The Importance of Light Collection Efficiency in Radiation Detection Systems That Use Organic Scintillators

by

Charles Stephen Sosa

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Nuclear Engineering and Radiological Sciences)
in the University of Michigan
2018

Doctoral Committee:

Professor Sara A. Pozzi, Chair
Dr. David L. Chichester, Idaho National Laboratory
Dr. Shaun D. Clarke
Professor Zhong He
Professor Herbert Winful

"Twenty years from now you will be more disappointed by the things that you didn’t do than by the ones you did do. So throw off the bowlines. Sail away from the safe harbor. Catch the trade winds in your sails. Explore. Dream. Discover."

cssosa@umich.edu

ORCID iD: 0000-0002-0171-5636

DEDICATION

Para mi familia. Su paciencia y apoyo han sido de gran ayuda durante este tiempo dificil. Espero que disfrutes este trabajo. Muchas gracias para todo su ayuda, Carolina, Crae, Ignacio, y Elvira.

ACKNOWLEDGMENTS

First, I would like to acknowledge and thank my sponsors. This work was funded in-part by the Consortium for Verification Technology under the Department of Energy (DOE) National Nuclear Security Administration award number DE-NA0002534, and Idaho National Laboratory, through its Laboratory Directed Research and Development Program, under the DOE Idaho Operations Office contract DE-AC07-05ID14517. Second, I would like to extend a deep thanks and appreciation to my advisor, Professor Sara A. Pozzi. She saw my potential and gave me a chance to suceed. I am thankful for her mentorship, and for all those who have mentored me along the way. Mentorship plays a crucial role in the professional development and personal growth of any person. I have been fortunate to have many mentors who have dedicated both their time and energy to share their personal experiences and knowledge with me. I remember each of you fondly. I would also like to thank my family and friends - their kindness, compassion, and humor made this journey even more worthwhile.

TABLE OF CONTENTS

Chapter

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

CFD constant-fraction discriminator

CFT constant-fraction timing

DAQ data aquisition

FOM figure of merit

FWHM full width at half maximum

LCE light-collection efficiency

LLD lower-level discriminator

PMT photomultiplier tube

PSD pulse-shape discrimination

PTFE polytetrafluoroethylene

QE quantum efficiency

RI refractive index

SNM special nuclear material

TOA time of arrival

TOF time of flight

TRC total-reflectivity coefficient

ULD upper-level discriminator

VSC visual-scintillation counting

ABSTRACT

An organic scintillator is a transparent material that fluoresces when ionizing radiation interacts with it, making it a suitable radiation detector when coupled to a light-readout electronic device, such as a photomultiplier tube (PMT). The most commonly used organic scintillator shape is a cylinder: one face is coupled to the PMT, while the remaining surfaces are covered with a reflective material to increase the light collected by the PMT. This work demonstrates how modifying the shape and reflective-boundary of organic scintillators can improve light-collection efficiency (LCE) and, by extension, the performance of radiation detection systems, particularly for applications within nuclear nonproliferation and safeguard. Efforts to improve detector performance have historically focused on increasing light output through chemical means, the detection efficiency of light-sensing technology, and methods in data acquisition and processing. While vast research in these areas has shown improvements to detector performance, less research exists on the impact of the organic-scintillator shape and reflective-boundary conditions. This work compared the performance of conical and cylindrical organic scintillators of two materials (EJ200 and *trans*-stilbene) in three key areas of detector performance: energy resolution, time resolution, and particle identification; often referred to as pulse-shape discrimination (PSD). Conical EJ200 and stilbene outperformed their cylindrical counterparts in LCE by 17.6% and 18.0%, respectively. Gains in energy resolution were shown to be strongly dependent on the light output of the scintillator material and the quality of the spectral match between the wavelength emission of the scintillator material and PMT response. Conical EJ200 outperformed cylindrical EJ200 by 35% in time resolution. And conical stilbene outperformed cylindrical stilbene by 23% in PSD within a light-output range of 25 keVee to 100 keVee. The work also developed and validated a Geant4 model used to study the light-collection process in the cone, cylinder, and various other geometries.

CHAPTER 1

Introduction

1.1 Scintillator-based radiation detection systems

One of the oldest techniques for detecting ionizing radiation is the use of a scintillator material [1–6]. These unique materials produce light when ionizing radiation deposits energy inside of them. Introduced by Crookes and Regener in 1908, visual-scintillation counting (VSC) played an important role in nuclear physics research, particularly in understanding the nature of alpha particles [2]. VSC consists of a researcher using a thin sheet of activated zinc sulfide (ZnS(Ag)) in a dark room to visually record the number of optical photons (i.e., scintillation-light) produced from alpha-particle interactions. Since ZnS(Ag) is not transparent, it was crucial that the room be completely dark and that the researcher have excellent eye sight to collect the most amount of light possible from these faint scintillation events. Without the observer sufficiently collecting enough light to their eyes, little to no information regarding the true count rate from the ZnS(Ag) sheet could accurately be interpreted. The eye-sight clarity of the observer was therefore important and is analogous to the theme of topics discussed in this work; the importance of light-collection. Despite its tediousness, VSC remained in use for many years until the photomultiplier tube (PMT) was developed in the late 1930's [7]; a light-sensing device. The PMT was able to count optical photons at higher rates than humanly possibly in addition to producing an electrical pulse from a scintillation event.

In the late 1940's and early 1950's, organic liquid and plastic scintillators were developed [3].

Their transparency to optical photons allowed the scintillator volume to be much thicker in contrast to ZnS(Ag), thus increasing the count rates, making PMT use a necessity. Over the years, these materials became commercialized and their improvement was largely focused on their chemistry to produce a brighter scintillation response (i.e., light yield) to ionizing radiation rather than finding a shape better suited than a cylinder to collect more scintillation light to the coupled PMT. How and why scintillator geometry affects the performance of a scintillator is explored in this work. As the scintillator size grew in the later 1950's and 1960's, so did concerns regarding the impact of the reflector on the light-collection process [8].

With the rapid development of organic scintillators, some unique materials such as anthracene and *trans*-stilbene [1, 2, 4, 5, 9, 10] were observed to produce longer scintillation-decay times for certain types of radiation interactions. These types of organic scintillators are categorized as pulse-shape discrimination (PSD) capable. For example, a greater fraction of delayed light is produced from fast-neutron interactions than from gamma-ray interactions for the same amount of total scintillation-light produced. This unique feature was extremely attractive to the study of prompt gamma rays and neutrons from fissile sources, including special nuclear material (SNM): any material containing uranium-235 enriched to 20% or more, uranium-233, or plutonium-235. Therefore, PSD-capable organic scintillators became the subject of extensive research and are still used today in a variety of fields where particle identification is required in mixed radiation fields, for example in nuclear security. This work discusses the impact of organic-scintillator geometry and reflector conditions on detector performance, with a focus on detector systems applied to nuclear nonproliferation and international safeguards research [5, 11].

In their most basic form, these detectors are typically comprised of a cylindrical organic scintillator coupled to a light-readout electronic device, such as a PMT, which is designed to detect optical photons, convert them into electrons, and produce an electronic pulse. The remaining surfaces of the scintillator are typically wrapped in a diffuse reflector, such as white titanium-dioxide paint or polytetrafluoroethylene (PTFE) tape. A visual representation of the detection process is shown in Figure 1.1.

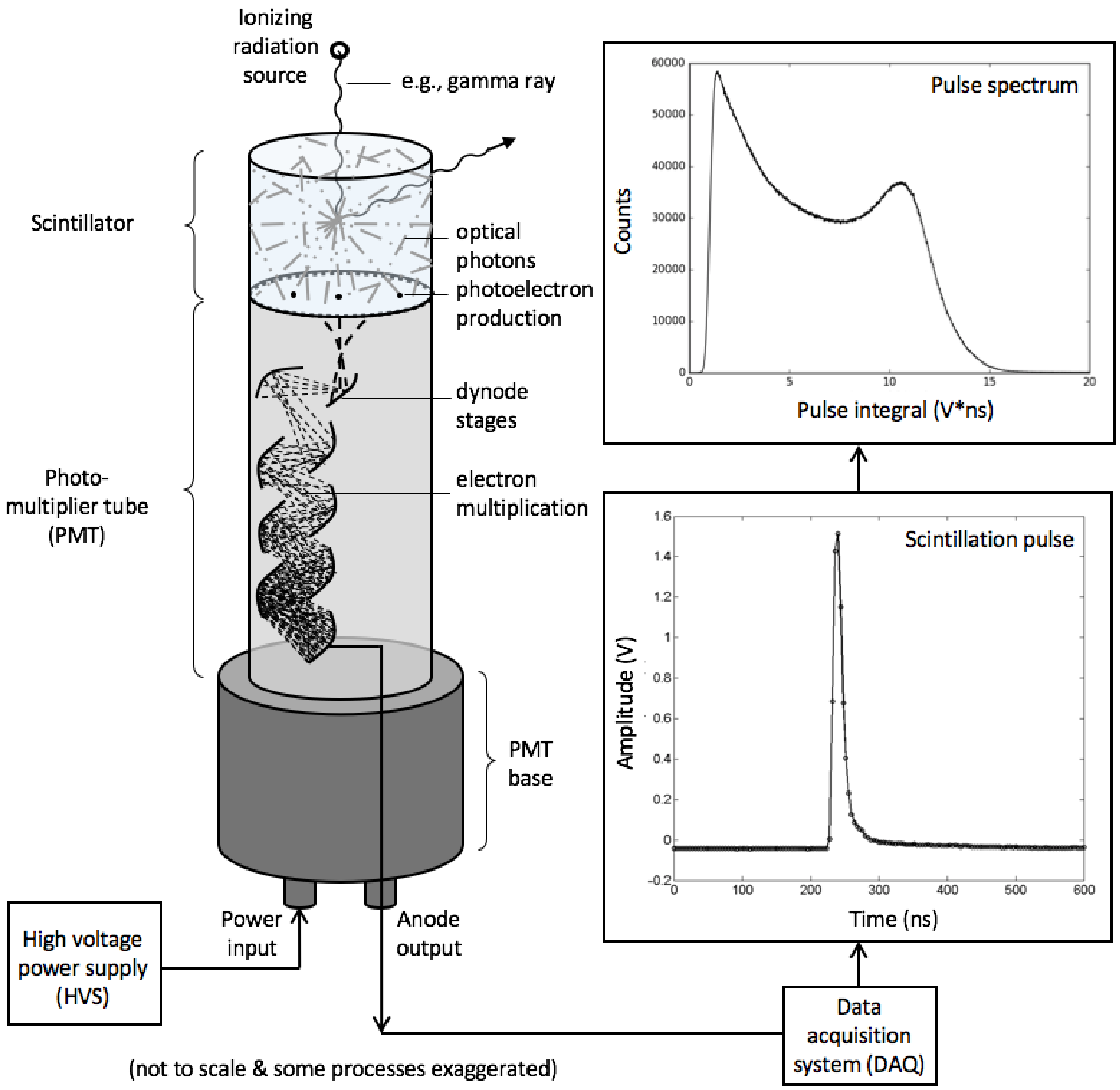


Figure 1.1: A representation of the detection process of a gamma ray as measured by an organic scintillator coupled to a PMT: optical photons are collected by the PMT that produces a pulse integral. The integral of the pulse contributes a single count to the pulse-integral distribution (top right). Over time, many pulses are collected and contribute more counts to the distribution.

Each pulse represents the time of arrival (TOA) of optical photons to the PMT, convolved with the scintillation process and the PMT response. The pulse integral is correlated to the energy deposited by the particle, where a histogram of many pulse integrals collected over time creates a distribution that is proportional to the energy spectrum of the detected radiation. Additionally, when a PSD-capable organic scintillator is coupled to a PMT, the shape of the pulse will vary according to the particle type. Therefore, this difference in pulse shape can be used as a way to identify the particle type [12–15]. The capability of a scintillator to identify particle types is very attractive to detector applications within the field of nuclear security.

Under the umbrella of nuclear-security research, detectors that use PSD-capable organic scintillators are often uniquely configured with other equipment, including but not limited to other types of detectors, data-acquisition hardware, nuclear-instrumentation modules, and data-processing techniques, to achieve a greater purpose than simply counting radiation interactions. These larger systems are more complex but are designed to return important information that can be used, for example, to study fission, characterize a fissile source, find radiological material, and thwart nuclear-material trafficking. Therefore, these systems are researched extensively at many universities and national laboratories in the United States before being deployed. For example, fast neutron-multiplicity counters [16, 17] are being developed to replace the current helium-3 based multiplicity counters; used to assay the fissile-mass content of an unknown neutron-emitting source.

The performance of these systems, or any scintillator-based detection system for that matter, is largely assessed by certain metrics. These metrics include the energy resolution, time resolution, and particle identification, which are all strongly dependent on light-collection efficiency (LCE). Detection efficiency can be equally important for certain applications that are count starved. However, it can be argued that one of the greatest impacts to the fundamental operation and performance of any scintillator-based system is the LCE.

1.2 Motivation

1.2.1 Light-collection efficiency

An ionizing-radiation particle such as a gamma ray or neutron that deposits energy inside an organic scintillator will typically produce photons from scintillation in the visible spectrum. Any impedance in the collection of these optical photons to the PMT will affect detector performance. LCE is defined as the fraction of optical photons detected by the PMT from the total amount produced following a radiation interaction.

The total LCE depends on several factors including but not limited to: the number of optical photons produced as a function of the energy deposited in the scintillator (i.e., light yield), re-absorptions in the scintillator, and reflections at the scintillator-reflector boundary [2, 5]. LCE is additionally influenced by variations in optical-photon detection efficiency (also referred to as quantum efficiency (QE)), which changes as a function of position along the optical window of a PMT. The spatial variation in QE is attributed to photocathode non-uniformity and photoelectron trajectories that miss the first dynode [18–21]. Moreover, typical peak-QE values are low to begin with (25-35%) [5]. As a result, typical LCE values for cylindrical scintillators coupled to a PMT with a bialkali photocathode can range from 10 to 20% [5]. An improvement in LCE would increase the pool of statistical information available per scintillation event for the PMT to detect and use to produce a pulse whose integral better represents the energy deposited.

