC/PE. In this way, each histogram can be given the same range and number of bins, facilitating combining the calibrated data from all four PMTs later. Figure 5.5 displays the resultant histogram. To verify the accuracy of the calibration, the linearity of the spacing between PE is investigated. In order to better expose the second and third PE, a Poisson distribution is fit to the right hand side of the calibrated spectrum (the red fit in Figure 5.5) and subtracted off. This method will leave some of the larger bins with a negative number of entries. However, since the area of interest is only the first few photoelectrons, these bins will

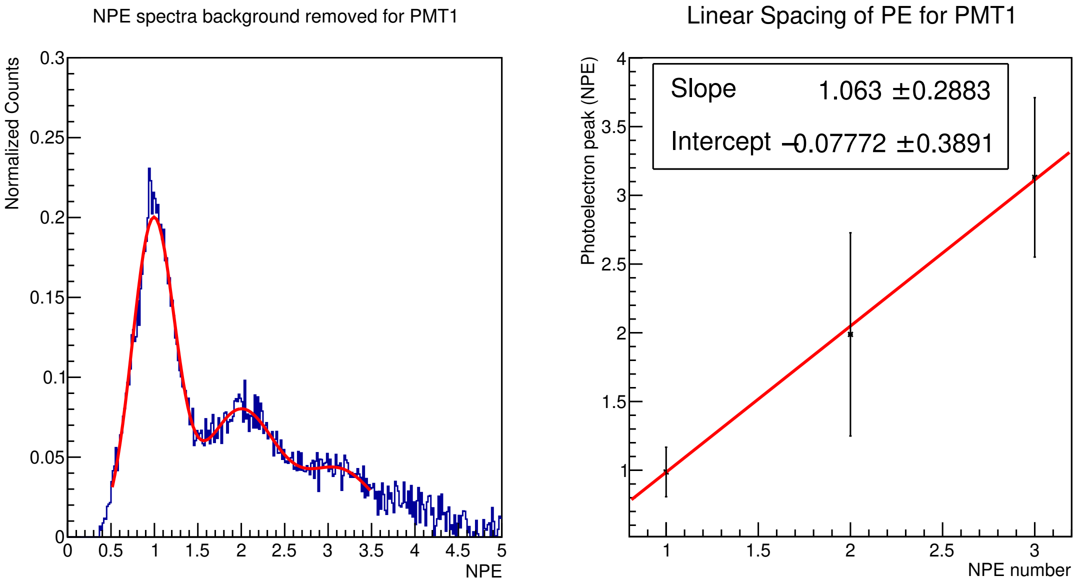


Figure 5.6: Result of subtracting off Poisson-like distribution at larger PE and the spacing of first three PE. The increasing uncertainties are due to lower statistics and non-Gaussian shape from contribution by higher order PE.

be ignored. The subtraction is performed, since at larger NPE the spectrum should be well described by a Poisson distribution, so by subtracting off this contribution one is left with only the lower NPE. Once the first three PE peaks are resolved, the spectrum is fit with a sum of three Gaussians to determine their positions and therefore their spacing. An example of this procedure is shown in Figure 5.6.

Examining the spacing of the first few peaks (right-hand plots of Figure 5.6 and Figure 5.7), there is a clear increase in the uncertainty of the location of the second and third peaks. This is due to two factors: lower statistics for those peaks, and greater contribution from the higher order NPE. Although the subtraction of a Poisson distribution, illustrated in Figure 5.5, helps to reveal the second and third peaks, it does not fully account for the tails of the various PE peaks contributing to the distribution. For example, refer to Figure 2.22 where the overlap of the various PE peaks can be clearly observed. This overlap causes the shape to deviate from the Gaussian shape, introducing greater uncertainty in the fit.

From here, a second estimate of the calibration is obtained by multiplying the calibration constant by the slope of the linear fit. The offset is ignored, since it is essentially zero, and would imply a correction to the pedestal subtraction which is unlikely. By doing this, the peak centered about one is expected to vary only slightly (since the slope is very close to one) while the spacing between PE may improve. The new calibration constant is obtained from the equation

$$\bar{\mu}' = m\bar{\mu}, \tag{5.6}$$

where $\bar{\mu}'$ is the new calibration, $\bar{\mu}$ is the previous calibration, and m is the slope of the spacing of the NPE peaks, a dimensionless value. The logic of Equation 5.6 is that a slope greater than one between PE peaks corresponds to a calibration constant that is too small, and vice-versa. Thus, multiplying the calibration by this slope can act as a self-correcting measure, yielding a more accurate calibration.

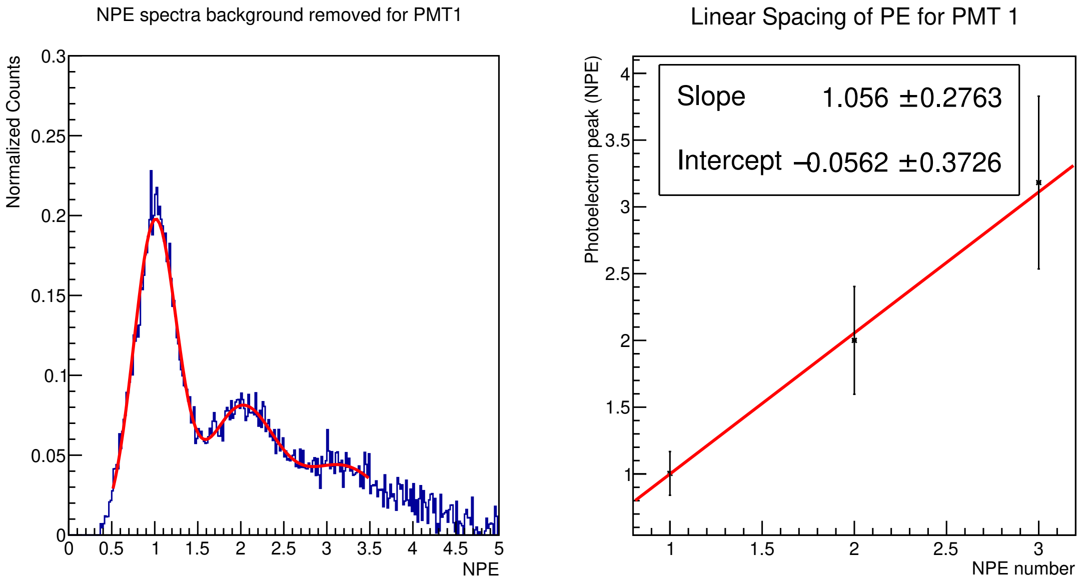


Figure 5.7: Second estimate of calibration constant. The updated calibration was obtained by multiplying the previous calibration constant by the slope of the spacing of the PE. The increasing uncertainties are due to lower statistics and non-Gaussian shape from contribution by higher order PE.

With this new calibration constant, the same procedure is repeated and gives similar results, as can be seen in Figure 5.7. For reference, the two calibration constants calculated in this example are 6.92 ± 0.02 pC/PE from Figure 5.4 and 6.66 ± 0.02 pC/PE from Equation 5.6. Between the two estimates, the actual calibration is taken as whichever better predicts the SPE, i.e. for the sum of three Gaussians fit, whichever calibration results in a χ^2/ndf closer to 1.0 for the first Gaussian.

Another metric of goodness-of-fit is how well the calibration characterizes the expected Poisson statistics of a properly calibrated detector. Here, each PMT is analyzed separately to ensure the calibration constants are correct. Cuts must be applied to expose the Poisson distribution characteristics as well: particle identification from the calorimeter was used to select electrons, and tracking from the drift chambers was used to remove all off-quadrant events. The results of fitting each PMT with these cuts are shown in Figure 5.8, where there is a strong agreement with the fit. The fits are not perfect due to the uncertainty in the calibration, alluded to earlier and shown

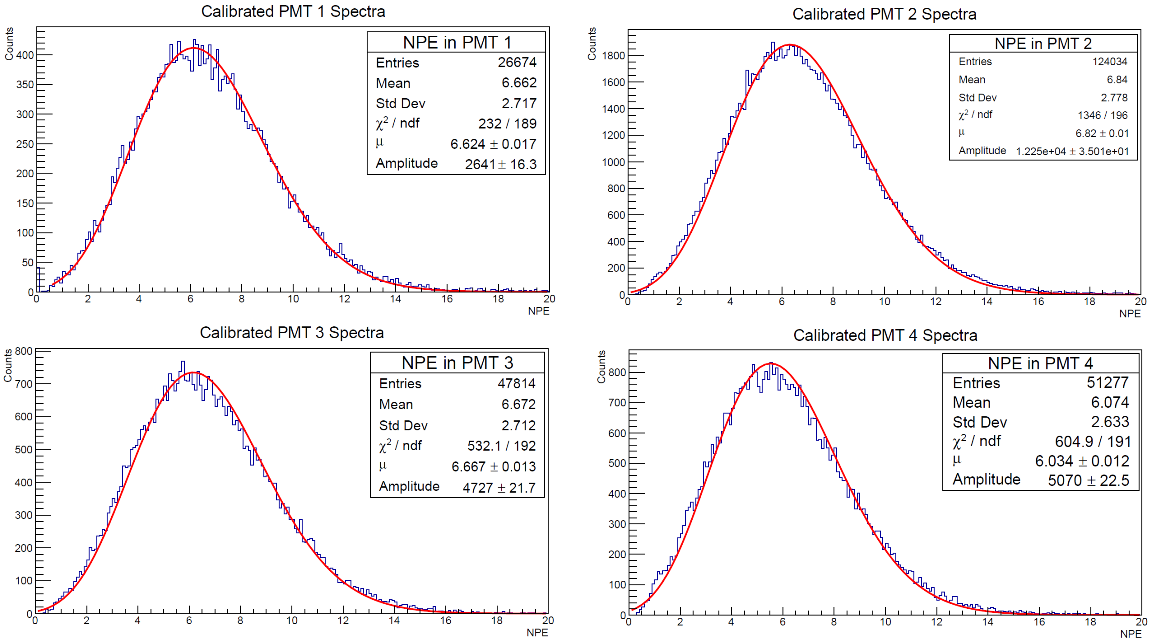


Figure 5.8: Poisson statistics of each PMT in the HGC. Particle ID was applied to select for electrons, and tracking cuts to eliminate off-quadrant events.

in Table 5.1. However, the calibration is sufficient for the experimental requirements.

Examining the results of Figure 5.8, one can see that a relatively low number of average photoelectrons ($\mu \sim 6.5$) are detected. Ideally, a much stronger signal would be detected as this facilitates the separation between particle species. To this end, run 3623 is analyzed, where the HGC was filled with C_4F_8O, as this will produce more light (for detailed discussion, consult Chapter 3). The details for this run can be found in logbook entry 3561974 (hclog, 2018e), but of importance now is that the SHMS was set to a central momentum of -2.53 GeV/c with the HGC filled to 1 atm of gas. Consulting Figure 3.4, one can see that both pions and electrons will produce Cherenkov radiation in this setting. Thus, particle separation is required to characterize the Poisson statistics. Unfortunately, the calorimeter calibration encountered issues in this running period and so is not sufficient to cleanly separate pions and electrons. Instead, the NGC, in combination with a loose calorimeter cut, is used to separate these two particles. This can be done, since the NGC was filled with 1 atm CO_2 and thus the pions are further from the threshold speed. The resultant HGC spectrum was fit with two Poisson distributions, as shown in Figure 5.9. The accuracy of this fit suggests that the light from the on-quadrant, and the light from the off-quadrants, may each contribute separate Poisson distributions of events. The possibility of this will be investigated in the future by altering the alignment of the mirror quadrants.

Additionally, run 3623 was recalibrated to ensure the stability of the calibration across runs. The results from the calibration strategy are summarized in Table 5.1 where it is clear the calibration methodology is reasonably consistent.

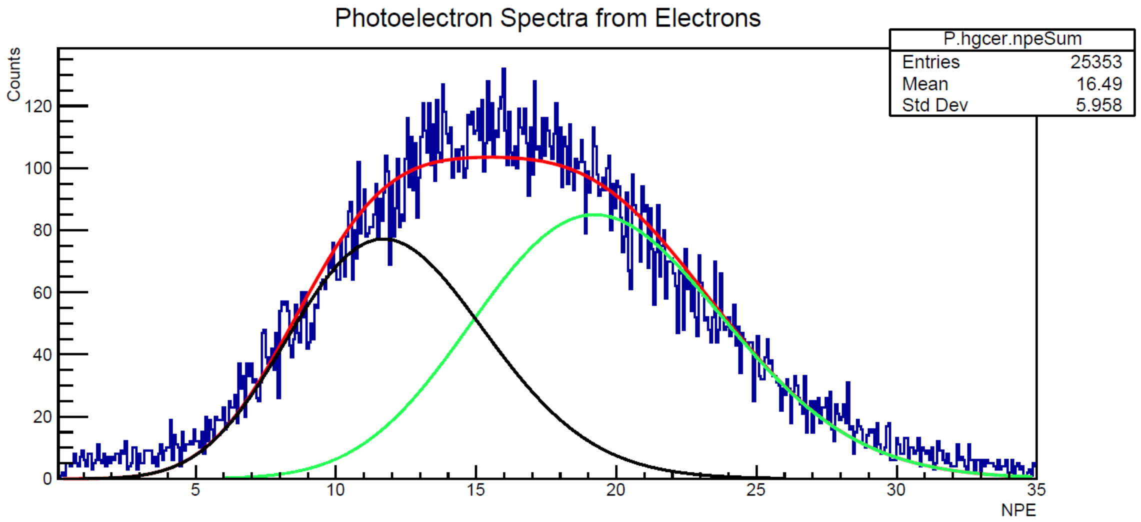


Figure 5.9: Poisson characteristics of the HGC. Particle ID was applied to select for electrons using the NGC and calorimeter. Green and black line are Poisson fits, suggesting two light sources.

	Run 1583		Run 3623	
PMT	First Estimate	Second Estimate	First Estimate	Second Estimate
1	6.92 pC/PE*	6.66 pC/PE	6.67 pC/PE*	6.46 pC/PE
2	7.30 pC/PE	7.06 pC/PE*	7.53 pC/PE	7.19 pC/PE*
3	6.80 pC/PE*	6.62 pC/PE	6.20 pC/PE*	6.02 pC/PE
4	7.25 pC/PE	6.69 pC/PE*	7.24 pC/PE	7.07 pC/PE*

Table 5.1: Results of the quadrant calibration strategy. The more accurate calibrations are marked with asterisks and are determined by the precision of the Gaussian fit of the SPE. Each calibration has a statistical uncertainty of ±0.02.

5.4 TracksFired Strategy

For the tracksfired strategy, the Process stage only requires that the quantity NumTracksFired is incremented for that event. As discussed in Chapter 4.3, this requires implicit tracking, NPE, and kinematic cuts. From here, the procedure is similar to that of the previous strategy. Again, the signal received by a PMT focusing on its own quadrant is ignored, as it is more difficult to resolve the SPE. Also identically, a peak finding algorithm locates the SPE, which is then fit with a Gaussian distribution, as can be seen in Figure 5.10. As an aside, this figure clearly demonstrates the

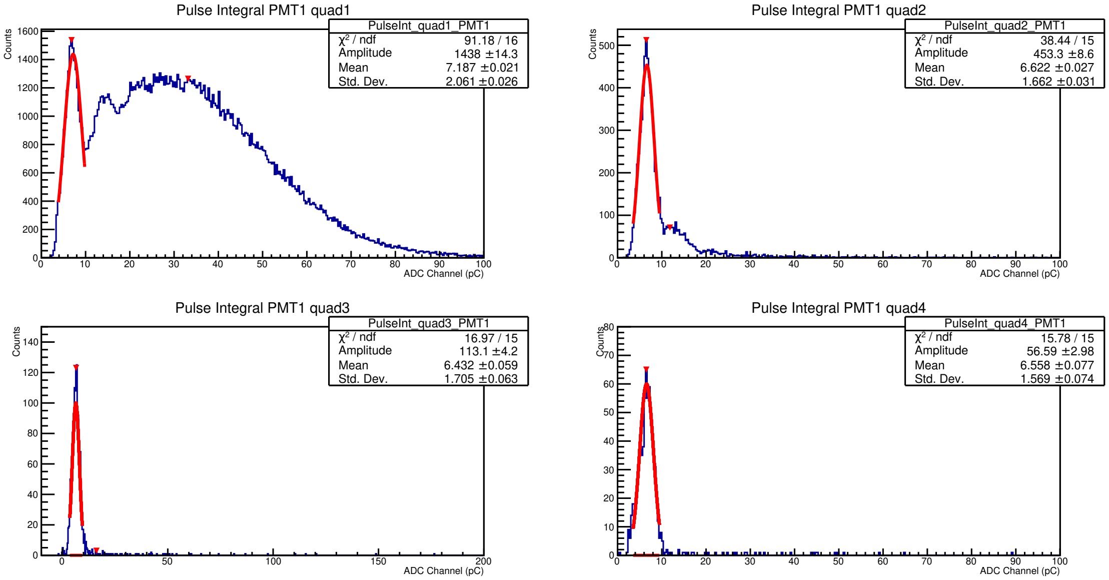


Figure 5.10: Illustration of determining the SPE in the HGC using tracksfired strategy. Each plot shows the signal received in each quadrant for PMT 1. While the SPE is clear in each quadrant, the location of the SPE in the mirror quadrant focused on the PMT is skewed to a higher value in pC due to the contribution of higher order NPE.

advantage in analyzing off quadrant signal to determine the calibration. The SPE from quadrant one is skewed due to contributions from higher order photoelectrons,

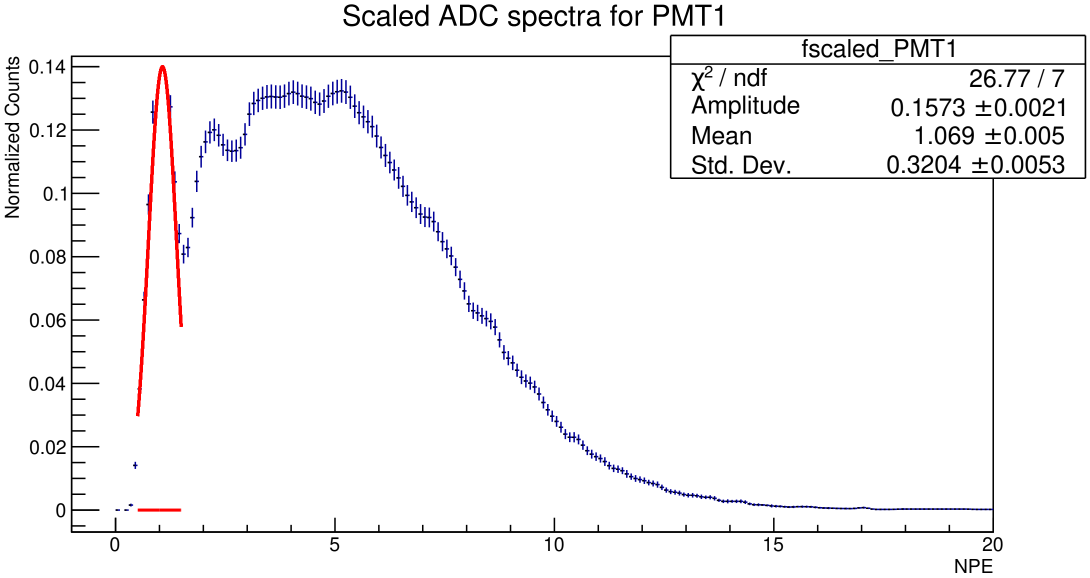


Figure 5.11: Result of calibration using the Tracksfired strategy. Data was normalized to facilitate fitting with the Gaussian in red.

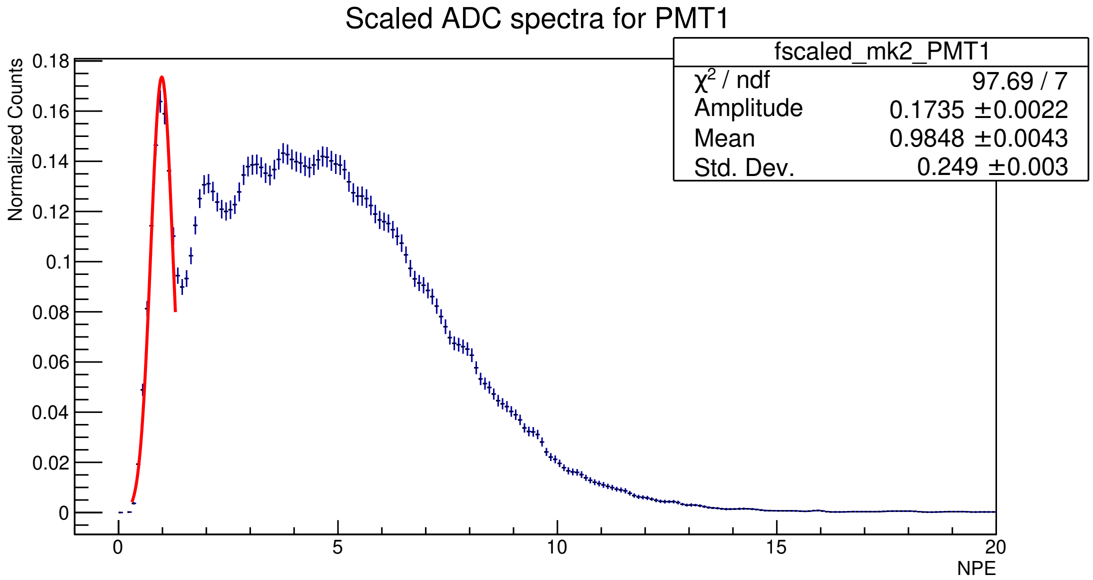


Figure 5.12: Second estimate of TracksFired calibration. The updated calibration was obtained by multiplying the previous calibration constant by position of the SPE.

resulting in an inaccurate peak. The other quadrants lack this effect, and give more consistent values. As before, this peak is taken to be the first estimate of a calibration and the ADC channels are scaled appropriately following Equation 5.5. To determine the second estimate, the previous method cannot be used, as it is too unreliable to isolate the second and third PE. Instead, the calibration constant is multiplied by the mean of a Gaussian fit of SPE, since this value should be close to 1.0. The reasoning is identical to that of Equation 5.6, an SPE peak larger than 1.0 indicates the calibration constant is too small and vice-versa. These methods are illustrated in Figures 5.11 and 5.12. The results from the calibration for both run 1583, and run 3623 are summarized in Table 5.2.

While the results of this calibration are reasonably consistent both with itself and with the other strategy, the additional cuts applied on track reconstruction and various kinematic factors limit the statistics available from a single run. As the detector checkouts are typically only a single run long, this particular strategy is inferior to manually imposing the tracking in the quadrant strategy.

	Run 1583		Run 3623	
PMT	First Estimate	Second Estimate	First Estimate	Second Estimate
1	6.53 pC/PE*	6.98 pC/PE	6.42 pC/PE*	6.79 pC/PE
2	7.22 pC/PE*	8.27 pC/PE	6.93 pC/PE*	7.35 pC/PE
3	6.66 pC/PE*	7.19 pC/PE	5.95 pC/PE	6.17 pC/PE*
4	6.67 pC/PE*	8.29 pC/PE	7.05 pC/PE*	7.33 pC/PE

Table 5.2: Results of the TracksFired calibration strategy. The more accurate calibration is marked with an asterisk and is determined by the precision of the Gaussian fit of the SPE. Each calibration has a statistical uncertainty of ±0.02.

5.5 Gain Matching

As mentioned in Chapter 2.1, ensuring each PMT has an identical response to incident light is an important component to the commissioning of the HGC. Investigations of the gain have been performed for the PMTs in the HGC (Fischer, 2012), where the results are summarized in Table 5.3. The gain was calculated by determining the location of the first two PE peaks, and the corresponding charge for each point. This will yield a linear relationship, the slope of which is the gain by definition. By repeating this procedure at several different voltages, one obtains a gain dependence on applied voltage which can be fit with Equation 2.6. The results displayed in Table 5.3 used this methodology in a highly controlled environment, over a wide range of voltages, to acquire a very precise relationship between gain and voltage. This is illustrated in the experimental data in Figure 5.13 (Fischer, 2012). Despite limitations in voltage range and resolution, this same procedure was performed in an

PMT	Exponent, αn	Coefficient, K	χ^2/ndf
PMT 1	10.54±0.12	$(3.80 \pm 3.47) \times 10^{-28}$	0.101
PMT 2	10.50±0.12	$(1.31 \pm 1.18) \times 10^{-27}$	0.410
PMT 3	10.63±0.13	$(6.30 \pm 6.27) \times 10^{-28}$	0.610
PMT 4	10.48±0.11	$(1.21 \pm 1.04) \times 10^{-27}$	0.348

Table 5.3: Gain characteristics for the PMTs in the HGC (Fischer, 2012). Values are given in reference to Equation 2.6

experimental setting as shown in Figure 5.14.

The results of this investigation, and a reiteration of the previous work, are displayed in Table 5.4. Experimental data are severely restricted in making any inferences, however it does reassure that the original exponents are accurate as the values are somewhat close in magnitude. The gain was measured at 2044 V, and a factor of two had to be applied to the gain to account for the passive 50:50 splitter the HGC signal passes through.

With the value of the exponent αn, it is straightforward to gain match the PMTs using Equation 2.6

$$V_{\text{new}} = V_{\text{old}} \left(\frac{\text{SPE Charge}_{\text{new}}}{\text{SPE Charge}_{\text{old}}} \right)^{\frac{1}{\alpha n}} \tag{5.7}$$

The new gains are chosen such that the SPE peak is large enough to clearly separate from the pedestal at zero, but not so large as to saturate the signal for the larger PE.

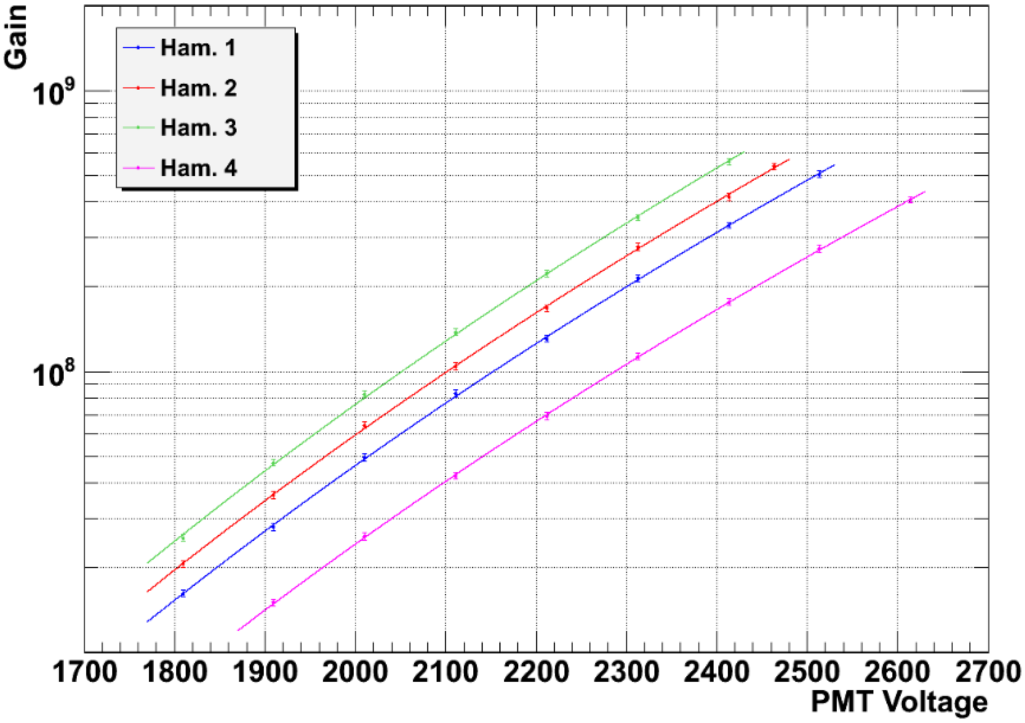


Figure 5.13: Original measurement of gain vs voltage for the HGC PMTs (Fischer, 2012). Relating the PMTs to their positions in the HGC: Ham. 1 is PMT 4, Ham. 2 is PMT 2, Ham. 3 is PMT 3, and Ham. 4 is PMT 1.

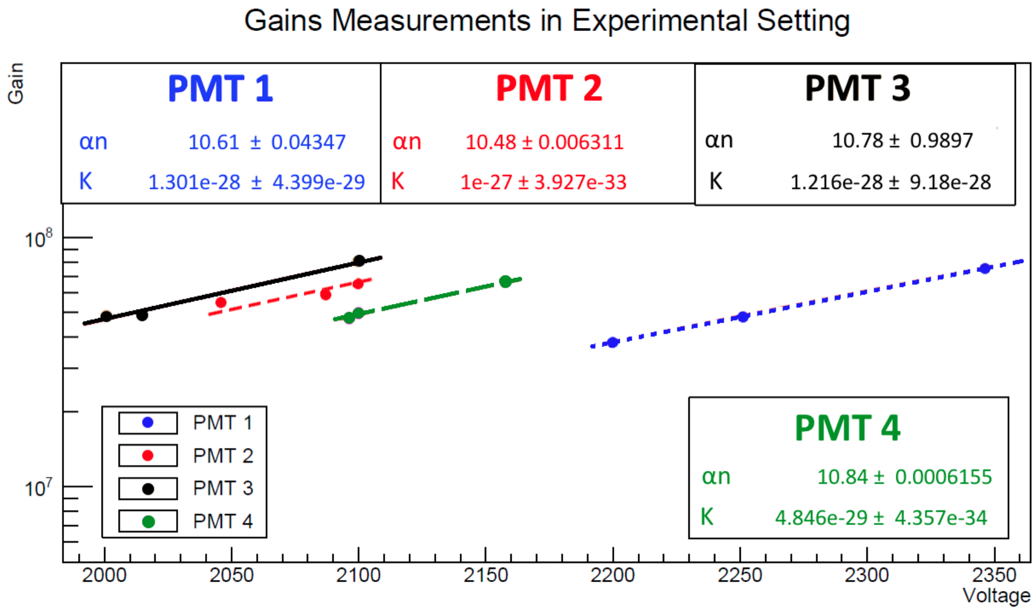


Figure 5.14: Experimental measurement of gain vs voltage for the HGC PMTs. Dotted line is PMT 1, dashed line is PMT 2, solid line is PMT 3, and long dashed line is PMT 4.

The appropriate value chosen is 6.825 pC, which yielded the new PMT voltages in Table 5.5. Also shown are the previous voltages used for the Key Performance Parameters (KPP) commissioning runs in the winter of 2017. The actual gain matching is shown in Figure 5.15, where one can see the alignment of the SPE peak, shown by the band near the bottom, improves dramatically from run 488 to run 1583.

Lastly, the recommendation (Degtiarenko, 2017c) to correct the calibration due to SPE events leaking into the pedestal, shifting the peak, will be commented on. Comparing the experimental raw FADC channel histogram of Figure 5.4 to those relevant

PMT	Experimental Exponent / Gain	Original Exponent & Gain
PMT 1	10.61±0.04 / 3.10 × 10^7	10.54±0.12 / 3.06 × 10^7
PMT 2	10.48±0.01 / 8.44 × 10^7	10.50±0.12 / 7.47 × 10^7
PMT 3	10.78±0.99 / 10.72 × 10^7	10.63±0.13 / 9.62 × 10^7
PMT 4	10.84±0.01 / 6.82 × 10^7	10.48±0.11 / 5.81 × 10^7

Table 5.4: Gain characteristics for the PMTs in the HGC between an experimental setting and controlled environment (Fischer, 2012). For gain measurements, voltage is set to 2044 V.

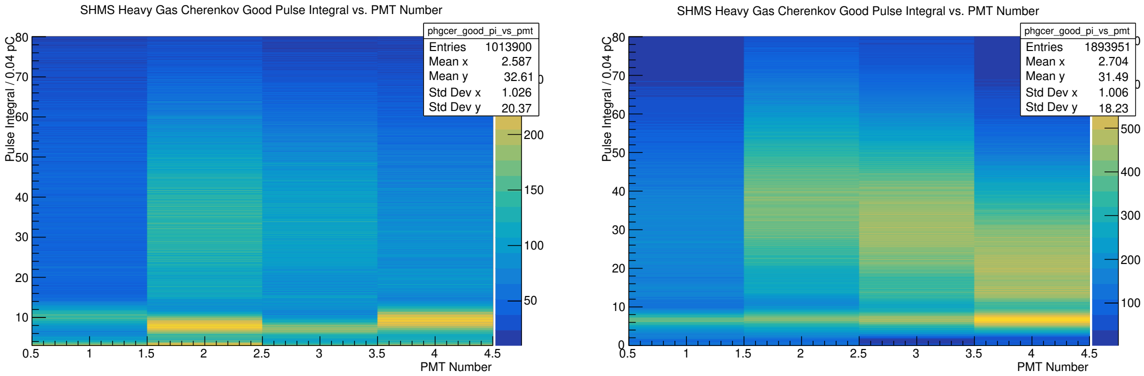


Figure 5.15: Comparison of gain matching between run 488 (left) and 1583 (right). Difference in voltages are shown on Table 5.5. Note that 488 Pulse Integral axis starts at 5 instead of 0.

to the low signal analysis in Figure 2.22, there is a clear difference in the possibility for the pedestal to have an effect on the SPE peak. Additionally, Figure 2.22 has a large contribution to the SPE from higher order PE peaks where this effect is specifically avoided by the quadrant selection.

PMT	KPP Run 488	Run 1583
PMT 1	2347 V	2251 V
PMT 2	2087 V	2051 V
PMT 3	2015 V	2001 V
PMT 4	2158 V	2086 V

Table 5.5: Comparison of voltages before gain matching in run 488, and after gain matching in run 1583.

5.6 Systematic Study

In the summer/fall of 2018, a study of the systematic uncertainty in the HGC calibration was performed. The motivation was to better understand the uncertainty, since the ±0.02 statistical uncertainty did not fully account for the variance in the calibration constant. This is best shown by the different locations of the SPE peak between mirror quadrants for a particular PMT as seen in Figure 5.4. In theory, since

	PMT 1	PMT 2	PMT 3	PMT 4
	(pC)	(pC)	(pC)	(pC)
Mean	6.88	6.93	6.62	6.64
RMS	0.15	0.14	0.16	0.19
% Uncertainty	2.18 %	1.99 %	2.39 %	2.81 %
Total % Systematic Uncertainty				4.73 %

Table 5.6: Summary of meta-analysis of SPE location in pC for each PMT across six runs, 2511 - 2516, all of which have an SHMS momentum setting of -4.0 GeV/c. The statistical uncertainty for the SPE peaks is ± 0.02 pC or $\approx 0.2\%$.

the signals are all for the same PMT, the location of these peaks should be identical. In practice, they differ due to higher order PE shifting the SPE peak to a higher value in pC, a trend that is quadrant specific. Since the average SPE peak between each quadrant is taken as the calibration, this variance yields a systematic uncertainty. To measure the systematic uncertainty of the calibration method, six runs at identical momentum (-4.0 GeV/c) were calibrated using the procedure established in Chapter 5.3. The runs used were 2511 - 2516, data were collected in February of 2018 and provided a large sample of runs at identical momentum. The results of the meta-analysis are shown in Table 5.6. While this uncertainty of 4.73% is not unreasonable, it is straightforward to reduce it by altering the calibration strategy. As the main contribution to the systematic uncertainty is the higher order PE, accounting

	PMT 1	PMT 2	PMT 3	PMT 4
	(pC)	(pC)	(pC)	(pC)
Mean	6.63	6.83	6.47	6.62
RMS	0.05	0.05	0.09	0.06
% Uncertainty	0.75 %	0.73 %	1.39 %	0.91 %
Total % Systematic Uncertainty				1.82 %

Table 5.7: Summary of meta-analysis of SPE location in pC for each PMT across six runs, 2511 - 2516, all of which have an SHMS momentum setting of -4.0 GeV/c. The statistical uncertainty for the SPE peaks is ± 0.02 pC or $\approx 0.2\%$. Presented are results for improved calibration method, where both the first and second PE peaks are fit.

for that background should reduce the systematic uncertainty. This is achieved by not only fitting the SPE in Figure 5.4, but the second PE peak as well, with a sum of two Gaussian distributions. By doing this, the mean of the SPE peak will properly compensate for the non-linear background of the second PE, where even higher order PE have already been removed by the calibration strategy. The results of this improved method are shown in Table 5.7, where the identical runs were calibrated and the improvement is evident by comparison to Table 5.6.

Chapter 6

Efficiency

Another important component to the heavy gas Cherenkov commissioning is determining the efficiency of the detector. The analysis code is available on GitHub in the same location as previously mentioned, or in Appendix A as Listing A.6. Recall from Chapter 3 that the objective of this detector is to separate between particle species based on the amount of light they produce. This is achieved by applying a threshold cut to the number of photoelectrons the HGC detects, as this should eliminate sub-threshold particles and δ-ray events. Therefore, the efficiency is a strong metric of how well this objective has been achieved. Before delving into how this is quantified and measured, it is important to establish what particle is being selected. First, similar to the calibration procedure, run 1583 (-3.0 GeV/c, CO_2 at 1 atm, 0.5% carbon target) (hclog, 2017a) will be analyzed where the HGC will be used to identify electrons by separating them from (predominantly) pions. This leaves two measures of interest, electron efficiency and pion contamination.

In this context, efficiency is how many electrons are correctly selected for a cut on NPE, while pion contamination is the number of pions that also pass this electron cut. In other words, pion contamination may be interpreted as the ratio of false positives to be expected. Therefore, the electron efficiency is ideally 100% and the pion contamination zero. The efficiencies are calculated by taking the ratio of electrons selected with HGC cuts, divided by the electrons selected without any HGC information

$$\text{efficiency} = \frac{\text{[Particle ID][Tracks][Beta][Timing][HGC]}}{\text{[Particle ID][Tracks][Beta][Timing]}} \times 100\%, \tag{6.1}$$

where quantities in brackets represent a cut. The efficiency is strictly less than one and therefore can be expressed as a percentage. The pion contamination is a ratio as well:

$$\pi^- \text{ contamination} = \frac{[e^- \text{ efficiency}]}{[\pi^- \text{ inefficiency}]} = \frac{\#\text{ of particles identified as } e^-}{\#\text{ of } \pi^- \text{ identified as } e^- \text{ by the HGC}} \tag{6.2}$$

where this quantity goes to infinity if the pion inefficiency is 0 (if a pion is correctly identified every time, it will take an infinite number of electrons until a pion slips through). Note that the various particle ID cuts will not produce a completely pure sample of particles, but sufficiently so to determine the HGC efficiency. However, this will introduce a systematic uncertainty in the efficiency.

Similarly to the calibration script, the efficiency script adheres to the TSelector framework. In the Process stage, the timing, beta, tracking, and particle ID cuts are applied, identical to those in the calibration. In the Terminate stage, the cut criteria are presented along with the results of the cuts on electrons and pions per PMT as well as for the whole detector. These results for run 1583 are shown in Figures 6.1, 6.2. With these figures, the definitions of electron efficiency and pion contamination are made clearer. For reference, the efficiency and contamination are listed in Table 6.1.

With Table 6.1 one can see how threshold cuts are chosen in the HGC. The choice is made as a compromise between the efficiency of the detector, and the contamination. Having a larger threshold cut (i.e. a cut at a larger NPE value) ensures very little

NPE Cut	2.0		1.5		1.0		0.5	
	e^- eff.	$\pi^- : e^-$	e^- eff.	$\pi^- : e^-$	e^- eff.	$\pi^- : e^-$	e^- eff.	$\pi^- : e^-$
PMT 1	84.37%	1:30497	89.14%	1:16110	95.38%	1:10343	99.90%	1:9028
PMT 2	95.63%	1:29772	97.64%	1:30400	99.13%	1:30864	99.98%	1:31128
PMT 3	89.42%	1:37001	93.09%	1:25680	97.11%	1:16074	99.89%	1:15501
PMT 4	89.64%	1:18795	94.27%	1:19765	97.98%	1:17975	99.99%	1:16307
Full	90.35%	1:28350	93.95%	1:23162	97.57%	1:17270	99.94%	1:16044

Table 6.1: Electron efficiency is calculated by taking the ratio of the right and left plots in Figure 6.1 for each PMT and the total detector. The pion to electron ratio is obtained by taking the ratio of the right hand plot in Figure 6.1 over the right hand plot in Figure 6.2 for each PMT and the whole detector. Results are for run 1583.

contamination, however, many real events are also discarded leading to a smaller efficiency. A lower threshold cut will give a better efficiency, but the contamination may be larger as well. Typically, a value of 1.5 NPE is chosen as the threshold to eliminate δ-rays from the sample while retaining a high efficiency.

An unexpected feature of the efficiency results are the asymmetric values between the PMTs. This is due in part to the distribution of particles, which is shown in Fig-

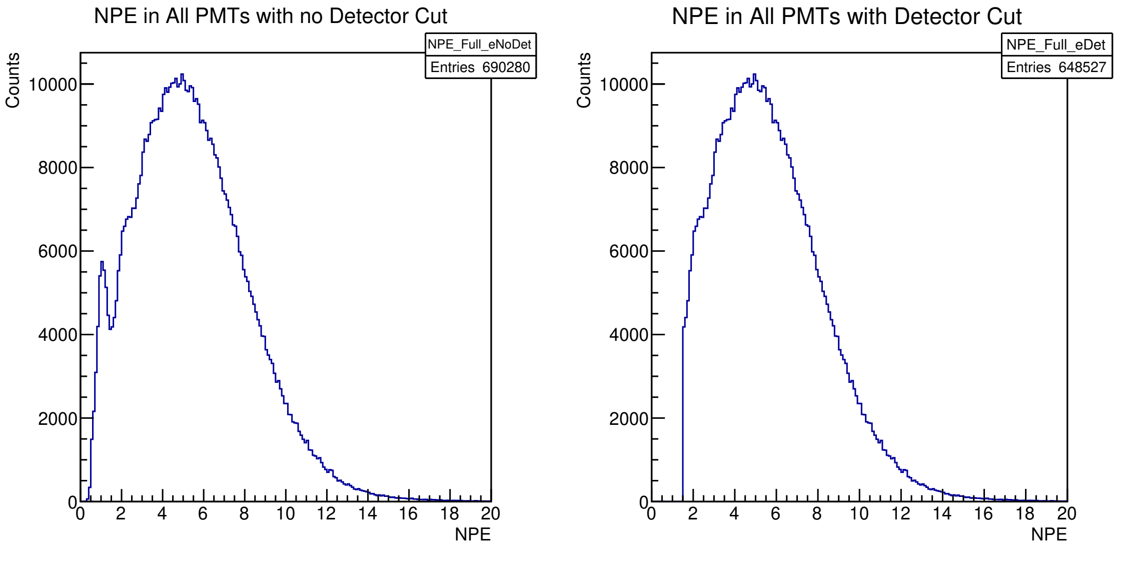


Figure 6.1: Spectra used to determine electron efficiency for entire HGC detector. The left plot is the data without a NPE cut and the right plot is the data with a NPE cut of 1.5. The efficiency is the ratio between the two spectra, 648527/690280 = 94%.

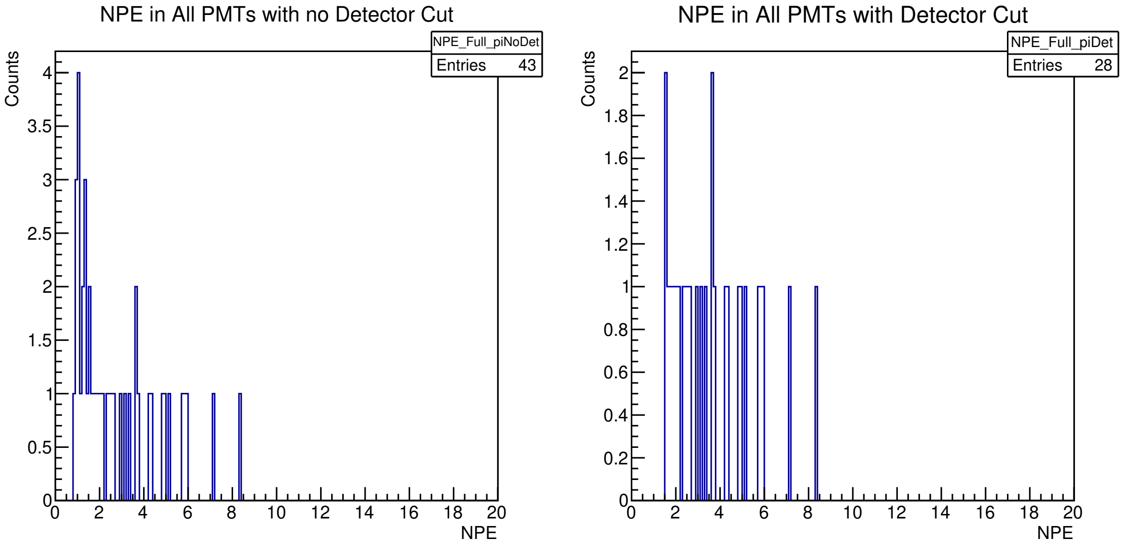


Figure 6.2: Spectra used to determine pion contamination for entire HGC detector. The left plot is the data without a NPE cut and the right plot is the data with a NPE cut of 1.5. The contamination is the ratio of particles identified as electrons over the number of pions identified as electrons, 1 pion per 648527/28 = 23161 electrons.

ure 6.3. Here, it is clear that mirror quadrant one receives very little light, yielding a lower efficiency. Recalling the discussion at the end of Chapter 2 on the mirror's shape, and the profiles shown in Figure 2.9, one would expect a poor efficiency for events around the mirror boundaries (recall as well the mirror interleaving will hide some mirror boundaries). Keeping this in mind, Figure 6.3 reveals that mirror quadrants three and four are nearly identical, with mirror quadrant two receiving a greater concentration of events, hence the asymmetry in the efficiencies.

Although run 1583 provides a good baseline to judge the HGC performance, it may not adequately characterize the eventual operating conditions as it is filled with CO_2 instead of a heavy gas. Therefore, the electron efficiency was determined a second time for the detector checkout run 3623 (hclog, 2018e), when the HGC was filled to 1 atm of C_4F_8O. The efficiency results are shown in Table 6.2, where it can be seen that the efficiency has improved compared to run 1583, when the HGC was filled with CO_2. Identical to before, there is an asymmetry between PMT performances

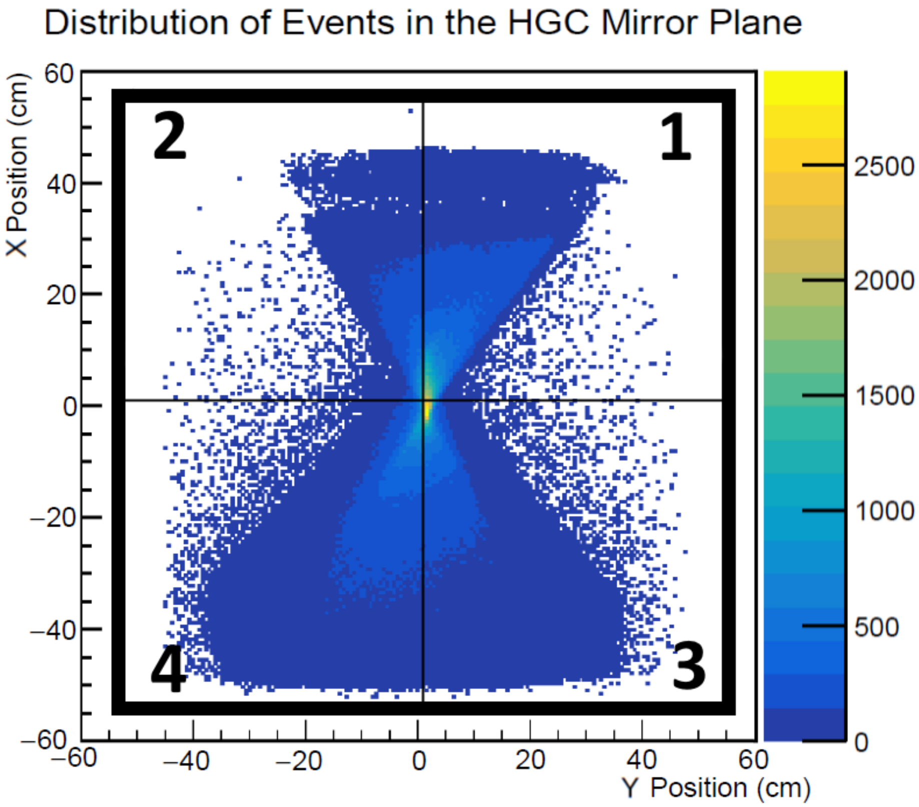


Figure 6.3: Distribution of events in the x-y axis of the HGC mirror plane. Lines indicate the separate mirror quadrants.

NPE Cut	0.5		2.0		8.0		10.0	
	e^- eff.	$\pi^- : e^-$	e^- eff.	$\pi^- : e^-$	e^- eff.	$\pi^- : e^-$	e^- eff.	$\pi^- : e^-$
PMT 1	99.93%	3:1	96.14%	2:1	73.05%	1:2	60.95%	1:5
PMT 2	99.99%	3:1	97.92%	2:1	92.59%	1:2	85.91%	1:4
PMT 3	99.29%	2:1	97.97%	1:1	88.62%	1:11	81.55%	1:31
PMT 4	99.75%	2:1	99.08%	1:1	87.00%	1:15	75.57%	1:38
Full	99.72%	3:1	97.82%	2:1	86.76%	1:3	77.75%	1:7

Table 6.2: Electron efficiency measurements for run 3623. Note that the ratio of $\pi^- : e^-$ is reversed at low NPE cut, due to the large pion background at this setting.

arising from the nature of the particle envelope. Despite the HGC being filled with a different gas, the voltage applied to the PMTs is unchanged and so the calibration should remain constant. The SPE location was checked, and the calibration remained unchanged.

A startling difference between the two runs is the stark difference in the pion contamination. This is due to two factors, the number of pion events and the amount of light they now produce. Comparing Figure 6.5 to Figure 6.2, it is easily seen that under these running kinematics, the number of pions events have increased dramatically. In fact, the majority of events change from electrons in run 1583 to pions in run 3623 resulting in any contamination drowning out the electron signal. Secondly, it can be seen that the average amount of light from pion events increases with a heavy gas media, as expected. This increases the overlap between pion and electron signals, requiring careful particle identification criteria from other detectors. Fortunately, efficiency is the more important metric as it represents the intrinsic performance of a device independent of running conditions. Particle contamination more-so represents the difficulty to cleanly separate particle species for a particular kinematic. For example, the pion contamination of run 3623 indicates that stricter calorimeter cuts should be applied, or perhaps a cut on δ needs to be included to remove the pions that are above Cherenkov threshold.

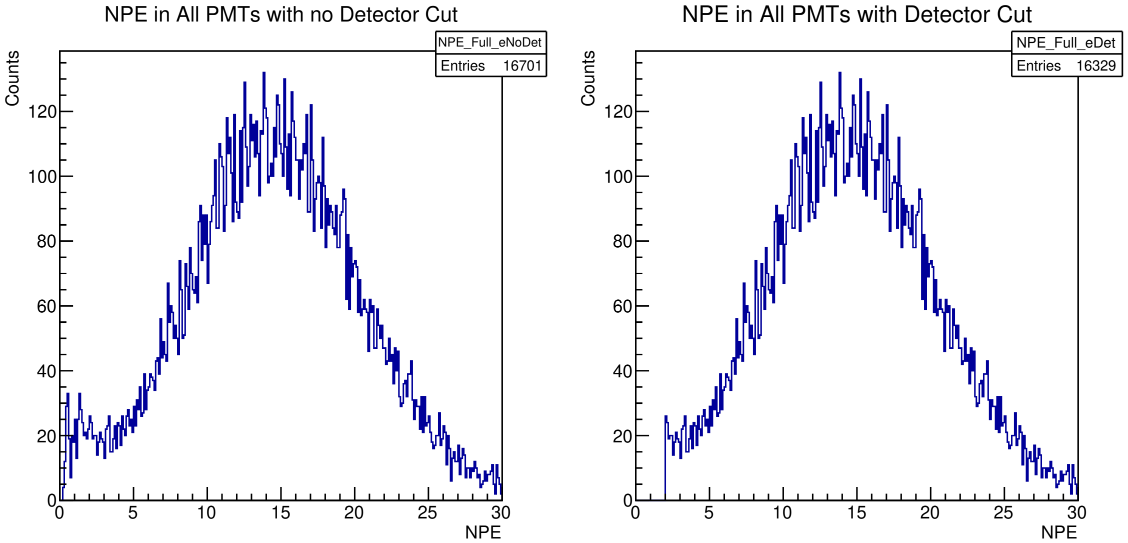


Figure 6.4: Spectra used to determine electron efficiency for entire Cherenkov detector. The left plot is the data without a NPE cut and the right plot is the data with a NPE cut of 2.0. The efficiency is the ratio between the two spectra.

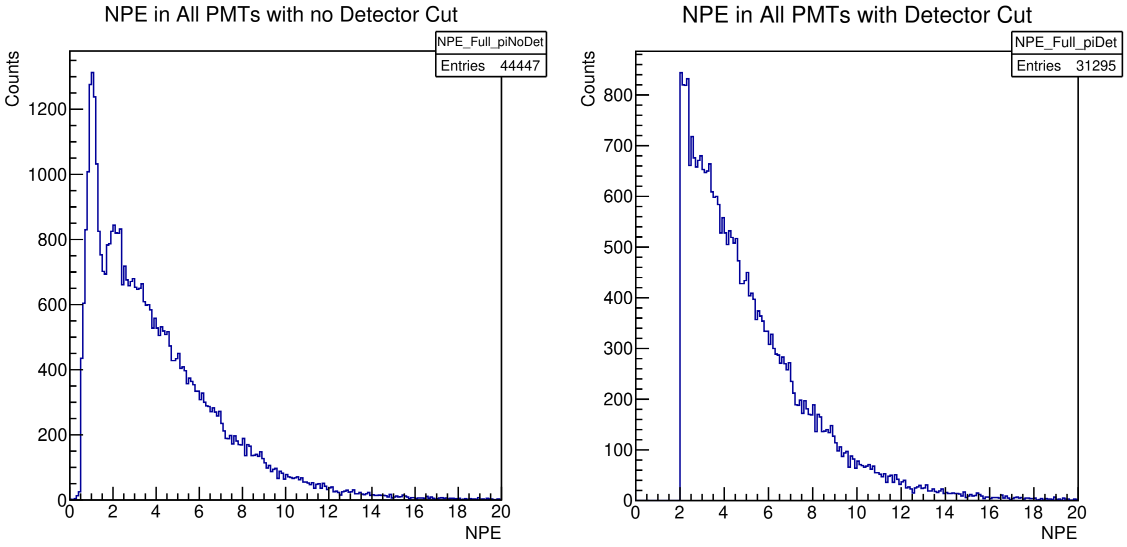


Figure 6.5: Spectra used to determine pion contamination for entire Cherenkov detector. The left plot is the data without a NPE cut and the right plot is the data with a NPE cut of 2.0. The contamination is the ratio of particles identified as electrons over the number of pions identified as electrons.

6.1 Delta & Position Scans

Another important characterization of efficiency is how it varies as a function of: delta, the fraction of deviation from the central momentum, and the position inside the detector. The efficiencies for these quantities are found in an identical manner as the electron efficiencies. Two histograms are filled, where one has a cut that all particle identification, including the HGC, identified that event as an electron, and the other is filled if the event is identified as an electron without any HGC information. For delta, the range is taken as $-30\% < \delta < 30\%$ with each bin spaced by 0.5%. The result is then trimmed to the nominal acceptance, $-10\% < \delta < 20\%$. For the position information, the two dimensional histogram is filled in the range of -50 cm $< x/y <$ 50 cm with bin spacing of 0.5 cm. The run considered is SHMS detector checkout 3623, where the run parameters are described in the previous section. The scan was performed for two HGC NPE cuts, one at 1.0 and another at 10.0.

First considering Figure 6.6, it is clear that the efficiency is relatively constant

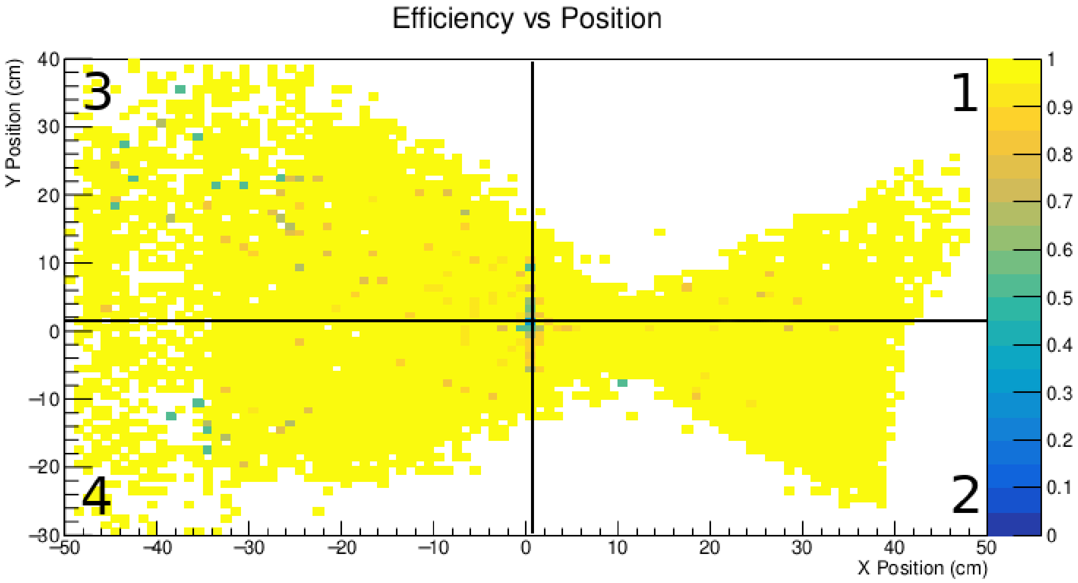


Figure 6.6: Position scan of efficiency for run 3623, with a HGC NPE cut of 1.0. Electrons are selected, where additional cuts are in text. Mirror quadrants are explicitly indicated.

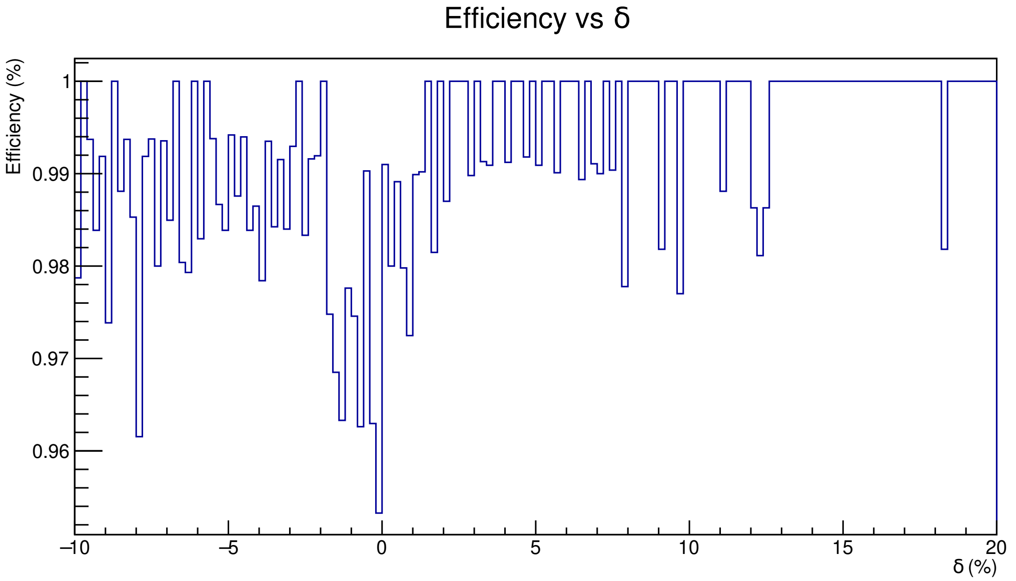


Figure 6.7: Delta scan of efficiency for run 3623, with a HGC NPE cut of 1.0. Electrons are selected, where additional cuts are in text.

across the mirror plane, except across two lines: a horizontal line across the y axis at 1 cm, and a vertical line across the x axis at 0 cm. Consulting Figure 2.11, this is likely due to mirror interleaving causing a drop in the light reflected to the PMTs.

This effect is also reflected in the delta scan of efficiency, shown in Figure 6.7, where a drop is observed at $\delta = 0$. The optics of the SHMS are tuned such that the $\delta = 0$ ray passes the center of the HGC, where the mirror interleaving is maximized and the poorest signal is expected to be observed.

To emphasize these inefficiencies, Figures 6.8 and 6.9 has an NPE cut on 10.0 applied. Consulting the position efficiency scan, the inefficiencies of the mirrors are more pronounced. The inefficiency along the x axis is centered about 0 cm, as expected, and has a width of approximately 2 cm. Similarly, the inefficient band in the y axis is centered around 1 cm and also has a width of 2 cm. As mentioned in Chapter 2 and shown in Figure 2.9, the edges of the mirrors are irregular in shape and so this band of inefficiency is expected. The interleaving of the mirrors is now obvious, reflected by the drop in efficiency at the coordinate (0 cm, 0 cm) and in the delta scan at $\delta = 0$ as before. One particularity revealed in Figure 6.8 are additional regions of inefficiency emerging in the mirror quadrants.

These emergent regions of inefficiency are not related to mirror interleaving. To

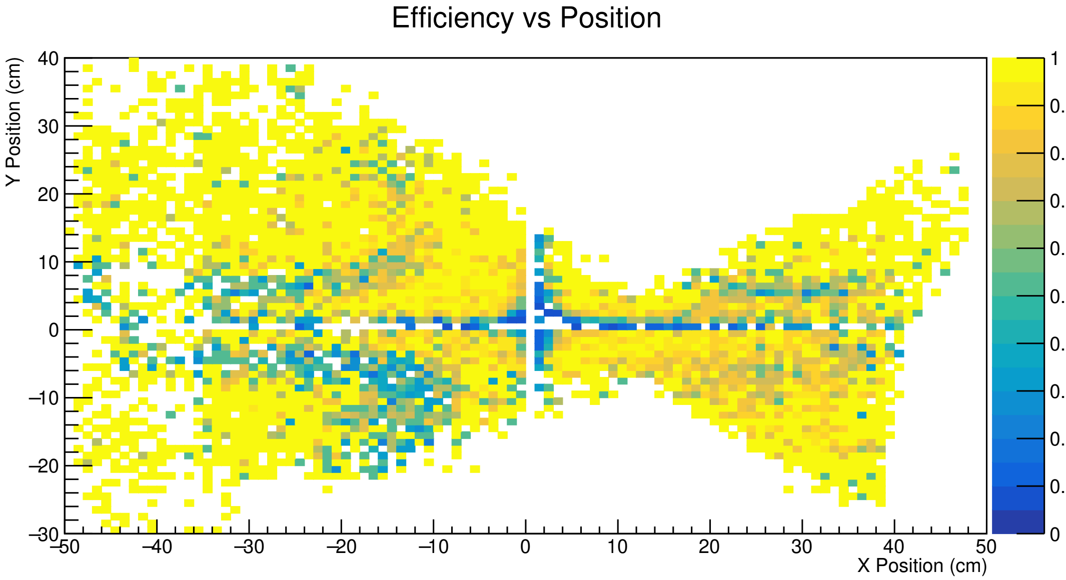


Figure 6.8: Position scan of efficiency for run 3623, with a Cherenkov cut of 10.0. Electrons are selected, where additional cuts are in text.