One method for improving LCE is to modify the scintillator geometry to reduce optical-photon reflections prior to detection. Each time an optical photon reflects at the scintillator-reflector boundary, its chance of detection at the photocathode is probabilistically reduced by the increased likelihood of transmission or absorption in the reflector material. The reflection probability (R_{T}) at a scintillator-reflector boundary is influenced by the optical-photon wavelength, the reflector material, the quantity and thickness of the layers used, as well as the scintillator-surface treatment [2, 5, 22, 23]. Experimentally measured reflection probabilities at a scintillator-reflector boundary have been reported as high as 0.95 [22]. However, the detection probability of a single optical photon is reduced by the reflection probability raised to a power given by the number of times (N) it reflects. Eq. 1.1 describes a simplified analytical expression for optical-photon detection probability (P_{d}), where the importance of reducing reflections is emphasized. In reality, Eq. 1.1 includes additional factors that further reduce the P_{d}, for example the transmission probabilities of the optical-coupling grease and PMT window. However, the greatest impact to the detection probability of an optical photon is the QE of a PMT, followed by the boundary reflectivity, and the number of reflections. Figure 1.2 uses Eq. 1.1 to show how the detection probability of a single photon is: (1) limited by QE, (2) sensitive to the reflection coefficient, and (3) reduced by the number of times it reflects.

$$P_d(N) = \frac{QE}{100\%} * {R_{\mathrm{T}}}^N \tag{1.1}$$

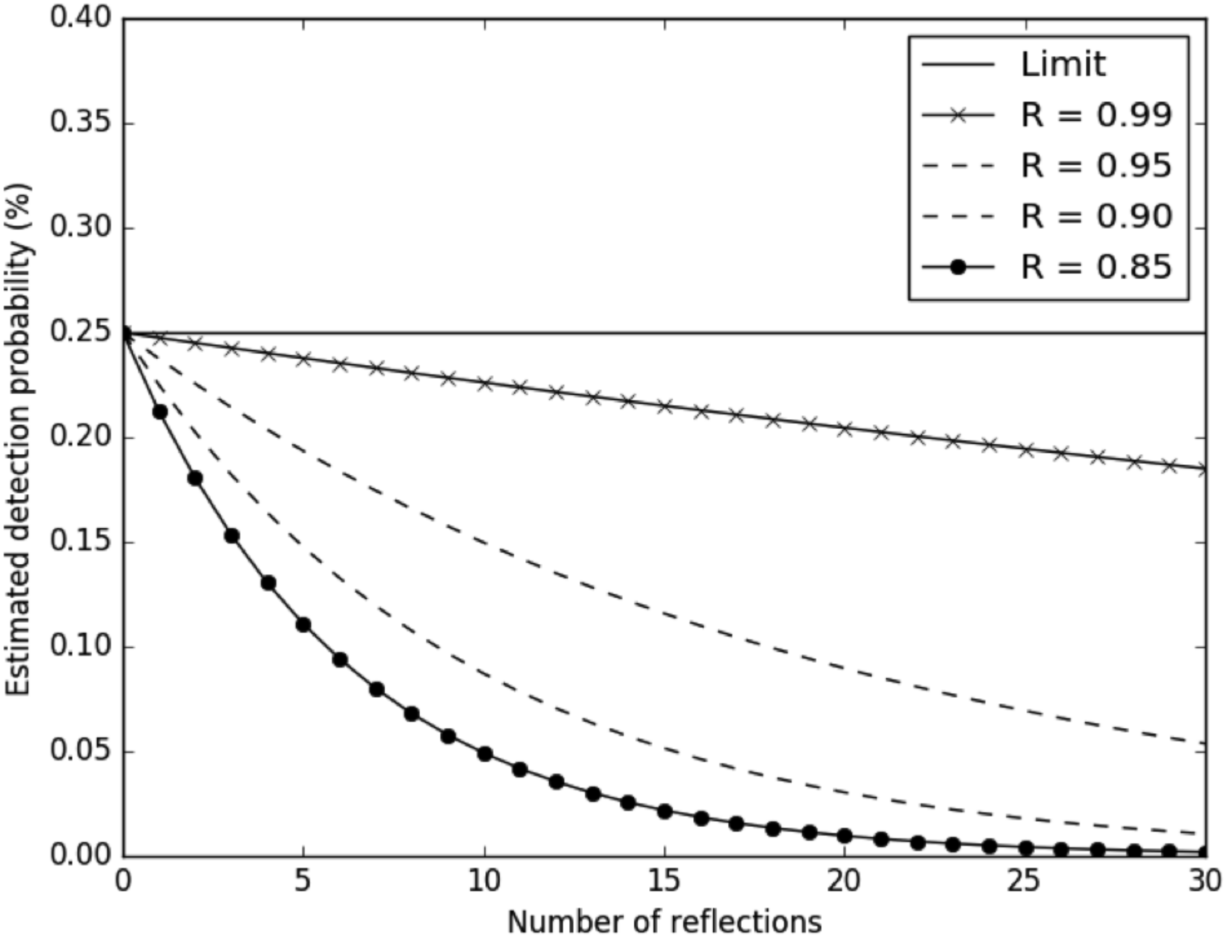


Figure 1.2: A simplified expression of the optical-photon detection probability as a function of: (1) the reflection probability at the scintillator-reflector boundary, and (2) the reflection number. An important aspect of this work is to reduce reflections in order to increase the optical-photon detection probability.

Figure 1.3 shows the base of an organic scintillator being exposed to a black light to promote illumination. The scintillator has been painted with a thick layer of titanium-dioxide paint for use as a reflector. The importance of the reflectivity at the scintillator-reflector boundary is emphasized by this figure. Note how a non-trivial quantity of light leaks through the reflector. The same phenomenon occurs for scintillation events generated by radiation interactions. These transmissions represent a loss in statistical information.

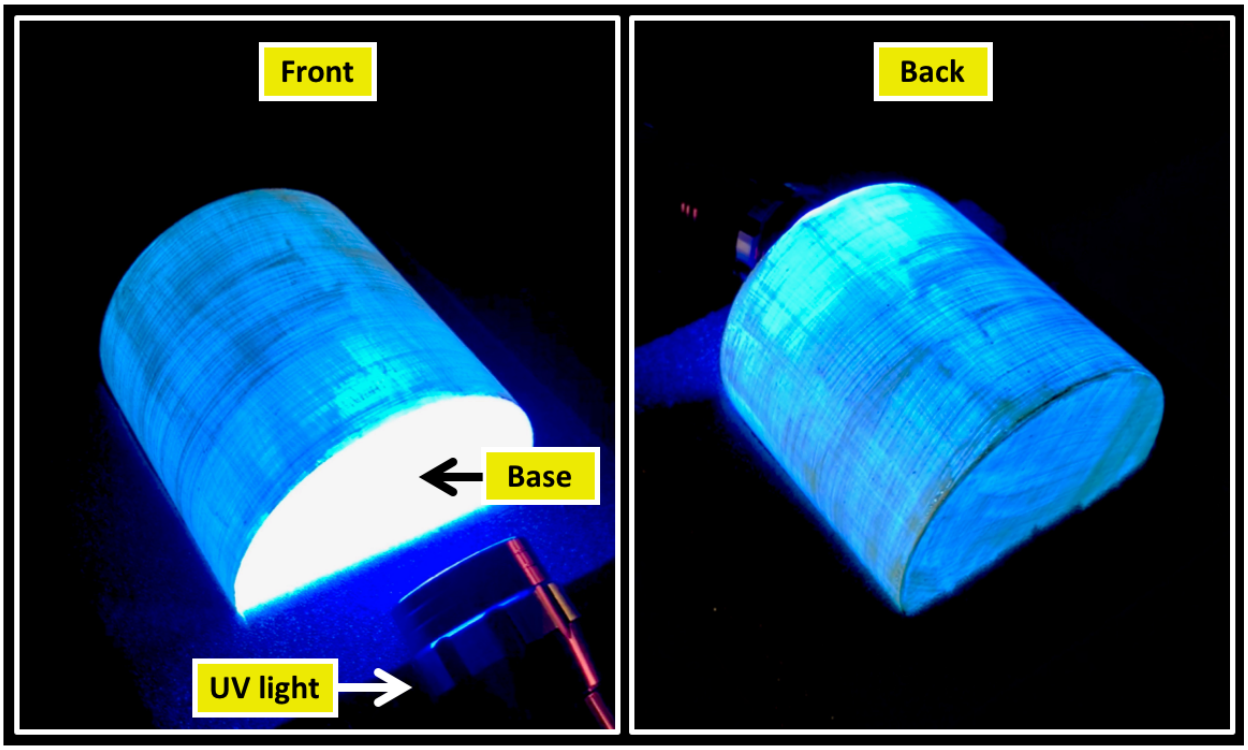


Figure 1.3: A plastic scintillator is shown transmitting its blue-wavelength emission through the reflector. Transmission through the reflector degrades LCE and detector performance.

1.2.2 The majestic elk analogy

An analogy can be made that highlights the importance of LCE in radiation detection systems that use any scintillator-based material. Figure 1.4 shows four pictures of the same elk taken in Yellowstone National Park. The top left picture is the best case scenario where the sun is out and eye sight of the person viewing the elk is unaffected by cataracts. The conditions are analogous to a bright scintillation event captured by a PMT with perfect QE.

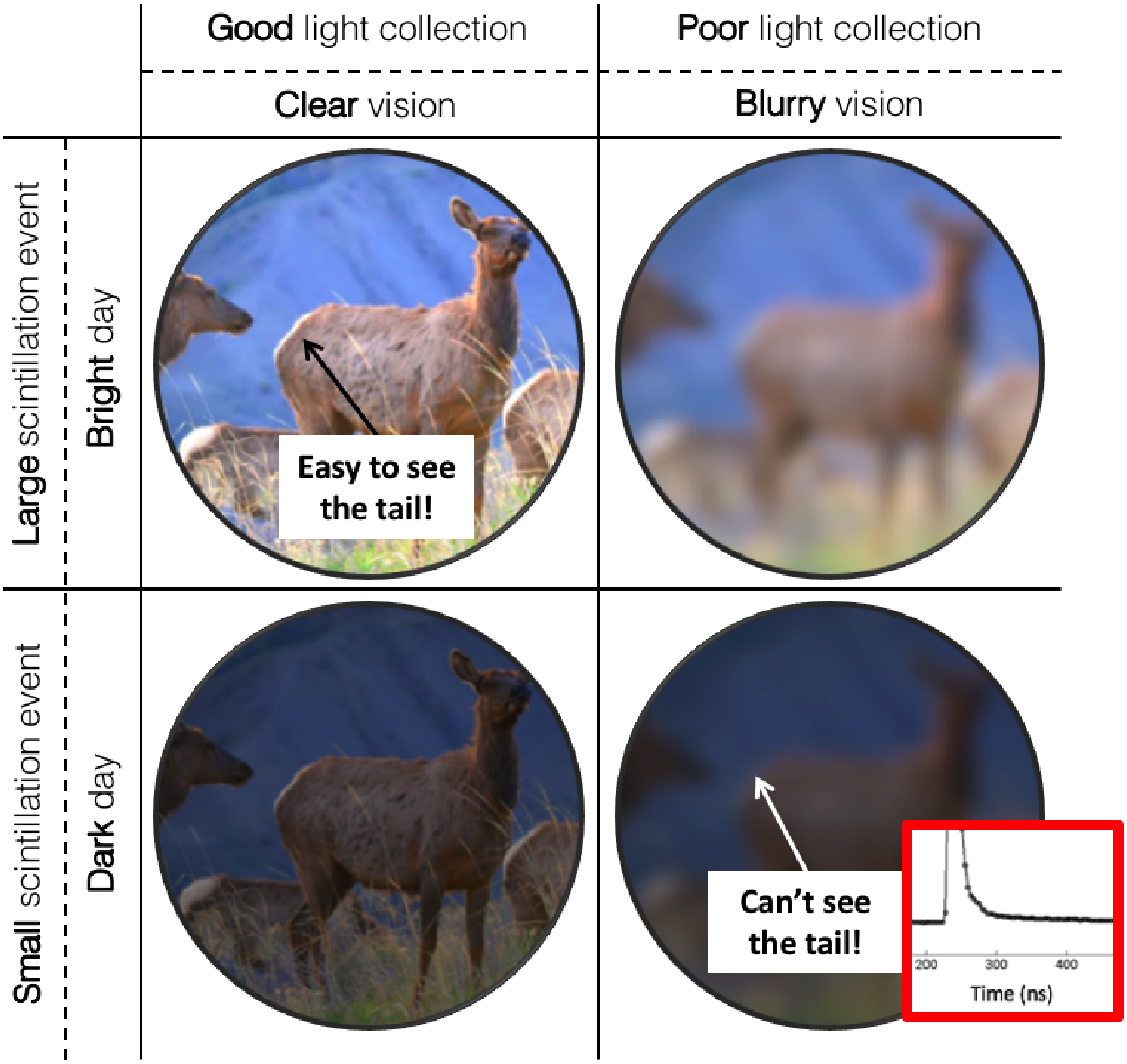


Figure 1.4: An analogy that describes the importance of LCE in everyday life; for example interpreting unique features of a majestic elk.

The bottom right picture is the worst case scenario where the sun has almost set and the person trying to view the elk suffers from cataracts. These conditions are analogous to a low-light scintillation event captured by with a PMT with a realistic QE. Overlaid on the bottom right image is the tail-end of a scintillation pulse, which is analogous to the bushy-white tail of the elk. In the best-case scenario, the elk tail is clearly visible, however, in the worst-case scenario, the tail is barely visible. The importance of this tail discussion will become more evident with discussions of how particle identification works in a PSD-capable organic scintillator. In the same way LCE is important for accurately interpreting the features of any object in dimly-lit conditions, such as the elk tail, it is equally important for interpreting features of a scintillator pulse for particle identification.

1.2.3 Literature review

Previous works have examined the effects of detector geometry and reflectivity conditions on detector performance for a variety of applications [24–28]. Another article, written by the author of this thesis, was recently published, and discussed the measured improvement in energy resolution when using a conical organic scintillator instead of the standard right circular cylinder [29]. These results are reported in Chapter 4. Only two journal articles were found that compared measured detector-performance metrics of a truncated cone and cylinder: one on energy resolution [30] and the other on time resolution [31]. The main results from both studies are discussed below. However, no work has been found that investigates the PSD performance of a conical scintillator.

The closest related work on energy resolution reports on a comparison of the energy resolution of a tapered cone and cylinder of a BGO inorganic scintillator [30]: both were ground, wrapped in PTFE tape, had a base diameter and height of 1.0 cm, and were optically coupled to the center of a 3.8 cm diameter PMT (Hamamatsu R980). Results demonstrated that the cone had worse energy resolution (10.5%) than the cylinder (9.5%), despite having an approximately 10% increase in pulse amplitude over the cylinder at 662 keVee. Three reasons are strongly suspected for the unexpected results reported in their study. First, the authors described wrapping the reflector around the cone as a difficult process. Any variability in the thickness of the PTFE tape would alter the uniformity of the reflection probability for optical photons as a function of position along the scintillator-reflector boundary [23], thus making it difficult to fairly compare the two geometries. PTFE tape was not used for any experiment in this work to avoid the possibility of applying an unequal amount of layers to the cylinder and cone. Doing so would bias performance-comparison tests. Secondly, the authors processed their data using the pulse-height technique instead of the pulse-integral technique. Previous work has shown that the pulse-integral technique is a more accurate representation of the energy resolution [32]. As previously illustrated in Figure 1.1, a detector pulse is a convolution of processes, including the TOA of optical photon detections. The integral of the pulse is a more accurate representation of the scintillation event and when comparing the subtleties in the energy resolution between two geometries it is conceivable that this data-processing technique could have an affect on the results. And third, as it will be shown in the simulation component of this work, some geometries, including some tapered cones, produce a poor energy resolution in comparison to a cylinder of equal base diameter and height due to effects from Gamma-ray scatter and optical-photon reflections (see Chapter 7).

The other closely related work compares the time resolution of a tapered cone and cylinder of an organic scintillator (EJ228) [31]: both shapes were polished, covered in white paint, and had a base diameter and height of 2.5 cm. The PMT model used in the study was an RCA C31024 with an outer diameter of 5.1 cm. Results demonstrated that the cone had improved time resolution over the cylinder by approximately 20.6%.

1.3 Contribution and overview

1.3.1 Thesis demonstration

Historically, efforts to improve LCE have largely focused on the chemical composition of organic scintillators to produce greater light output per energy deposited. The goal of this work was to demonstrate measurable gains in three key detector-performance metrics (energy resolution, time resolution, and PSD) by modifying the organic scintillator geometry from the standard cylindrical shape to a cone of equal base diameter and height. A description of how LCE impacts each metric is enumerated below, along with the primary results demonstrated in this thesis.

1. Energy resolution

   This performance metric describes how well a scintillator can identify features of an energy spectrum from a radiation source (see the top right plot of Figure 1.5). A cone is expected to reduce optical-photon reflections inside the scintillator volume, thereby lowering the chance of optical-photon transmission or absorption at the reflector; thus increasing LCE. An increase in LCE is expected to improve energy resolution. This hypothesis was confirmed by demonstrating a 16% increase in energy resolution by using conical E200 (see Chapter 4).

2. Time resolution

   This performance metric describes how quickly a scintillator can respond to a radiation interaction and still identify a second interaction (see the top left plot of Figure 1.5). If optical photons reflect less in the cone than in the cylinder, then it is reasonable to expect that their average path length and time for detection is also reduced. Moreover, the faster light-collection process also suggests that the pulse rise-time may be reduced with the additional presence of optical-photon contributions towards the front end of the pulse. An increase in counting statistics along the rising edge of the pulse is expected to improve time resolution. This hypothesis was confirmed by demonstrating a 35% increase in time resolution by using conical E200 (see Chapter 5).

3. Pulse shape discrimination

   Only some organic scintillators are capable of PSD, which is also referred to as particle identification. This performance metric describes how well a scintillator can distinguish between different types of radiation-particle interactions as a function of the light output they generate (see the bottom left plot of Figure 1.5). It is known that PSD performance degrades at low-light output events (<100 keVee) [14, 15] where gamma-ray and neutron pulses appear similar. An increase in LCE may counter the degrading PSD performance for low-light output events. This hypothesis was confirmed by demonstrating a 23% increase in PSD by using conical stilbene (see Chapter 6).

4. Simulation

   A simulation package in Geant4 [33] was developed and validated with experimental techniques to study the transport of optical-photon in any scintillator geometry. The simulated percent-gain in LCE (17.5%), energy resolution (16.2%), and percent-loss in absolute detection efficiency (70%) by using conical EJ200 agreed well with measured results within less than (0.6%). The loss in ionizing-radiation detection efficiency (refer to bottom right plot in Figure 1.5) was expected due to the decreased volume of the cone. Although this may not be ideal for certain applications where high-count rates are required, the gains achieved by the cone in the three other metrics as compared to the standard cylinder are significant and can improve certain detection systems that require excellent energy resolution, time resolution, and PSD performance (see Section 8.2).

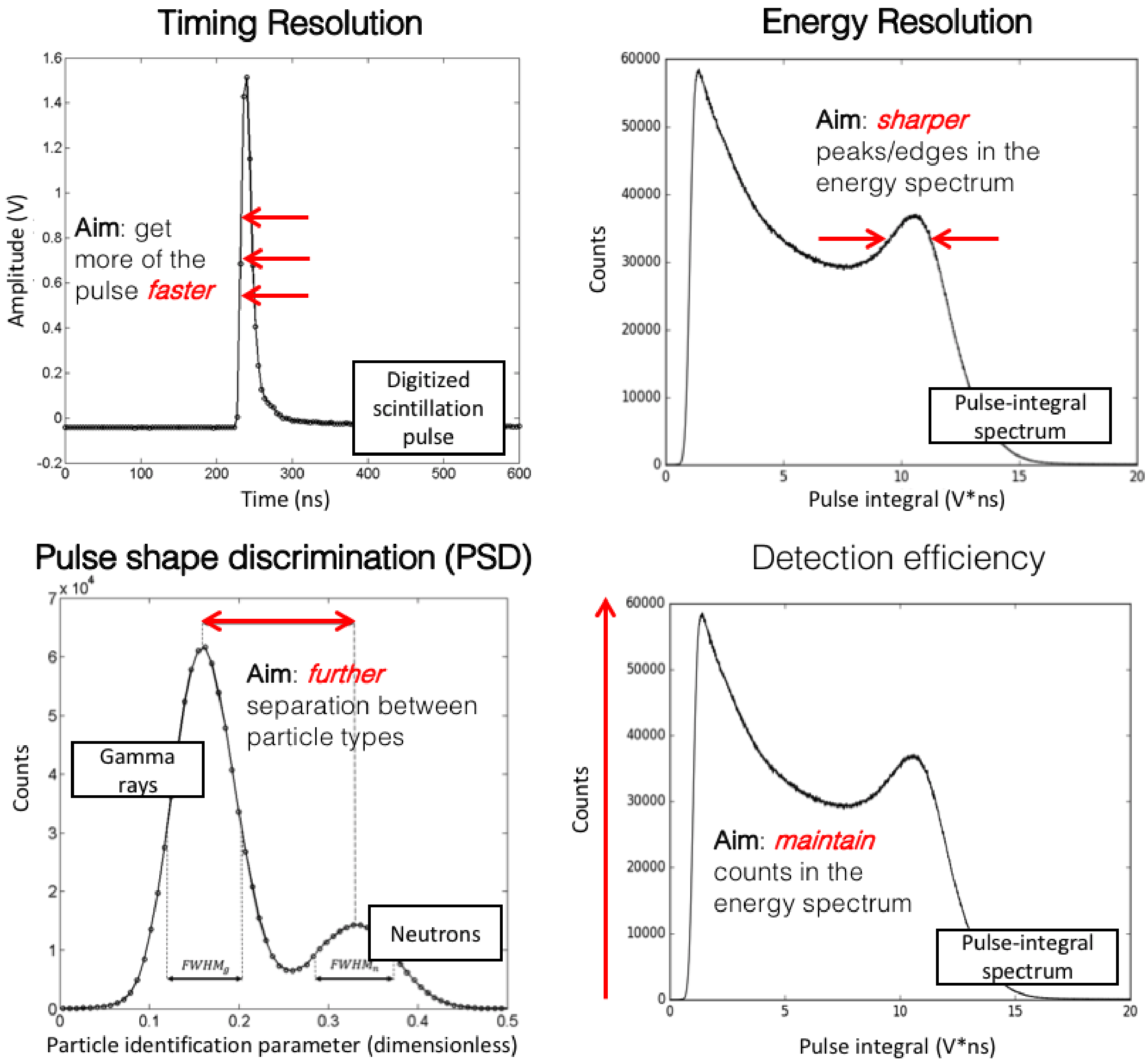


Figure 1.5: The performance metrics in bold are criteria that this thesis aims to improve upon by using a conical scintillator instead of the standard cylinder. Detection efficiency will also be assessed but is not the primary focus of this work.

1.3.2 Potential applications

As demonstrated in this thesis, an increase in LCE positively impacts the three detector performance metrics discussed earlier. These improvements carry significant contributions to the overall performance for a number of detection systems that use scintillators. For example, an increase in LCE would enable any scintillator-based detection system to operate at a lower light-output threshold which is synonymous with a lower minimum-detectable activity threshold, improve the calibration accuracy from light output to energy deposited, and enhance spectroscopic capability [34]. These changes would especially have a positive impact on applied systems in nuclear nonproliferation and safeguards research that make use of PSD-capable organic scintillators, for example fast-neutron multiplicity counters and fast-neutron scatter cameras. Both are further described in the list below.

1. Fast-Neutron Multiplicity Counters

   These systems can be used to measure the fissile mass content of a multiplying source, such as SNM [16, 17]. A typical system configuration consists of a source surrounded by multiple PSD-capable organic scintillators. The simultaneous detection of two or three fast neutrons, called neutron multiples, are said to be correlated with the same fission event. Neutron multiples are counted over time and can be used to estimate the fissile-mass content.

2. Fast-Neutron Scatter Cameras

   These systems can be used to localize neutron sources [35–39] and can vary in size. Planar arrays of PSD-capable organic scintillators are used to correlate the single scatter of a neutron in one detector with its subsequent interaction in another. Since the position and distance between each scintillator volume is known, correlated neutron scatters can use this information along with the neutron time of flight (TOF) to calculate the incident-neutron energy and location. Fast response times are important for neutron-scatter cameras. The time resolution of each detector must be narrower than the flight time between scatters to achieve reasonable localization accuracy.

In both applications and potentially many others, conical organic scintillators capable of PSD could be used in place of the current standard cylindrical geometries to improve light-output calibration, reduce time pick-off uncertainty, and achieve greater separation between neutron and gamma-ray pulses at low-light output events to counter PSD degradation.

1.3.3 Chapter descriptions

The following contents include a brief summary of each chapter:

Chapter 2: A background on the production of scintillation light in organic scintillators and a discussion about why some scintillators are capable of uniquely responding to particular types of ionizing particles. Optical-photon physics is discussed as well as the impact that a PMT can have on detector performance.

Chapter 3: A detailed explanation of the equipment and experimental techniques used in this work to measure the energy resolution (Compton-coincidence), time resolution (cross-correlation), and PSD performance (charge integration).

Chapter 4: A discussion of the methods and results from the energy-resolution experiments of conical and cylindrical EJ200. Photocathode non-uniformity is assessed and its impact on the energy resolution is demonstrated and discussed.

Chapter 5: A discussion of the methods and results from the time-resolution experiments of conical and cylindrical EJ200.

Chapter 6: A discussion of the methods and results from the PSD experiments of conical and cylindrical *trans*-stilbene.

Chapter 7: A discussion of optical-photon transport methodologies in Geant4, the model development, the model validation techniques, and a discussion of various simulated results, including an efficiency trade-off study.

Chapter 8: A detailed summary and discussion of the main results is presented along with general conclusions and suggestions for future work.

CHAPTER 2

Background

2.1 Organic scintillators

This section offers a condensed and simplified explanation of optical-photon production resulting from ionizing-particle interactions in organic-scintillator material. For a more comprehensive description of how and why light is produced in organic scintillators, the reader is pointed to the following references that describe the associated processes in much greater detail [2, 5, 13].

2.1.1 Overview

Organic scintillators include liquid and solid (plastic or crystalline) materials (see Figure 2.1) that can be used to detect radiation. When a sufficient amount of energy is deposited from radiation, organic scintillators will produce a measurable quantity of optical photons that is proportional to the energy deposited by the particle, which is often referred to as the light yield [1–6]. The phrase, light output, is sometimes used to describe the same quantity. Figure 2.2 shows a graphical representation of the electromagnetic spectrum and the small region occupied by the visible spectrum (390 nm to 780 nm) [40]. Most organic scintillators will typically emit a spectrum of optical-photon wavelengths that peak near 425 nm [2, 5].

For most nuclear-security applications, organic scintillators are commonly associated with the detection of fast neutrons, gamma rays, and sometimes beta particles. This work is primarily concerned with gamma rays that range in energy from approximately 25 keV to a few MeV and fast neutrons that range from approximately 450 keV to a few MeV. Some organic scintillators that are PSD-capable [1, 2, 4, 5], for example *trans*-stilbene [9, 10], can be used to identify particle types. The amount of light produced as a function of time in Stilbene will differ based on the type of radiation, and this difference in the pulse shape can be used by PSD techniques, such as digital charge integration or analog zero crossing [12–15], to identify the particle type. These types of organic scintillators are categorized as PSD capable and are very important for nuclear-security research where gamma-ray and neutron discrimination is necessary in mixed-radiation fields.

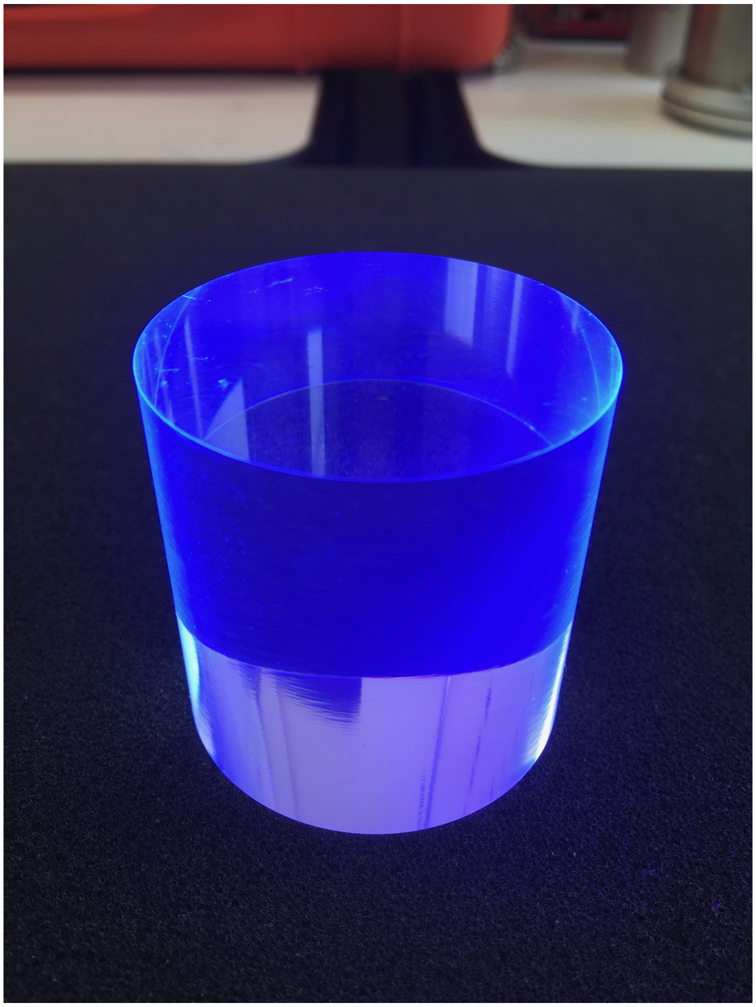


Figure 2.1: A bare machine-polished non-PSD capable plastic scintillator (EJ200), without its reflector, is shown exposed to ultraviolet light to promote illumination. The peak emission of EJ200 is approximately 425 nm.

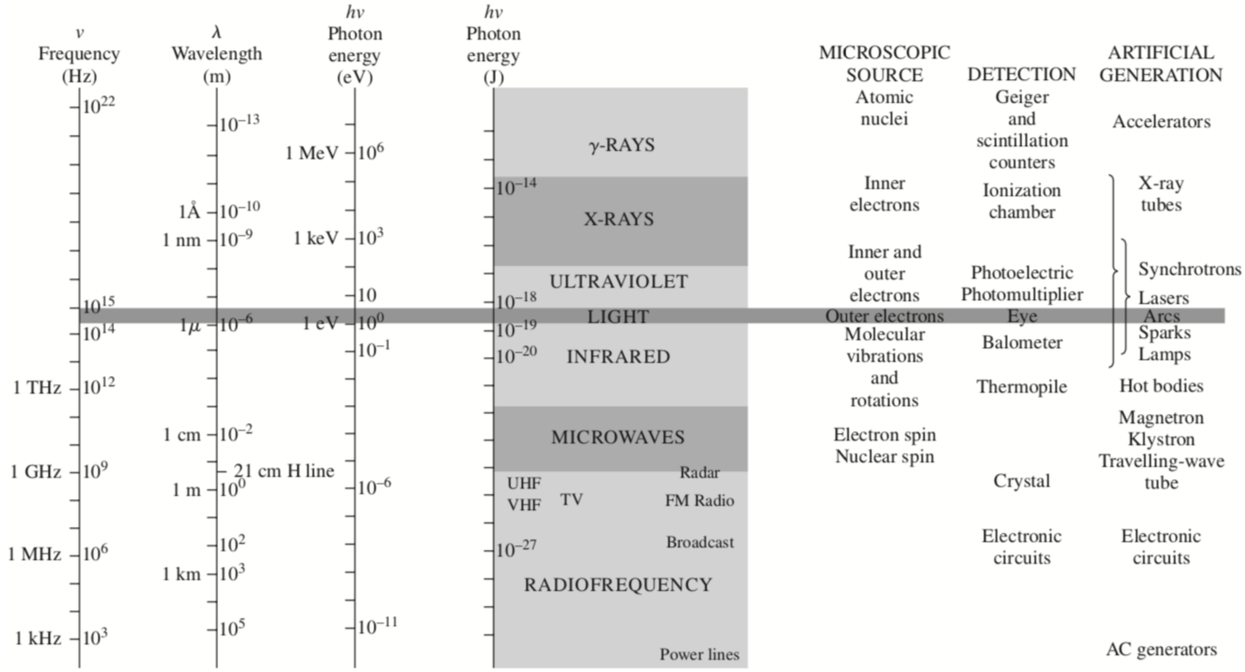


Figure 2.2: A diagram that encapsulates the vast range of photon energies in the electromagnetic spectrum; taken from Hecht's textbook on *Optics [40].*

2.1.2 Particle interactions

Figure 2.3 shows a relationship between the atomic number of a material, the energy of a gamma ray and its most likely interaction mechanism. Due to the low atomic number of organic scintillators, the predominant gamma-ray interaction is Compton Scattering. The photoelectric process is also greatly suppressed. With sufficient initial energy, pair production can occur if a gamma-ray scatter deposits an energy that meets or exceed the pair-production threshold (1022 keV), however this process is rare and is not commonly observed in organic scintillators.