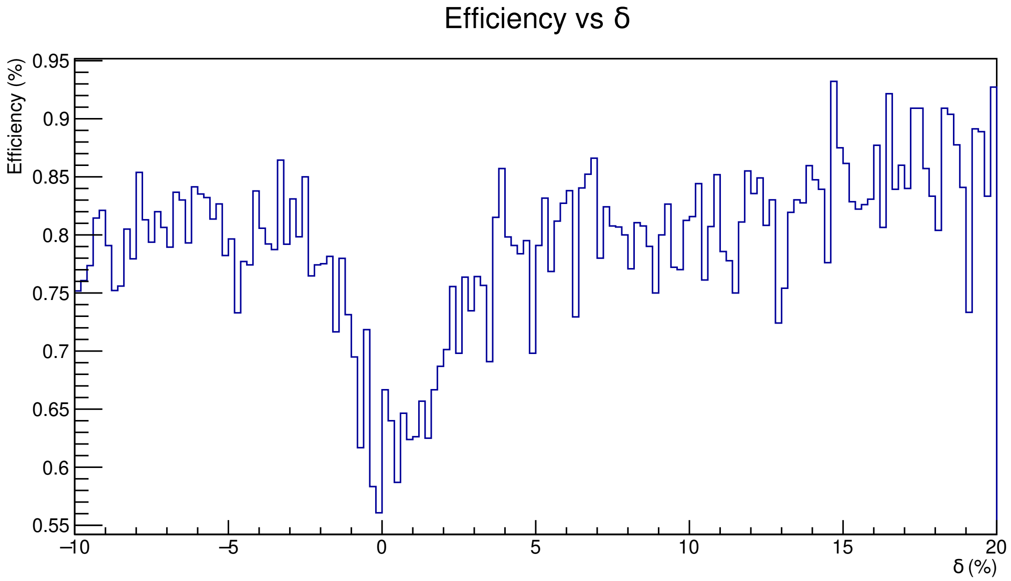


Figure 6.9: Delta scan of efficiency for run 3623, with a Cherenkov cut of 10.0. Electrons are selected, where additional cuts are in text.

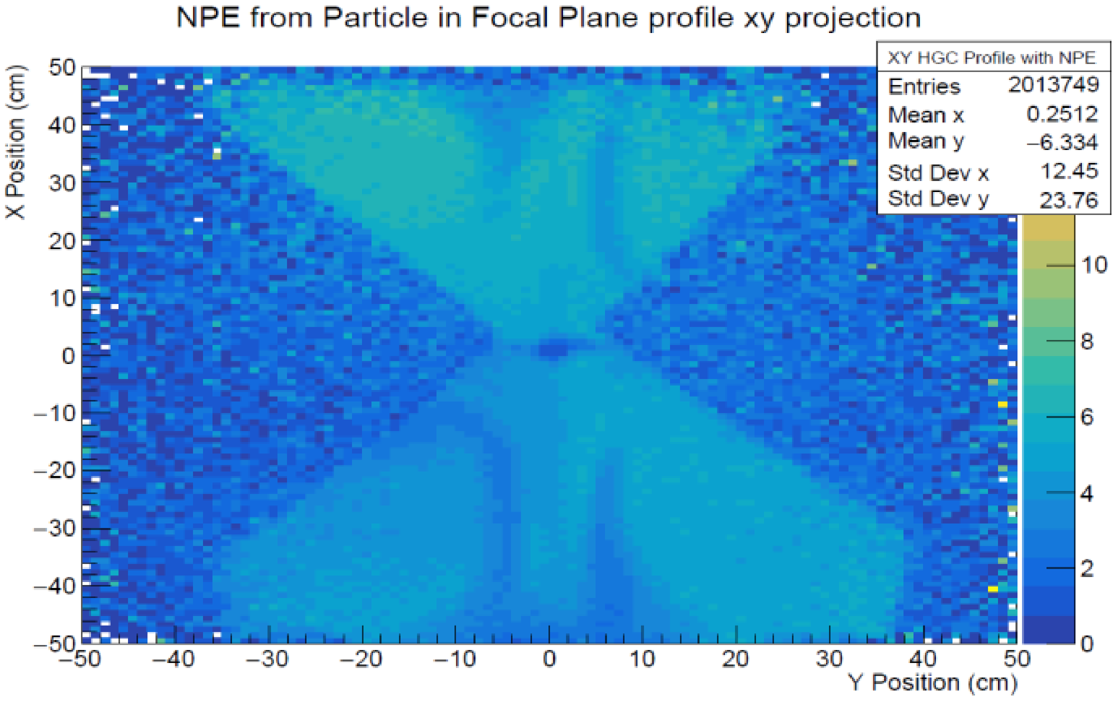


Figure 6.10: Position in the HGC mirror plane with a colour axis representing NPE. Of note are the apparent structures of lower counts. Electrons are selected, where additional cuts are in text.

try and identify the source of these regions, a map of the x and y distribution of events in the HGC mirror plane with associated NPE is generated in Figure 6.10. To

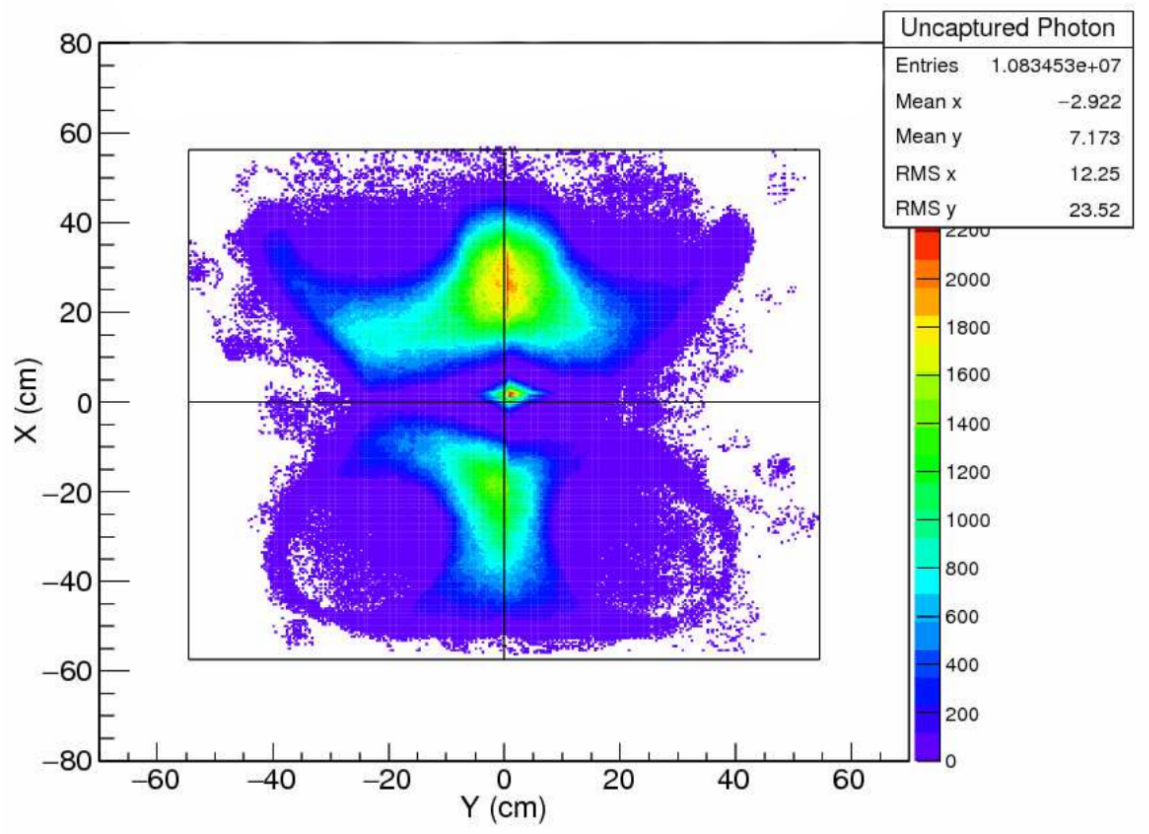


Figure 6.11: Uncaptured photons from a simulation of the HGC. Of note are the curved boundaries.

compare with Figure 6.8, note that the x and y axes are reversed. Examining both of these figures, the regions of inefficiency are caused by the curved regions of lower NPE. These regions are less prominent in mirrors one and two, possibly explaining why these regions of inefficiency are not easily observed in Figure 6.8. This may be a side effect of the optical configuration, or mirror alignment. To this end, the mirror alignment has been reoptimized to better focus the light incident on the mirror interleaving, and the effect on these darker bands will be analyzed.

Interestingly, these regions may be predicted by the detector simulation, shown in Figure 6.11. Here, the colour map refers to the number of photons left uncaptured in the mirror plane. While this is the opposite of the trend observed in Figure 6.10, what is striking is that the circular boundaries in the simulation align almost perfectly with those observed experimentally. This supports the hypothesis that the regions are a result of either alignment, or optical configuration. However, the root cause of these darker regions is not well understood as the simulation does not correctly characterize the observed trends.

6.2 Index of Refraction

The final metric obtained from studying the efficiency of the HGC is an independent measurement of the index of refraction of the Cherenkov media. Mentioned in Chapter 2.1.2, confirming the index of refraction is an established method of verifying the accuracy of a detector's calibration. To perform the measurement, the relationship between a particle efficiency and momentum is fit with Equation 2.10

$$\eta_{\text{HGC}} = 1 - e^{-(p-p_o)/\Gamma},$$

where p_o and Γ are free parameters, and η_{HGC} is the detector efficiency. In this case, the pion efficiency is used since kinematics are available for pions close to, and far away from, threshold momentum. Pion efficiency is calculated in an identical manner as electron efficiency, given by Equation 6.1, where it is the ratio of events identified as a pion by all detectors including the HGC, over the events identified as a pion

Run Number	SHMS Momentum	π Efficiency
3623	-2.53 GeV/c	7.69 %
4167	-2.65 GeV/c	23.35 %
4079	+2.65 GeV/c	22.65 %
3565	-3.37 GeV/c	81.47 %
3566	-3.37 GeV/c	81.20 %
3547	-5.05 GeV/c	99.13 %
3423	+5.05 GeV/c	93.83 %

Table 6.3: Summary of the runs used and the resultant pion efficiencies used to determine the index of refraction. Note that positive and negative polarity runs were used to verify the stability of the efficiency calculation, and account for the electron background.

without any HGC information. To try to account for electron background in the pion sample, data were analyzed at both positive and negative spectrometer polarity, as electrons should not be detected if the SHMS is set to a positive polarity. The runs chosen, the momentum setting of the SHMS, and pion efficiency are shown in Table 6.3. Additional particle identification criteria were used to select the pions, namely that the aerogel Cherenkov registered a signal (NPE > 1.5) and the noble gas Cherenkov did not (NPE < 9.0). Something of notice in Table 6.3 is the stability of the efficiency at a momentum of -3.37 GeV/c, and at ± 2.65 GeV/c, but the discrepancy at ± 5.05 GeV/c. At this high of a momentum setting, it is more likely to incorrectly identify electrons as pions through the various detectors. This may yield an artificially large efficiency for the negative polarity run. The result of fitting the data in Table 6.3 with Equation 2.10 is shown in Figure 6.12. Note that the uncertainty in the pion efficiency arises from Poisson statistical uncertainty and is too small to have an appreciable effect due to the large number of events. Calculating the threshold momentum with Equation 2.10 setting $\eta_{\mathrm{HGC}} = 0.5$ yields $p_{\mathrm{threshold}} = 2.87 \pm 0.04$ GeV/c. Using this threshold momentum, and the pion mass, with Equation 3.21 gives the index of refraction of C_4F_8O to be $n = 1.001 \pm 0.002$, in agreement with the accepted value $n = 1.00139$ (Artuso et al., 2006).

Further work will be done to minimize the errors on the index of refraction measurement, as currently the value $n = 1.001 \pm 0.002$ cannot discern between C_4F_8O or C_4F_{10}. This will be done by taking additional runs in this momentum range to further restrict the fit. Furthermore, it is worrying that the positive polarity run at $+5.05$ GeV/c gave an efficiency that fit poorly with the rest of the data. Both runs had approximately the same average NPE in the HGC with the HGC pion cut applied (run 3547 had 10.66 NPE, run 3423 had 10.92 NPE). The difference in efficiency arises from the larger proportion of events not registering a hit in the HGC, and therefore giving an NPE of zero. The cause of this is yet unknown, and will be explored with

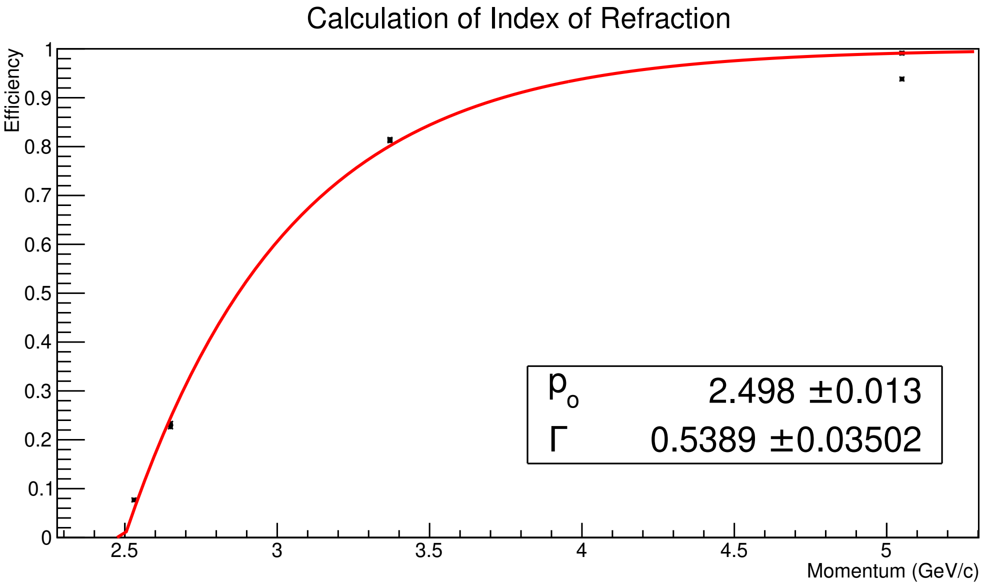


Figure 6.12: Determination of the index of refraction of the HGC. Data represents pion efficiency taken at several momentum settings.

runs obtained in fall 2018.

Chapter 7

Detector Simulation

In the commissioning process, it is important to understand the source of all signals and anomalies. This is the strength of simulation, one can compare their assumptions to how a device will perform with reality. To this end, the heavy gas Cherenkov is simulated using the Geant4 software toolkit (Geant4, 2017) to identify possible improvements to design. The original simulation was developed by University of Virginia, and then heavily modified at the University of Regina (Li, 2012).

Similar to the ROOT analysis framework described in Chapter 4.3, Geant4 is a simulation toolkit designed at CERN and is written in the C++ programming language. The simulation package is designed to model the passage of particles through matter as well as accurately simulate the process of Cherenkov radiation. Objects are simulated by specifying their physical dimensions, optical and radiological bulk properties, and surface/interface effects.

Until recently, no experimental information has been available to compare with expected performance. To begin, the original configuration of the HGC will be presented and the simulated signal will be compared to experimental values. Next, alterations to the detector will be described, as well as the impact on performance. Lastly, ongoing work to fully understand the simulated signal will be presented.

Before going into the simulation details, it is important to keep in mind the coordinate system used for the simulation. It is the same as that of Hall C, the z axis is defined as the trajectory of the central momenta of the beam, the x axis points downwards, and the y axis points to the left.

7.1 Original Configuration

The simulated HGC has specifications identical to those established in Chapter 2, the tank has an inner diameter of 170 cm, length 117.1 cm, and offset by 5 cm in the $+x$ direction. Three Cherenkov media are able to be simulated: CO_2, C_4F_8O, and C_4F_{10}, up to the gas vapour pressure. The resultant index of refraction is determined by a qualitative fit of experimental values and is valid in the region of interest of approximately 1 atm. For example, the C_4F_{10} refractive index is given by

$$n_{\mathrm{C_4F_{10}}} - 1.0 = P\left(\frac{2.5324 \times 10^{-7}}{73.7^{-2} - \lambda^{-2}}\right), \tag{7.1}$$

where λ is the photon wavelength in nm and P is the pressure of the gas in atm. This relation is calculated at 273 K and 1 atm, reasonably close to our operating conditions (RuggeroTurra, 2009). The dispersion formula for CO_2 is given by

$$\begin{aligned} n_{\mathrm{CO_2}} - 1.0 = P\Bigg(&\frac{6.991 \times 10^{-2}}{166.175 - \lambda^{-2}} + \frac{1.4472 \times 10^{-3}}{79.609 - \lambda^{-2}} + \frac{6.42941 \times 10^{-5}}{56.3064 - \lambda^{-2}} \\ &+ \frac{5.21306 \times 10^{-5}}{46.0196 - \lambda^{-2}} + \frac{1.46847 \times 10^{-6}}{0.0584738 - \lambda^{-2}}\Bigg), \end{aligned} \tag{7.2}$$

where again P refers to the gas pressure in atm, λ is the photon wavelength in nm, and data were taken at 273 K and 1 atm (Bideau-Mehu et al., 1973). Lastly, the index of refraction for C_4F_8O is given by

$$n_{\mathrm{C_4F_8O}} - 1.0 = P\left(1.3178 \times 10^{-3} + 1.7488 \times 10^{-7}E + 5.5785 \times 10^{-6}E^2\right), \tag{7.3}$$

where E is the energy in MeV of the photon and P is the gas pressure in atm (Fuchy, 2016).

The absorption length of each gas is determined from a simple application of the Beer-Lambert law

$$T(\lambda) = e^{-\lambda/\alpha}, \tag{7.4}$$

where $T(\lambda)$ is the transmission of light of a particular wavelength λ and α is the absorption length. By convention, α is defined by the distance into a material where the incident flux of particles has dropped by $1/e$. Thus, an infinitely large α refers to no absorption, while $\alpha = 0$ refers to complete absorption. The values for the absorption length are taken from Figure 3.2, where the plots are sampled across 15 photon wavelengths in the UV region of interest and the corresponding transmission is used with Equation 7.4 to calculate the absorption length.

As discussed in Chapter 2, the mirrors in the HGC have an oblate ellipsoid shape with radius 112.96 cm and conic constant 0.94 (Li, 2012). This allows each mirror to reflect incident Cherenkov light at $\pm 42^o$ with respect to the x axis, focusing it onto the PMT arrangement. The positioning of the mirrors is established in Figure 2.11, where the central interleaving is in the order 4, 3, 2, 1 and this overlap is approximately 5 cm. A visualization of the HGC layout is shown in Figure 7.1 where: mirror 1 is blue, mirror 2 is red, mirror 3 is yellow, and mirror 4 is pink. Also shown is the focusing of the Cherenkov cone depicted in green. Note that the mirrors are not identical in the simulation, but specified according to the actual elliptical fit results for the individual mirrors (Li, 2012).

The optical properties for the mirrors were implemented in a similar manner as the radiator gases. The results of measuring the reflectivity of a physical mirror, as shown in Figure 2.10, were coded into the simulation. Since this measurement was performed by the University of Regina (Li, 2012), a higher resolution of data was available, and thus the reflectivity is specified for 33 photon wavelengths.

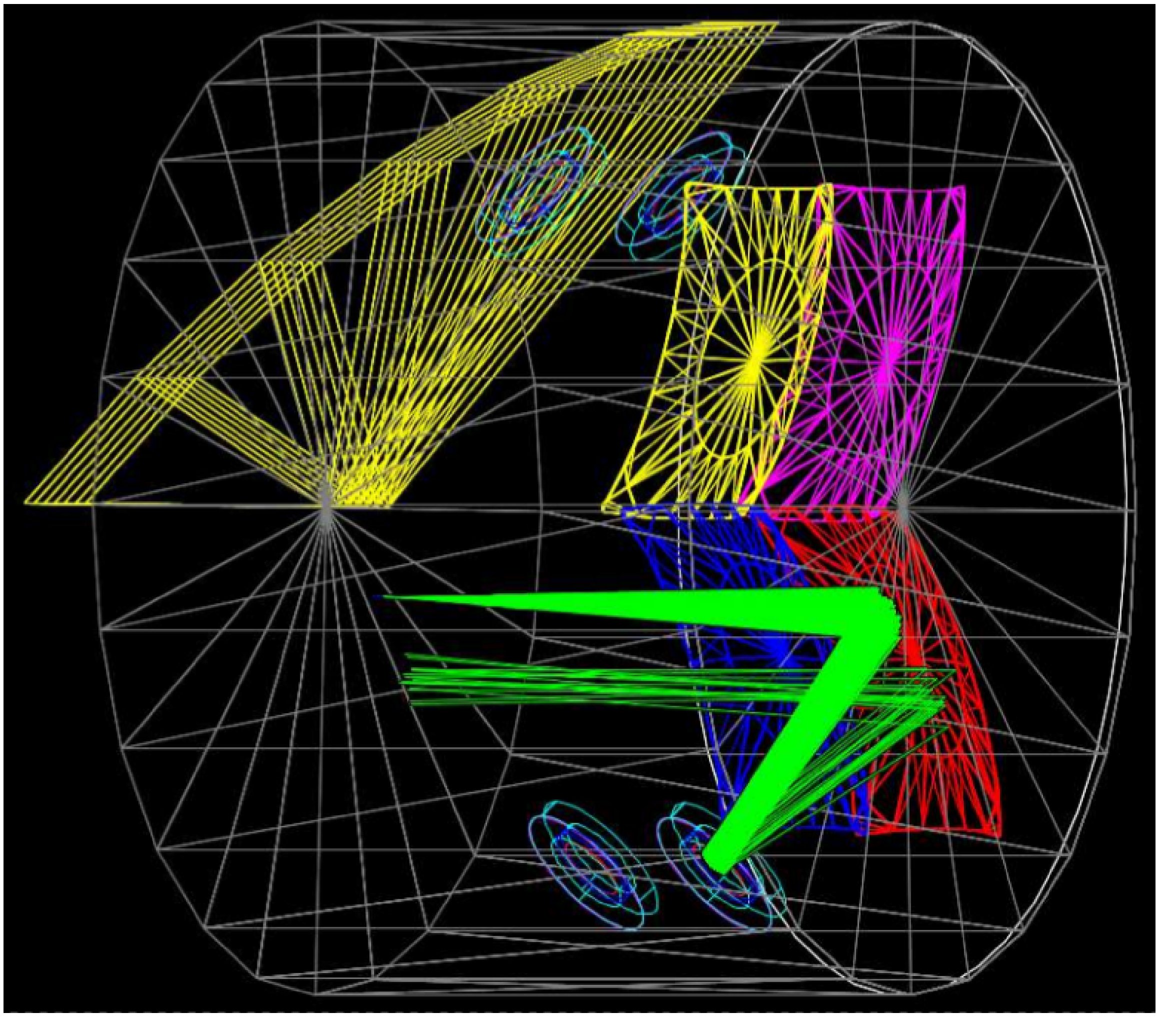


Figure 7.1: Layout of the heavy gas Cherenkov in the Geant4 simulation. Each mirror is coloured yellow, pink, blue, and red. Also visible are the PMT locations as the concentric blue circles. Photon trajectories are in green. Tracking planes are given in yellow.

The original configuration for the simulated PMT and associated optics consisted of four components: cathode, quartz window, flange, and backing. The flange and backing serve to restrict the dimension of the PMTs and are simulated to absorb any incident light. The quartz window has the chemical composition and absorption length specified by Corning 7980 quartz (Corning, 2018), with an additional 10% photon loss to compensate for Fresnel reflection at the surface. The transmission for the window is given in Figure 2.14. The photocathode is flat and positioned such that the leading edge is the same as the real photocathode. Wavelength dependent quantum efficiency, and radial dependent efficiency were taken from the manufacturer (Li, 2012) and can be observed in Figure 2.19.

To facilitate comparison with physical data, simulations take as input the position, angle, and momentum of each event in the focal plane. This information is easily available from a replayed run, where it needs to be translated into an ASCII format. As the events (i.e. charged particles) are propagated through the detector, they produce Cherenkov radiation as dictated by the simulation calculations. These secondary photons are then reflected by the mirrors onto the PMT arrangement where they pass seven tracking planes to record photon trajectory near the PMT surface. Three planes are located before and after the cathode plane, while the last lies directly on the flat cathode. These seven planes are displayed in Figure 7.1 as the yellow lines. Each plane is separated by 2 cm in the z direction and lie parallel to each other.

Currently, the simulation is capable of producing $e^{\pm}$, $\pi^{\pm}$, $K^{\pm}$ or proton events where the amount of detected photons, their position, and energy are recorded. Functionality was added to the simulation to separate the PMT signals into the specific mirror quadrant, as is done for the experimental data. Additionally, the initial particle envelope is stored to verify its accuracy to the input trajectories.

The first experimental run used to compare with simulated data was run 488, taken during the Key Performance Parameter (KPP) commissioning of the SHMS. For this setting, the HGC was filled with CO_2 at 1 atm and the energy of the beam was 10 GeV/c (hclog, 2017b). The comparison between simulation and experiment is shown in Figure 7.2, where the red histogram refers to experimental data and the blue is simulated data. In this figure, the total response from all four PMTs are summed together, and electrons are selected. For the simulation this simply requires specifying electrons are produced, while for the experimental data the particle identification criteria outlined in Chapter 5 were used. There is clear discrepancy between these two results, the cause of which is hypothesized to be due to the simplicity of the optical configuration of the PMT. As can be seen in Figure 2.13, several components of the PMT couplings are absent: a layer of grease, a quartz adaptor, and the RTV 615. This may allow photons that realistically would be absorbed to pass through to the PMT. This is supported by examining the profile of photon energies registered at the PMT shown in Figure 7.3. Above photon energies of 6.2 eV, corresponding to wavelengths less than 200 nm, a significant amount of light passes to the PMT, although the transmission curves of RTV 615 and optical grease shown in Figures 2.15 2.16, and 2.17 suggest this may not be the case. To test this theory, a hard photon cutoff of 6.2 eV (200 nm), shown by the red curve of Figure 7.3, was applied to the simulation. The resultant distribution is shown in Figure 7.4, where the average number of photoelectrons is significantly closer to the experimental value than it is in Figure 7.2. This is highly suggestive of the importance of the additional UV absorbing components, and thus they were implemented in the Geant4 simulation as well.

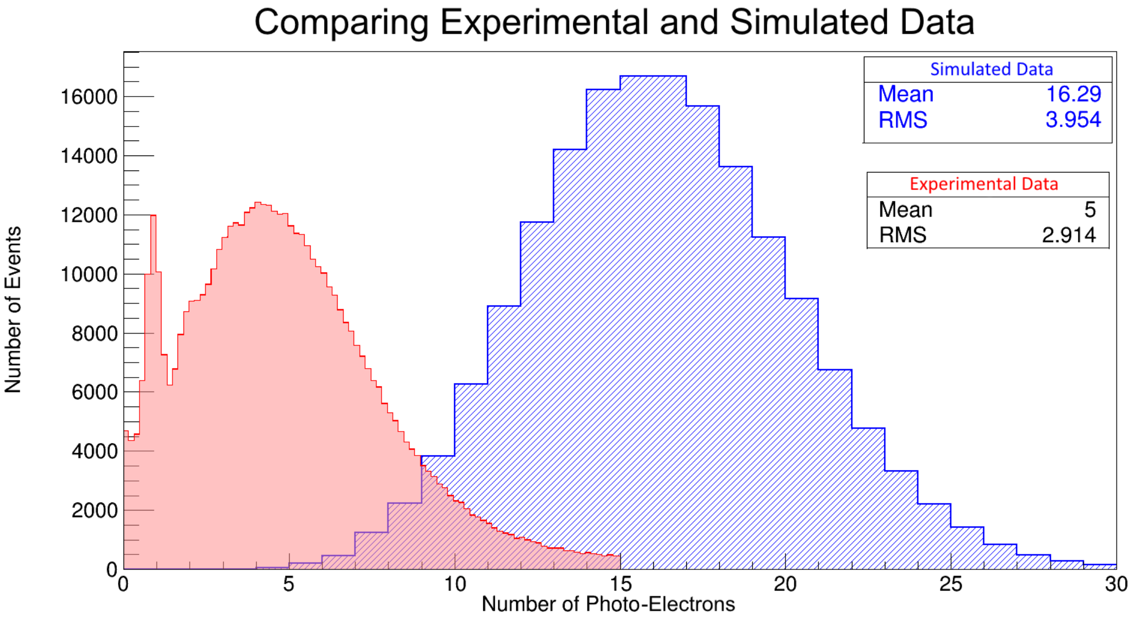


Figure 7.2: Comparison between experimental and simulated HGC response for run 488. Combined PMT response is shown, with electrons being selected. Simulated data is in blue and experimental data is in red, where the SPE is clearly observed. Discrepancy between the two is attributed to optical configuration of the PMTs.

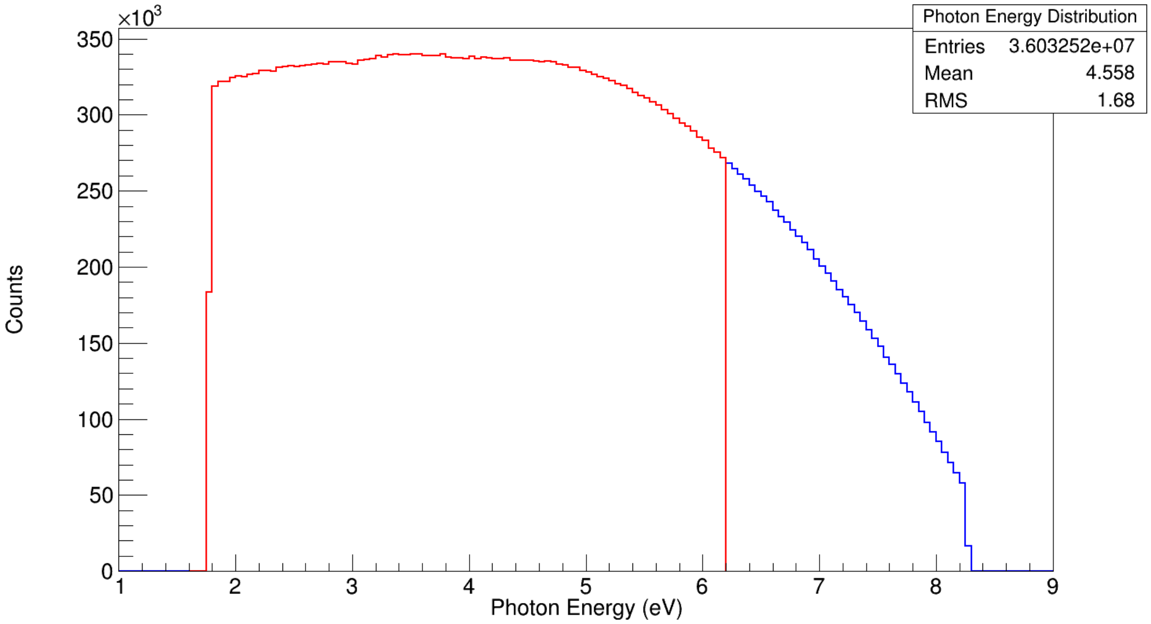


Figure 7.3: Simulated profile of photon energies at the PMT surface. Blue refers to the profile observed in Figure 7.2 and the red illustrates the restriction applied to yield Figure 7.4.