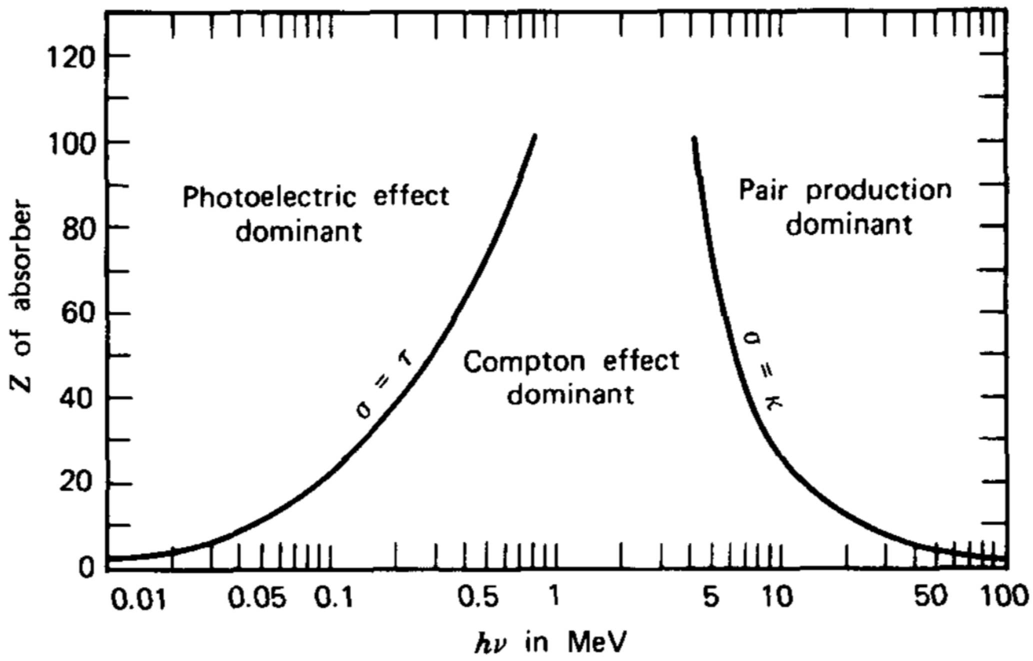


Figure 2.3: The predominance of gamma-ray scatter regimes as a function of energy and atomic number; taken from Knoll's textbook on *Radiation Detection and Measurement [5].*

The most common interaction mechanism for fast neutrons in organic scintillators is elastic scattering on hydrogen, which produces recoiled protons. Based on scatter kinematics, fast neutrons can deposit any fraction of their energy to the proton. Neutrons can elastically or inelastically scatter on carbon nuclei. Although both processes are equally likely, carbon recoils following inelastic scatter produce a negligible fraction of light output as compared to elastic scattering on protons [41]. Figure 2.4 illustrates the predominant scatter process for both gamma rays and fast neutrons in organic scintillators, and includes an equation that relates the initial energy and scatter angle to the final energy of the particle.

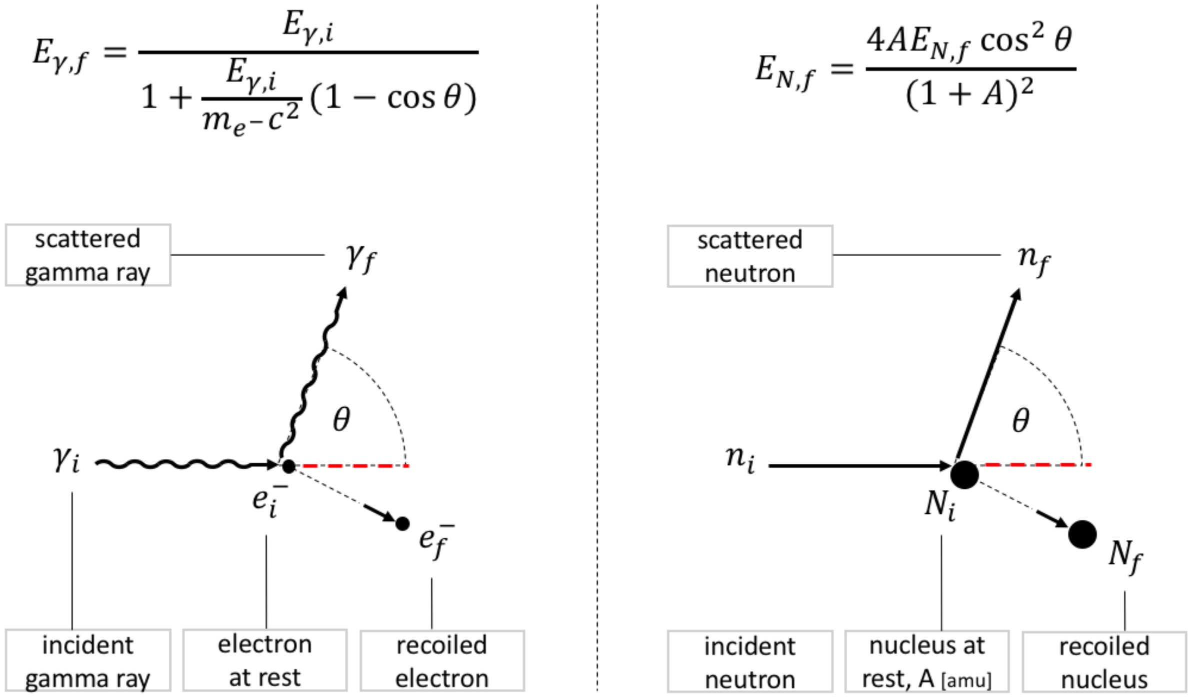


Figure 2.4: The predominant mode of gamma ray and fast neutron interactions on a free electron and hydrogen nucleus, respectively.

When ionizing-radiation particles interact in the scintillator, a fraction of the energy that gets deposited is converted into molecular excitations and ionizations that can emit optical photons, where the quantity and time-scale of optical-photon emission can depend on the incident particle [2]. Examples of ionizing radiation includes beta particles (electrons), alpha particles, x-rays, gamma rays, and neutrons. In this work, the particles of interest are primarily gamma rays and fast neutrons. A gamma ray that Compton scatters will produce a recoil electron and a fast neutron that elastically scatters will produced a recoil proton, as shown in Figure 2.4.

2.1.3 The benzene ring

Whether an organic scintillator is a plastic or solid material, the shared characteristic that enables scintillation to occur is the aromatic structure of the hydrocarbon bonds. In organic chemistry, the word *aromatic* is used to describe a molecular bond that is connected in a circular fashion.

In the case of an organic scintillation, the aromatic hydrocarbons responsible for the scintillation process are based on the benzene ring (C_6H_6). Benzene rings can be interlinked in a variety of ways to produce different scintillation properties. The unpaired valence electrons of each carbon atom in the benzene ring de-localize above and below the ring to form π-bonds where electron sharing can occur across the molecule. These de-localized electrons are also referred to as π-electrons. The sharing of π-electrons produces a π-electronic system that includes a singlet and triplet state with quantized energy and vibrational levels. These states are electron-spin dependent, have unique excitation mechanisms and scintillation-emission times, can migrate in the material, and are partially responsible for the PSD capability of some organic scintillators.

2.1.4 Light emission

Singlet state de-excitations are responsible for the scintillation light observed from a radiation interaction in an organic scintillator. An excited singlet state that relaxes to the ground state produces an optical photon on the order of 10^{-8} s, and this process is referred to in literature as prompt florescence. Due to the migration capability of these electronic states, another florescence emission can be observed, referred to as delayed florescence, and is essential to PSD. Electronic-state migration can also quench (i.e., reduce) the prompt-singlet light output from a radiation interaction.

2.1.5 Electronic-state migration

If the track length of a directly-ionizing particle is very small, then the excitation density is increased per unit length, and so is the likelihood that singlet states can migrate near each other [2]. Two excited singlet states that are in close proximity to each other may combine in a process called singlet quenching. In this process, a single optical photon is produced instead of two had the states not been in close proximity, and thus the amount of light that could have been produced is reduced. Singlet quenching in organic scintillators is the reason why the light output as a function of energy deposited becomes non-linear at low-energy deposition events where either the electron or proton recoil tracks are short enough to promote singlet-state annihilation. Singlet quenching especially affects neutron interactions where the excitation density along the proton-recoil track is high. Thus, the neutron light output is less than a gamma ray that deposits the same amount of energy. Some electronic-state migrations are useful, however, because they enable an organic scintillator to be PSD capable. When two triplet states migrate in close proximity to each other, they will annihilate and produce an excited singlet state that will immediately de-excite to the ground state by the prompt emission of an optical photon. Triplet-triplet annihilation is therefore associated with delayed florescence, which play an important role in PSD.

2.1.6 Pulse-shape discrimination

Electron and proton recoils will produce both molecular excitations and ionizations along their track [2] in an organic scintillator. Molecular excitations will largely result in excited singlet states that will immediately produce prompt florescence. Direct population of the triplet state can occur, however only through recombination. After ionization occurs, a π-electron is ejected and thus a vacancy is created. The recombination of this vacancy with another π-electron shared in the π-bonds of the molecule will produce either an excited singlet or triplet state. The probability of producing a singlet or triplet state from recombination is reported in literature to be approximately 25% and 75%, respectively [2]. Triplet states produced from recombinations can migrate, annihilate, and produce an excited singlet state. In other words, triplet-triplet annihilations from recombinations are the reason delayed florescence can occur. Therefore, if triplet migration is not impeded, then the organic scintillator may be PSD-capable and respond uniquely to electron and proton recoils. Therefore, if electron-state migration is impeded in a particular organic scintillator material, then PSD cannot occur.

If triplet migration in a particular organic scintillator is uninhibited, then the final piece of the puzzle to understand why the scintillator can produce a unique scintillation response to gamma ray (electron recoil) and neutron (proton recoil) events, is the penetration depth of the recoil particles. The higher linear-energy transfer of the proton means that the proton will travel further in the material than the electron. In other words, the proton will deposit its energy over a longer track than the electron. Therefore, the excitation and ionization density along the proton track is less than the electron track.

2.2 Optical photons

This section offers a condensed and simplified explanation of optical-photon interactions in organic-scintillator material. For a more detailed and comprehensive description of how and why light interacts with matter the way it does, the reader is pointed to the field of quantum-electro dynamics (QED); where the study of quantum mechanics and electromagnetic radiation is combined.

2.2.1 Definition

A single optical photon is considered non-ionizing electromagnetic radiation due to its low energy (a few eV) [2,5,40]. Optical photons are commonly associated with the visible spectrum, however, not all optical photons are visible to humans. For example, ultraviolet photons (10 nm to 390 nm) and infrared photons (780 nm to 1 mm) are invisible. A photon is considered “optical" when its energy corresponds to a wavelength that is much larger than the distance between adjacent nuclei; referred to as the atomic spacing in a material (typically a few Angstroms). The distance between atoms in a covalent bond, for example a hydrocarbon bond in benzene, is approximately 1-2 Angstroms (1.5 nm). Note how this distance is much shorter than the peak-wavelength emission of a typical organic scintillator (425 nm). As with any photon along the electromagnetic spectrum, its interaction mechanism and probability is strongly dependent on its energy and the material it traverses. For optical photons in a scintillator-based detection system, the interaction mechanisms that can occur are two modes of elastic scatter (Rayleigh and Mie scatter), absorption, reflection, and refraction.

2.2.2 Elastic scattering

An optical photon can elastically scatter off molecules in organic scintillators [40]. This process conserves the initial energy of the photon, however its direction is changed. The scatter direction of the photon is influenced by the photon wavelength and the size of the particle it scattered from. Mie scattering favors a forward-scatter direction, where the directionality is weakly dependent on wavelength. Mie scattering dominates when particle sizes in a material approach the photon wavelength. On the other hand, Rayleigh scattering favors a directionality that is strongly dependent on the photon energy. Violet and blue light, for example, is laterally scattered from its incident direction. Rayleigh scatter dominates when the photon wavelength is much larger than the particle size. The size of an organic-scintillator molecule is larger but on the order of its atomic spacing (a few Angstrom). Therefore an organic scintillator that produces optical photons from scintillation (425 nm) will predominantly scatter with lateral directionality and no change in wavelength, which is important for spectral matching (see Section 2.3).

2.2.3 Absorption

The molecular structure of an organic scintillator produces a continuous (as opposed to discrete) optical-photon wavelength emission spectrum [2, 5]. The same continuous nature applies to the absorption spectrum of the molecular structure of an organic scintillator. As shown in Figure 2.5, the emission and absorption spectra can sometimes overlap. Optical photons that are emitted from scintillation with wavelengths that overlap the absorption spectrum can be absorbed.

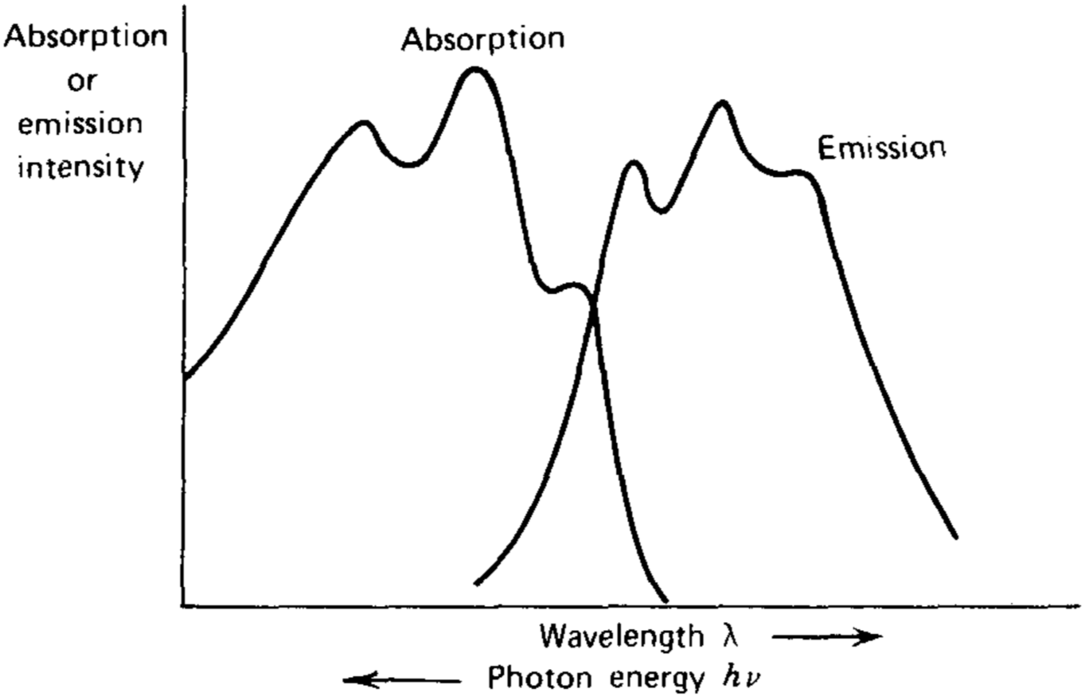


Figure 2.5: A sample absorption and emission distribution of optical photons in an organic scintillator; taken from Knoll's textbook on *Radiation Detection and Measurement [5].*

2.2.4 Reflection and refraction

In a vacuum, optical photons travel at a constant velocity c (29.28 cm/ns). When optical photons traverse through any material, their velocity c is reduced by some fraction to a new velocity v. This fraction is referred to as the refractive index (RI) of the material, and is wavelength dependent [40]. For example, the RI of a typical organic scintillator at its peak-emission wavelength (425 nm) is 1.58, which tells us that the speed of the optical-photon in the scintillator is approximately 18.53 cm/ns. More importantly, the RI of two materials can be used to understand how an optical photon of a particular wavelength will reflect or refract at their interface. Reflection and refraction can be loosely defined as an optical photon that bounces off of or transmits through a material interface, respectively.

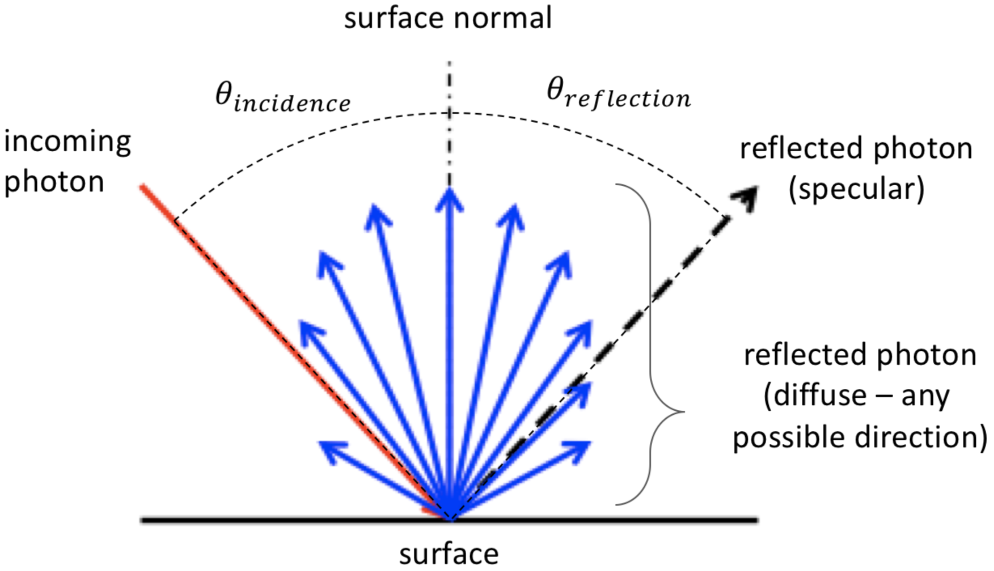

surface normal
$\theta_{incidence}$
$\theta_{reflection}$
incoming photon
reflected photon (specular)
reflected photon (diffuse – any possible direction)
surface

Figure 2.6: The two modes of reflection considered in this work are shown: perfectly specular and perfectly diffuse (Lambertian) reflection.