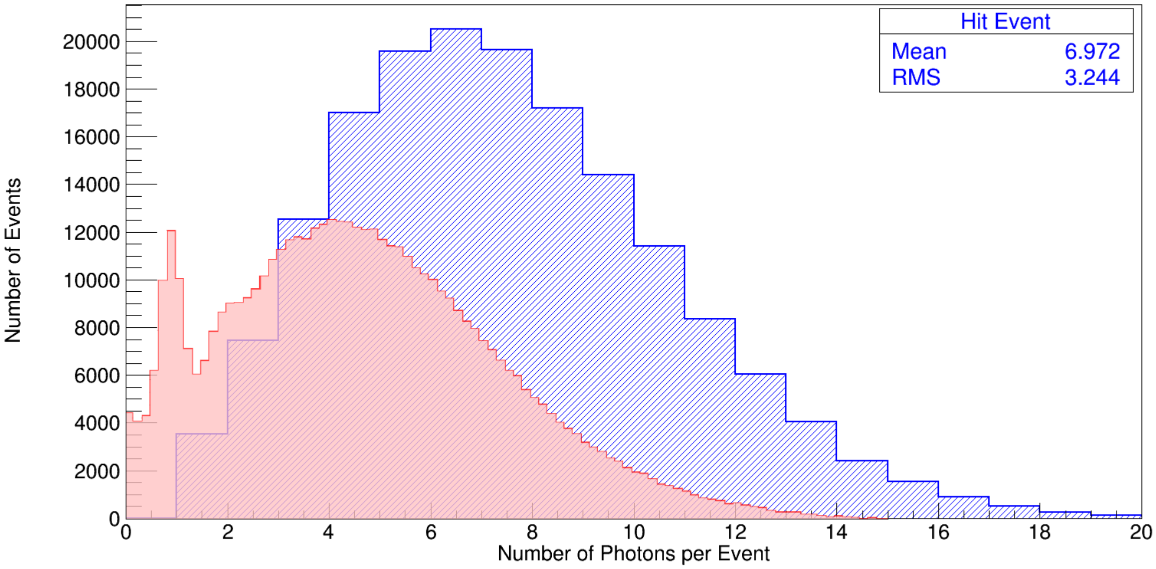


Figure 7.4: Simulated number of photoelectrons measured after the 200 nm photon cutoff of Figure 7.3 was applied. Simulated data is in blue and experimental data is in red, where again total PMT response is shown with electron particle identification. Comparing to Figure 7.2, the agreement between simulation and experiment is much improved.

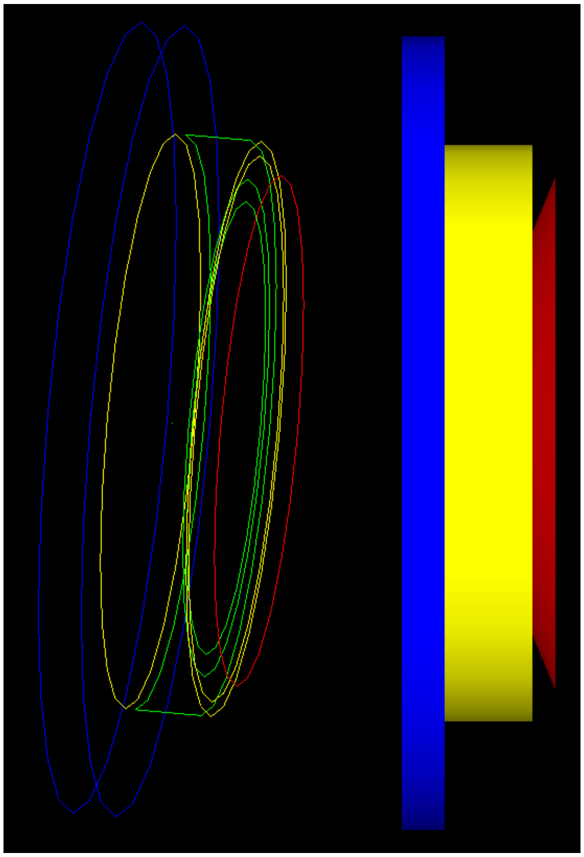


Figure 7.5: Visualization of updated PMT optical configuration for the HGC simulation. The left side shows a wireframe depiction of the PMT ensemble while the right shows the filled surfaces. The blue component is the quartz window, yellow is the quartz adaptor, green is the RTV 615, and red is the cathode surface. A layer of grease is present between the window and adaptor, but not visible. Also note the RTV is only visible in the wireframe view on the left.

7.2 Updated Configuration

The optical grease, quartz adaptor, and RTV 615 were simulated by the author in an identical manner as the other PMT components. Absorption length for the optical grease is obtained from Figure 2.15, where a linear extrapolation was performed to reach photon wavelengths of 200 nm. The layer of grease is assumed to be 500 μm thick, with an index of refraction $n = 1.4658$ (Dow Corning, 1998). Following the grease is a quartz adaptor to couple the flat window to the curved cathode surface. Naturally, this component has identical optical properties as the quartz window, given in Figure 2.14, and the curved surface has a radius of curvature of 132.0 mm. This radius is larger than that of the cathode, thus the space is filled with RTV 615 due to its index of refraction, $n = 1.406$, being reasonably close to that of quartz (Momentive, 2018). Two separate measurements of the absorption length of RTV 615 were performed by the University of Regina (Li and Huber, 2014) and North Carolina A&T State University (Wilson, 2014), shown in Figures 2.16 and 2.17, respectively. Although both measurements are accurate, the RTV 615 used in the HGC is approximately 100 μm thick, meaning the latter measurement is more appropriate as they measure down to 250 μm thickness. The cathode surface was also changed from flat to curved, with a radius of curvature of 125 mm. At each interface, a 10% photon loss to compensate for Fresnel reflection is included as well. A visualization of the new PMT configuration is presented in Figure 7.5.

Similar to before, this new optical configuration is compared to experimental data. In this case, Run 1583 is used for comparison, where the central momentum is 2.2 GeV/c and the HGC was filled with CO_2 at 1.0 atm (hclog, 2017a). As well, the PMT response is separated according to each mirror quadrant, yielding a 4×4 grid of histograms shown in Figure 7.6. Here each row corresponds to a PMT, and each column to a mirror quadrant. The thick blue histogram with a fill is simulated data and the thin histogram without a fill is experimental data. Between the two data sets one should compare the mean of the distribution but not the number of counts. This is reflected by the normalized counts being plotted as opposed to the raw number of counts.

Before discussing Figure 7.6, the change in the accepted photon energy spectrum will be commented on. This is shown in Figure 7.7 where, after comparing with Figure 7.3, one can see that fewer photons are accepted below 200 nm wavelength as expected with the inclusion of optical grease and RTV 615.

Performing the comparison of Figure 7.6 PMT-by-PMT, there is excellent agree-

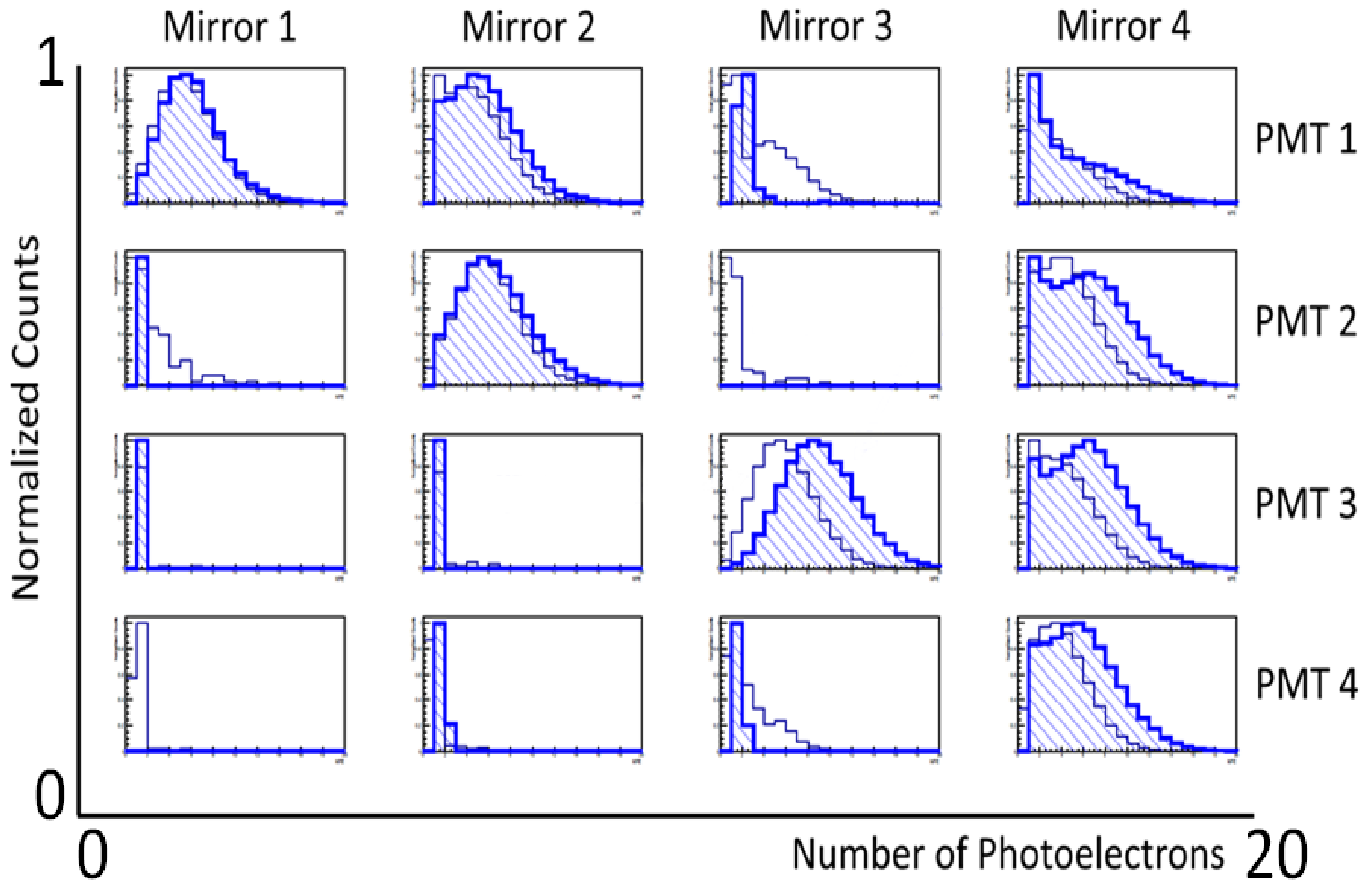


Figure 7.6: Simulated detector response versus experimental data with the updated optical configuration given in Figure 7.5 implemented. The thick blue filled histogram is simulated data, and the thin blue histogram is experimental data. Each column refers to a particular mirror quadrant, and each row to a unique PMT.

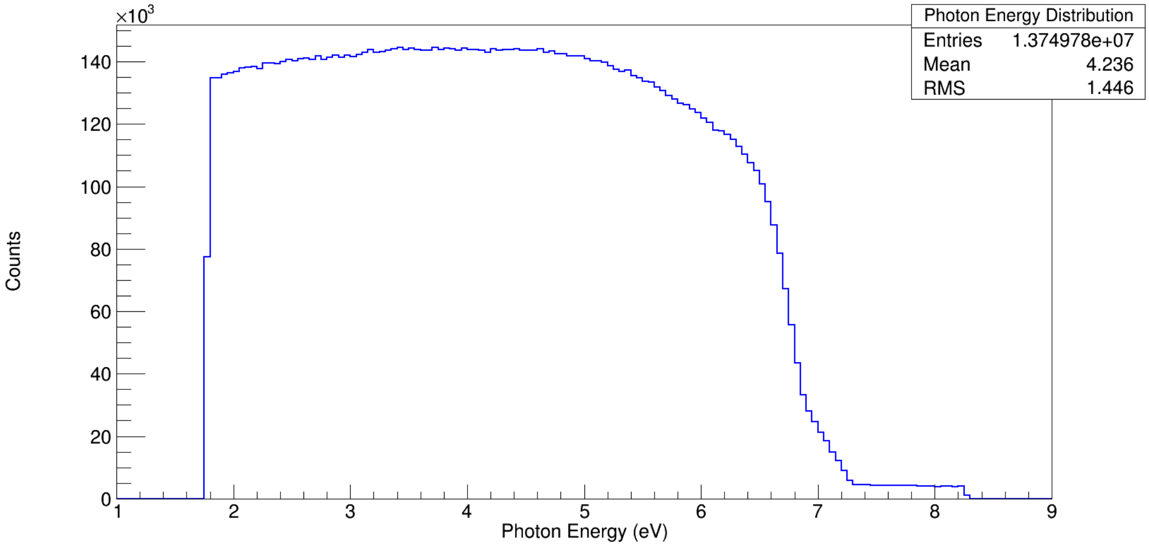


Figure 7.7: Profile of photon energies measured at PMT surface with the updated optical configuration.

ment with PMT 1 response from mirrors 1 and 2. The response from mirror 3 is somewhat worse where the simulation anticipates almost no light to be received by PMT 1, but a definite signal is obtained. Mirror 3 lies above mirror 1, suggesting that this mirror boundary needs to be readjusted or the alignment of mirror 3 has changed. The signal from mirror 4 is similar between simulation and experiment, with simulation overpredicting by a small amount. PMT 2 is in good agreement for mirrors 1, 2, and 3. An identical discrepancy is present in mirror 4, where the simulation overpredicts the amount of light by roughly the same amount as for PMT 1. PMT 3 has excellent agreement for mirrors 1 and 2. However, the PMT response from its on-quadrant mirror is different between experiment and simulation, unlike PMT 1 and 2. This difference is similar in magnitude as the discrepancy seen between PMT 1 and mirror 3, suggesting a mirror alignment issue is the culprit. Another explanation is that the grease implementation is inaccurate, and should be cutting more light out of the simulation. This may also explain the discrepancy between the two data sets for mirror 4. Lastly, PMT 4 is in good agreement for mirrors 1, 2, and 3. Similar to PMT 3, there is an overestimation by the simulation for the on-quadrant mirror.

Now general trends and improvements suggested by Figure 7.6 will be discussed. Based on the accuracy of the PMT 1 and 2 simulation in all mirror quadrants except 3 and 4, it is reasonable to assume that these PMTs are now implemented correctly in the simulation. Mirror quadrant 4, corresponding to the furthest upstream mirror, gave systematically larger NPE (approximately 1.0 - 1.5) in the simulation than experimentally. This suggests that the reflectivity of mirror 4 is worse in reality than in the simulation, since such a uniform drop is observed. The discrepancy present in mirror 3 is harder to resolve. For PMTs 2 and 4, there is reasonably good agreement, while PMT 1 is underestimated and PMT 3 is overestimated. One possible explanation is that mirror 3 has tilted due to gravity to shift its focus slightly more to PMT 1, which lies further in the x direction. However, this is improbable due to the constrained space and an alignment test performed in May 2018 indicated the mirror alignment has not changed within measurement uncertainties (Huber, 2018b). The next likely explanation is a combination of effects, such as mirror 3 edge effects giving less dispersion in the simulation than in reality, and inaccuracy in the grease and RTV simulation yielding fewer NPE for the on-quadrant PMT. Nevertheless, the discrepancy between simulation and reality is on the order of 1 - 2 NPE, making it a reasonable representation of the detector response.

7.3 Detector Improvements

With a more realistic Geant4 simulation of the HGC, aspects of the detector design were altered to try to increase the measured NPE. If the amount of light received by the PMTs increases, and the spread between the NPE response of pions, kaons, and δ-rays widens, a higher efficiency and lower contamination will be achieved. Due to the design of the HGC, the easiest way to improve the response is by changing the coupling of the PMT to the window, as this lies outside of the HGC tank. As seen in Chapter 7.2, the inclusion or removal of optical components in the simulation has dramatic effects on the number of measured NPE.

To achieve the large number of photoelectrons observed in the original simulation (consult Figure 7.2), the coupling was simplified to feature an acrylic ring holding the PMT surface close to the quartz window. This way, the optical grease, quartz adaptor, and RTV are removed and replaced with a small air gap. The simulated ring is presented in Figure 7.8, where the dimensions are: inner radius of 61.5 mm, outer radius of 70.0 mm, thickness of 8.5 mm, and a length of 46.0 mm. Starting at 36.0 mm down the ring, it has a beveled surface of 30^o which will rest against the PMT glass. The dimensions are chosen such that the ring edge in contact with the PMT surface will not infringe on the active cathode area. Additionally, the ring has a strip of aluminized Mylar along the inner circumference to enhance reflection of photons onto the PMT cathode. Care was taken to ensure this strip does not come into actual contact with the cathode, as this may lead to PMT discharge. It is also worth noting that the PMT window was simulated with the ring, where the properties of UV transmitting glass were taken from Figure 2.20 and the gap between the quartz window and cathode surface was taken to be dry air. The simulated performance of the ring is presented in Figure 7.9, where it is compared to the identical experimental data set of Figure 7.6. The simulation predicts a significant improvement to the NPE signal, however, it is unlikely that this degree of improvement will be observed. A more pragmatic perspective is that the experimental signal will increase if everything is replaced with the acrylic ring.

This new optical configuration was tested experimentally in the spring of 2018 during the semi-inclusive deep inelastic scattering experiment (SIDIS) (hclog, 2018d). Unfortunately, the aluminized Mylar strip was omitted since the adhesion to the ring was not adequate. Two runs are compared: run 4167 where the SHMS was set to -2.65 GeV/c central momentum and the HGC was filled with 1 atm C_4F_8O without the

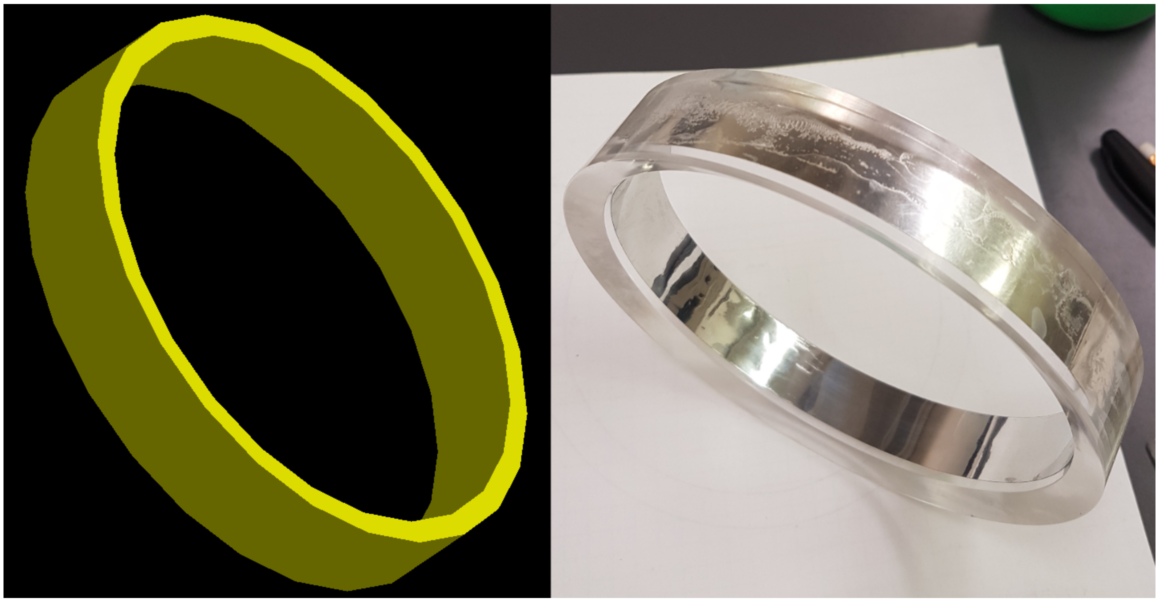


Figure 7.8: Visualization of acrylic ring used in the HGC simulation. The left side is how the ring is implemented in the Geant4 simulation, and the right is the actual ring.

ring (hclog, 2018b), and run 4201 with identical settings except for the newly installed ring (hclog, 2018c). Due to the kinematics of these settings, few electrons passed through the HGC and other charged particles are below the Cherenkov threshold, which yielded fairly low statistics runs. Subsequent runs had the SHMS at positive polarity, which causes statistics to become even worse. The results of this comparison are presented in Figure 7.10. Unfortunately, the signal appears relatively unchanged with a difference of, at most, 1 photoelectron. Despite this, all four PMTs will be changed to feature the acrylic ring instead of the optical grease, quartz adaptor, and RTV 615 combination. The primary motivation is that after testing the two configurations, a marked increase in transmission of UV photons was observed, shown in Figure 7.11 (Sawatzky et al., 2018). Here, the bold lines refer to PMT 2, where the green and dark blue have a layer of RTV 615 present and the light blue and purple have the RTV removed and instead have just an adaptor (light blue) or an air gap (purple). Comparing this profile to the transmission of RTV 615 measured by the University of Regina in Figure 2.16, or North Carolina A&T State University

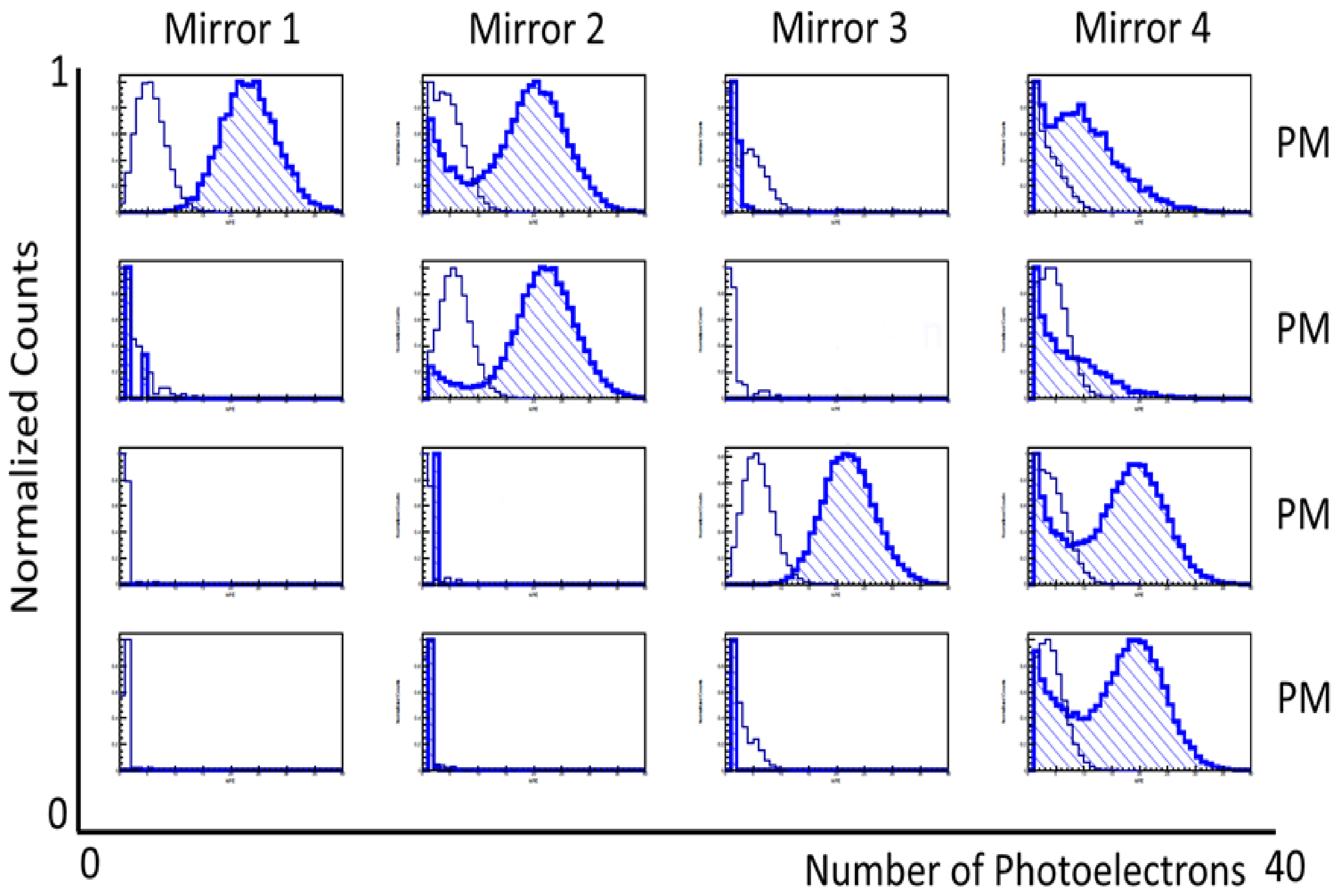


Figure 7.9: Simulated detector response versus experimental data with the acrylic ring shown in Figure 7.8 installed. The thick blue filled histogram is simulated data, and the thin blue histogram is experimental data. Each column refers to a particular mirror quadrant, and each row to a unique PMT.

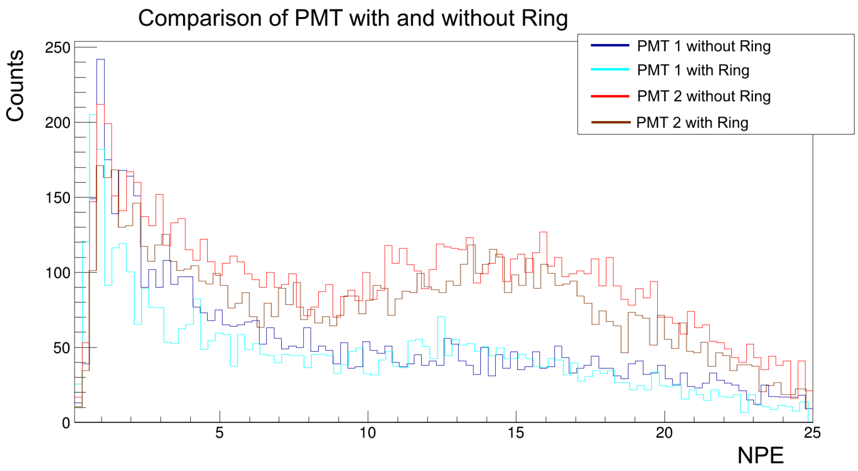


Figure 7.10: Comparison of NPE counts in PMTs with and without the acrylic ring installed. Dark blue and red lines refer to PMT 1 and 2 respectively without the ring, and light blue and dark orange refer to PMT 1 and 2 respectively with the ring installed.

in Figure 2.17, the influence of the RTV is made obvious. Lastly, a new test of the acrylic ring with a reflector cone design will be tried later.

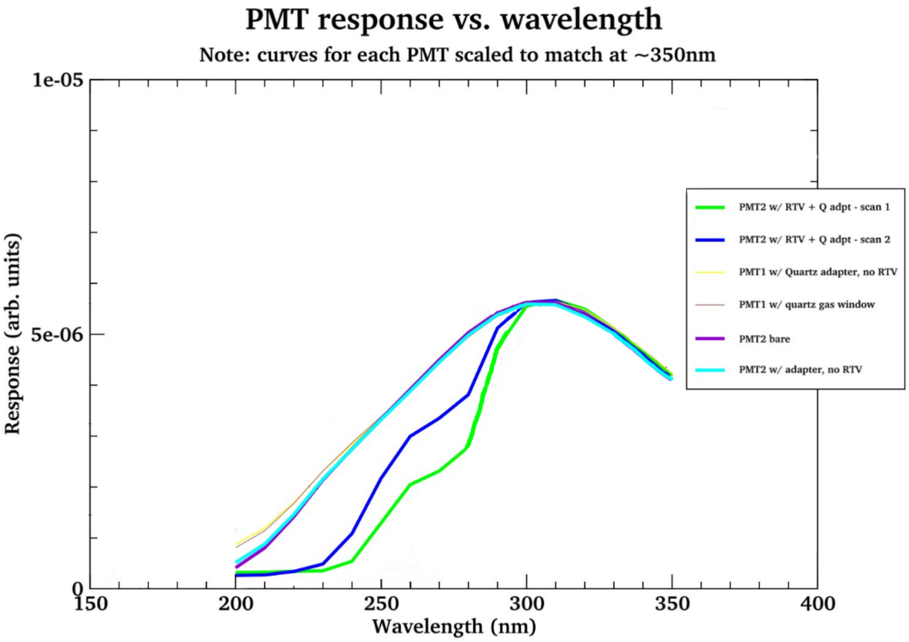


Figure 7.11: Effect of removing the RTV 615, optical grease, and quartz adaptor from the PMT and replacing it with an acrylic ring and a 5 mm air gap (Sawatzky et al., 2018).

Chapter 8

Experimental Performance

With the commissioning work for the heavy gas Cherenkov completed, the scientific role it will play will be described, along with some initial performance results. Of particular interest to our research group is the upcoming L-T separated kaon electroproduction experiment, scheduled to begin August 2018 (Horn et al., 2008). The reaction of interest for this experiment is $p(e, e'K^+)\Lambda, \Sigma^o$, with the objective of studying quantum chromodynamic (QCD) model building with strangeness degrees of freedom, and measuring the kaon-hyperon coupling constants in nucleon-meson and quark models.

This study is of scientific significance, because of the insight it provides into the strong interaction. This interaction, one of the four fundamental forces of nature, dictates the structure of hadronic matter. Unlike the other forces, the strong interaction is feeble at small distance scales and gets asymptotically stronger as the distance increases. This unique property leads to several physical phenomena, all of which are attempted to be characterized by QCD. This theory describes the interaction between the quanta of the strong interaction, quarks and gluons.

While highly successful, some aspects of QCD need to be better understood. First, there is a lack of physical data to restrict the various parameters and predictions of the theory. This limits the predictability of the theory as a whole. Second, many calculations are intractable in QCD since the coupling constants depend strongly on momentum transfer. This leads to a regime called perturbative quantum chromodynamics (pQCD), where the physical processes are exactly calculable at small distance scales. The point at which pQCD becomes valid is not well known, and a better understanding will shed light on the transition between hadronic (long range, ∼1 fm) and partonic (short range, <0.1 fm) degrees of freedom.

8.1 L-T Separated Kaon Electroproduction Experiment

Probing the kaon electroproduction channel to investigate the effects of strangeness on QCD models is attractive for two main reasons: the kaon's relative simplicity, and the existing body of knowledge. Kaons are mesonic particles, meaning they have a valence quark structure of a quark-antiquark pair. In particular, a K^+ has an up and anti-strange valence quark structure. Since bare quarks cannot be produced, this particular structure is one of the simplest to study. Additionally, the up quark is the lightest of the quark family, resulting in the K^+ being the easiest strange meson to produce, hence is responsible for the main part of the nucleon-hyperon interaction that binds atomic nuclei. The proposed technique has been very successful in probing the QCD structure of another meson, the pion. This particle has a valence quark structure of an up and anti-down pair, making it very light and easy to produce. Comparing the pion to the kaon, one can see that the anti-down quark is exchanged for an anti-strange quark. Thus, repeating these experiments to produce a kaon directly explores the effect of introducing strangeness to QCD.