On a microscopic scale, one can consider a material interface to be composed of many tiny microfacet surfaces, each with their own surface normal vector. Reflection (and refraction) occur at these microfacets. The reflected photon will always be in the same plane made by the incident vector and the microfacet surface normal, but on the opposite side of the microfacet normal [40]. Additionally, with respect to the normal, the incident angle will always equal the reflected angle. A rough (or “ground") surface has microfacet normals that are randomly oriented and thus rough to the touch, and slightly opaque. For example, consider the surface of a machine-polished scintillator. This surface is extremely smooth to the touch and resulting from microfacet normals all pointing in the same outward direction. If sand paper is used to grind the same polished surface by hand, the microfacet orientation will become random. Therefore on a macroscopic scale, optical photons that reflect from a smooth surface, for example a polished scintillator, will essentially maintain the same reflection behavior that exists at the scale of each microfacet; namely that the angle of incidence will equal the angle of reflection. This mode of reflection is called “specular" reflection [40]. On the other hand, optical photons that reflect from a rough surface, for example a ground scintillator, will also maintain their “specular” behavior at the scale of the microfacet, but on an aggregate scale, incident photons are reflected from the surface at almost any angle. This mode of reflection is called “diffuse" reflection [40]. A reflector that is perfectly diffuse is also called a Lambertian reflector, which allows for any reflected angle to occur with a probability that is proportional to the cosine of the incident angle to the average surface normal. The two modes of reflection are shown in Figure 2.5 and can influence the light-collection process in an organic scintillator and will be further discussed in the energy resolution experiments reported in Chapter 4. It is also worth noting that two material surfaces in contact with each other that have a similar RI will transmit light more readily than two materials that have a very different RI. In practical terms, it is important to match the RI of a scintillator with the optical window of a PMT to enhance optical-photon transmission from the scintillator through the window and reduce reflection. RI-matching is achieved using an optical-coupling grease, preferably one that has a RI between that of the scintillator and optical window.

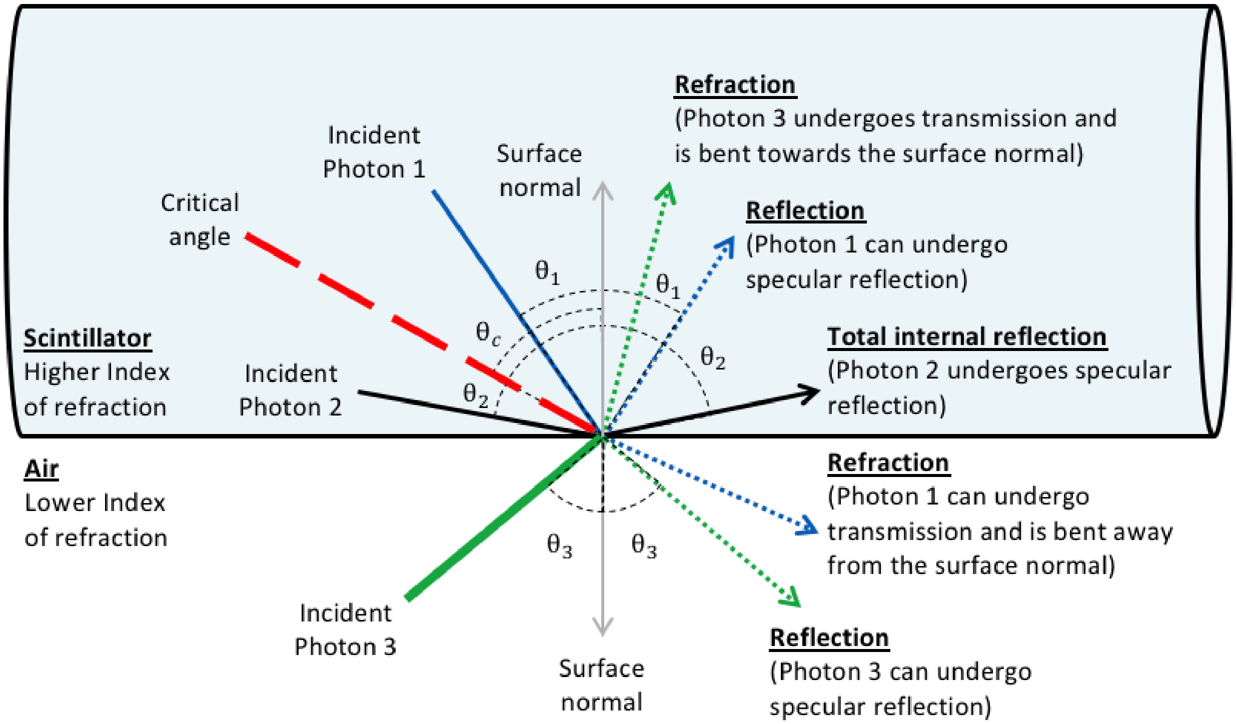

Refraction
(Photon 3 undergoes transmission and is bent towards the surface normal)
Incident Photon 1
Surface normal
Critical angle
Reflection
(Photon 1 can undergo specular reflection)
θ_1
θ_1
θ_c
θ_2
θ_2
Scintillator
Higher Index of refraction
Incident Photon 2
Total internal reflection
(Photon 2 undergoes specular reflection)
Air
Lower Index of refraction
Refraction
(Photon 1 can undergo transmission and is bent away from the surface normal)
θ_3
θ_3
Incident Photon 3
Surface normal
Reflection
(Photon 3 can undergo specular reflection)

Figure 2.7: A description of Fresnel reflection and refraction at a scintillator-reflector boundary. Following the optical photon with numbers 1, 2, and 3 to understand their possible interaction pathways at the surface.

Fresnel refraction, reflection, and transmission at the scintillator boundary (see Figure 2.7) are the most common interaction mechanisms [22,23,42], and it is worth noting that these interactions do not change the energy of the optical photon, although they can greatly influence its direction [40]. Optical-photons losses (before reaching the PMT) are largely attributed to transmission through the reflector or re-absorption within the scintillator material. Re-absorption are not a great concern for organic scintillators where their size is much smaller in comparison to their optical-photon attenuation length. The greatest concern for optical-photon losses is therefore transmission through the reflector, and the probability for this occurrence is increased each time an optical photon reflects at the scintillator-reflector boundary. The surface roughness of the scintillator can also greatly influence the collection profile onto the PMT surface.

2.3 Photomultiplier tubes

2.3.1 Overview

By coupling a light-sensing device such as a PMT [7] to a scintillator, optical photons can be collected and converted into a voltage pulse, allowing for data acquisition and processing. Figure 1.1 illustrated a simplified explanation for how a PMT operates. In short, optical photons that transmit through the optical window of the PMT reach a thin and transparent layer known as the photocathode. Once optical photons have a chance to interact in this layer, the photoelectric process can occur, thus emitting a photoelectron. Photoelectrons that enter into the vacuum chamber are guided by an electric field to the first of many dynode stages that further multiply the electrons to quantities that are sufficient to produce an electronic pulse that can be digitized by most oscilloscopes and data-acquisition hardware.

2.3.2 Spectral matching

Figure 2.8 shows a wavelength-emission spectrum of a scintillator overlaid with the wavelength-dependent PMT response spectrum; often referred to as the QE curve. The goal of spectral matching is to ensure that both spectra overlap each other to avoid collecting optical photons that the PMT simply cannot detect. Preferably, the most ideal match is one where both the peak-emission directly lines up with the peak-response, and where the emission spectrum is fully encompassed by response spectrum.

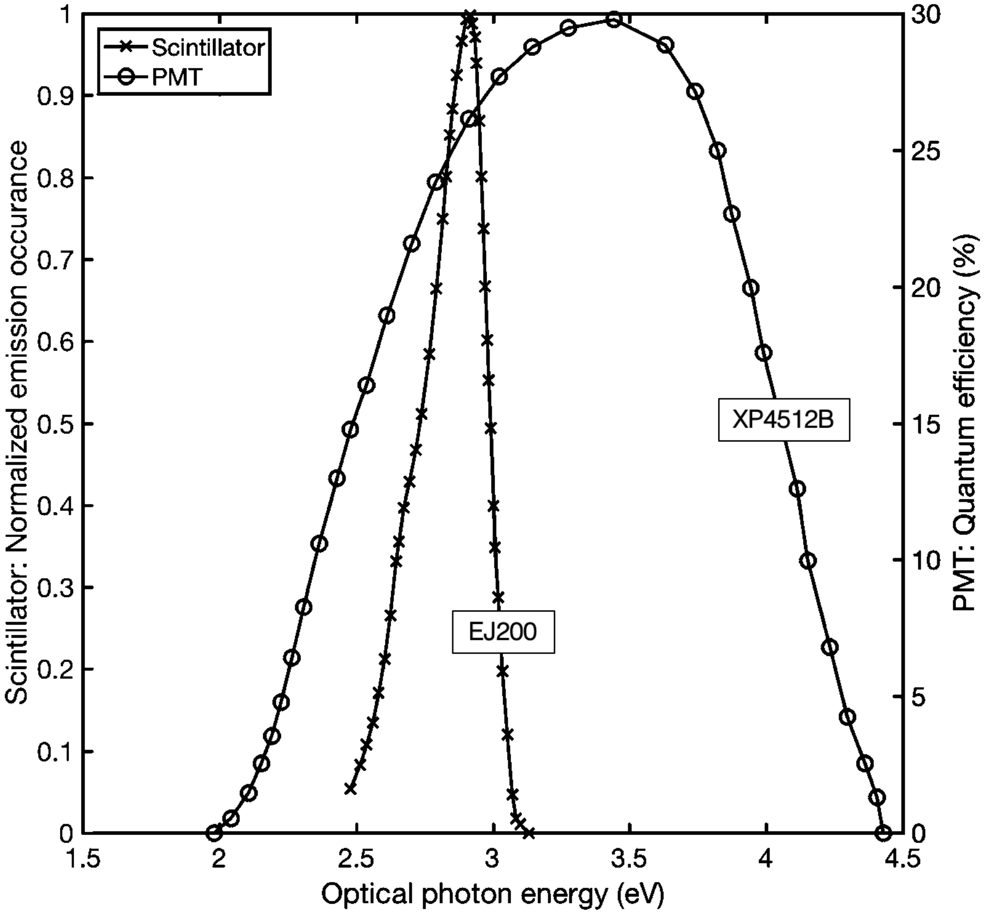


Figure 2.8: Spectral matching of an EJ200 emission spectrum to the response spectrum (QE curse) of a PMT (Photonis XP4512B).

2.3.3 Photocathode non-uniformity

Problems exist, however, in the localized QE of the PMT of a single optical photon as a function of its entrance position along the optical window, as depicted in Figure 2.9. The cause of this variability within the PMT is not explored in this work but has been discussed extensively in literature [18–21, 29]. The variability is attributed to the non-uniform deposition of the photocathode along the back layer of the optical window as well as electron trajectories that miss the first dynode stage for further multiplication. Generally speaking, the highest QE exists near the center of the window and worsens towards the outer edge. The non-uniformity in the conversion process produces a slightly different PMT response for the same amount of energy deposited, depending upon where the greatest number of optical photons were detected from a given scintillation event. Photocathode non-uniformity therefore contributes to degradations in the true energy resolution that is achievable by any scintillator material [29]. Efforts were made to assay the PMT response as a function of position along the optical window for all PMT's used in this work. Figure 2.10 shows a preliminary study using a small cube (5 mm height) of EJ200 placed in different locations along the optical window of a PMT with constant gain, demonstrated the impact of photocathode non-uniformity on the location of the Compton edge for cesium-137 in a pulse-integral spectrum. Greater LCE would produce a Compton edge furthest to the right of the plot. As expected, note how the right-most distribution in Figure 2.10 corresponds to the center of the PMT. The variation in the Compton-edge position represents changes to the overall LCE.

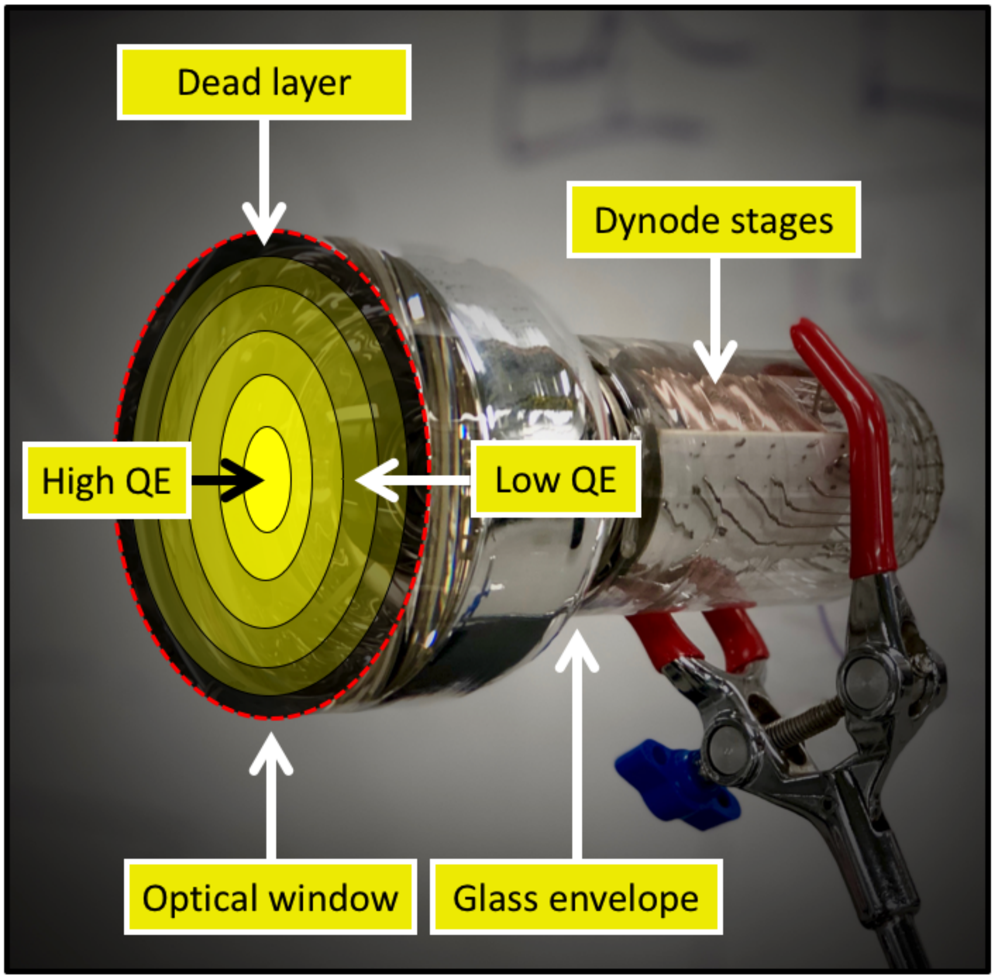


Figure 2.9: The QE of a PMT is spatially dependent along the optical window. The greatest QE exists near the center and drops off towards the outer perimeter of the window.

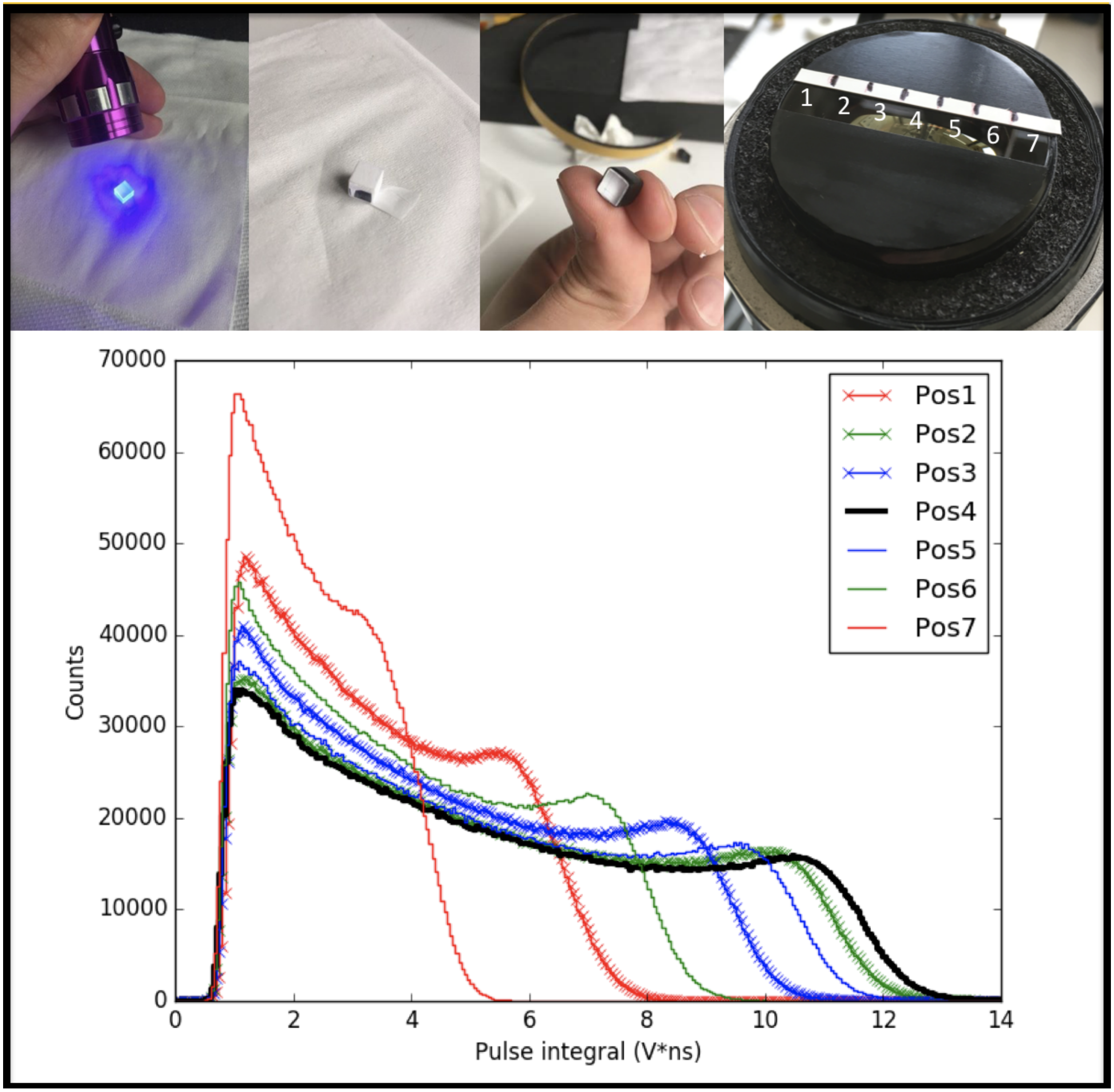


Figure 2.10: The energy resolution on the PMT (ETL 9214B) is spatially dependent due to photocathode non-uniformity that impact the localized QE. The same gain was used for all experiments (-1450 V).

CHAPTER 3

Experiment techniques

3.1 Overview

Three primary experiments were conducted. An energy-resolution experiment described in Chapter 4, a time-resolution experiment described in Chapter 5, and a PSD experiment described in Chapter 6. All experiments were conducted inside of a light-tight compartment described in Section 3.2.2). Each experiment compared the performance of a conical organic scintillator against a cylinder of equal base diameter (50 mm), height (50 mm), and material (EJ200 or $trans$-stilbene). A summary of the motivation and methods are included in each of the three chapters, along with a detailed discussion of the results. The subsections that follow (3.3.1, 3.3.2, and 3.3.3) describe the general techniques used in those experiments with greater detail than what is discussed in the methods section of those chapters.

3.2 Equipment

Table 3.1, shows a listed of all the main equipment used in this thesis work. Figure 3.1 and 3.2 show a picture of all the scintillators and PMTs used in this work.

Equipment	Provider	Model	Description			
Source	Ludlum	disk source	cesium-137	Activity = 97.61 μCi +/- 5%		
			sodium-22	Activity = 97.61 μCi +/- 5%		
Inorganic scintillator	Saint Gobain	$LaBr_3$	pre-packed detector	63,000 p/MeV	cylinder	equal base diameter and height (50 mm)
Organic scintillator	Inrad Optics	*trans*-stilbene	ground	15,000 p/MeV		
					cone	
	Eljen	EJ200		10,000 p/MeV	cylinder	
					cone	
					cube	height (5 mm)
			polished		cylinder	equal base diameter and height (50 mm)
					cone	
Grease	Eljen	EJ550	silicone-based optical coupling grease			
Reflector	3M	D2000	specular	non-adhesive back side		reflectivity (97% at 425 nm)
		D50		adhesive back side		
PMT	ETL	9214B	named: "PMT-50"	outer diameter (50 mm)		operating gain (-1425 V)
		9821B	named: "PMT-76"	outer diameter (76.2 mm)		operating gain (-1830 V)
	Photonis	XP4512B	named: "PMT-127"	outer diameter (127 mm)		operating gain (-1700 V)
			named: "PMT-127v"			varied
DAQ	CAEN	DT5730	4 channels	sampling rate (500 MHz / 2 ns steps) resolution (14-bits across 2 V range)		
Power		DT1470ET		output range (6 kV / 300 μA)		

Table 3.1: List of equipment used in experimental work.

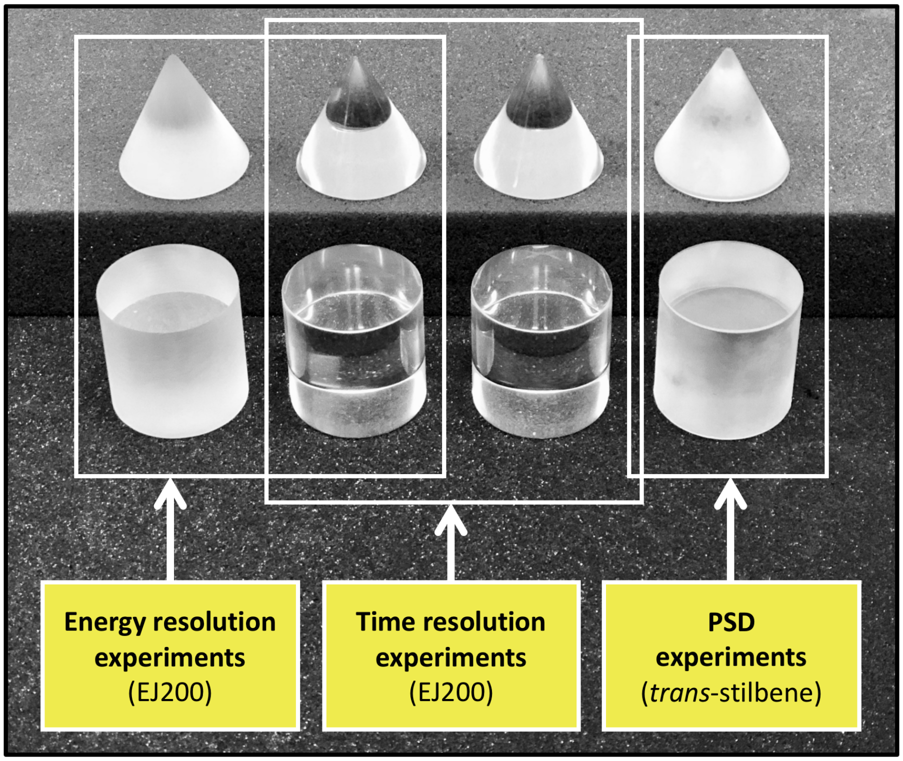

Energy resolution experiments (EJ200)
Time resolution experiments (EJ200)
PSD experiments (*trans*-stilbene)

Figure 3.1: Shows all the scintillators used in the various experiments.

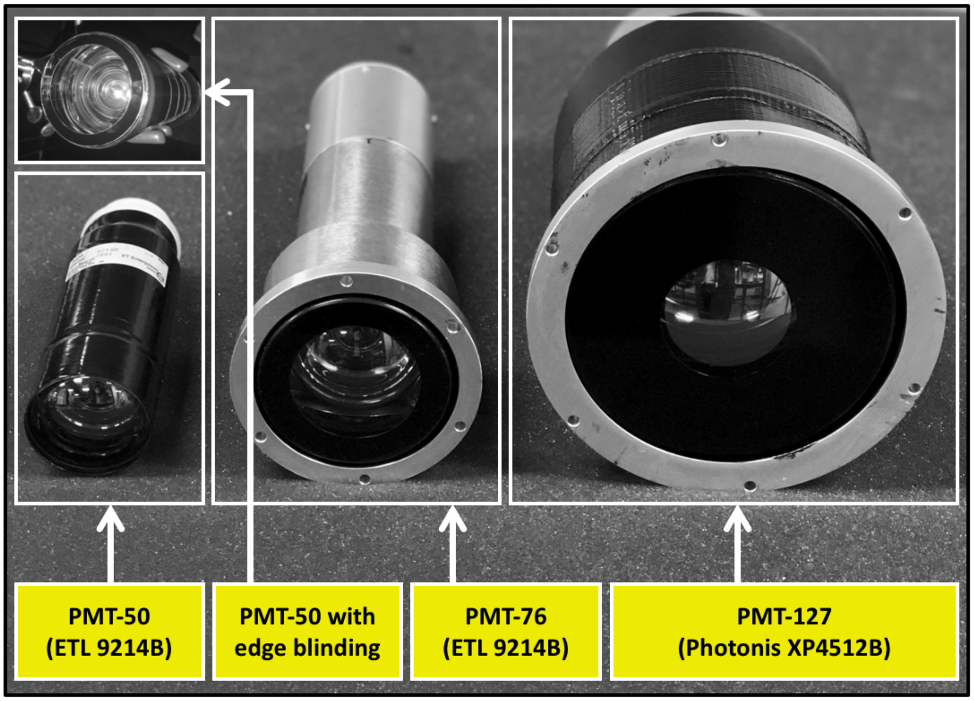


Figure 3.2: Shows all the PMTs used in the various experiments.

3.2.1 Reflector encasements

For reproducibility, an encasement for both geometries was carefully constructed (see Figure 3.3) by hand using a single layer of a specular reflector (3M D2000). Unlike the cylinder, it is difficult to apply the same quantity of PTFE tape to the cone given its geometry [30]. Therefore, PTFE tape was not used to avoid potential changes to the total-reflectivity coefficient between experiments. It was important to ensure that the only variation in the results presented between the cylinder and cone was due to their geometry. Although a specular reflector was used, the optical-photon reflection profile at the scintillator-reflector boundary remained diffuse (i.e., Lambertian) for the ground cases. Previous work has shown that a diffuse or specular reflector applied to a ground scintillator surface makes no difference to the reflection profile from that surface [22].

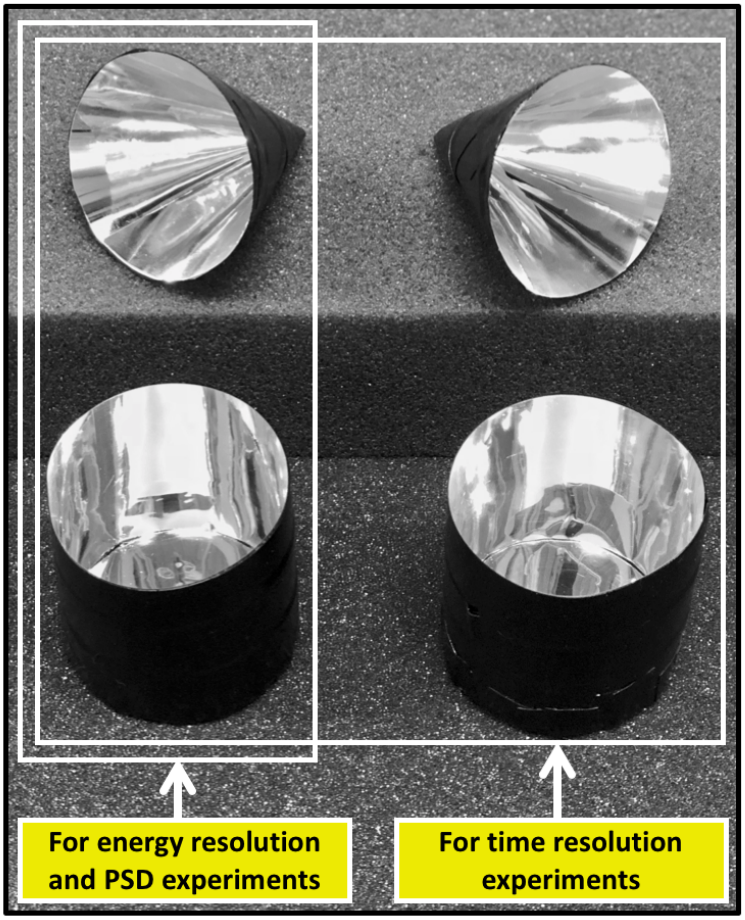

For energy resolution and PSD experiments
For time resolution experiments

Figure 3.3: Specular reflector encasements.

3.2.2 Light-tight compartment

The purpose of conducting experiments inside the compartment was to avoid light leakage to the PMT when a large-diameter PMT was used relative to the smaller-diameter scintillators. For example, the diameter of the scintillators used in this study were all 50 mm. However, various PMTs were used. The diameter of the largest PMT used was 127 mm. The light-tight compartment allowed for the space to conduct various experiments while avoiding light leakage that could potentially damage the PMT.

The compartment consists of an aluminum cage covered by thick window-curtain sheets. The curtains are designed to block as much sunlight as possible. The cage is 183 cm long, 150 cm tall, 90 cm wide, and sits on a movable flat table made of aluminum. A total of three curtains were used to wrap the entire cage. One curtain was used to cover the top and two curtains were used to cover the sides. After the curtains were clipped together using small vice grips to hold them in place, white duck tape was used to wrap them into place. White instead of black duck tape was chosen to prevent the inside of the cage from getting too hot, thus potentially causing a shift in PMT operation. It is worth mentioning that no study was conducted regarding the temperature effect on PMT operation in the cage over time. Figure 3.4 shows the partial construction of the cage on the left and the final product on the right. The ability of the cage to block light was tested by applying a gain to various PMT's and examining their response on an oscilloscope. During PMT operation, the room lights were turned on and off. No noticeable shift in baseline was observed and therefore the light-tight compartment worked as expected.