Before delving into the experimental details, a brief digression into the meaning behind an L-T separated experiment will be explained. It is well known that photons have degrees of freedom in how their electric (or magnetic) field is oriented in space. Real photons must be transversely polarized, resulting in two unique configurations. Virtual photons have no such restriction, and thus have an extra degree of freedom for longitudinally polarized fields. The cross-section is dependent on whether the photon is transversely or longitudinally polarized as,

$$2\pi\frac{\mathrm{d}^2\sigma}{\mathrm{d}t\mathrm{d}\phi} = \epsilon\frac{\mathrm{d}\sigma_L}{\mathrm{d}t} + \frac{\mathrm{d}\sigma_T}{\mathrm{d}t} + \sqrt{2\epsilon\left(\epsilon+1\right)}\frac{\mathrm{d}\sigma_{LT}}{\mathrm{d}t}\cos\phi + \epsilon\frac{\mathrm{d}\sigma_{TT}}{\mathrm{d}t}\cos 2\phi, \tag{8.1}$$

$$\epsilon = \left(1 + 2\frac{|\mathbf{q}^2|}{Q^2}\tan^2\frac{\theta}{2}\right)^{-1} \tag{8.2}$$

where ϵ is a measure of the proportion of longitudinally polarized photons. The subscripts represent either transverse, longitudinal, or a superposition of orthogonal polarizations, ϕ is the azimuthal dependence, t is the Mandelstam variable representing four momentum transfer, and $\mathbf{q}^2$ is the square of the three-momentum transferred to the nucleon. One can then perform a Rosenbluth separation by measuring the cross-section at fixed invariant mass, Q^2, $-t$, and at two beam energies corresponding to different ϵ. By performing a simultaneous fit of the four cross-sections, the contribution from the various polarizations can be isolated.

As established in Chapter 2, the experimental facility for this experiment features an electron beam incident onto a proton target. Since neither a kaon target nor suitable kaon beam is available, due to the low Q^2 restriction imposed by the Lorentz boost to the center of mass frame, the structure of the meson must be probed indirectly. This is performed via the kaon pole diagram, depicted in Figure 8.1. Here, the fact that the proton exists as a superposition of several quantum states,

$$|p\rangle = |p\rangle_o + |n\pi^+\rangle + |\Lambda K^+\rangle + |\Sigma^o K^+\rangle + \cdots, \tag{8.3}$$

is taken advantage of, where the off-shell virtual meson cloud is interacted with. Note

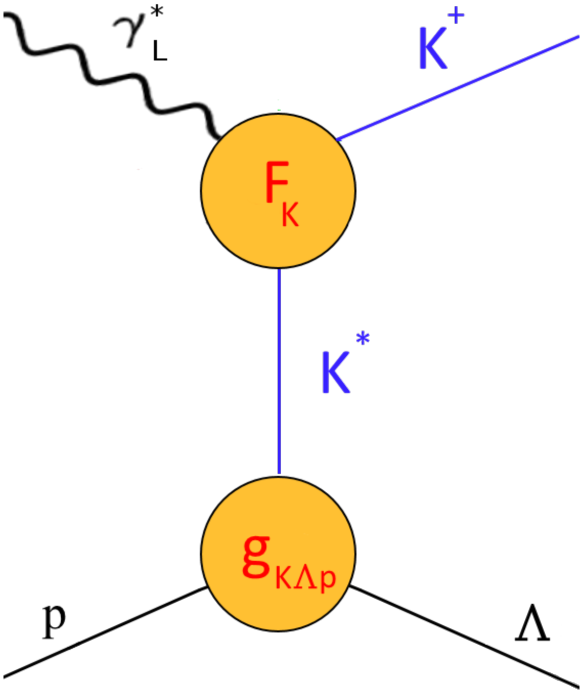


Figure 8.1: Kaon pole diagram.

that in the previous pion electroproduction experiment, the same interaction is used except with the virtual pion cloud.

If the kaon pole diagram is the dominant interaction, the Born term model predicts that the resultant cross-section for longitudinally polarized photons is,

$$\sigma_L \propto \frac{-tQ^2}{(t-m_K^2)^2}\left(g_{K\Lambda p}\right)^2 F_K^2\left(Q^2\right), \tag{8.4}$$

where t corresponds to the Mandelstam variable $(p_\gamma - p_K)^2$, the four-momentum transfer to the nucleon or transverse size of the target, and Q^2 is the four-momentum of the virtual photon $(p_e - p_{e'})^2$ and represents the photon's virtuality or the transverse size of the probe. Note that these are kinematic variables which are experimentally controllable. As well, m_K is the on-shell mass of the kaon, $g_{K\Lambda p}$ is the coupling constant between the virtual meson cloud and the on-shell baryons, and $F_K^2\left(Q^2\right)$ is the form factor of the kaon.

While Equation 8.4 offers a direct way to probe the kaon's inner structure, it is only accurate if the interaction of Figure 8.1 is dominant. This may be determined by comparing two independent processes for kaon electroproduction, a final state baryon of Λ or Σ^o. Since either may be produced via a kaon pole diagram like Figure 8.1, they have an identical form for their cross-section as Equation 8.4. The only difference will be in the interaction between the baryon and virtual meson cloud, encoded in the coupling constant $g_{K\Lambda p}$ or $g_{K\Sigma^o p}$. Thus, by taking a ratio of the cross-sections, one will obtain, if the kaon pole diagram dominates,

$$\frac{\sigma_L\left(\gamma^* p \rightarrow K^+\Lambda\right)}{\sigma_L\left(\gamma^* p \rightarrow K^+\Sigma^o\right)} \propto \frac{g_{K\Lambda p}}{g_{K\Sigma^o p}}, \tag{8.5}$$

assuming the cross-sections are measured at identical kinematics. Thus, if it is found that the ratio of Equation 8.5 remains constant across the kinematic regime, the Q^2 dependence of the kaon form factor will be extracted with high confidence. In the case that Equation 8.5 is not constant, the insight into the cross-sections of a virtual kaon cloud will provide new restrictions to QCD models.

Another objective for the L-T separated kaon electroproduction experiment is to determine the onset of pQCD. Previously mentioned, pQCD is the exactly calculable regime of QCD and represents a transition from hadronic degrees of freedom (baryons exchanging mesons, intractable to calculate) to partonic degrees of freedom (quarks exchanging gluons, perturbatively calculable). Indicative of this transition is the validity of factorization, depicted in Figure 8.2. In basic terms, factorization implies that the contribution from hard gluon interactions may be separated from those of soft gluons. In this way, the combined effect of soft gluons may be characterized by a generalized parton distribution function (GPD) and can be "matched" with various hard gluon processes. This regime has well defined Q^2 dependencies from pQCD for the cross-sections: $\sigma_L \propto Q^{-6}$ and $\sigma_T \propto Q^{-8}$. Thus, these relationships will be tested along with the limit that at large Q^2, $\sigma_L >> \sigma_T$.

8.2 Heavy Gas Cherenkov Performance

With the scientific motivation established, the details of the heavy gas Cherenkov will be discussed. The HGC will perform as a component of the particle identification system, specifically to separate between pion and kaon signals. As the final state of the interaction includes a kaon, the importance of the detector is evident.

To perform the actual identification, a threshold cut is placed on the photoelectron spectra of the HGC. The reasoning can be observed in Figure 3.4, at a set gas pressure and momentum only certain particle species may produce a signal. However, this does not imply a cut on the HGC will generate a clean separation of particles. For example, once again consulting Figure 3.4, attempting to separate between pions and kaons will result in a kaon and proton signal. Hence the need for additional particle identification detectors, threshold cuts between the various Cherenkovs and calorimeter may cleanly separate particle species. This methodology of particle identification is illustrated in Figure 8.3. By placing threshold cuts on both the heavy gas Cherenkov and

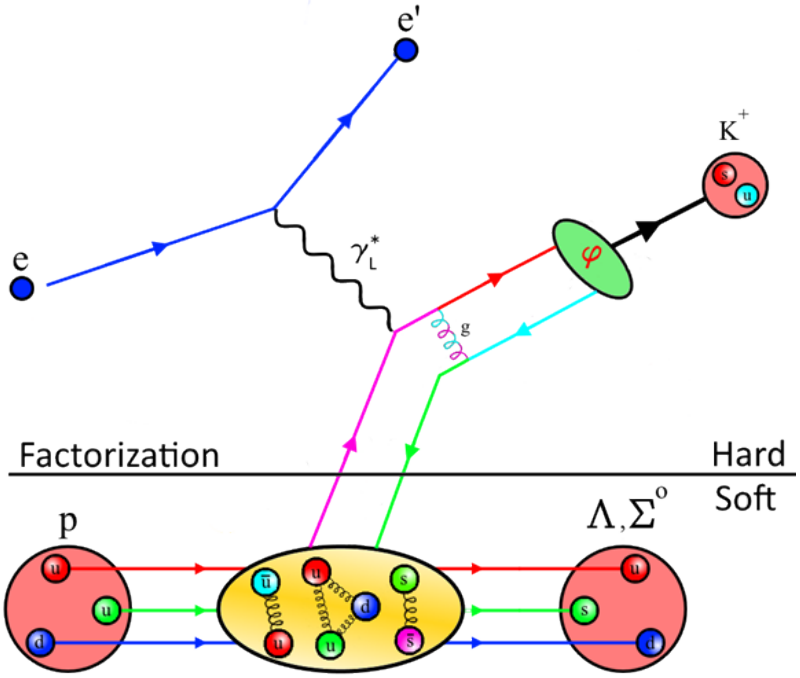


Figure 8.2: Visualization of factorization in perturbative quantum chromodynamics.

aerogel Cherenkov, events may be separated into pions, kaons, or protons according to whether they did or did not generate a signal.

A test of the particle identification capabilities of the heavy gas Cherenkov was performed with run 3423(hclog, 2018a), in which the Hall C DAQ was operating in coincident mode between the two spectrometer arms, the HMS and SHMS. The running experiment was semi-inclusive deep inelastic scattering (SIDIS), where a 10.6 GeV/c electron beam was incident on a liquid hydrogen target. The HMS was set to -5.27 GeV/c central momentum at 13.505 degrees, and the SHMS was set to +5.05 GeV/c central momentum at 11.995 degrees. Identical to our group's upcoming experiment, the p(e,e'K^+)Λ reaction was selected by specifying an electron in the HMS and a kaon in the SHMS. An identical cut as in Figure 8.3 was applied to the SHMS to select kaons, and a calorimeter cut was applied to the HMS to select electrons. The results of this analysis are shown in Figure 8.4 (Basnet, 2018). The blue histogram is the raw data, the red histogram are data after the particle selection, and

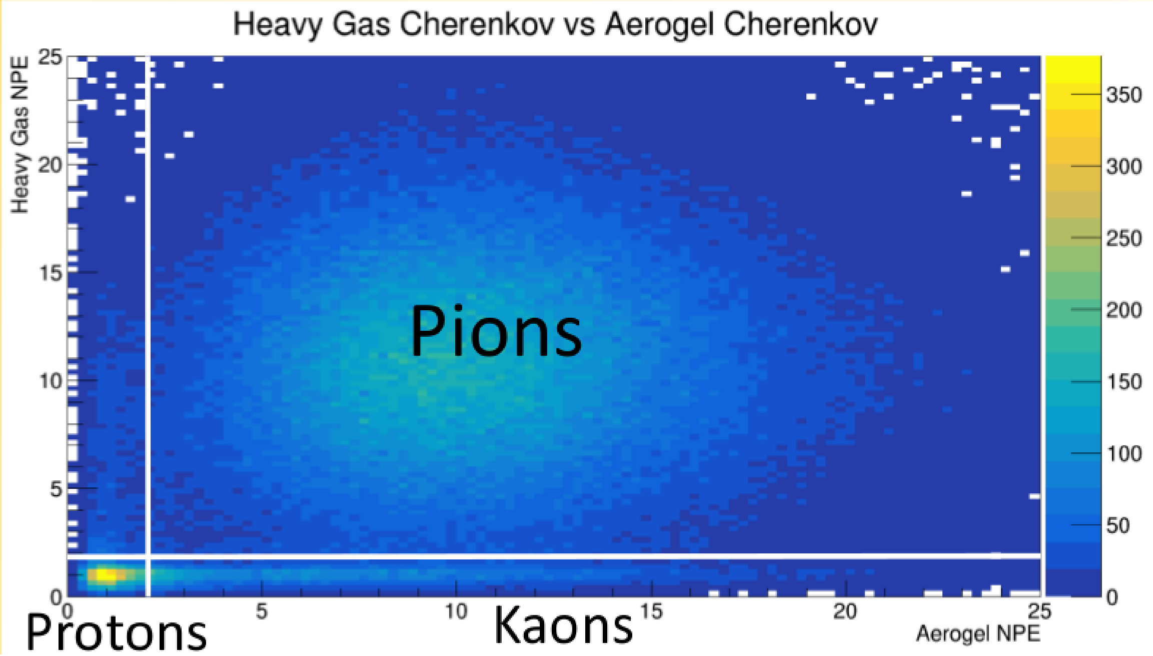


Figure 8.3: Number of photoelectrons (NPE) in the heavy gas Cherenkov against the aerogel Cherenkov.

the black histogram are the simulated results for the Λ. The vertical black lines represent the location of the reconstructed Λ and neutron masses. Clearly, the kaon separation ability of the HGC is demonstrated as the Λ peak can be isolated. The emergence of a neutron peak is indicative of pion leakage into the kaon selection. If pions are selected, the resulting reaction is $p(e, e'\pi^+)n$, explaining why the neutron mass is approximately reconstructed. The events filling the larger mass bins correspond to other semi-inclusive reactions filling the phase space, for example reactions like $p(e, e'K^+)\pi^0\Lambda$, $p(e, e'\pi^+)\pi^0 n$, and so on. As these are preliminary tests, they do not pose an issue, and in future analysis will be removed by stringent restrictions on kinematics.

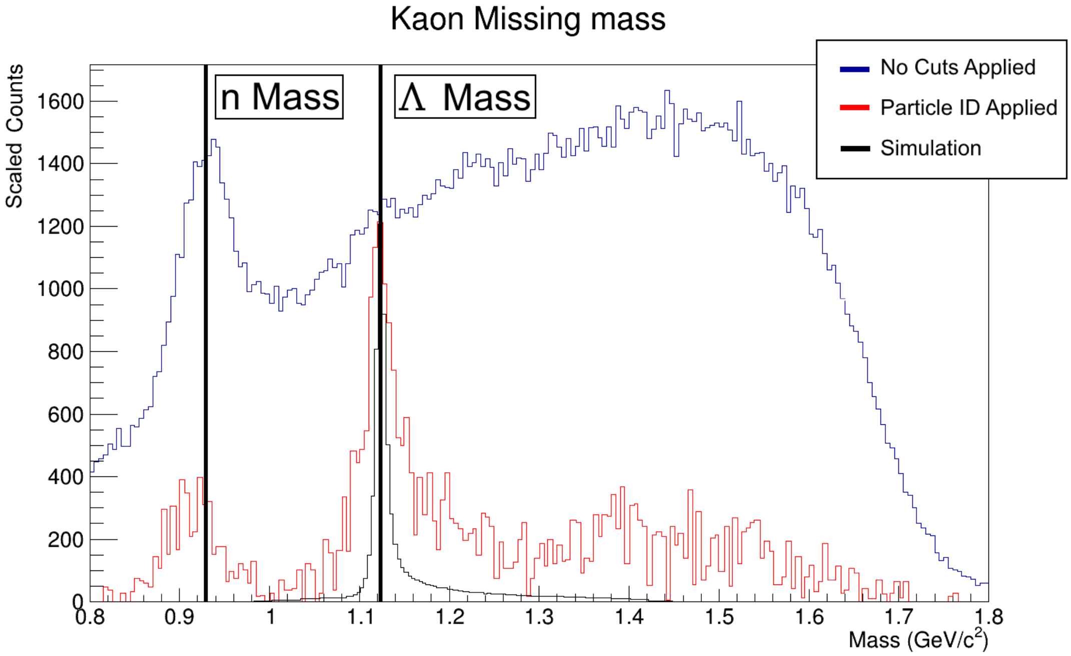


Figure 8.4: Results of applying particle identification from the heavy gas Cherenkov and aerogel Cherenkov on raw data. Blue represents original data, red the results after selecting for kaon in SHMS and electron in HMS, and black is the simulated result for the Λ. Indicated by black lines are the Λ and neutron mass. The experimental value for the neutron mass does not agree with the known value since the reconstructed mass is calculated using the K^+ mass, not the π^+ mass.

Chapter 9

Summary

The heavy gas Cherenkov detector was designed and built at the University of Regina and serves an important role in the particle identification capabilities of the Super High Momentum Spectrometer (SHMS) in Hall C of the Thomas Jefferson National Accelerator Facility. This detector can provide reliable π/K separation between 3-11 GeV/c central momenta, or e/π separation below 3 GeV/c. The Cherenkov media used is C_4F_8O or C_4F_{10} at a pressure of 1.00 - 0.95 atm between 3-7 GeV/c, and steadily decreased at larger momenta. Cherenkov light is collected by reflecting off four oblate ellipsoid mirrors onto a corresponding PMT.

For the first time, experimental data are available to gauge the performance of the HGC. Firstly, the calibration of the detector was determined, as efficiency measurements require a reliable calibration. To perform this measurement, the single photoelectron peak is isolated, as this yields a simple translation from ADC channel to number of photoelectrons. Due to the finite peak of Cherenkov radiation, the single photoelectron was isolated by examining the signal a PMT receives from a particle traversing different mirror quadrants. This yielded a clean spectrum of two or three photoelectrons, allowing the calibration to be determined as well as the linearity of the gain response. The absolute value of the gain was also measured, and found to be in good agreement with previous measurements.

With an accurate calibration, the efficiency of the detector was measured. This is determined from a simple equation: take the ratio of events identified by everything except the HGC under the events identified by everything including the HGC. This method shows that the HGC is 98% - 99% efficient in e/π separation for a photoelectron cut of 0.5. The efficiency was also found to have a positional dependence related to the mirror edges. However, the mirror edges are imperfect in construction and so this is expected. Additional regions of localized inefficiency were discovered whose source is uncertain. The optical configuration and mirror alignment have been adjusted to observe the effect on these regions.

The HGC has been fully simulated using the Geant4 software package. After comparisons with experimental data, it became apparent that the optical configuration had to be modified to better account for absorption of UV photons. Once a better description of the RTV 615 and optical grease was implemented, the simulation agreed with experimental data. Based on this, the simulation was again modified to test the effect of replacing the complicated optical configuration with a simpler one. Improvements to performance were observed, and so the optical configuration of the HGC will be changed.

To test the particle identification capabilities of the HGC in an experimental setting, the mass of the Λ hyperon was measured by isolating the reaction $p(e, e'K)\Lambda$. This is done by using the HGC and aerogel detector together, kaon events were taken as producing a signal in the aerogel while giving no signal in the HGC. With this cut the Λ mass was indeed reconstructed, clearly demonstrating the effectiveness of the HGC. This has particular import to our research group, as the upcoming L-T separated kaon electroproduction experiment requires very clean identification of kaons in the SHMS.

Thus, the HGC has been successfully commissioned and is currently in a state to provide π/K separation at high momentum as intended. Changes to the optical configuration have been made to improve performance, and the effect will be analyzed during the fall run of 2018. The HGC, and other new instruments available from the 12 GeV/c upgrade, make experiments such as the Kaon L-T separation experiment possible furthering our knowledge of the strong interaction.

Appendix A

Analysis Code

Listing A.1: README.md; Text File of Options for Scripts and How to Run Them

~~~
# Calibrating the Cherenkovs & Determining Efficiency
* To automatically run the calibration & efficiency
```
root -l run_cal.C
```
* To manually run the scripts:
* Link your ROOT file
* Run the script with options (listed after script is executed)
```
root -l ../../ROOTfiles/shms_replay_488_-1.root
T->Process("calibration.C+", "options");
T->Process("efficiencies.C+", "options");
```
* Or, if you want to run using PROOF
```
root -l
TChain ch("T");
ch.Add("/path/to/ROOTfile");
TProof::Open(""); \\For PROOF-lite, adjust accordingly for PROOF
ch.SetProof();
ch.Process("calibration.C+", "options");
```
### Options for the calibration script are (case/spelling important):
* **readall** - read all data from ROOT tree (very slow)
* **showall** - display all calibration details (lots of windows)
* **tracksfired** - alternate calibration method using the tracksfired leaf
* **cut** - use particle ID
* **pions** - use particle ID selecting pions

### Options for the efficiencies script are (case/spelling important):
* **showall** - display all calibration details (lots of windows)
* **Chercut** - use the other Cherenkov detector in particle ID
* **NGC** - calibrate the nobel gas Cherenkov
~~~

```
* **[0-9].[0-9]** - number of photoelectrons to cut on. If a cut on the other Cherenkov is desired, the second
    number entered will be taken as that Cherenkov's cut parameter.
```

Listing A.2: run_cal.C; Initialization Script to Grab File and Run Analysis Code

```
#include <TProof.h>
#include <iostream>
#include <fstream>
#include <string>
#include <stdio.h>

void run_cal(Int_t RunNumber = 0, Int_t NumEvents = 0, Int_t coin = 0) {
  if (RunNumber == 0) {
    cout << "Enter a Run Number (-1 to exit): ";
    cin >> RunNumber;
    if (RunNumber <= 0) return;
  }
  if (NumEvents == 0) {
    cout << "\nNumber of Events to analyze: ";
    cin >> NumEvents;
  }
  if (coin == 0) {
    cout << "\nIf this is a coincident run enter 1: ";
    cin >> coin;
  }

  cin.ignore(numeric_limits<streamsize>::max(), '\n');

  string calib_raw;
  cout << "\nEnter options for calibration  (enter NA to skip): ";
  getline(std::cin, calib_raw);
  TString calib_option = calib_raw;

  string eff_raw;
  cout << "\nEnter options for efficiency (enter NA to skip): ";
  getline(std::cin, eff_raw);
  TString eff_option = eff_raw;

  cout << "\n\n";

  TChain ch("T");
  if (coin == 1) ch.Add(Form("../../ROOTfiles/shms_coin_replay_production_all_%d_%d.root", RunNumber, NumEvents));
  else ch.Add(Form("../../ROOTfiles/shms_replay_production_all_%d_%d.root", RunNumber, NumEvents));
  TProof *proof = TProof::Open("workers=4");
  proof->SetProgressDialog(0);
  ch.SetProof();

  if (calib_option != "NA") {
    //Start calibration process
    ch.Process("calibration.C+",calib_option);

    cout << "\n\nUpdate calibration constants with the better estimate (y/n)? ";

    TString user_input;
    cin >> user_input;
    if (user_input == "y") {
      ifstream temp;
      temp.open("calibration_temp.txt", ios::in);
      if (temp.is_open()) {
        if (calib_option.Contains("NGC")) rename("calibration_temp.txt", Form("../../PARAM/SHMS/NGCER/CALIB/pngcer_calib_%d.param", RunNumber));

        else rename("calibration_temp.txt", Form("../../PARAM/SHMS/HGCER/CALIB/phgcer_calib_%d.param", RunNumber));
      }
      else cout << "Error opening calibration constants, may have to update constants manually!" << endl;
    }
    else {
      remove("calibration_temp.txt");
    }
  }
  if (eff_option != "NA") ch.Process("efficiencies.C+",eff_option);
}
```

Listing A.3: calibration.h; Header File for Calibration Code

```
//////////////////////////////////////////////////////////
// This class has been automatically generated on
// Tue Jul  4 10:23:34 2017 by ROOT version 5.34/13
// from TTree T/Hall A Analyzer Output DST
// found on file: shms_replay_488.root
//////////////////////////////////////////////////////////

#ifndef calibration_h
#define calibration_h

#include <TROOT.h>
#include <TChain.h>
#include <TFile.h>
#include <TSelector.h>
#include <TTreeReader.h>
#include <TTreeReaderValue.h>
#include <TTreeReaderArray.h>
#include <TH1.h>
#include <TH2.h>
#include <TNtuple.h>

const Int_t        fhgc_pmts = 4;
const Double_t     fhgc_zpos = 156.27;

// Header file for the classes stored in the TTree if any.

// Fixed size dimensions of array or collections stored in the TTree if any.

class calibration : public TSelector {
public :
   TTreeReader    fReader;
   TTree          *fChain = 0;   //!pointer to the analyzed TTree or TChain
   Bool_t          fFullRead;
   Bool_t          fFullShow;
   Bool_t          fTrack;
   Bool_t          fCut;
   Bool_t          fPions;

   // Declaration of histograms
   TH1F          **fPulseInt;
   TH1F         ***fPulseInt_quad;
   TH2F           *fCut_everything;
   TH1F           *fCut_enorm;
   TH2F           *fCut_electron;
   TH2F           *fCut_pion;
   TH1F           *fBeta_Cut;
   TH1F           *fBeta_Full;
   TH1F           *fTiming_Cut;
   TH1F           *fTiming_Full;

   // Declaration of histograms used in fitting/analysis
   TH1F *fscaled[4];
   TH1F *fscaled_nobackground[4];
   TH1F *fscaled_mk2[4];
```

```
  TH1F *fscaled_mk2_nobackground[4];
  TH1F *fscaled_temp[4];
  TH1F *fscaled_combined[4];
  TH1F *fscaled_total;
  TH1F *fscaled_temp_mk2[4];
  TH1F *fscaled_combined_mk2[4];
  TH1F *fscaled_total_mk2;

  // Declaration of canvases used in fitting/analysis
  TCanvas *quad_cuts_ipmt;
  TCanvas *low_stats_ipmt;
  TCanvas *background_ipmt;
  TCanvas *final_spectra_ipmt;
  TCanvas *background_mk2_ipmt;
  TCanvas *final_spectra_mk2_ipmt;
  TCanvas *final_spectra_combined;
  TCanvas *final_spectra_combined_mk2;
  TCanvas *scaled_poisson;
  TCanvas *scaled_total;

  // Declaration of preprocessing quantities
  Double_t  timing_mean[4];
  Double_t  timing_std[4];

  // Readers to access the data
  TTreeReaderValue<Int_t>    Ndata_P_tr_beta            = {fReader, "Ndata.P.tr.beta"};
  TTreeReaderArray<Double_t> P_tr_beta                  = {fReader, "P.tr.beta"};
  TTreeReaderArray<Double_t> P_hgcer_goodAdcTdcDiffTime = {fReader, "P.hgcer.goodAdcTdcDiffTime"};
  TTreeReaderArray<Double_t> P_hgcer_goodAdcPulseInt    = {fReader, "P.hgcer.goodAdcPulseInt"};
  TTreeReaderArray<Double_t> P_hgcer_goodAdcPulseAmp    = {fReader, "P.hgcer.goodAdcPulseAmp"};
  TTreeReaderArray<Double_t> P_hgcer_numTracksFired     = {fReader, "P.hgcer.numTracksFired"};
  TTreeReaderValue<Double_t> P_cal_fly_earray           = {fReader, "P.cal.fly.earray"};
  TTreeReaderValue<Double_t> P_cal_pr_eplane            = {fReader, "P.cal.pr.eplane"};
  TTreeReaderValue<Double_t> P_cal_etotnorm             = {fReader, "P.cal.etotnorm"};
  TTreeReaderArray<Double_t> P_gtr_dp                   = {fReader, "P.gtr.dp"};
  TTreeReaderArray<Double_t> P_tr_x                     = {fReader, "P.tr.x"};
  TTreeReaderArray<Double_t> P_tr_ph                    = {fReader, "P.tr.ph"};
  TTreeReaderArray<Double_t> P_tr_y                     = {fReader, "P.tr.y"};
  TTreeReaderArray<Double_t> P_tr_th                    = {fReader, "P.tr.th"};

calibration(TTree * /*tree*/ =0) : fChain(0) {fPulseInt = 0, fPulseInt_quad = 0, fCut_everything = 0, fCut_enorm =0, fCut_electron = 0, fCut_pion = 0, fBeta_Cut = 0, fBeta_Full = 0, fTiming_Cut = 0, fTiming_Full = 0, fFullRead = kFALSE, fFullShow = kFALSE, fTrack = kFALSE, fCut = kFALSE, fPions = kFALSE;}
  virtual ~calibration() { }
  virtual Int_t   Version() const { return 2; }
  virtual void    Begin(TTree *tree);
  virtual void    SlaveBegin(TTree *tree);
  virtual void    Init(TTree *tree);
  virtual Bool_t  Notify();
  virtual Bool_t  Process(Long64_t entry);
  virtual Int_t   GetEntry(Long64_t entry, Int_t getall = 0) { return fChain ? fChain->GetTree()->GetEntry(entry, getall) : 0; }
  virtual void    SetOption(const char *option) { fOption = option; }
  virtual void    SetObject(TObject *obj) { fObject = obj; }
  virtual void    SetInputList(TList *input) { fInput = input; }
  virtual TList  *GetOutputList() const { return fOutput; }
```

```
   virtual void    SlaveTerminate();
   virtual void    Terminate();

   ClassDef(calibration,0);
};

#endif

#ifdef calibration_cxx
void calibration::Init(TTree *tree)
{
   // The Init() function is called when the selector needs to initialize
   // a new tree or chain. Typically here the branch addresses and branch
   // pointers of the tree will be set.
   // It is normally not necessary to make changes to the generated
   // code, but the routine can be extended by the user if needed.
   // Init() will be called many times when running on PROOF
   // (once per file to be processed).

   // Set branch addresses and branch pointers

  fReader.SetTree(tree);

}

Bool_t calibration::Notify()
{
   // The Notify() function is called when a new file is opened. This
   // can be either for a new TTree in a TChain or when when a new TTree
   // is started when using PROOF. It is normally not necessary to make changes
   // to the generated code, but the routine can be extended by the
   // user if needed. The return value is currently not used.

   return kTRUE;
}


//Poisson distribution is used to remove background from larger values of NPE
Double_t poisson(Double_t *x, Double_t *par)
{
  Double_t result1 = (par[1]*pow(par[0],x[0])*exp(-par[0]))/(tgamma(x[0]+1));
  return result1;
}
//Gaussian distribution is used to find the mean of the SPE and determine spacing between subsequent NPE
Double_t gauss(Double_t *x, Double_t *par)
{
  Double_t result1 = par[0]*exp((-0.5)*(pow((x[0] - par[1]),2)/pow((par[2]),2)));
  Double_t result2 = par[3]*exp((-0.5)*(pow((x[0] - par[4]),2)/pow((par[5]),2)));
  Double_t result3 = par[6]*exp((-0.5)*(pow((x[0] - par[7]),2)/pow((par[8]),2)));
  Double_t result4 = par[9]*exp((-0.5)*(pow((x[0] - par[10]),2)/pow((par[11]),2)));
  Double_t result5 = par[12]*exp((-0.5)*(pow((x[0] - par[13]),2)/pow((par[14]),2)));
  return result1 + result2 + result3 + result4 + result5;
}

//A simple linear equation is used to determine how linear the means of the NPE are
Double_t linear(Double_t *x, Double_t *par)
```

```
{
  Double_t result1 = par[0]*x[0] + par[1];
  return result1;
}

#endif // #ifdef calibration_cxx
```