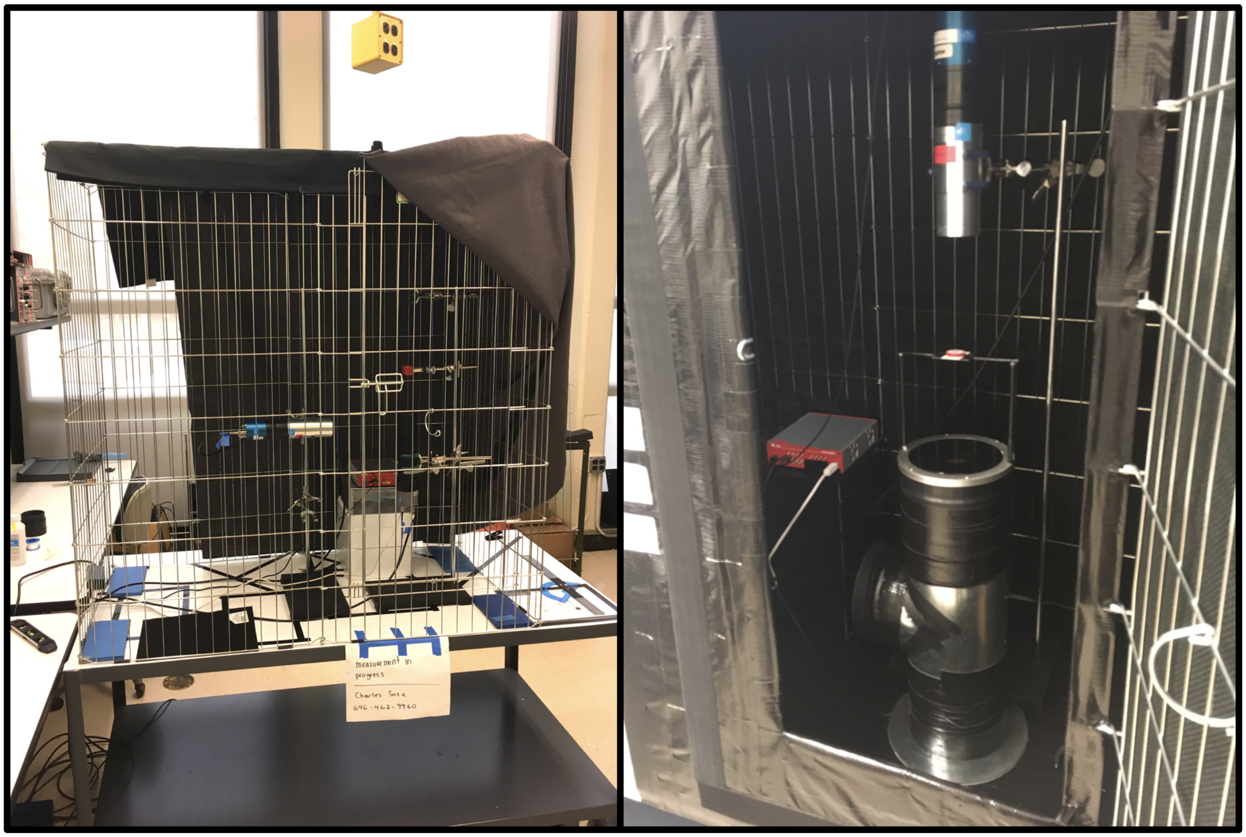


Figure 3.4: A light-tight compartment to support all experiments discussed in this work. The cage was large enough to accomodate all experiments and was necessary to avoid light leakage to certain PMTs that were large in diameter as compared to the tested scintillators.

3.3 Techniques

3.3.1 Energy resolution assessment

As mentioned previously in Chapter 2, gamma rays (< a few MeV) will predominantly Compton scatter in organic scintillators, without ever fully depositing their total energy [5]. In other words, the measured energy spectrum of a mono-energetic gamma ray source (< a few MeV) using an organic scintillator will only consist of the Compton continuum. An inorganic scintillator can measure the photopeak of a gamma ray and the width of the photopeak can be used as an assessment of the energy resolution. The lack of a photopeak in the measured energy spectrum taken with an organic scintillator presents a challenge for assessing its energy resolution, which serves as the motivation for the Compton coincidence technique. In this work, Compton-coincidence [29,43–45] was used to measure and compare the energy resolution at the Compton edge of cesium-137 for various organic scintillator (EJ200) and PMT configurations. The technique involves placing a mono-energetic gamma ray source in between an organic and inorganic scintillator. A gamma ray that backscatters in the organic scintillator can be captured by the inorganic scintillator and these two events can be correlated both in time and energy. Compton-edge events in the organic scintillator can be isolated by correlating scatter-capture events in both detectors with the same data aquisition (DAQ) system time-stamp, in addition to applying a narrow gate on the capture peak in the pulse-integral distribution of the inorganic scintillator. A Gaussian function can then be fit to Compton-edge events after subtracting accidental correlations, from which the energy resolution can be found using Eq. 3.1. Energy resolution is defined as the full width at half maximum (FWHM) of the fitted Gaussian divided by its centroid. The uncertainty in energy resolution was obtained by propagating the error on the centroid and FWHM from the fit parameters using 95% confidence intervals.

$$R = \frac{FWHM}{Centroid} * 100\% \tag{3.1}$$

Figure 3.5 illustrates the experiment setup. A point isotropic cesium-137 source (97.61 μCi) was placed in-between an organic and inorganic scintillator. The inorganic scintillator was a cylindrical (50 mm diameter and height) lanthanum bromide ($LaBr_3$) manufactured by Saint Gobain. It is worth noting that not all correlated counts were produced by gamma rays that scattered exactly at 180° in the organic scintillator; this effect is referred to as geometric broadening and is illustrated in Figure 3.6. Some gamma rays could scatter as low as 166° and still produce correlated counts. Although there is slight variation in the scatter angles that could result in correlated counts due to the position of both detectors and their size, there is a negligible effect on the energy deposited at the Compton edge (a variation of 2 keV from 478 keV).

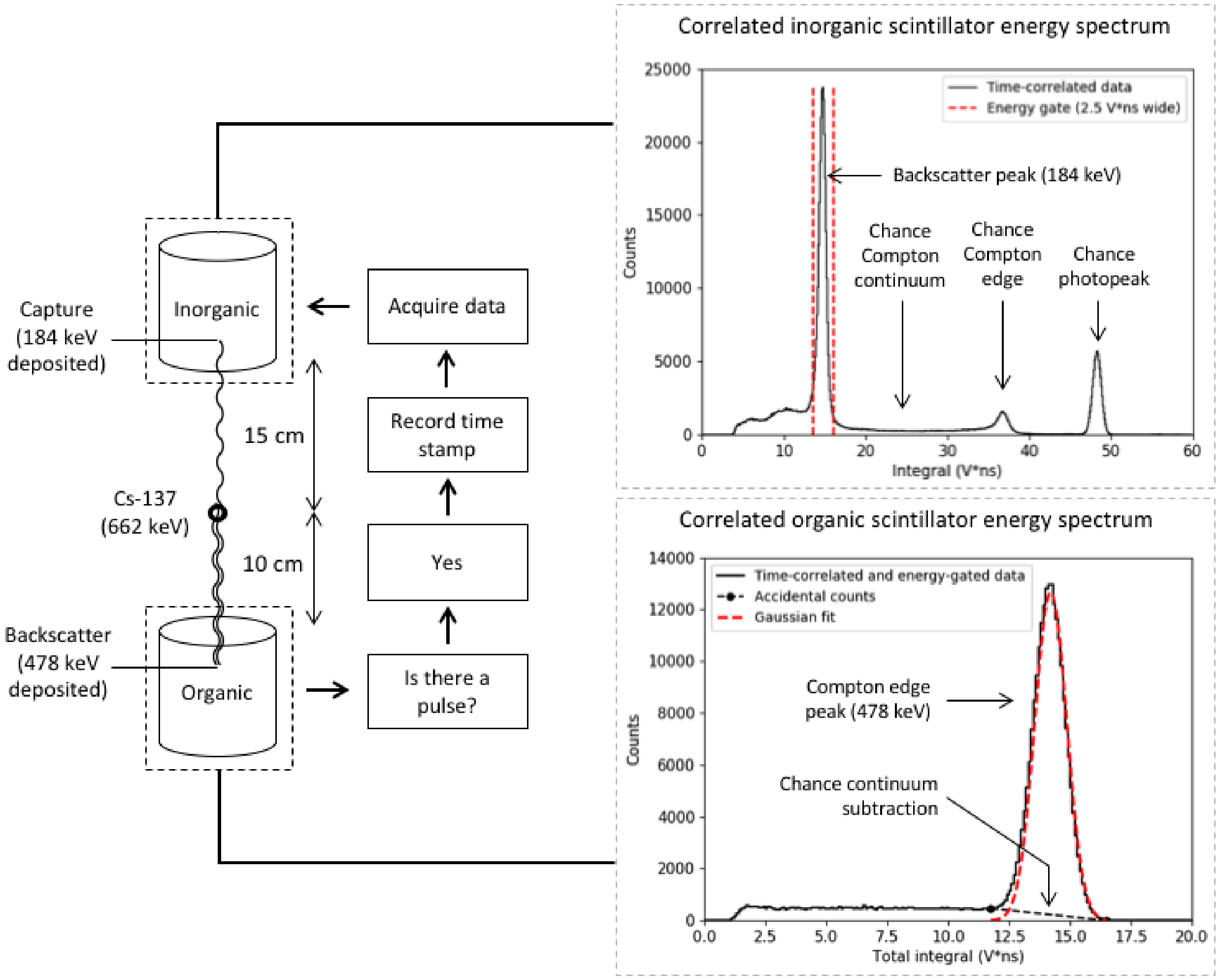


Figure 3.5: A graphical representation of the Compton-coincidence experiment and its associated data output. Note that the figure is not to scale.

The contribution of geometric broadening to the correlated distribution shown in the bottom left plot of Figure 3.5 can be expressed analytically through the Klein-Nishina formula [5]. The formula is used to calculate the differential-scattering cross section for a gamma-ray that Compton scatters on a free electron. Eq. 3.2 is component of the Klein-Nishina formula that relates the initial energy of a gamma ray and its scattering angle to its final energy. The figure shows how negligible the difference is for the final energy of a gamma ray that scatters between 166° and 180°. This physical process is unique and extremely important in Compton coincidence because it allows for additional scatters to correlate that may not be exactly 478, however, those events will still produce a very similar light yield in the scintillator. For example, a cesium-137 gamma ray (662 keV) that scatters at 180° from its incident direction will have a final energy of 184 keV and therefore the energy deposited is 478 keV (the Compton Edge). If the same gamma ray back scatters at 166°, then the energy it will deposit is 476. If the light yield of a scintillator produces 10,000 photons per energy deposited (in keV), then the number of optical photons expected from those energy-deposition events are 4,780 and 4,760, respectively; a difference of less than half a percent. Geometric broadening is therefore a negligible effect on the results reported in this work.

$$E(\theta) = \frac{662\,keV}{1 + \frac{662\,keV}{511\,keV} * [1 - cos(\theta)]} \tag{3.2}$$

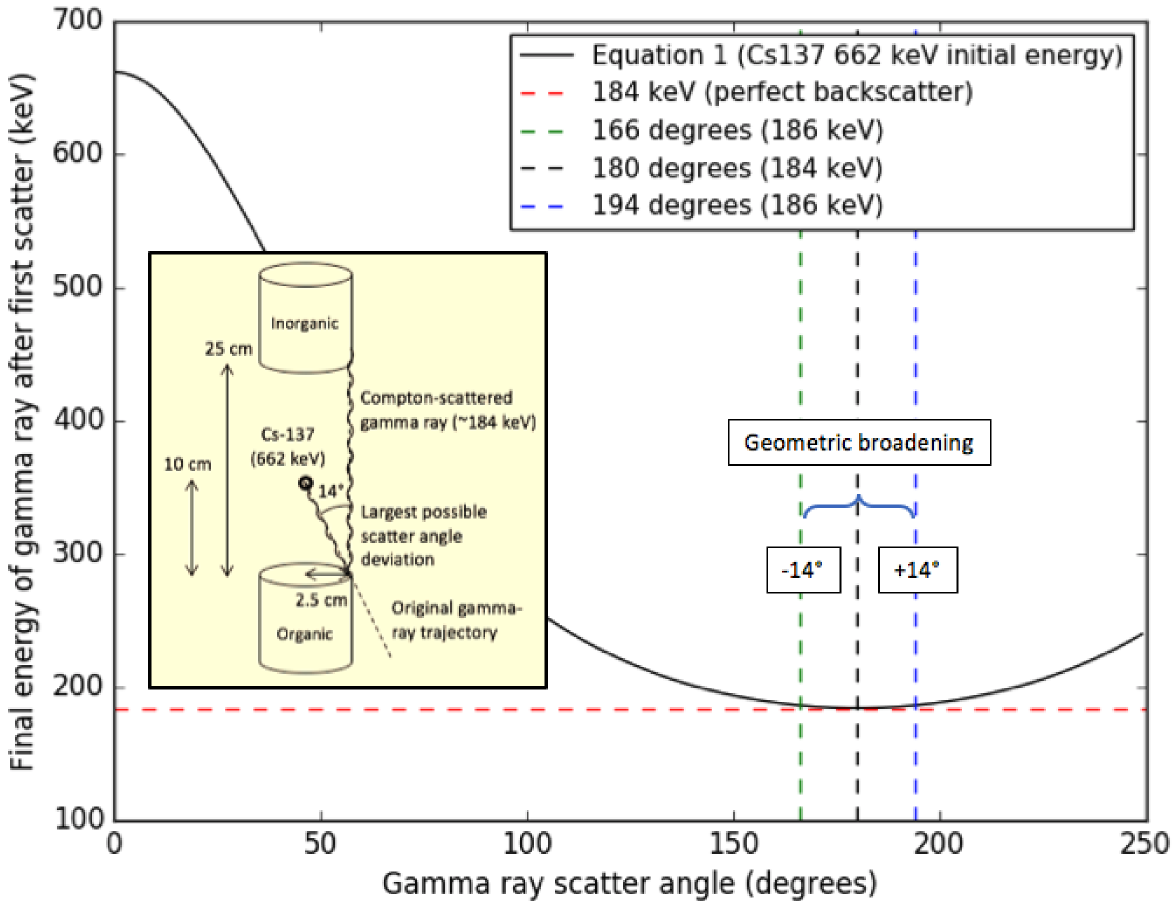


Figure 3.6: Eq. 3.2 is plotted as a function of initial scatter angle for a cesium-137 gamma ray. Geometric broadening yields a negligible uncertainty at the Compton edge.

3.3.2 Time resolution assessment

Cross correlation is commonly used in applied detection systems within nuclear safeguards and medical physics. In their most basic form, these systems are designed to provide the user with information that depends on the detection of correlated particles from the same decay event. In general, cross correlation is used to identify the simultaneous detection of particles within a short time in two or more detectors that originated from the same decay event. Cross correlation can extract a wealth of information depending upon the system. For example, within the nuclear safeguards, novel fast neutron-multiplicity counters use PSD-capable organic scintillators with cross correlation to count neutron multiples from fission to estimate fissile mass content [16, 17]. And within medical physics, positron emission tomography (PET) scanners use radioisotope tracers and cross correlation to image disease in living tissue.

In this work, cross correlation was used to measure and compare the time resolution of cylindrical and conical organic scintillator (EJ200) detectors. To measure the time resolution of a specific detector using cross correlation, a second detector with the same scintillating material, geometry, dimensions, reflector, and PMT, is needed. A sodium-22 source, which emits two oppositely-directed gamma rays (each 511 keV) from positron annihilation, is placed in between the two detectors. The goal of the experiment is to detect the simultaneous interaction in both detectors (Compton-scatter events) of the two correlated gamma rays from annihilation. The purpose behind matching the characteristics of both detectors is to ensure that both respond identically to incident radiation of same energy; for example 511 keV gamma rays that Compton scatter within a fixed energy range along the continuum.

The setup for the cross correlation experiment used in this work is illustrated in Figure 3.7. Gamma rays from annihilation can Compton scatter in both detectors and be correlated in time. The DAQ system used in these experiments allowed each detector channel to trigger the other. In other words, if one channel triggered first, then data from the other channel was recorded; and vice versa. After the data was collected and calibrated, a cross-correlation analysis script sorted through the data and flagged correlated events using the following procedure: (1) a pulse was present in both channels, (2) the light output in both channels are both above and below a fixed lower-level discriminator (LLD) (200 keVee) and and upper-level discriminator (ULD) (370 keVee), respectively. This light-output range encompasses the Compton-edge (341 keV) of a 511 keV gamma ray. Correlated events where at least one gamma ray Compton scatters at an angle shallower than 69.10°, which corresponds in energy to the LLD, do not contribute counts to the time-resolution analysis. Correlated events that satisfy these requirements are further processed by time pick-off methods used to assess the time resolution of the system.