Listing A.4: calibration.C; Calibration Script

```cpp
#define calibration_cxx
// The class definition in calibration.h has been generated automatically
// by the ROOT utility TTree::MakeSelector(). This class is derived
// from the ROOT class TSelector. For more information on the TSelector
// framework see \$ROOTSYS/README/README.SELECTOR or the ROOT User Manual.

// The following methods are defined in this file:
//    Begin():        called every time a loop on the tree starts,
//                    a convenient place to create your histograms.
//    SlaveBegin():   called after Begin(), when on PROOF called only on the
//                    slave servers.
//    Process():      called for each event, in this function you decide what
//                    to read and fill your histograms.
//    SlaveTerminate: called at the end of the loop on the tree, when on PROOF
//                    called only on the slave servers.
//    Terminate():    called at the end of the loop on the tree,
//                    a convenient place to draw/fit your histograms.
//
// To use this file, try the following session on your Tree T:
//
// Root > T->Process("calibration.C+")
// Root > T->Process("calibration.C+","some options")
//


#include "calibration.h"
#include <TH1.h>
#include <TH2.h>
#include <TF1.h>
#include <TStyle.h>
#include <TCanvas.h>
#include <TPaveText.h>
#include <TSpectrum.h>
#include <TList.h>
#include <TPolyMarker.h>
#include <TGraphErrors.h>
#include <TMath.h>
#include <iostream>
#include <iomanip>
#include <fstream>

using namespace TMath;

void calibration::Begin(TTree * /*tree*/)
{
  // The Begin() function is called at the start of the query.
  // When running with PROOF Begin() is only called on the client.
  // The tree argument is deprecated (on PROOF 0 is passed).
  printf("\n\n");

  TString option = GetOption();
  TString report_option = option(0,option.Length()-79);
  Info("Begin", "Script will fail unless ’calibration.C+’ is used");
  Info("Begin", "Starting calibration process with option: %s", report_option.Data());
  Info("Begin", "To see details of calibration, use option showall");
  Info("Begin", "To calibrate using TrackFired leaf, use option trackfired");
  Info("Begin", "Default is no particle cut, use option cut if desired");
  Info("Begin", "Default particle ID is electrons, use option pions if desired");
  printf("\n\n");

  //Check option
  if (option.Contains("showall")) fFullShow = kTRUE;
  if (option.Contains("trackfired")) fTrack = kTRUE;
  if (option.Contains("pions") || option.Contains("pion")) fPions = kTRUE;
  if (option.Contains("cut") || fPions || option.Contains("cuts")) fCut = kTRUE;
}

void calibration::SlaveBegin(TTree * /*tree*/)
{
  // The SlaveBegin() function is called after the Begin() function.
  // When running with PROOF SlaveBegin() is called on each slave server.
  // The tree argument is deprecated (on PROOF 0 is passed).

  printf("\n\n");
  TString option = GetOption();

  //Check option
  if (option.Contains("showall")) fFullShow = kTRUE;
  if (option.Contains("trackfired")) fTrack = kTRUE;
  if (option.Contains("pions") || option.Contains("pion")) fPions = kTRUE;
  if (option.Contains("cut") || fPions || option.Contains("cuts")) fCut = kTRUE;

  Info("SlaveBegin", "’%s’ showing", (fFullShow ? "full" : "minimal"));
  Info("SlaveBegin", "’%s’ strategy", (fTrack ? "tracking" : "quadrant"));
  Info("SlaveBegin", "cuts %s performed", (fCut ? "are" : "are not"));
  if (fCut) Info("SlaveBegin", "cutting for ’%s’", (fPions ? "pions" : "electrons"));

  // Inintialize the histograms. Note they are binned per ADC channel which will be changed in the calibration analysis.
  Int_t ADC_min;
  Int_t ADC_max;
  Int_t bins;


  ADC_min = 0;
  ADC_max = 400;
  bins = 2*(abs(ADC_min) + abs(ADC_max));


  fPulseInt = new TH1F*[4];
  fPulseInt_quad = new TH1F**[4];
  fPulseInt_quad[0] = new TH1F*[4];
  fPulseInt_quad[1] = new TH1F*[4];
  fPulseInt_quad[2] = new TH1F*[4];
  fPulseInt_quad[3] = new TH1F*[4];
  for (Int_t ipmt = 0; ipmt < 4; ipmt++)
    {
      fPulseInt[ipmt] = new TH1F(Form("PulseInt_PMT%d",ipmt+1),Form("Pulse Integral PMT%d; ADC Channel (pC); Counts",ipmt+1), bins, ADC_min, ADC_max);
      GetOutputList()->Add(fPulseInt[ipmt]);
      for (Int_t iquad = 0; iquad < 4; iquad++)
        {
          fPulseInt_quad[iquad][ipmt] = new TH1F(Form("PulseInt_quad%d_PMT%d",iquad+1,ipmt+1),Form("Pulse Integral PMT%d quad%d; ADC Channel (pC); Counts",ipmt+1,iquad+1),bins,ADC_min,ADC_max);
          GetOutputList()->Add(fPulseInt_quad[iquad][ipmt]);
        }
    }

  //Timing and Beta cut visualizations
  fBeta_Cut = new TH1F("Beta_Cut", "Beta cut used for ’good’ hits;Beta;Counts", 100, -0.1, 1.5);
  GetOutputList()->Add(fBeta_Cut);

  fBeta_Full = new TH1F("Beta_Full", "Full beta for events;Beta;Counts", 100, -0.1, 1.5);
  GetOutputList()->Add(fBeta_Full);

  fTiming_Cut = new TH1F("Timing_Cut", "Timing cut used for ’good’ hits;Time (ns);Counts", 500, -100, 100);
  GetOutputList()->Add(fTiming_Cut);

  fTiming_Full = new TH1F("Timing_Full", "Full timing information for events;Time (ns);Counts", 500, -100, 100);
  GetOutputList()->Add(fTiming_Full);

  //Particle ID cut visualization
  fCut_everything = new TH2F("Cut_everything", "Visualization of no cuts; Calorimeter Energy (GeV); Pre-Shower Energy (GeV)", 250, 0, 1.0, 250, 0, 1.0);
  GetOutputList()->Add(fCut_everything);
  fCut_enorm = new TH1F("Cut_enorm", "Visualization of normalized energy cuts; Normalized Energy; Counts", 200, 0, 2.0);
  GetOutputList()->Add(fCut_enorm);
  fCut_electron = new TH2F("Cut_electron", "Visualization of electron cut; Calorimeter Energy (GeV); Pre-Shower Energy (GeV)", 250, 0, 1.0, 250, 0, 1.0);
  GetOutputList()->Add(fCut_electron);
  fCut_pion = new TH2F("Cut_pion", "Visualization of pion cut; Calorimeter Energy (GeV); Pre-Shower Energy (GeV)", 250, 0, 1.0, 250, 0, 1.0);
  GetOutputList()->Add(fCut_pion);

  printf("\n\n");
}

Bool_t calibration::Process(Long64_t entry)
{
  // The Process() function is called for each entry in the tree (or possibly
  // keyed object in the case of PROOF) to be processed. The entry argument
  // specifies which entry in the currently loaded tree is to be processed.
  // It can be passed to either calibration::GetEntry() or TBranch::GetEntry()
  // to read either all or the required parts of the data. When processing
  // keyed objects with PROOF, the object is already loaded and is available
  // via the fObject pointer.
  //
  // This function should contain the "body" of the analysis. It can contain
  // simple or elaborate selection criteria, run algorithms on the data
  // of the event and typically fill histograms.
  //
  // The processing can be stopped by calling Abort().
  //
  // Use fStatus to set the return value of TTree::Process().
  //
```

```
// The return value is currently not used.

fReader.SetEntry(entry);

//Output to verify script is working, and store the total number of events
if (entry % 100000 == 0) printf("Processing Entry number %lld\n",entry);

//Define quantities to loop over
Int_t fpmts;
fpmts = fhgc_pmts;   //Note HGC & NGC have the same # of PMTS

//Require only one good track reconstruction for the event
if (*Ndata_P_tr_beta != 1) return kTRUE;

//Redundant, but useful if multiple tracks are eventually allowed
for (Int_t itrack = 0; itrack < *Ndata_P_tr_beta; itrack++)
  {
    //Require loose cut on particle velocity
    fBeta_Full->Fill(P_tr_beta[itrack]);
    if (TMath::Abs(P_tr_beta[itrack] - 1.0) > 0.2) return kTRUE;
    fBeta_Cut->Fill(P_tr_beta[itrack]);

    //Filling the histograms
    for (Int_t ipmt = 0; ipmt < fpmts; ipmt++)
      {
        //Perform a loose timing cut
        fTiming_Full->Fill(P_hgcer_goodAdcTdcDiffTime[ipmt]);
        if (P_hgcer_goodAdcTdcDiffTime[ipmt] > 38.0 || P_hgcer_goodAdcTdcDiffTime[ipmt] < 30.0) continue;
        fTiming_Cut->Fill(P_hgcer_goodAdcTdcDiffTime[ipmt]);

        //Cuts to remove entries corresponding to a PMT not registering a hit
        if (P_hgcer_goodAdcPulseInt[ipmt] == 0.0) continue;

        //For quadrant cut strategy with particle ID cuts. In this case electrons are selected
        if (!fTrack && fCut && !fPions)
          {
            //Retrieve particle ID information
            Float_t central_p = 2.2;
            Float_t p = ((P_gtr_dp[0]/100.0)*central_p) + central_p;

            //Fill histogram visualizaing the electron selection
            fCut_everything->Fill(*P_cal_fly_earray/p, *P_cal_pr_eplane/p);
            fCut_enorm->Fill(*P_cal_etotnorm);

            //Cut on Shower vs preshower is a tilted ellipse, this requires an angle of rotation (in radians), x/y center, semimajor and semiminor axis
            //Float_t eangle = 3.0*3.14159/4.0;
            //Float_t ex_center = 0.66;
            //Float_t ey_center = 0.35;
            //Float_t esemimajor_axis = 0.28;
            //Float_t esemiminor_axis = 0.04;
            Float_t eangle = 3.0*3.14159/4.0;
            Float_t ex_center = 0.375;
            Float_t ey_center = 0.360;
            Float_t esemimajor_axis = 0.38;
            Float_t esemiminor_axis = 0.05;
            if (pow((*P_cal_fly_earray/p - ex_center)*cos(eangle) + (*P_cal_pr_eplane/p - ey_center)*sin(eangle),2)/pow(esemimajor_axis,2) +
                  pow((*P_cal_fly_earray/p - ex_center)*sin(eangle) - (*P_cal_pr_eplane/p - ey_center)*cos(eangle),2)/pow(esemiminor_axis,2) < 1
                /*P_cal_etotnorm > 0.4*/)
              {
                //Fill histogram visualizing the electron selection
                fCut_electron->Fill(*P_cal_fly_earray/p, *P_cal_pr_eplane/p);

                //Fill histogram of the full PulseInt spectra for each PMT
                fPulseInt[ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);

                //Fill histograms of what each PMT registers from each quadrant, this requires tracking the particle from the focal plane. Each quadrant is defined from the parameter files
                Float_t y_pos = P_tr_y[0] + P_tr_ph[0]*fhgc_zpos;
                Float_t x_pos = P_tr_x[0] + P_tr_th[0]*fhgc_zpos;

                //Condition for quadrant 1 mirror
                if (y_pos >= 4.6 && x_pos >= 9.4) fPulseInt_quad[0][ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);

                //Condition for quadrant 2 mirror
                if (y_pos < 4.6 && x_pos >= 9.4) fPulseInt_quad[1][ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);

                //Condition for quadrant 3 mirror
                if (y_pos >= 4.6 && x_pos < 9.4) fPulseInt_quad[2][ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);

                //Condition for quadrant 4 mirror
                if (y_pos < 4.6 && x_pos < 9.4) fPulseInt_quad[3][ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);
              }
          }//Marks end of electron selection condition


        //For quadrant cut strategy with particle ID cuts. In this case pions are selected
        if (!fTrack && fCut && fPions)
          {
            //Retrieve particle ID information
            Float_t central_p = 3.0;
            Float_t p = ((P_gtr_dp[0]/100.0)*central_p) + central_p;

            //Fill histogram visualizaing the pion selection
            fCut_everything->Fill(*P_cal_fly_earray/p, *P_cal_pr_eplane/p);
            fCut_enorm->Fill(*P_cal_etotnorm);

            //Cut on Shower vs preshower is a tilted ellipse, this requires an angle of rotation (in radians), x/y center, semimajor and semiminor axis
            Float_t piangle = 0.0;
            Float_t pix_center = 0.26;
            Float_t piy_center = 0.03;
            Float_t pisemimajor_axis = 0.1;
            Float_t pisemiminor_axis = 0.02;
            if (pow((*P_cal_fly_earray/p - pix_center)*cos(piangle) + (*P_cal_pr_eplane/p - piy_center)*sin(piangle),2)/pow(pisemimajor_axis,2) +
                pow((*P_cal_fly_earray/p - pix_center)*sin(piangle) - (*P_cal_pr_eplane/p - piy_center)*cos(piangle),2)/pow(pisemiminor_axis,2) < 1)
              {
                //Fill histogram visualizaing the pion selection
                fCut_pion->Fill(*P_cal_fly_earray/p, *P_cal_pr_eplane/p);

                //Fill histogram of the full PulseInt spectra for each PMT
                fPulseInt[ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);

                //Fill histograms of what each PMT registers from each quadrant, this requires tracking the particle from the focal plane. Each quadrant is defined from the parameter files
                Float_t y_pos = P_tr_y[0] + P_tr_ph[0]*fhgc_zpos;
                Float_t x_pos = P_tr_x[0] + P_tr_th[0]*fhgc_zpos;

                //Condition for quadrant 1 mirror
                if (y_pos >= 4.6 && x_pos >= 9.4) fPulseInt_quad[0][ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);

                //Condition for quadrant 2 mirror
                if (y_pos < 4.6 && x_pos >= 9.4) fPulseInt_quad[1][ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);

                //Condition for quadrant 3 mirror
                if (y_pos >= 4.6 && x_pos < 9.4) fPulseInt_quad[2][ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);

                //Condition for quadrant 4 mirror
                if (y_pos < 4.6 && x_pos < 9.4) fPulseInt_quad[3][ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);
              }
          }//Marks end of pion selection condition

        //For quadrant cut strategy with no particle ID cut
        if (!fTrack && !fCut)
          {
            //Fill histogram of the full PulseInt spectra for each PMT
            fPulseInt[ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);

            //Retrieve information for particle tracking from focal plane

            //Fill histograms of what each PMT registers from each quadrant, this requires tracking the particle from the focal plane. Each quadrant is defined from the parameter files
            Float_t y_pos = P_tr_y[0] + P_tr_ph[0]*fhgc_zpos;
            Float_t x_pos = P_tr_x[0] + P_tr_th[0]*fhgc_zpos;

            //Condition for quadrant 1 mirror
            if (y_pos >= 4.6 && x_pos >= 9.4) fPulseInt_quad[0][ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);

            //Condition for quadrant 2 mirror
            if (y_pos < 4.6 && x_pos >= 9.4) fPulseInt_quad[1][ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);

            //Condition for quadrant 3 mirror
            if (y_pos >= 4.6 && x_pos < 9.4) fPulseInt_quad[2][ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);

            //Condition for quadrant 4 mirror
            if (y_pos < 4.6 && x_pos < 9.4) fPulseInt_quad[3][ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);
          }//Marks end of no particle ID strategy

        //For TracksFired cut strategy with no particle ID cut
        if (fTrack && !fCut)
          {
            //Fill histogram of the full PulseInt spectra for each PMT
            fPulseInt[ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);

            //Fill histograms with TracksFired cut, note that quadrant cuts are included
            for (Int_t iregion = 0; iregion < 4; iregion++)
              {
                if (P_hgcer_numTracksFired[iregion] == (iregion + 1)) fPulseInt_quad[iregion][ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);
              }
          }//Marks end of tracksfired strategy with no particle ID

        //For TracksFired cut strategy selecting electrons
        if (fTrack && fCut && !fPions)
          {
            //Retrieve particle ID information
            Float_t central_p = 2.2;
            Float_t p = ((P_gtr_dp[0]/100.0)*central_p) + central_p;

            //Fill histogram visualizaing the electron selection
            fCut_everything->Fill(*P_cal_fly_earray/p, *P_cal_pr_eplane/p);

            //Cut on Shower vs preshower is a tilted ellipse, this requires an angle of rotation (in radians), x/y center, semimajor and semiminor axis
            Float_t eangle = 3.0*3.14159/4;
            Float_t ex_center = 0.375;
            Float_t ey_center = 0.360;
            Float_t esemimajor_axis = 0.38;
            Float_t esemiminor_axis = 0.05;
            if (pow((*P_cal_fly_earray/p - ex_center)*cos(eangle) + (*P_cal_pr_eplane/p - ey_center)*sin(eangle),2)/pow(esemimajor_axis,2) +
                pow((*P_cal_fly_earray/p - ex_center)*sin(eangle) - (*P_cal_pr_eplane/p - ey_center)*cos(eangle),2)/pow(esemiminor_axis,2) < 1)
              {
                //Fill histogram visualizing the electron selection
                fCut_electron->Fill(*P_cal_fly_earray/p, *P_cal_pr_eplane/p);

                //Fill histogram of the full PulseInt spectra for each PMT
                fPulseInt[ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);

                //Fill histograms with TracksFired cut, note that quadrant cuts are included so any off quadrant histograms will be empty
                for (Int_t iregion = 0; iregion < 4; iregion++)
                  {
                    if (P_hgcer_numTracksFired[iregion] != 0.0) fPulseInt_quad[iregion][ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);
                  }
              }
          }//Marks end of tracksfired with electrons

        //For TracksFired cut strategy selecting pions
        if (fTrack && fCut && fPions)
          {
            //Retrieve particle ID information
            Float_t central_p = 3.0;
            Float_t p = ((P_gtr_dp[0]/100.0)*central_p) + central_p;

            //Fill histogram visualizaing the electron selection
            fCut_everything->Fill(*P_cal_fly_earray/p, *P_cal_pr_eplane/p);

```

```
              //Cut on Shower vs preshower is a tilted ellipse, this requires an angle of rotation (in radians), x/y center, semimajor and semiminor axis
              Float_t piangle = 0.0;
              Float_t pix_center = 0.26;
              Float_t piy_center = 0.03;
              Float_t pisemimajor_axis = 0.1;
              Float_t pisemiminor_axis = 0.02;
              if (pow((*P_cal_fly_earray/p - pix_center)*cos(piangle) + (*P_cal_pr_eplane/p - piy_center)*sin(piangle),2)/pow(pisemimajor_axis,2) +
                  pow((*P_cal_fly_earray/p - pix_center)*sin(piangle) - (*P_cal_pr_eplane/p - piy_center)*cos(piangle),2)/pow(pisemiminor_axis,2) < 1)
                {
                  //Fill histogram visualizing the electron selection
                  fCut_pion->Fill(*P_cal_fly_earray/p, *P_cal_pr_eplane/p);

                  //Fill histogram of the full PulseInt spectra for each PMT
                  fPulseInt[ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);

                  //Fill histograms with TracksFired cut, note that quadrant cuts are included
                  for (Int_t iregion = 0; iregion < 4; iregion++)
                    {
                      if (P_hgcer_numTracksFired[iregion] != 0.0) fPulseInt_quad[iregion][ipmt]->Fill(P_hgcer_goodAdcPulseInt[ipmt]);
                    }
                }
            }//Marks end of tracksfired with electrons

        }//Marks end of loop over PMTs

    }//Marks end of loop over tracks

  return kTRUE;
}

void calibration::SlaveTerminate()
{
   // The SlaveTerminate() function is called after all entries or objects
   // have been processed. When running with PROOF SlaveTerminate() is called
   // on each slave server.
}

void calibration::Terminate()
{
  // The Terminate() function is the last function to be called during
  // a query. It always runs on the client, it can be used to present
  // the results graphically or save the results to file.

  printf("\n");
  Info("Terminate", "'%s' showing", (fFullShow ? "full" : "minimal"));
  Info("Terminate", "'%s' strategy", (fTrack ? "tracking" : "quadrant"));
  Info("Terminate", "cuts %s performed", (fCut ? "are" : "are not"));
  if (fCut) Info("Terminate", "cutting for '%s'", (fPions ? "pions" : "electrons"));
  printf("\n");
  Info("Terminate", "Histograms formed, now starting calibration.\n'Peak Buffer full' is a good warning!\n");
  printf("\n");

```

```
gStyle->SetOptStat(1000000001);

//Have to extract the histograms from the OutputList
TH1F* PulseInt[4];
TH1F* PulseInt_quad[4][4];
for (Int_t ipmt = 0; ipmt < 4; ipmt++)
  {
    PulseInt[ipmt] = dynamic_cast<TH1F*> (GetOutputList()->FindObject(Form("PulseInt_PMT%d",ipmt+1)));
    for (Int_t iquad = 0; iquad < 4; iquad++)
      {
        PulseInt_quad[iquad][ipmt] = dynamic_cast<TH1F*> (GetOutputList()->FindObject(Form("PulseInt_quad%d_PMT%d",iquad+1,ipmt+1)));
      }
  }

//Rebin the histograms, add functionality to bin HGC & NGC independently
/*if (fTrack) {
  for (Int_t ipmt=0; ipmt < (fhgc_pmts); ipmt++)
    {
      for (Int_t iquad=0; iquad<4; iquad++)
        {
          PulseInt_quad[iquad][ipmt]->Rebin(4);
        }
      PulseInt[ipmt]->Rebin(4);
    }
    }*/

//Canvases to display cut information
if (fFullShow)
  {
    //Canvas to show beta cut information
    TCanvas *Beta;
    Beta = new TCanvas("Beta", "Beta information for events");
    Beta->Divide(2,1);
    Beta->cd(1);
    fBeta_Full->Draw();
    Beta->cd(2);
    fBeta_Cut->Draw();

    //Canvas to show timing cut information
    TCanvas *Timing;
    Timing = new TCanvas("Timing", "Timing information for events");
    Timing->Divide(2,1);
    Timing->cd(1);
    fTiming_Full->Draw();
    Timing->cd(2);

    fTiming_Cut->Draw();
 }

//Show the particle cuts performed in the histogram forming
if (fCut)
  {
    TCanvas *cut_enorm = new TCanvas("cut_enorm", "Visualization of etotnorm");
    fCut_enorm->Draw();

    TCanvas *cut_visualization = new TCanvas("cut_visualization", "Visualization of the particle ID cuts performed");
    cut_visualization->Divide(2,1);
    cut_visualization->cd(1);
    fCut_everything->Draw("Colz");
    cut_visualization->cd(2);
    fPions ? fCut_pion->Draw("Colz") : fCut_electron->Draw("Colz");
  }

gStyle->SetOptFit(111);

//Single Gaussian to find mean of SPE
TF1 *Gauss1 = new TF1("Gauss1",gauss,-10,200,3);
Gauss1->SetParNames("Amplitude","Mean","Std. Dev.");

//Sum of two Gaussians to determine SPE with minimal systematics
TF1 *Gauss2 = new TF1("Gauss2",gauss,2,10,6);
Gauss2->SetParNames("Amplitude 1","Mean 1","Std. Dev. 1","Amplitude 2","Mean 2","Std. Dev. 2");

//Sum of three Gaussians to determine NPE spacing
TF1 *Gauss3 = new TF1("Gauss3",gauss,0.5,3.5,9);
Gauss3->SetParNames("Amplitude 1","Mean 1","Std. Dev. 1","Amplitude 2","Mean 2","Std. Dev. 2","Amplitude 3","Mean 3","Std. Dev. 3");

//Poisson distribution to remove high NPE background
TF1 *Poisson = new TF1("Poisson",poisson,0,20,2);
Poisson->SetParNames("Mean", "Amplitude");

//Note about Poisson background, the mean varies between detectors/operating conditions so this quantity may require user input
Double_t Poisson_mean = 0;
Poisson_mean = 5.5;

//Linear function used to determine goodness-of-fit for NPE spacing
TF1 *Linear = new TF1("Linear",linear,0,4,2);
Linear->SetParNames("Slope", "Intercept");

//An array is used to store the means for the SPE, and to determine NPE spacing
Double_t mean[3];
Double_t x_npe[3], y_npe[3], x_err[3], y_err[3];

//Two more arrays are used to store the estimates for the calibration constants and another two to store goodness of calibration
Double_t calibration_mk1[4], calibration_mk2[4], pmt_calib[4], pmt_calib_mk2[4];

//Array to hold the Poisson character of the calibrations
Double_t Pois_Chi[2];
Pois_Chi[0] = 0.0, Pois_Chi[1] = 0.0;

gStyle->SetOptFit(111);

//Main loop for calibration
for (Int_t ipmt=0; ipmt < (fhgc_pmts); ipmt++)
  {
    //Initialize the various arrays (calibration arrays are explicitly filled)
    for (Int_t i=0; i<3; i++)
```

```
      {
        mean[i] = 0.0;
        x_npe[i] = 0, y_npe[i] = 0, x_err[i] = 0, y_err[i] = 0;
      }

    //Begin strategy for quadrant cut calibration
    if (!fTrack)
      {
        //TSpectrum class is used to find the SPE peak using the search method
        TSpectrum *s = new TSpectrum(2);

        //Create Canvas to see the search result for the SPE
        if (fFullShow) quad_cuts_ipmt = new TCanvas(Form("quad_cuts_%d",ipmt), Form("First Photoelectron peaks PMT%d",ipmt+1));
        if (fFullShow) quad_cuts_ipmt->Divide(3,1);

        Int_t ipad = 1; //Variable to draw over pads correctly
        for (Int_t iquad=0; iquad<4; iquad++)
          {
            if (iquad == ipmt) continue; //ignore a PMT looking at its own quadrant
            if (fFullShow) quad_cuts_ipmt->cd(ipad);

            if (PulseInt_quad[iquad][ipmt]->GetEntries() > 0)
              {
                //Perform search for the SPE and save the peak into the array xpeaks
                fFullShow ? s->Search(PulseInt_quad[iquad][ipmt], 2.5, "nobackground", 0.001) : s->Search(PulseInt_quad[iquad][ipmt], 2.5, "nobackground&&nodraw", 0.001);
                TList *functions = PulseInt_quad[iquad][ipmt]->GetListOfFunctions();
                TPolyMarker *pm = (TPolyMarker*)functions->FindObject("TPolyMarker");
                Double_t *xpeaks = pm->GetX();
                if (xpeaks[1] < xpeaks[0]) xpeaks[1] = xpeaks[0];

                //Use the peak to fit the SPE with a Gaussian to determine the mean
                Gauss2->SetRange(xpeaks[0]-3, xpeaks[0]+10);
                Gauss2->SetParameter(1, xpeaks[0]);
                Gauss2->SetParameter(2, 10.);
                Gauss2->SetParameter(4, xpeaks[1]);
                Gauss2->SetParameter(5, 10.);
                Gauss2->SetParLimits(0, 0., 2000.);
                Gauss2->SetParLimits(1, xpeaks[0]-3, xpeaks[0]+3);
                Gauss2->SetParLimits(2, 0.5, 10.);
                Gauss2->SetParLimits(3, 0., 2000.);
                Gauss2->SetParLimits(4, xpeaks[1]-3, xpeaks[1]+3);
                Gauss2->SetParLimits(5, 0.5, 10.);
                fFullShow ? PulseInt_quad[iquad][ipmt]->Fit("Gauss2","RQ") : PulseInt_quad[iquad][ipmt]->Fit("Gauss2","RQN");
                //if (fFullShow) PulseInt_quad[iquad][ipmt]->GetXaxis()->SetRangeUser(0,20);

                //Store the mean of the SPE in the mean array provided it is not zero and passes a loose statistical cut. Note that indexing by ipad-1 is for convienience
                //cout << xpeaks[0] << "   " << PulseInt_quad[iquad][ipmt]->GetBinContent(PulseInt_quad[iquad][ipmt]->GetXaxis()->FindBin(xpeaks[0])) << endl;
                if (xpeaks[0] > 2.0 && PulseInt_quad[iquad][ipmt]->GetBinContent(PulseInt_quad[iquad][ipmt]->GetXaxis()->FindBin(xpeaks[0])) > 90) mean[ipad-1] = Gauss2->GetParameter(1);
                ipad++;
              }
      }

    //Obtain the conversion from ADC to NPE by taking the average of the SPE means
    Double_t xscale = 0.0;
    Double_t num_peaks = 0.0;
    for (Int_t i=0; i<3; i++)
      {
        if (mean[i] == 0.0) continue;
        xscale += mean[i];
        num_peaks += 1.0;
      }
    if (num_peaks != 0.0) xscale = xscale/num_peaks;

    //Perform check if the statistics were too low to get a good estimate of the SPE mean
    if (xscale < 1.0)
      {
        //Repeat the exact same procedure for the SPE of each quadrant, except now its for the full PMT spectra
        if(fFullShow) low_stats_ipmt = new TCanvas(Form("low_stats_%d",ipmt),Form("Low stats analysis for PMT%d",ipmt+1));
        if(fFullShow) low_stats_ipmt->cd(1);
        PulseInt[ipmt]->GetXaxis()->SetRangeUser(0,20);
        fFullShow ? s->Search(PulseInt[ipmt], 3.5, "nobackground", 0.001) : s->Search(PulseInt[ipmt], 2.0, "nobackground&&nodraw", 0.001);
        TList *functions = PulseInt[ipmt]->GetListOfFunctions();
        TPolyMarker *pm = (TPolyMarker*)functions->FindObject("TPolyMarker");
        Double_t *xpeaks = pm->GetX();
        Gauss1->SetRange(xpeaks[0]-3, xpeaks[0]+3);
        Gauss1->SetParameter(1, xpeaks[0]);
        Gauss1->SetParameter(2, 10.);
        Gauss1->SetParLimits(0, 0., 10000.);
        Gauss1->SetParLimits(1, xpeaks[0]-3, xpeaks[0]+3);
        Gauss1->SetParLimits(2, 0.5, 20.);
        PulseInt[ipmt]->GetXaxis()->SetRangeUser(0,200);
        fFullShow ? PulseInt[ipmt]->Fit("Gauss1","RQ") : PulseInt[ipmt]->Fit("Gauss1","RQN");
        xscale = Gauss1->GetParameter(1);
      }
    //Scale full ADC spectra according to the mean of the SPE. This requires filling a new histogram with the same number of bins but scaled min/max
    Int_t nbins;
    nbins = (PulseInt[ipmt]->GetXaxis()->GetNbins());

    //With the scale of ADC to NPE create a histogram that has the conversion applied
    fscaled[ipmt] = new TH1F(Form("fscaled_PMT%d", ipmt+1), Form("Scaled ADC spectra for PMT%d; NPE; Normalized Counts",ipmt+1), 200, 0, 30);

    //Fill this histogram bin by bin
    for (Int_t ibin=0; ibin<nbins; ibin++)
      {
        Double_t y = PulseInt[ipmt]->GetBinContent(ibin);
        Double_t x = PulseInt[ipmt]->GetXaxis()->GetBinCenter(ibin);
        Double_t x_scaled = x/xscale;
        Int_t bin_scaled = fscaled[ipmt]->GetXaxis()->FindBin(x_scaled);
        fscaled[ipmt]->SetBinContent(bin_scaled,y);
      }

```

```
        //Normalize the histogram for ease of fitting
        fscaled[ipmt]->Scale(1.0/fscaled[ipmt]->Integral(), "width");

        //Begin the removal of the Poisson-like background
        if (fFullShow) background_ipmt = new TCanvas(Form("backgrounf_pmt%d",ipmt), Form("NPE spectra for PMT%d with Poisson-like background",ipmt+1));
        if (fFullShow) background_ipmt->cd(1);
        Poisson->SetParameter(0, 23.0);
        Poisson->SetParameter(1, 0.25);
        Poisson->SetParLimits(0, 18.0, 30.0);
        fFullShow ? fscaled[ipmt]->Fit("Poisson","RQ") : fscaled[ipmt]->Fit("Poisson","RQN");

        //Make and fill histogram with the background removed
        fscaled_nobackground[ipmt] = new TH1F(Form("fscaled_nobackground_pmt%d", ipmt+1), Form("NPE spectra background removed for PMT%d; NPE; Normalized Counts",ipmt+1), 200, 0, 30);

        for (Int_t ibin=1; ibin<nbins; ibin++)
          {
            Double_t y = Poisson->Eval(fscaled[ipmt]->GetXaxis()->GetBinCenter(ibin));
            Double_t y2 = fscaled[ipmt]->GetBinContent(ibin) - y;
            //if (ipmt == 1) cout << "Poisson Value: " << y << "\nfscaled Value: " << fscaled[ipmt]->GetBinContent(ibin) << endl;
            fscaled_nobackground[ipmt]->SetBinContent(ibin,y2);
          }

        if (fFullShow) final_spectra_ipmt = new TCanvas(Form("final_Spectra_%d",ipmt), Form("NPE spectra Background Removed for PMT%d",ipmt+1));
        if (fFullShow) final_spectra_ipmt->Divide(2,1);
        if (fFullShow) final_spectra_ipmt->cd(1);
        Gauss3->SetParameters(0.08, 1.0, 0.22, 0.029, 2, 0.5, 0.15, 3, 0.26);
        Gauss3->SetParLimits(1, 0.5, 1.5);
        Gauss3->SetParLimits(2, 0.0, 1.0);
        Gauss3->SetParLimits(3, 0.0, 1.0);
        Gauss3->SetParLimits(4, 1.5, 2.5);
        Gauss3->SetParLimits(5, 0.2, 0.6);
        Gauss3->SetParLimits(6, 0.0, 1.0);
        Gauss3->SetParLimits(7, 2.5, 3.5);
        Gauss3->SetParLimits(8, 0.2, 0.5);
        fFullShow ? fscaled_nobackground[ipmt]->Fit("Gauss3","RQ") : fscaled_nobackground[ipmt]->Fit("Gauss3","RQN");
        if (fFullShow) fscaled_nobackground[ipmt]->GetXaxis()->SetRangeUser(0,5);
        if (fFullShow) fscaled_nobackground[ipmt]->GetYaxis()->SetRangeUser(0,0.3);

        //Create a TGraphErrors to determine the spacing of the NPE
        y_npe[0] = Gauss3->GetParameter(1), y_npe[1] = Gauss3->GetParameter(4), y_npe[2] = Gauss3->GetParameter(7);
        y_err[0] = Gauss3->GetParError(1), y_err[1] = Gauss3->GetParError(4), y_err[2] = Gauss3->GetParError(7);
        x_npe[0] = 1, x_npe[1] = 2, x_npe[2] = 3;
        TGraphErrors *gr_npe = new TGraphErrors(3, x_npe, y_npe, x_err, y_err);
        gr_npe->SetTitle(Form("Linear Spacing of PE for PMT%d",ipmt+1));
        gr_npe->GetXaxis()->SetTitle("NPE number");
        gr_npe->GetYaxis()->SetTitle("Photoelectron peak (NPE)");

        //Plot this graph with the NPE spectra
        if (fFullShow) final_spectra_ipmt->cd(2);
```

```
      Linear->SetParameters(1.0, 0.0);
      fFullShow ? gr_npe->Fit("Linear","RQ") : gr_npe->Fit("Linear","RQN");
      if (fFullShow) gr_npe->Draw("A*");
      calibration_mk1[ipmt] = xscale;
      pmt_calib[ipmt] = abs(1.0 - Gauss3->GetParameter(1));

      //Initial calibration constant has been obtained. Now I multiply it by the slope of the spacing of the NPE (should be approx. 1) for a second estimate

      Double_t xscale_mk2 = xscale * Gauss3->GetParameter(1);

      //Take this new xscale and repeat the exact same procedure as before
      fscaled_mk2[ipmt] = new TH1F(Form("fhgc_scaled_mk2_PMT%d", ipmt+1), Form("Scaled ADC spectra for PMT%d; NPE; Normalized Counts",ipmt+1), 200, 0, 30);

      //Fill this histogram bin by bin
      for (Int_t ibin=0; ibin<nbins; ibin++)
        {
          Double_t y = PulseInt[ipmt]->GetBinContent(ibin);
          Double_t x = PulseInt[ipmt]->GetXaxis()->GetBinCenter(ibin);
          Double_t x_scaled = x/xscale_mk2;
          Int_t bin_scaled = fscaled_mk2[ipmt]->GetXaxis()->FindBin(x_scaled);
          fscaled_mk2[ipmt]->SetBinContent(bin_scaled,y);
        }

      //Normalize the histogram for ease of fitting
      fscaled_mk2[ipmt]->Scale(1.0/fscaled_mk2[ipmt]->Integral(), "width");

      //Begin the removal of the Poisson-like background
      if (fFullShow) background_mk2_ipmt = new TCanvas(Form("background_mk2_pmt%d",ipmt), Form("NPE spectra for PMT%d with Poisson-like background",ipmt+1));
      if (fFullShow) background_mk2_ipmt->cd(1);
      Poisson->SetParameter(0, 23.0);
      Poisson->SetParameter(1, 0.25);
      Poisson->SetParLimits(0, 19.0, 30.0);
      fFullShow ? fscaled_mk2[ipmt]->Fit("Poisson","RQ"):fscaled_mk2[ipmt]->Fit("Poisson","RQN");

      //Make and fill histogram with the background removed
      fscaled_mk2_nobackground[ipmt] = new TH1F(Form("fscaled_mk2_nobackground_pmt%d", ipmt+1), Form("NPE spectra background removed for PMT%d; NPE; Normalized Counts",ipmt+1), 200, 0, 30);

      for (Int_t ibin=0; ibin<nbins; ibin++)
        {
          Double_t y = Poisson->Eval(fscaled_mk2[ipmt]->GetXaxis()->GetBinCenter(ibin));
          Double_t y2 = fscaled_mk2[ipmt]->GetBinContent(ibin) - y;
          fscaled_mk2_nobackground[ipmt]->SetBinContent(ibin,y2);
        }

      if (fFullShow) final_spectra_mk2_ipmt = new TCanvas(Form("final_Spectra_mk2_%d",ipmt), Form("NPE spectra Background Removed for PMT%d",ipmt+1));
      if (fFullShow) final_spectra_mk2_ipmt->Divide(2,1);
      if (fFullShow) final_spectra_mk2_ipmt->cd(1);
      Gauss3->SetParameters(0.08, 1.0, 0.22, 0.029, 2, 0.5, 0.15, 3, 0.26);
      Gauss3->SetParLimits(1, 0.5, 1.5);
      Gauss3->SetParLimits(2, 0.0, 1.0);
      Gauss3->SetParLimits(3, 0.0, 1.0);
      Gauss3->SetParLimits(4, 1.5, 2.5);
      Gauss3->SetParLimits(5, 0.2, 0.6);
      Gauss3->SetParLimits(6, 0.0, 1.0);
      Gauss3->SetParLimits(7, 2.5, 3.5);
      Gauss3->SetParLimits(8, 0.2, 0.5);
      fFullShow ? fscaled_mk2_nobackground[ipmt]->Fit("Gauss3","RQ") : fscaled_mk2_nobackground[ipmt]->Fit("Gauss3","RQN");
      if (fFullShow) fscaled_mk2_nobackground[ipmt]->GetXaxis()->SetRangeUser(0,5);
      if (fFullShow) fscaled_mk2_nobackground[ipmt]->GetYaxis()->SetRangeUser(0,0.2);

      //Create a TGraphErrors to determine the spacing of the NPE
      y_npe[0] = Gauss3->GetParameter(1), y_npe[1] = Gauss3->GetParameter(4), y_npe[2] = Gauss3->GetParameter(7);
      y_err[0] = Gauss3->GetParError(1), y_err[1] = Gauss3->GetParError(4), y_err[2] = Gauss3->GetParError(7);
      x_npe[0] = 1, x_npe[1] = 2, x_npe[2] = 3;
      TGraphErrors *gr_npe_mk2 = new TGraphErrors(3, x_npe, y_npe, x_err, y_err);
      gr_npe_mk2->GetXaxis()->SetTitle("NPE number");
      gr_npe_mk2->GetYaxis()->SetTitle("Photoelectron peak (NPE)");

      //Plot this graph with the NPE spectra
      if (fFullShow) final_spectra_mk2_ipmt->cd(2);
      Linear->SetParameters(1.0, 0.0);
      fFullShow ? gr_npe_mk2->Fit("Linear","RQ") : gr_npe_mk2->Fit("Linear","RQN");
      if (fFullShow) gr_npe_mk2->Draw("A*");
      calibration_mk2[ipmt] = xscale_mk2;
      pmt_calib_mk2[ipmt] = abs(1.0 - Gauss3->GetParameter(1));
    } // This brance marks the end of the quadrant cut strategy




  //Begin the TrackFired cut calibration
  if (fTrack)
    {
      //TSpectrum class is used to find the SPE peak using the search method
      TSpectrum *s = new TSpectrum(2);

      //Create Canvas to show the search result for the SPE
      if (fFullShow) quad_cuts_ipmt = new TCanvas(Form("quad_cuts_%d",ipmt), Form("First Photoelectron peaks PMT%d",ipmt+1));
      if (fFullShow) quad_cuts_ipmt->Divide(2,2);
      //if (fFullShow) quad_cuts_ipmt->cd(1);

      for (Int_t iquad = 0; iquad < 4; iquad++)
        {
          //Perform search for the SPE and save the peak into the array xpeaks
          if (fFullShow) quad_cuts_ipmt->cd(iquad+1);

          PulseInt_quad[ipmt][ipmt]->GetXaxis()->SetRangeUser(0,30);
          fFullShow ? s->Search(PulseInt_quad[iquad][ipmt], 1.0, "nobackground", 0.001) : s->Search(PulseInt_quad[iquad][ipmt], 1.5, "nobackground&&nodraw", 0.001);
          TList *functions = PulseInt_quad[iquad][ipmt]->GetListOfFunctions();
          TPolyMarker *pm = (TPolyMarker*)functions->FindObject("TPolyMarker");
          Double_t *xpeaks = pm->GetX();
          PulseInt_quad[iquad][ipmt]->GetXaxis()->SetRangeUser(-1,200);

          //Use the peak to fit the SPE with a Gaussian to determine the mean
          Gauss1->SetRange(xpeaks[0]-3, xpeaks[0]+3);
          Gauss1->SetParameter(1, xpeaks[0]);
          Gauss1->SetParameter(2, 10.);
          Gauss1->SetParLimits(0, 0., 2000.);
          Gauss1->SetParLimits(1, xpeaks[0]-3, xpeaks[0]+3);
          Gauss1->SetParLimits(2, 0.5, 10.);
          fFullShow ? PulseInt_quad[iquad][ipmt]->Fit("Gauss1","RQ") : PulseInt_quad[iquad][ipmt]->Fit("Gauss1","RQN");

          //Store the mean of the SPE in the mean array provided it is not zero, passes a loose statistical cut, and is above a minimum channel number
          if (xpeaks[0] != 0.0 && PulseInt_quad[iquad][ipmt]->GetBinContent(PulseInt_quad[iquad][ipmt]->GetXaxis()->FindBin(xpeaks[0])) > 10 && ipmt != iquad) mean[iquad] = Gauss1->GetParameter(1);
        }

      Double_t xscale = 0.0;
      Double_t num_peaks = 0.0;
      for (Int_t i = 0; i < 3; i++)
        {
          if (mean[i] == 0.0) continue;
          xscale += mean[i];
          num_peaks += 1.0;
        }

      calibration_mk1[ipmt] = xscale/num_peaks;

      //Scale full ADC spectra according to the mean of the SPE. This requires filling a new histogram with the same number of bins but scaled min/max
      Int_t nbins;
      nbins = PulseInt_quad[ipmt][ipmt]->GetXaxis()->GetNbins();

      fscaled[ipmt] = new TH1F(Form("fscaled_PMT%d", ipmt+1), Form("Scaled ADC spectra for PMT%d; NPE; Normalized Counts",ipmt+1), 210, -1, 20);

      //Fill this histogram bin by bin
      for (Int_t ibin=0; ibin<nbins; ibin++)
        {
          Double_t y = PulseInt_quad[ipmt][ipmt]->GetBinContent(ibin);
          Double_t x = PulseInt_quad[ipmt][ipmt]->GetXaxis()->GetBinCenter(ibin);
          Double_t x_scaled = x/calibration_mk1[ipmt];
          Int_t bin_scaled = fscaled[ipmt]->GetXaxis()->FindBin(x_scaled);
          fscaled[ipmt]->SetBinContent(bin_scaled,y);
        }

      //Normalize the histogram for ease of fitting
      fscaled[ipmt]->Scale(1.0/fscaled[ipmt]->Integral(), "width");

      if (fFullShow) final_spectra_ipmt = new TCanvas(Form("final_Spectra_%d",ipmt), Form("Calibrated spectra for PMT%d; NPE; Normalized Counts",ipmt+1));
      if (fFullShow) final_spectra_ipmt->cd(1);

      //Find the location of the SPE and subtract from 1.0 to determine accuracy of calibration
      Gauss1->SetRange(0.50, 1.50);
      Gauss1->SetParameter(0, 0.05);
```

```
        Gauss1->SetParameter(1, 1.0);
        Gauss1->SetParameter(2, 0.3);
        Gauss1->SetParLimits(0, 0.0, 1.0);
        Gauss1->SetParLimits(1, 0.5, 1.5);
        Gauss1->SetParLimits(2, 0.1, 0.5);
        fFullShow ? fscaled[ipmt]->Fit("Gauss1","RQ") : fscaled[ipmt]->Fit("Gauss1","RQN");

        calibration_mk2[ipmt] = calibration_mk1[ipmt]*Gauss1->GetParameter(1);
        pmt_calib[ipmt] = abs(1.0 - Gauss1->GetParameter(1));

        //Scale full ADC spectra according to the mean of the SPE. This requires filling a new histogram with the same number of bins but scaled min/max
        fscaled_mk2[ipmt] = new TH1F(Form("fscaled_mk2_PMT%d", ipmt+1), Form("Scaled ADC spectra for PMT%d; NPE; Normalized Counts",ipmt+1), 210, -1, 20);

        //Fill this histogram bin by bin
        for (Int_t ibin=0; ibin<nbins; ibin++)
          {
            Double_t y = PulseInt_quad[ipmt][ipmt]->GetBinContent(ibin);
            Double_t x = PulseInt_quad[ipmt][ipmt]->GetXaxis()->GetBinCenter(ibin);
            Double_t x_scaled = x/calibration_mk2[ipmt];
            Int_t bin_scaled = fscaled_mk2[ipmt]->GetXaxis()->FindBin(x_scaled);
            fscaled_mk2[ipmt]->SetBinContent(bin_scaled,y);
          }

        //Normalize the histogram for ease of fitting
        fscaled_mk2[ipmt]->Scale(1.0/fscaled_mk2[ipmt]->Integral(), "width");

        if (fFullShow) final_spectra_mk2_ipmt = new TCanvas(Form("final_Spectra_mk2_%d",ipmt), Form("Calibrated spectra for PMT%d; NPE; Normalized Counts",ipmt+1));
        if (fFullShow) final_spectra_mk2_ipmt->cd(1);

        //Find the location of the SPE and subtract from 1.0 to determine accuracy of calibration
        Gauss1->SetRange(0.50, 1.50);
        Gauss1->SetParameter(0, 0.05);
        Gauss1->SetParameter(1, 1.0);
        Gauss1->SetParameter(2, 0.3);
        Gauss1->SetParLimits(0, 0.0, 0.1);
        Gauss1->SetParLimits(1, 0.5, 1.5);
        Gauss1->SetParLimits(2, 0.1, 0.5);
        fFullShow ? fscaled_mk2[ipmt]->Fit("Gauss1","RQ") : fscaled_mk2[ipmt]->Fit("Gauss1","RQN");

        pmt_calib_mk2[ipmt] = abs(1.0 - Gauss1->GetParameter(1));

      } //This brace marks the end of TracksFired strategy


    //Begin investigation of Poisson-like behaviour of calibrated spectra..only valid if particle ID is applied
    if (fCut)
      {
        fscaled_combined[ipmt] = new TH1F(Form("fscaled_combined%d",ipmt+1), Form("Scaled ADC spectra for PMT %d", ipmt+1), 300, 0, 20);

        fscaled_combined_mk2[ipmt] = new TH1F(Form("fscaled_combined_mk2%d",ipmt+1), Form("Scaled ADC spectra with Second Calibration for PMT %d", ipmt+1), 300, 0, 20);

        Int_t nbins = PulseInt[ipmt]->GetXaxis()->GetNbins();
        Double_t xmean = calibration_mk1[ipmt];
        Double_t xmean_mk2 = calibration_mk2[ipmt];

        fscaled_temp[ipmt] = new TH1F(Form("fscaled_temp_pmt%d",ipmt+1), Form("Scaled ADC spectra for PMT %d", ipmt+1), 300, 0, 20);
        fscaled_temp_mk2[ipmt] = new TH1F(Form("fscaled_temp_mk2_pmt%d",ipmt+1), Form("Scaled ADC spectra for PMT %d", ipmt+1), 300, 0, 20);

        //Fill this histogram bin by bin
        for (Int_t ibin=0; ibin < nbins; ibin++)
          {
            Double_t y = PulseInt[ipmt]->GetBinContent(ibin);
            Double_t x = PulseInt[ipmt]->GetXaxis()->GetBinCenter(ibin);
            Double_t x_scaled_mk1 = x/xmean;
            Double_t x_scaled_mk2 = x/xmean_mk2;
            Int_t bin_scaled_mk1 = fscaled_temp[ipmt]->GetXaxis()->FindBin(x_scaled_mk1);
            Int_t bin_scaled_mk2 = fscaled_temp_mk2[ipmt]->GetXaxis()->FindBin(x_scaled_mk2);
            fscaled_temp[ipmt]->SetBinContent(bin_scaled_mk1,y);
            fscaled_temp_mk2[ipmt]->SetBinContent(bin_scaled_mk2,y);
          }
        fscaled_combined[ipmt]->Add(fscaled_temp[ipmt]);
        fscaled_combined_mk2[ipmt]->Add(fscaled_temp_mk2[ipmt]);

        //Normalize the histogram for ease of fitting
        fscaled_combined[ipmt]->Scale(1.0/fscaled_combined[ipmt]->Integral(), "width");
        fscaled_combined_mk2[ipmt]->Scale(1.0/fscaled_combined[ipmt]->Integral(), "width");
      }

  } // This brace marks the end of the loop over PMTs

//Combine each PMT into one final histogram
if (fCut)
  {
    fscaled_total = new TH1F("fscaled_total", "Scaled ADC spectra for all PMTs;NPE;Normalized Counts", 300, 0, 20);
    fscaled_total_mk2 = new TH1F("fscaled_total_mk2", "Scaled ADC spectra for all PMTs;NPE;Normalized Counts", 300, 0, 20);
    for (Int_t i=0; i<4; i++)
      {
        fscaled_total->Add(fscaled_combined[i]);
        fscaled_total_mk2->Add(fscaled_combined_mk2[i]);
      }

    fscaled_total->Scale(1.0/fscaled_total->Integral(), "width");
    fscaled_total_mk2->Scale(1.0/fscaled_total_mk2->Integral(), "width");

    //Display the Poisson characteristics of the ADC spectra
    if (fFullShow) scaled_total = new TCanvas("scaled_total", "Scaled ADC of all PMTs showing Poisson Fit");
    if (fFullShow) scaled_total->Divide(2,1);
    if (fFullShow) scaled_total->cd(1);
    Poisson->SetRange(0, 20);
    Poisson->SetParameter(0, Poisson_mean);
    Poisson->SetParameter(1, 0.8);
    Poisson->SetParLimits(0, Poisson_mean - 1.0, Poisson_mean + 3.0);
    fFullShow ? fscaled_total->Fit("Poisson","RQ") : fscaled_total->Fit("Poisson","RQN");
```

```
      Pois_Chi[0] = Poisson->GetChisquare();
      if (fFullShow) scaled_total->cd(2);
      fFullShow ? fscaled_total_mk2->Fit("Poisson","RQ") : fscaled_total_mk2->Fit("Poisson","RQN");
      Pois_Chi[1] = Poisson->GetChisquare();
    }

  printf("\n\n");

  //Output the actual calibration information
  cout << "Calibration constants are (where the '*' indicates the better value)\nPMT#: First Guess  Second Guess\n" << endl;
  for (Int_t i=0; i<4; i++)
    {
      cout << Form("PMT%d:", i+1) << setw(8) << Form("%3.3f", calibration_mk1[i]) << (pmt_calib[i] < pmt_calib_mk2[i] ? "*" : " ") << setw(13) << Form("%3.3f", calibration_mk2[i]) << (pmt_calib[i] > pmt_calib_mk2[i] ? "*\n" : "\n");
    }

  printf("\n");

  if (fCut) cout << (Pois_Chi[0] < Pois_Chi[1] ? "First Guess":"Second Guess") << " better characterizes the full Poisson character" << endl;

  //Start the process of writing the calibration information to file

  ofstream calibration;
  calibration.open("calibration_temp.txt", ios::out);

  if (!calibration.is_open()) cout << "Problem saving calibration constants, may have to update constants manually!" << endl;

  else
    {
      calibration << "phgcer_adc_to_npe = ";
      for (Int_t ipmt = 0; ipmt < 4; ipmt++)
        {
          calibration << Form("1./%3.3f. ", (pmt_calib[ipmt] < pmt_calib_mk2[ipmt]) ? calibration_mk1[ipmt] : calibration_mk2[ipmt]);
        }

      calibration.close();
    }

}
```

Listing A.5: efficiencies.h; Header File for Efficiency Script

```
//////////////////////////////////////////////////////////
// This class has been automatically generated on
// Thu Jul 20 10:25:29 2017 by ROOT version 5.34/13
// from TTree T/Hall A Analyzer Output DST
// found on file: ../../ROOTfiles/shms_replay_488_-1.root
//////////////////////////////////////////////////////////

#ifndef efficiencies_h
#define efficiencies_h

#include <TROOT.h>
#include <TChain.h>
#include <TFile.h>
#include <TSelector.h>
#include <TTreeReader.h>
#include <TTreeReaderValue.h>
#include <TTreeReaderArray.h>
#include <TH1.h>
#include <TH2.h>

const Int_t hgc_pmts = 4;

// Header file for the classes stored in the TTree if any.

// Fixed size dimensions of array or collections stored in the TTree if any.

class efficiencies : public TSelector {
public :
   TTreeReader    fReader;
   TTree          *fChain = 0;   //!pointer to the analyzed TTree or TChain
   Bool_t          fShowall;
   Bool_t          fChercut;
   Bool_t          fNGC;
   Float_t         fHGC_cut;
   Float_t         fNGC_cut;

   // Declaration of histograms
   TH1F           **fNPE_eNoDet;
   TH1F           **fNPE_eDet;
   TH1F           *fNPE_Full_eNoDet;
   TH1F           *fNPE_Full_eDet;
   TH1F           **fNPE_piNoDet;
   TH1F           **fNPE_piDet;
   TH1F           *fNPE_Full_piNoDet;
   TH1F           *fNPE_Full_piDet;
   TH1F           *fBeta_Cut;
   TH1F           *fBeta_Full;
   TH1F           *fTiming_Cut;
   TH1F           *fTiming_Full;
   TH2F           *fFly_Pr_Full;
   TH2F           *fFly_Pr_eCut;
   TH2F           *fFly_Pr_piCut;

   // Readers to access the data
   TTreeReaderValue<Int_t>    Ndata_P_tr_beta            = {fReader, "Ndata.P.tr.beta"};
   TTreeReaderArray<Double_t> P_tr_beta                  = {fReader, "P.tr.beta"};
   TTreeReaderArray<Double_t> P_hgcer_goodAdcTdcDiffTime = {fReader, "P.hgcer.goodAdcTdcDiffTime"};
   TTreeReaderArray<Double_t> P_hgcer_goodAdcPulseInt    = {fReader, "P.hgcer.goodAdcPulseInt"};
   TTreeReaderArray<Double_t> P_hgcer_goodAdcPulseAmp    = {fReader, "P.hgcer.goodAdcPulseAmp"};
   TTreeReaderArray<Double_t> P_hgcer_numTracksFired     = {fReader, "P.hgcer.numTracksFired"};
   TTreeReaderArray<Double_t> P_hgcer_npe                = {fReader, "P.hgcer.npe"};
   TTreeReaderValue<Double_t> P_cal_fly_earray           = {fReader, "P.cal.fly.earray"};
   TTreeReaderValue<Double_t> P_cal_pr_eplane            = {fReader, "P.cal.pr.eplane"};
   TTreeReaderValue<Double_t> P_cal_etotnorm             = {fReader, "P.cal.etotnorm"};
   TTreeReaderArray<Double_t> P_gtr_dp                   = {fReader, "P.gtr.dp"};
   TTreeReaderArray<Double_t> P_tr_x                     = {fReader, "P.tr.x"};
   TTreeReaderArray<Double_t> P_tr_ph                    = {fReader, "P.tr.ph"};
   TTreeReaderArray<Double_t> P_tr_y                     = {fReader, "P.tr.y"};
   TTreeReaderArray<Double_t> P_tr_th                    = {fReader, "P.tr.th"};

 efficiencies(TTree * /*tree*/ =0) : fChain(0) {fNPE_eNoDet = 0, fNPE_eDet = 0, fNPE_piNoDet = 0, fNPE_piDet = 0,
       fNPE_Full_eNoDet = 0, fNPE_Full_eDet = 0, fNPE_Full_piNoDet = 0, fNPE_Full_piDet = 0, fBeta_Cut = 0,
      fBeta_Full = 0, fTiming_Cut = 0, fTiming_Full = 0, fFly_Pr_Full = 0, fFly_Pr_eCut = 0, fFly_Pr_piCut = 0,
      fShowall = kFALSE, fChercut = kFALSE, fNGC = kFALSE, fHGC_cut = 2.0, fNGC_cut = 2.0;}
   virtual ~efficiencies() { }
   virtual Int_t   Version() const { return 2; }
   virtual void    Begin(TTree *tree);
   virtual void    SlaveBegin(TTree *tree);
   virtual void    Init(TTree *tree);
   virtual Bool_t  Notify();
   virtual Bool_t  Process(Long64_t entry);
   virtual Int_t   GetEntry(Long64_t entry, Int_t getall = 0) { return fChain ? fChain->GetTree()->GetEntry(entry
        , getall) : 0; }
   virtual void    SetOption(const char *option) { fOption = option; }
   virtual void    SetObject(TObject *obj) { fObject = obj; }
   virtual void    SetInputList(TList *input) { fInput = input; }
   virtual TList  *GetOutputList() const { return fOutput; }
   virtual void    SlaveTerminate();
   virtual void    Terminate();

   ClassDef(efficiencies,0);
};

#endif

#ifdef efficiencies_cxx
void efficiencies::Init(TTree *tree)
{
   // The Init() function is called when the selector needs to initialize
   // a new tree or chain. Typically here the branch addresses and branch
   // pointers of the tree will be set.
   // It is normally not necessary to make changes to the generated
   // code, but the routine can be extended by the user if needed.
   // Init() will be called many times when running on PROOF
   // (once per file to be processed).

   // Set branch addresses and branch pointers

  fReader.SetTree(tree);
}
```

```

Bool_t efficiencies::Notify()
{
   // The Notify() function is called when a new file is opened. This
   // can be either for a new TTree in a TChain or when when a new TTree
   // is started when using PROOF. It is normally not necessary to make changes
   // to the generated code, but the routine can be extended by the
   // user if needed. The return value is currently not used.

   return kTRUE;
}

#endif // #ifdef efficiencies_cxx
```

Listing A.6: efficiencies.C; Script to determine the efficiency of the HGC

```
#define efficiencies_cxx
// The class definition in efficiencies.h has been generated automatically
// by the ROOT utility TTree::MakeSelector(). This class is derived
// from the ROOT class TSelector. For more information on the TSelector
// framework see \$ROOTSYS/README/README.SELECTOR or the ROOT User Manual.

// The following methods are defined in this file:
//    Begin():        called every time a loop on the tree starts,
//                    a convenient place to create your histograms.
//    SlaveBegin():   called after Begin(), when on PROOF called only on the
//                    slave servers.
//    Process():      called for each event, in this function you decide what
//                    to read and fill your histograms.
//    SlaveTerminate: called at the end of the loop on the tree, when on PROOF
//                    called only on the slave servers.
//    Terminate():    called at the end of the loop on the tree,
//                    a convenient place to draw/fit your histograms.
//
// To use this file, try the following session on your Tree T:
//
// Root > T->Process("efficiencies.C+","some options")
// Root > T->Process("efficiencies.C+")
//

#include "efficiencies.h"
#include <TH2.h>
#include <TStyle.h>
#include <TCanvas.h>
#include <TPaveStats.h>
#include <iostream>
#include <iomanip>
#include <TRegexp.h>

void efficiencies::Begin(TTree * /*tree*/)
{
   // The Begin() function is called at the start of the query.
   // When running with PROOF Begin() is only called on the client.
   // The tree argument is deprecated (on PROOF 0 is passed).
  printf("\n\n");

  TString option = GetOption();
  //Regular expression to extract the specified photoelectron cutoff
  TRegexp re("[0-9]+[.][0-9]+");

  Info("Begin", "Starting efficiency process with option: %s", option.Data());
  Info("Begin", "To show details of calculation, use option showall (will open lots of windows)");
  Info("Begin", "Default detector is HGC, use option NGC if desired");
  Info("Begin", "If cut desired on other Cherenkov, use option Chercut");
  Info("Begin", "Photoelectron cut is taken as first decimal number entered. Default is 2.0");
  Info("Begin", "If cut on other Chenekov is requested, it will be taken as the second decimal number. Default is 2.0");
  printf("\n\n");

  //Check option

  if (option.Contains("showall")) fShowall = kTRUE;
  if (option.Contains("Chercut")) fChercut = kTRUE;
  if (option.Contains("NGC")) fNGC = kTRUE;

  //if a number is entered as %f.%f take that as the photoelectron cutoff
  if (option.Contains(re))
    {
      TString cut_val = option.operator()(re);
      fNGC ? fNGC_cut =  cut_val.Atof() : fHGC_cut = cut_val.Atof();
      if (fChercut)
        {
          TString cut_val = option.operator()(re, option.Index(re)+1);
          fNGC ? fHGC_cut =  cut_val.Atof() : fNGC_cut = cut_val.Atof();
        }
    }
}

void efficiencies::SlaveBegin(TTree * /*tree*/)
{
  // The SlaveBegin() function is called after the Begin() function.
  // When running with PROOF SlaveBegin() is called on each slave server.
  // The tree argument is deprecated (on PROOF 0 is passed).

  printf("\n\n");
  TString option = GetOption();
  TRegexp re("[0-9]+[.][0-9]+");
  //Check option
  if (option.Contains("showall")) fShowall = kTRUE;
  if (option.Contains("Chercut")) fChercut = kTRUE;
  if (option.Contains("NGC")) fNGC = kTRUE;
  if (option.Contains(re))
    {
      TString cut_val = option.operator()(re);
      fNGC ? fNGC_cut =  cut_val.Atof() : fHGC_cut = cut_val.Atof();
      if (fChercut)
        {
          TString cut_val = option.operator()(re, option.Index(re)+1);
          fNGC ? fHGC_cut =  cut_val.Atof() : fNGC_cut = cut_val.Atof();
        }
    }

  Info("SlaveBegin", "%s showing", fShowall ? "full" : "minimal");
  Info("SlaveBegin", "%s other Cherenkov for particle ID", fChercut ? "using" : "ignoring");
  Info("SlaveBegin", "efficiency for %s is found", fNGC ? "NGC" : "HGC");
  Info("SlaveBegin", "Number of photoelectrons to cut on is %f", fNGC ? fNGC_cut : fHGC_cut);
  if (fChercut) Info("SlaveBegin", "Number of photoelectrons to cut on in other Cherenkov %f", fNGC ? fHGC_cut : fNGC_cut);

  printf("\n");


  //Initialize the histograms.
  Int_t NPE_min;
  Int_t NPE_max;
  Int_t bins;
```

```

NPE_min = 0;
NPE_max = 20;
bins = (NPE_min + NPE_max)*10;

fNPE_eNoDet = new TH1F*[4];
fNPE_eNoDet[0] = new TH1F("NPE_eNoDet_PMT1", "NPE in PMT1 No Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
fNPE_eNoDet[1] = new TH1F("NPE_eNoDet_PMT2", "NPE in PMT2 No Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
fNPE_eNoDet[2] = new TH1F("NPE_eNoDet_PMT3", "NPE in PMT3 No Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
fNPE_eNoDet[3] = new TH1F("NPE_eNoDet_PMT4", "NPE in PMT4 No Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
GetOutputList()->Add(fNPE_eNoDet[0]);
GetOutputList()->Add(fNPE_eNoDet[1]);
GetOutputList()->Add(fNPE_eNoDet[2]);
GetOutputList()->Add(fNPE_eNoDet[3]);


fNPE_eDet = new TH1F*[4];
fNPE_eDet[0] = new TH1F("NPE_eDet_PMT1", "NPE in PMT1 with Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
fNPE_eDet[1] = new TH1F("NPE_eDet_PMT2", "NPE in PMT2 with Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
fNPE_eDet[2] = new TH1F("NPE_eDet_PMT3", "NPE in PMT3 with Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
fNPE_eDet[3] = new TH1F("NPE_eDet_PMT4", "NPE in PMT4 with Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
GetOutputList()->Add(fNPE_eDet[0]);
GetOutputList()->Add(fNPE_eDet[1]);
GetOutputList()->Add(fNPE_eDet[2]);
GetOutputList()->Add(fNPE_eDet[3]);

fNPE_Full_eNoDet = new TH1F("NPE_Full_eNoDet", "NPE in All PMTs with no Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
GetOutputList()->Add(fNPE_Full_eNoDet);

fNPE_Full_eDet = new TH1F("NPE_Full_eDet", "NPE in All PMTs with Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
GetOutputList()->Add(fNPE_Full_eDet);

fNPE_piNoDet = new TH1F*[4];
fNPE_piNoDet[0] = new TH1F("NPE_piNoDet_PMT1", "NPE in PMT1 No Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
fNPE_piNoDet[1] = new TH1F("NPE_piNoDet_PMT2", "NPE in PMT2 No Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
fNPE_piNoDet[2] = new TH1F("NPE_piNoDet_PMT3", "NPE in PMT3 No Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
fNPE_piNoDet[3] = new TH1F("NPE_piNoDet_PMT4", "NPE in PMT4 No Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
GetOutputList()->Add(fNPE_piNoDet[0]);
GetOutputList()->Add(fNPE_piNoDet[1]);
GetOutputList()->Add(fNPE_piNoDet[2]);
GetOutputList()->Add(fNPE_piNoDet[3]);

fNPE_piDet = new TH1F*[4];
fNPE_piDet[0] = new TH1F("NPE_piDet_PMT1", "NPE in PMT1 with Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
fNPE_piDet[1] = new TH1F("NPE_piDet_PMT2", "NPE in PMT2 with Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
fNPE_piDet[2] = new TH1F("NPE_piDet_PMT3", "NPE in PMT3 with Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
fNPE_piDet[3] = new TH1F("NPE_piDet_PMT4", "NPE in PMT4 with Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
GetOutputList()->Add(fNPE_piDet[0]);
GetOutputList()->Add(fNPE_piDet[1]);
GetOutputList()->Add(fNPE_piDet[2]);
```

```
  GetOutputList()->Add(fNPE_piDet[3]);

  fNPE_Full_piNoDet = new TH1F("NPE_Full_piNoDet", "NPE in All PMTs with no Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
  GetOutputList()->Add(fNPE_Full_piNoDet);

  fNPE_Full_piDet = new TH1F("NPE_Full_piDet", "NPE in All PMTs with Detector Cut;NPE;Counts", bins, NPE_min, NPE_max);
  GetOutputList()->Add(fNPE_Full_piDet);

  //Histograms examining various cuts for "good" hits
  fBeta_Cut = new TH1F("Beta_Cut", "Beta cut used for ’good’ hits;Beta;Counts", 1000, -5, 5);
  GetOutputList()->Add(fBeta_Cut);

  fBeta_Full = new TH1F("Beta_Full", "Full beta for events;Beta;Counts", 1000, -5, 5);
  GetOutputList()->Add(fBeta_Full);

  fTiming_Cut = new TH1F("Timing_Cut", "Timing cut used for ’good’ hits;Time (ns);Counts", 500, -100, 100);
  GetOutputList()->Add(fTiming_Cut);

  fTiming_Full = new TH1F("Timing_Full", "Full timing information for events;Time (ns);Counts", 500, -100, 100);
  GetOutputList()->Add(fTiming_Full);

  //Histograms examining particle ID cuts
  fFly_Pr_Full = new TH2F("Fly_Pr_Full", "Particle ID from calorimeter & preshower;Calorimeter (GeV);Pre-Shower (GeV)", 250, 0.0, 0.7, 250, 0.0, 0.8);
  GetOutputList()->Add(fFly_Pr_Full);

  fFly_Pr_eCut = new TH2F("Fly_Pr_eCut", "calorimeter & preshower electrons;Calorimeter (GeV);Pre-Shower (GeV)", 250, 0.0, 0.7, 250, 0.0, 0.8);
  GetOutputList()->Add(fFly_Pr_eCut);

  fFly_Pr_piCut = new TH2F("Fly_Pr_piCut", "calorimeter & preshower pions;Calorimeter (GeV);Pre-Shower (GeV)", 250, 0.0, 0.7, 250, 0.0, 0.8);
  GetOutputList()->Add(fFly_Pr_piCut);

  printf("\n\n");
}

Bool_t efficiencies::Process(Long64_t entry)
{
   // The Process() function is called for each entry in the tree (or possibly
   // keyed object in the case of PROOF) to be processed. The entry argument
   // specifies which entry in the currently loaded tree is to be processed.
   // It can be passed to either efficiencies::GetEntry() or TBranch::GetEntry()
   // to read either all or the required parts of the data. When processing
   // keyed objects with PROOF, the object is already loaded and is available
   // via the fObject pointer.
   //
   // This function should contain the "body" of the analysis. It can contain
   // simple or elaborate selection criteria, run algorithms on the data
   // of the event and typically fill histograms.
   //
   // The processing can be stopped by calling Abort().
   //
   // Use fStatus to set the return value of TTree::Process().
```

```
 //
 // The return value is currently not used.

//Output to verify script is working, and store the total number of events
//if (entry % 100000 == 0) printf("Processing Entry number %lld\n",entry);

//Get the entry to loop over
fReader.SetEntry(entry);

//Require only one good track reconstruction for the event
if (*Ndata_P_tr_beta != 1) return kTRUE;

//Redundant requirement, but useful if multiple tracks are eventually allowed
for (Int_t itrack = 0; itrack < *Ndata_P_tr_beta; itrack++)
  {
    //Require loose cut on particle velocity
    fBeta_Full->Fill(P_tr_beta[itrack]);
    if (TMath::Abs(P_tr_beta[itrack] - 1.0) > 0.2) return kTRUE;
    fBeta_Cut->Fill(P_tr_beta[itrack]);

    //Filling the histograms
    for (Int_t ipmt = 0; ipmt < hgc_pmts; ipmt++)
      {
        //Require the signal passes a timing cut
        fTiming_Full->Fill(/*fNGC ?  P_ngcer_goodAdcPulseTime[ipmt] : */P_hgcer_goodAdcTdcDiffTime[ipmt]);
        if (/*fNGC ? P_ngcer_goodAdcPulseTime[ipmt] < 50 || P_ngcer_goodAdcPulseTime[ipmt] > 125 :*/
        P_hgcer_goodAdcTdcDiffTime[ipmt] > 38.0 || P_hgcer_goodAdcTdcDiffTime[ipmt] < 30.0) continue;
        fTiming_Cut->Fill(/*fNGC ?  P_ngcer_goodAdcPulseTime[ipmt] : */P_hgcer_goodAdcTdcDiffTime[ipmt]);

        //Require the signal passes a tracking cut, with a threshold NPE cut as well
        if (/*fNGC ? P_ngcer_numTracksFired[ipmt] == 0.0 :*/ P_hgcer_numTracksFired[ipmt] == 0.0) continue;

        //Begin particle identification using calorimeter and NGC
        Float_t central_p = 2.2;
        Float_t p = ((P_gtr_dp[itrack]/100.0)*central_p) + central_p;

        //Visualize what we have to cut with
        fFly_Pr_Full->Fill(*P_cal_fly_earray/p, *P_cal_pr_eplane/p);

        //Perform cut for electrons
        //Cut on Shower vs preshower is a tilted ellipse, this requires an angle of rotation (in radians), x/y center, semimajor and semiminor axis
        /*Float_t eangle = 3.0*3.14159/4.0;
        Float_t ex_center = 0.66;
        Float_t ey_center = 0.35;
        Float_t esemimajor_axis = 0.28;
        Float_t esemiminor_axis = 0.04;*/
        Float_t eangle = 3.0*3.14159/4.0;
        Float_t ex_center = 0.490;
        Float_t ey_center = 0.400;
        Float_t esemimajor_axis = 0.38;
        Float_t esemiminor_axis = 0.05;
        if (pow((*P_cal_fly_earray/p - ex_center)*cos(eangle) + (*P_cal_pr_eplane/p - ey_center)*sin(eangle),2)/pow(esemimajor_axis,2) +
            pow((*P_cal_fly_earray/p - ex_center)*sin(eangle) - (*P_cal_pr_eplane/p - ey_center)*cos(eangle),2)/pow(esemiminor_axis,2) < 1)
            {
              /*if (!fChercut || (fChercut && (fNGC ? P_hgcer_npeSum > fHGC_cut : P_ngcer_npeSum > fNGC_cut)))*/ //condition if cut on other Cherenkov if desired
                {
                  fFly_Pr_eCut->Fill(*P_cal_fly_earray/p, *P_cal_pr_eplane/p);
                  fNPE_eNoDet[ipmt]->Fill(/*fNGC ? P_ngcer_npe[ipmt] :*/ P_hgcer_npe[ipmt]);
                  fNPE_Full_eNoDet->Fill(/*fNGC ? P_ngcer_npe[ipmt] :*/ P_hgcer_npe[ipmt]);

                  if (/*fNGC ? P_ngcer_npe[ipmt] > fNGC_cut : */P_hgcer_npe[ipmt] > fHGC_cut)  //Cut on detector to determine efficiency
                    {
                      fNPE_eDet[ipmt]->Fill(/*fNGC ? P_ngcer_npe[ipmt] :*/ P_hgcer_npe[ipmt]);
                      fNPE_Full_eDet->Fill(/*fNGC ? P_ngcer_npe[ipmt] :*/ P_hgcer_npe[ipmt]);
                    }
                }
            }

          //Perform cut for pions
          //Cut on Shower vs preshower is a tilted ellipse, this requires an angle of rotation (in radians), x/y center, semimajor and semiminor axis
          Float_t piangle = 0.0;/*
          Float_t pix_center = 0.26;
          Float_t piy_center = 0.03;
          Float_t pisemimajor_axis = 0.1;
          Float_t pisemiminor_axis = 0.02;*/
          Float_t pix_center = 0.30;
          Float_t piy_center = 0.05;
          Float_t pisemimajor_axis = 0.08;
          Float_t pisemiminor_axis = 0.02;
          if (pow((*P_cal_fly_earray/p - pix_center)*cos(piangle) + (*P_cal_pr_eplane/p - piy_center)*sin(piangle),2)/pow(pisemimajor_axis,2) +
              pow((*P_cal_fly_earray/p - pix_center)*sin(piangle) - (*P_cal_pr_eplane/p - piy_center)*cos(piangle),2)/pow(pisemiminor_axis,2) < 1)
            {
              /*if (!fChercut || (fChercut && (fNGC ? P_hgcer_npeSum < fHGC_cut : P_ngcer_npeSum < fNGC_cut)))*/ //condition if cut on other Cherenkov is desired
                {
                  fFly_Pr_piCut->Fill(*P_cal_fly_earray/p, *P_cal_pr_eplane/p);
                  fNPE_piNoDet[ipmt]->Fill(/*fNGC ? P_ngcer_npe[ipmt] :*/ P_hgcer_npe[ipmt]);
                  fNPE_Full_piNoDet->Fill(/*fNGC ? P_ngcer_npe[ipmt] :*/ P_hgcer_npe[ipmt]);

                  if (/*fNGC ? P_ngcer_npe[ipmt] > fNGC_cut :*/ P_hgcer_npe[ipmt] > fHGC_cut)   //Cut on detector to determine efficiency
                    {
                      fNPE_piDet[ipmt]->Fill(/*fNGC ? P_ngcer_npe[ipmt] :*/ P_hgcer_npe[ipmt]);
                      fNPE_Full_piDet->Fill(/*fNGC ? P_ngcer_npe[ipmt] :*/ P_hgcer_npe[ipmt]);
                    }
                }
            }

        }//End loop on PMTs
    }//End loop over tracks

   return kTRUE;
}

```

```
void efficiencies::SlaveTerminate()
{
   // The SlaveTerminate() function is called after all entries or objects
   // have been processed. When running with PROOF SlaveTerminate() is called
   // on each slave server.
}

void efficiencies::Terminate()
{
   // The Terminate() function is the last function to be called during
   // a query. It always runs on the client, it can be used to present
   // the results graphically or save the results to file.
  printf("\n\n");
  Info("Terminate", "%s showing", fShowall ? "full" : "minimal");
  Info("Terminate", "%s other Cherenkov for particle ID", fChercut ? "using" : "ignoring");
  Info("Terminate", "efficiency for %s is found", fNGC ? "NGC" : "HGC");
  Info("Terminate", "Number of photoelectrons to cut on is %f", fNGC ? fNGC_cut : fHGC_cut);
  if (fChercut) Info("Terminate", "Number of photoelectrons to cut on in other Cherenkov %f", fNGC ? fHGC_cut : fNGC_cut);

  //Need to exact the histograms from the OutputList
  TH1F* NPE_eNoDet[4];
  TH1F* NPE_eDet[4];
  TH1F* NPE_piNoDet[4];
  TH1F* NPE_piDet[4];

  for (Int_t ipmt = 0; ipmt < 4; ipmt++)
    {
      NPE_eNoDet[ipmt] = dynamic_cast<TH1F*> (GetOutputList()->FindObject(Form("NPE_eNoDet_PMT%d",ipmt+1)));
      NPE_eDet[ipmt] = dynamic_cast<TH1F*> (GetOutputList()->FindObject(Form("NPE_eDet_PMT%d",ipmt+1)));
      NPE_piNoDet[ipmt] = dynamic_cast<TH1F*> (GetOutputList()->FindObject(Form("NPE_piNoDet_PMT%d",ipmt+1)));
      NPE_piDet[ipmt] = dynamic_cast<TH1F*> (GetOutputList()->FindObject(Form("NPE_piDet_PMT%d",ipmt+1)));
    }



  //Canvases to display efficiency information
  if (fShowall)
    {
      //Canvas to show beta cut information
      TCanvas *Beta;
      Beta = new TCanvas("Beta", "Beta information for events");
      Beta->Divide(2,1);
      Beta->cd(1);
      fBeta_Full->Draw();
      Beta->cd(2);
      fBeta_Cut->Draw();

      //Canvas to show timing cut information
      TCanvas *Timing;
      Timing = new TCanvas("Timing", "Timing information for events");
      Timing->Divide(2,1);
      Timing->cd(1);
      fTiming_Full->Draw();
      Timing->cd(2);

```

```
fTiming_Cut->Draw();

//Canvas to show Particle ID cut information
TCanvas *Fly_Pr;
Fly_Pr = new TCanvas("Fly_Pr", "Particle ID info from calorimeter & Preshower");
Fly_Pr->Divide(3,1);
Fly_Pr->cd(1);
fFly_Pr_Full->Draw("Colz");
Fly_Pr->cd(2);
fFly_Pr_eCut->Draw("Colz");
Fly_Pr->cd(3);
fFly_Pr_piCut->Draw("Colz");

//Canvases to show Effects of HGC cut
//Start with electrons
gStyle->SetOptStat(11);
TCanvas *Det_eCut;
Det_eCut = new TCanvas("Det_eCut","Effect of performing cut for electrons per PMT");
Det_eCut->Divide(2,4);
for (Int_t ipad = 0; ipad < 8; ipad += 2)
  {
    Det_eCut->cd(ipad+1);
    NPE_eNoDet[ipad/2]->Draw();
    gPad->Update();
    TPaveStats *s1 = (TPaveStats*) gPad->GetPrimitive("stats");
    s1->SetTextSize(0.1), s1->SetX1NDC(0.7), s1->SetY1NDC(0.5);

    Det_eCut->cd(ipad+2);
    NPE_eDet[ipad/2]->Draw();
    gPad->Update();
    TPaveStats *s2 = (TPaveStats*) gPad->GetPrimitive("stats");
    s2->SetTextSize(0.1), s2->SetX1NDC(0.7), s2->SetY1NDC(0.5);
  }

TCanvas *Det_eCut_Full;
Det_eCut_Full = new TCanvas("Det_eCut_Full","Effect of performing cut for electrons");
Det_eCut_Full->Divide(2,1);
Det_eCut_Full->cd(1);
fNPE_Full_eNoDet->Draw();
Det_eCut_Full->cd(2);
fNPE_Full_eDet->Draw();

//End with Pions
gStyle->SetOptStat(11);
TCanvas *Det_piCut;
Det_piCut = new TCanvas("Det_piCut","Effect of performing cut for pions per PMT");
Det_piCut->Divide(2,4);
for (Int_t ipad = 0; ipad < 8; ipad += 2)
  {
    Det_piCut->cd(ipad+1);
    NPE_piNoDet[ipad/2]->Draw();
    gPad->Update();
    TPaveStats *s3 = (TPaveStats*) gPad->GetPrimitive("stats");
    s3->SetTextSize(0.1), s3->SetX1NDC(0.7), s3->SetY1NDC(0.5);

    Det_piCut->cd(ipad+2);
```

```
          NPE_piDet[ipad/2]->Draw();
          gPad->Update();
          TPaveStats *s4 = (TPaveStats*) gPad->GetPrimitive("stats");
          s4->SetTextSize(0.1), s4->SetX1NDC(0.7), s4->SetY1NDC(0.5);
        }

      TCanvas *Det_piCut_Full;
      Det_piCut_Full = new TCanvas("Det_piCut_Full","Effect of performing cut for pions");
      Det_piCut_Full->Divide(2,1);
      Det_piCut_Full->cd(1);
      fNPE_Full_piNoDet->Draw();
      Det_piCut_Full->cd(2);
      fNPE_Full_piDet->Draw();
    }

  //Output the actual efficiency information i.e. ratio of detected particles
  cout << Form("\nEfficiencies for the %s with a cut at %.1f are:\nPMT#  electrons  pions:electron", fNGC ? "NGC" : "HGC", fNGC ? fNGC_cut : fHGC_cut) << endl;
  for (Int_t ipmt = 0; ipmt < 4; ipmt++)
    {
      printf("PMT%d: %2.2f%%         1:%.f\n", ipmt+1, (NPE_eDet[ipmt]->GetEntries()/NPE_eNoDet[ipmt]->GetEntries())*100.0, NPE_eDet[ipmt]->GetEntries()/NPE_piDet[ipmt]->GetEntries());
    }
  printf("Full: %2.2f%%         1:%.f\n\n\n",(fNPE_Full_eDet->GetEntries()/fNPE_Full_eNoDet->GetEntries())*100.0, fNPE_Full_eDet->GetEntries()/fNPE_Full_piDet->GetEntries());
}
```

Bibliography

Abrisa (2010), "Corning 7980 fused silica." http://abrisatechnologies.com/products-services/glass-products/quartz-fused-silica/corning-7980/.

Alexa, L.C., G.M. Huber, G.J. Lolos, F. Farzanpay, F. Garibaldi, M. Jodice, A. Leone, R. Perrino, Z. Papandreou, D.L. Humphrey, P. Ulmer, and R. DeLeo (1995), "Empirical tests and model of a silica aerogel cherenkov detector for cebaf." *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 365, 299 – 307, URL http://www.sciencedirect.com/science/article/pii/0168900295005153.

Arrington, John, Kees de Jager, and Charles F Perdrisat (2011), "Nucleon form factors – a jefferson lab perspective." *Journal of Physics: Conference Series*, 299, 012002, URL http://stacks.iop.org/1742-6596/299/i=1/a=012002.

Artuso, M. et al. (2006), "Performance of a C(4)F(8)O gas radiator ring imaging Cerenkov detector using multi-anode photomultiplier tubes." *Nucl. Instrum. Meth.*, A558, 373–387.

Basnet, Samip (2018). private communication.

Bergström, Kai and Stig Sunner (1976), "The relative permittivity of argon, carbon dioxide, and hydrogen, determined at 10 kHz using a transformer bridge method." *Physica Scripta*, 13, 51–52, URL https://doi.org/10.1088%2F0031-8949%2F13%2F1%2F007.

Bideau-Mehu, A., Y. Guern, R. Abjean, and A. Johannin-Gilles (1973), "Interferometric determination of the refractive index of carbon dioxide in the ultraviolet region." *Optics Communications*, 9, 432 – 434, URL `http://www.sciencedirect.com/science/article/pii/0030401873902897`.

Brash, E.J, J Hovdebo, G.J Lolos, G.M Huber, R van der Meer, and Z Papandreou (2002), "Operational performance of the hall a mirror aerogel cherenkov counter." *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 487, 346 – 352, URL `http://www.sciencedirect.com/science/article/pii/S0168900201021994`.

CERN, 2018 (2018), *ROOT Reference Guide*. CERN. `https://root.cern.ch/doc/v608/classTSpectrum.html#a5ba181a45b117934b48c4ef5f78d0b2b`.

Chem Spider (2018a), "Carbon dioxide." accessed at `http://www.chemspider.com/Chemical-Structure.274.html`, CSID:274.

Chem Spider (2018b), "Perfluorobutane." accessed at `http://www.chemspider.com/Chemical-Structure.13862701.html`, CSID:13862701.

Chem Spider (2018c), "Perfluorotetrahydrofuran." accessed at `http://www.chemspider.com/Chemical-Structure.144411.html`, CSID:144411.

Christy, M. Eric, Peter Monaghan, Narbe Kalantarians, Debaditya Biswas, and Mahlon Long (2016), "Hall c reference shms drift chambers." `https://hallcweb.jlab.org/document/howtos/shms_drift_chambers.pdf`.

Ciovati, G., P. Dhakal, and G. R. Myneni (2016), "Superconducting radio-frequency cavities made from medium and low-purity niobium ingots." *Superconductor Science Technology*, 29, 064002.

Corning (2018), "High purity fused silica." One Riverfront Plaza, Corning, NY 14831 USA, https://www.corning.com/in/en/products/advanced-optics/product-materials/semiconductor-laser-optic-components/high-purity-fused-silica.html.

Day, Donal (2017), "Preliminary design of the shms noble gas Čerenkov detector." https://hallcweb.jlab.org/DocDB/0009/000933/001/shms-cerv6.pdf.

Degtiarenko, Pavel (2017a). e-mail correspondence.

Degtiarenko, Pavel (2017b). e-mail correspondence.

Degtiarenko, Pavel (2017c), "Precision analysis of the photomultiplier response to ultra low signals." *Nucl. Instrum. Meth.*, A872, 1–15.

Dow Corning (1998), "Dow corning q2-3067 optical couplant." One Riverfront Plaza, Corning, NY 14831 USA, https://www.ellsworth.com/products/by-manufacturer/dow-corning/encapsulants/optical/dow-corning-q2-3067-optical-couplant-113-g-jar/.

Evaporated Coatings (2018). 2365 Maryland Road, Willoe Grove, PA 19090. http://www.evaporatedcoatings.com.

Fernow, Richard Clinton (1986), *Cerenkov counters*, 178–204. Cambridge University Press.

Fischer, Alex (2012), "Performance testing of 5 inch pmts." https://hallcweb.jlab.org/DocDB/0007/000738/001/gain_report.pdf.

Fuchy, Eric (2016). private correspondence.

Geant4, 2017 (2017), *Geant4 a Simulation Toolkit: User Documentation.* CERN. Accessed at http://geant4.web.cern.ch/support/user_documentation.

Gilman, R, R J Holt, and P Stoler (2011), "Transition to perturbative qcd." *Journal of Physics: Conference Series*, 299, 012009, URL `http://stacks.iop.org/1742-6596/299/i=1/a=012009`.

Griffiths, David J (2013), *Introduction to electrodynamics; 4th ed.* Pearson, Boston, MA, URL `https://cds.cern.ch/record/1492149`. Re-published by Cambridge University Press in 2017.

Gross, Franz (2011), "Making the case for jefferson lab." *Journal of Physics: Conference Series*, 299, 012001, URL `http://stacks.iop.org/1742-6596/299/i=1/a=012001`.

Hall-C, Thomas Jefferson National Research Facility (2018), "hcana." `https://github.com/JeffersonLab/hcana`, `https://github.com/JeffersonLab/hallc_replay`. Version 0.90.0.

Hamamatsu, 2006 (2006), *Photomultiplier Tubes: Photomultiplier Tubes and Related Products*. Hamamatsu.

Hardin Optical Company (2018). PO Box 219, Bandon, OR 97411, `https://www.hardinoptical.com`.

hclog (2017a), "Shms start of run 1583." `https://logbooks.jlab.org/entry/3501980`.

hclog (2017b), "Shms start of run 488." `https://logbooks.jlab.org/entry/3467029`.

hclog (2018a), "Coin start of run 3623." `https://logbooks.jlab.org/entry/3558820`.

hclog (2018b), "Coin start of run 4167." `https://logbooks.jlab.org/entry/3569530`.

hclog (2018c), “Coin start of run 4201.” https://logbooks.jlab.org/entry/3570132.

hclog (2018d), “Shms hgc pmt1 assembly modified and reinstalled.” https://logbooks.jlab.org/entry/3569752.

hclog (2018e), “Shms start of run 3623.” https://logbooks.jlab.org/entry/3561974.

Horn, T., G.M. Huber, and P. Markowitz (2008), “Studies of the l-t separated kaon electroproduction cross section from 5-11 gev.” Accessed at https://www.jlab.org/exp_prog/proposals/09/PR12-09-011.pdf.

Horn, Tanja (2017), “Shms aerogel detector operations.” https://hallcweb.jlab.org/DocDB/0008/000818/001/SHMS_aerogel_operations_manual_06202016.pdf.

Huber, Garth (2018a). e-mail correspondence. Values for C_4F_{10} and C_4F_8O are from the HADES collaboration.

Huber, Garth (2018b). e-mail correspondence.

Jackson, John David (1999), *Classical Electrodynamics*, 3rd ed. edition. John Wiley & Sons, Inc., New York, NY.

Jones, 2008 (2008), *FADC250 User’s Manual*. UConn. https://hallcweb.jlab.org/DocDB/0008/000814/001/FADC250UsersManual.pdf.

Leo, William R. (1987), *Techniques for Nuclear and Particle Physics Experiments A How-to Approach*, 1st ed. edition. Springer-Verlag, Lausanne, Sitzerland.

Li, Wenliang (2012), *Heavy Gas Cherenkov Detector Construction for Hall C at Thomas Jefferson National Accelerator Facility*. Master’s thesis, University of Regina.

Li, Wenliang (2018a). e-mail correspondence. Configuration is also located in simulation software.

Li, Wenliang (2018b). private e-mail correspondence.

Li, Wenliang and Garth Huber (2014), “Rtv 615 transmission results.” Accessed at `https://hallcweb.jlab.org/DocDB/0007/000772/001/transmivisity.pdf`.

Mariscotti, M. A. (1967), “A method for automatic identification of peaks in the presence of background and its application to spectrum analysis.” *Nuclear Instruments and Methods*, 50, 309–320.

Metzger, Bert (2012). SHMS engineering drawings, metzger@jlab.org.

Mkrtchyan, A., ed. (2012), *SHMS Calorimeter Status Update*, SHMS Detector Progress Meeting, Jefferson Lab. `https://hallcweb.jlab.org/DocDB/0007/000745/001/arthur-shms-calo-prog-aug27-2012.pdf`.

Mkrtchyan, A., ed. (2016), *Aerogel Detectors*, Hall C Experimental Readiness Review, Jefferson Lab. `https://hallcweb.jlab.org/doc-private/ShowDocument?docid=808`.

Momentive (2018), “Rtv615.” 180 East Broad Street, Columbus, OH 43215, `https://www.momentive.com/en-us/products/tds/rtv615/`.

Montgomery, Hugh (2011), “Jefferson lab: A long decade of physics.” *Journal of Physics: Conference Series*, 299, 011001, URL `http://stacks.iop.org/1742-6596/299/i=1/a=011001`.

Morháč, M., J. Kliman, V. Matoušek, M. Veselský, and I. Turzo (2000), “Identification of peaks in multidimensional coincidence /γ-ray spectra.” *Nuclear Instruments and Methods in Physics Research A*, 443, 108–125.

Pooser, Eric (2016), *The GlueX Start Counter & Beam Asymmetry* Σ *in Single* π^0 *Photoproduction*. Ph.D. thesis, Florida International University.

Pooser, Eric, ed. (2017), *Hall C Analyzer*, Hall C Winter Collaboration Meeting, Jefferson Lab. `https://hallcweb.jlab.org/DocDB/0008/000816/001/Pooser_HallC_Analyzer.pdf`.

Pooser, Eric (2018a). e-mail correspondence.

Pooser, Eric (2018b), "Daq/trigger run check list." `https://hallcweb.jlab.org/wiki/index.php/DAQ/Trigger_Run_Check_Lists`. Accessed: 15-05-2018.

Roche, Julie, Willem T H van Oers, and Ross D Young (2011), "Searches for physics beyond the standard model." *Journal of Physics: Conference Series*, 299, 012012, URL `http://stacks.iop.org/1742-6596/299/i=1/a=012012`.

ROOT, 2018 (2018), *ROOT Data Analysis Framework User's Guide*. CERN. Accessed at `https://root.cern.ch/root/htmldoc/guides/users-guide/ROOTUsersGuide.html`.

RuggeroTurra (2009). Accessed July 2018 at `https://twiki.cern.ch/twiki/bin/view/LHCb/C4F10`.

Sawatzky, Brad, D. Day, A. Conover, and P. McCormick (2018), "Shms hgc cerenkov pmt sanity checks at uva." Accessed at `https://hallcweb.jlab.org/DocDB/0009/000979/001/HGC_PMT_check-Jul2018.pdf`.

Silagadze, Z. K. (1996), "A New algorithm for automatic photopeak searches." *Nucl. Instrum. Meth.*, A376, 451–454.

Sinclair Glass (2018). 105 N Wabash Ave, P.O. Box 527, hartford City, IN 47348-2366. `http://www.sinclairglass.com`.

Jefferson Lab (2018a). https://www.jlab.org/12-gev-upgrade.

Jefferson Lab (2018b). https://www.jlab.org/Hall-C/upgrade/.

Vacek, Vic (1999), “P - T diagram for C4F10.” https://detector-cooling.web.cern.ch/detector-cooling/data/C4F10PTDiagram.htm.

Vovenko, A S, B A Kulakov, M F Likhachev, Yu A Matulenko, I A Savin, and V S Stavinskiĭ (1964), “Gas filled cerenkov counters.” *Soviet Physics Uspekhi*, 6, 794, URL http://stacks.iop.org/0038-5670/6/i=6/a=R02.

Wenliang Li, Keith Wolbaum, Garth Huber (2013), “Hamamatsu r1584 pmt modifications.” https://hallcweb.jlab.org/DocDB/0007/000765/001/pmt_modification_report.pdf.

Wilson, Julian T. (2014), *Quartz Hodoscope: Assembly, Calibration, and Data Analysis*. Master’s thesis, North Carolina A&T State University.

Yan, C., N. Sinkine, and R. Wojcik (2005), “Linear beam raster for cryogenic targets.” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 539, 1 – 15, URL http://www.sciencedirect.com/science/article/pii/S0168900204021874.

Yero, Carlos (2017), “Hall c 12 gev trigger set-up.” https://github.com/JeffersonLab/Hall-C-Trigger-Setup/blob/master/trigger_doc/trigger.pdf.

Yero, Carlos (2018), “Brief document on the hall c hodoscopes calibration.” https://hallcweb.jlab.org/DocDB/0009/000970/001/hodo_calib.pdf.