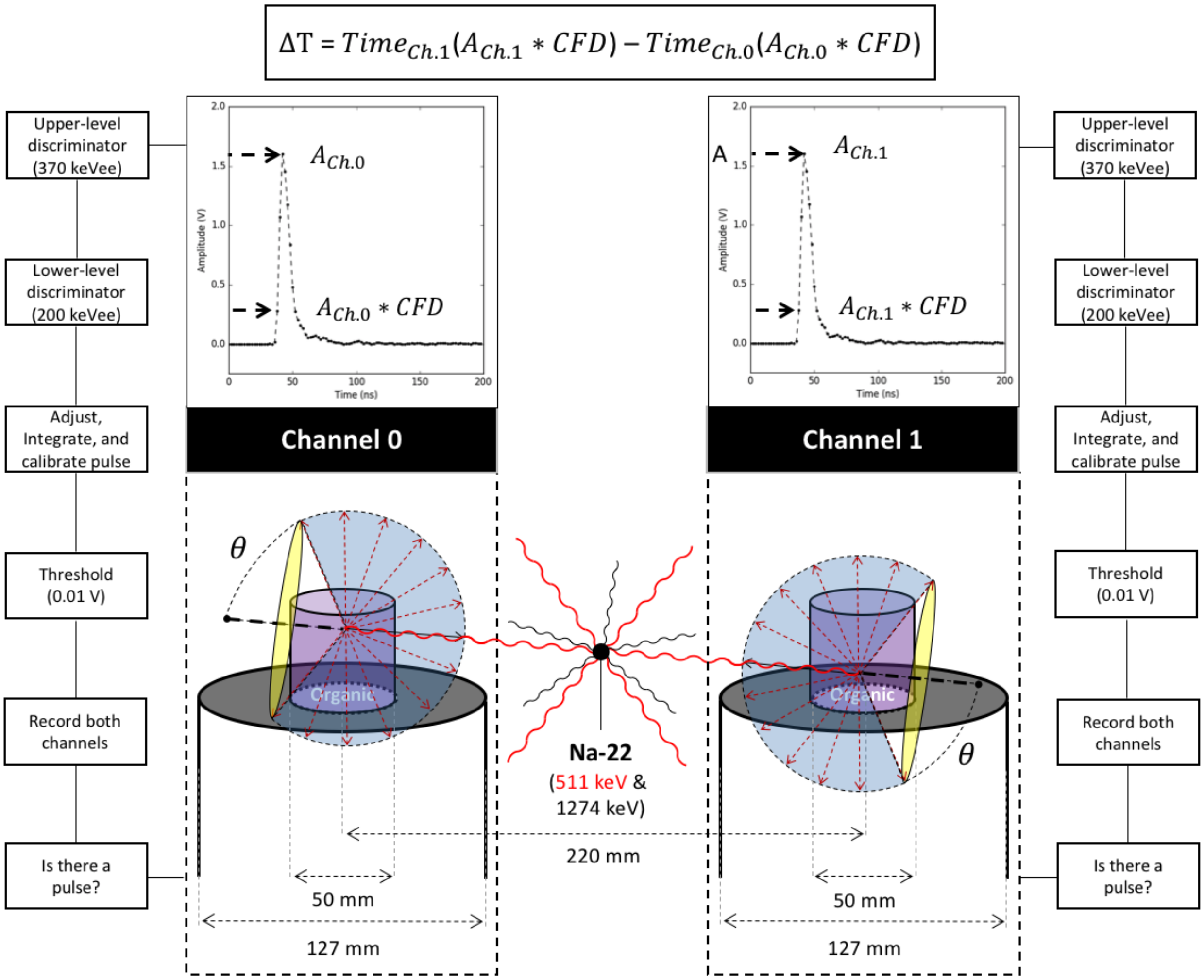


Figure 3.7: A graphical representation of the cross-correlation experiment and its associated processes. Note that the figure is not to scale.

The pulse-start time in both channels for correlated events was found using a time pick-off method referred to in this work as constant-fraction timing (CFT) [5]. In CFT, a fraction of the pulse amplitude is used to associate its position on the leading edge with time. This fraction is fixed for all pulses and is often referred to as the constant-fraction discriminator (CFD), and requires some optimization described further in Chapter 4. The CFD provides a position along the rising edge of the pulse for time pick-off that is independent of pulse shape and is therefore insensitive to amplitude walk or time jitter when a pulse is triggered by the fixed-voltage threshold of the DAQ [5]. The start time of a pulse in one channel (referred to as the $start$ detector) is then subtracted from the start time of its correlated pulse in the other channel (referred to as the $stop$ detector), yielding a time interval (ΔT) between the two pulses. In this work, the $start$ and $stop$ detectors correspond to channel 0 and channel 1, respectively. A histogram of ΔT values produces a cross-correlation distribution with a Gaussian-like structure. Figure 3.8 shows a sample cross-correlation distribution. The time resolution of is defined as the FWHM of a fitted Gaussian function to the cross-correlation distribution. The uncertainty in time resolution was obtained from the fit parameters using 95% confidence intervals.

It is worth clarifying that historically speaking, CFT was performed using analog circuitry. However, as it was just described, the same technique can be implemented in the post processing of digitized pulses recorded by a DAQ system. The same clarification is made for the CFD; the fractional amplitude is calculated from digitized pulses rather than using on-board electronics. For clarity, CFT is the method for time-pickoff, of which there are many. CFD is the name given to the fractional-amplitude specifically chosen for time pick-off along the leading edge for correlated pulses to calculate their time interval (see Figure 3.7).

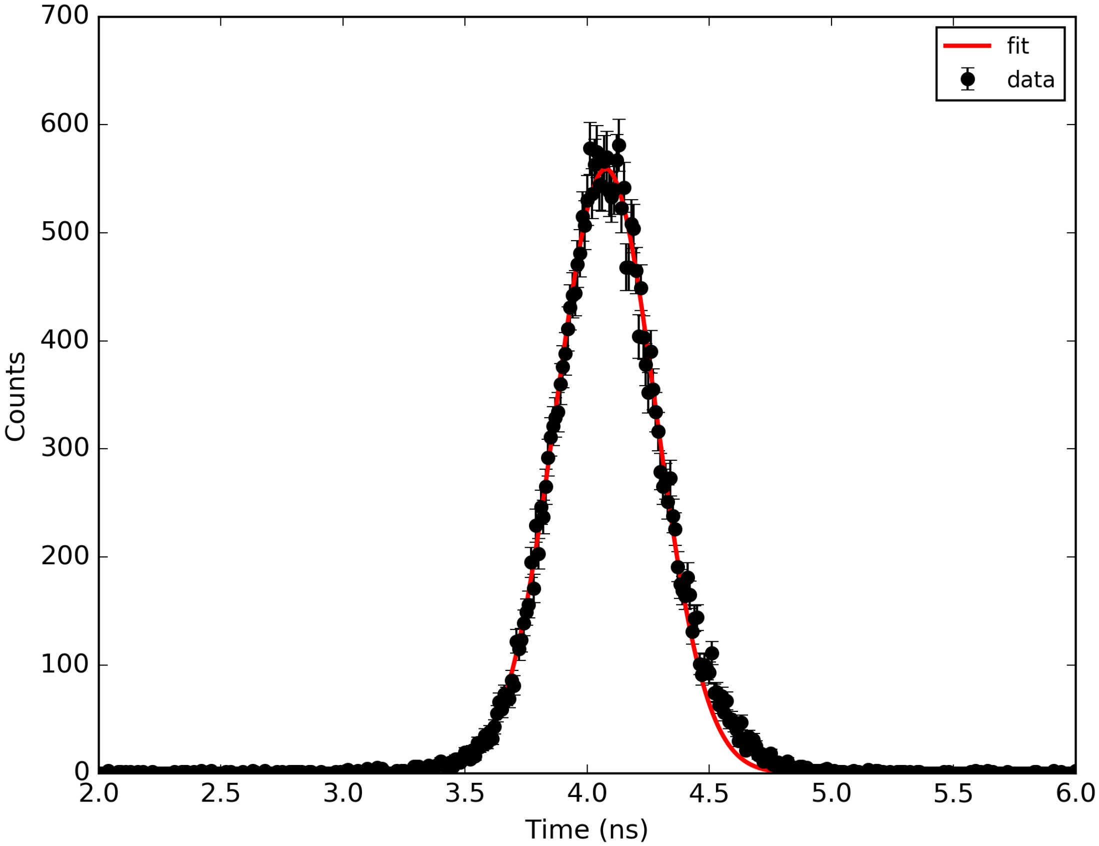


Figure 3.8: A sample cross-correlation distribution.

3.3.3 Pulse shape discrimination assessment

Neutron-gamma PSD of a californim-252 source was performed using the charge integration method, illustrated in Figure 3.9. In this method, the ratio between the tail integral and total integral of the pulse is calculated, where the start time of the tail integral requires optimization to achieve the best possible separation between neutron and gamma events [12–15]. Note that the ratio of the tail integral to the total integral is referred to in this work as the pulse-shape (S) parameter. Previous work also refers to particle-type identification in this manner [13]. A sample histogram of S parameters taken from californium-252 data is shown in Figure 3.10.

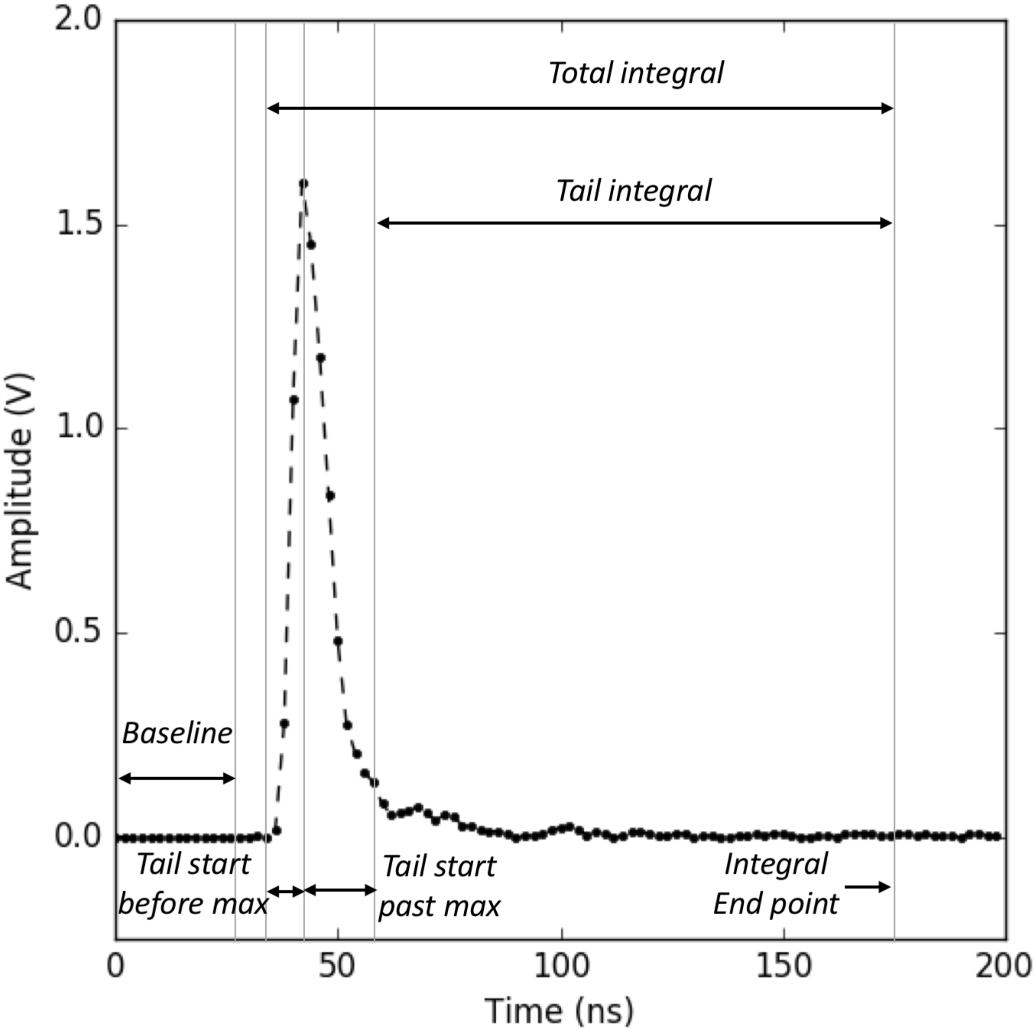


Figure 3.9: A diagram that represents the charge-integration technique for PSD.

A double-Gaussian function is applied to the S distribution, where the centroid and FWHM values for the gamma-ray and neutron clusters are found. These values, as determined by the fit, are used to calculate a figure of merit (FOM), used to assess PSD performance by quantifying the degree of separation between the neutron and gamma-ray regions in the S distribution, as shown in Figure 3.10. The equation for the FOM is also shown in Figure 3.10, where a higher FOM value indicates better PSD performance.

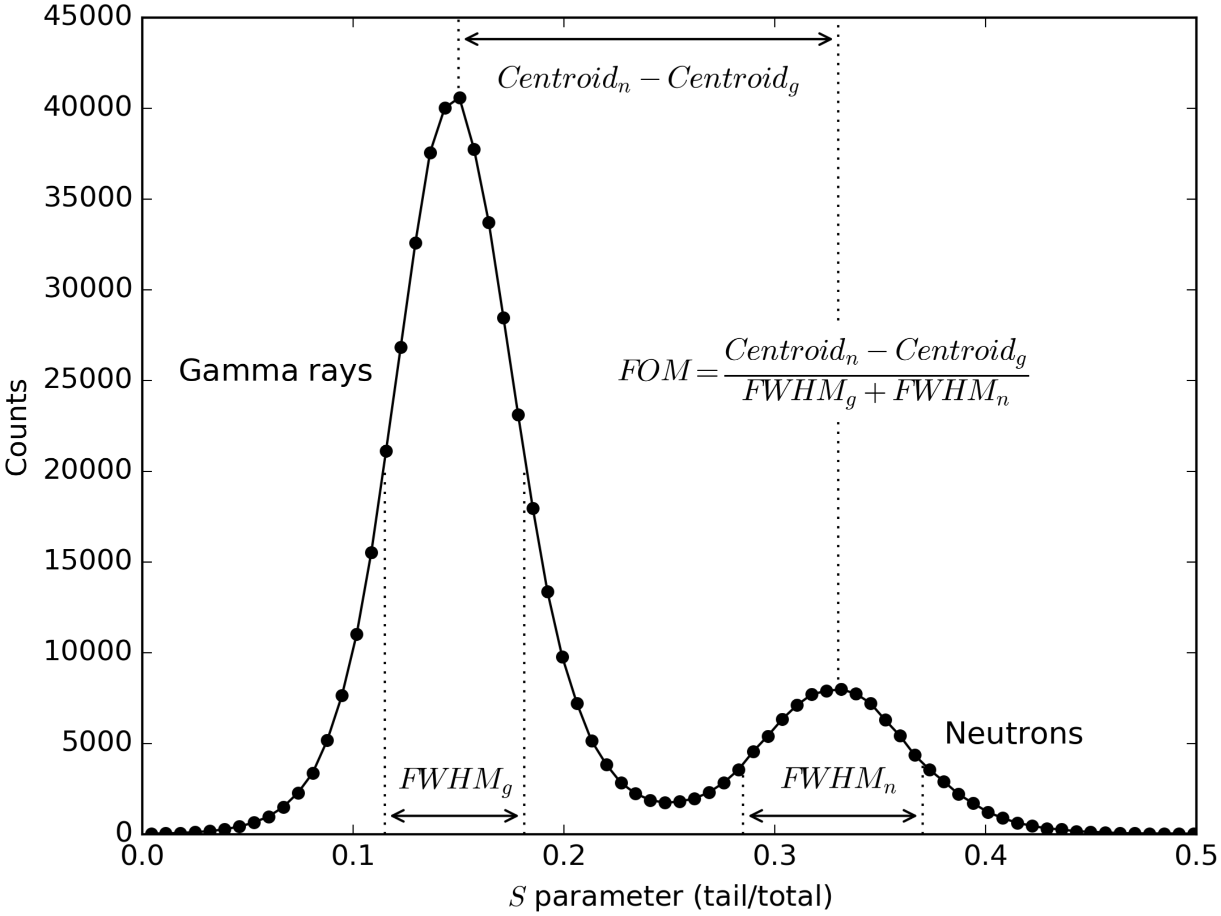


Figure 3.10: Sample pulse-shape (S) parameter distribution for events across a specific energy range, as produced with the charge integration method.

CHAPTER 4

Experiment: Energy resolution

4.1 Motivation

The purpose of this work is to measure and compare the energy resolution of conical and cylindrical organic scintillator detectors of equal base diameter, height, and material. The cone is expected to show an improvement in energy resolution over the cylinder by reducing optical-photon reflections at the scintillator-reflector boundary. By reducing reflections, the combined absorption and transmission probabilities at the reflector surface are decreased, thereby increasing the optical-photon detection probability. In other words, greater LCE will increase the counting statistics available per scintillation pulse, which helps to resolve the pulse shape and integral more accurately. Recall that the pulse integral is proportional to the energy deposited. An improvement in the energy resolution can lower the measurement threshold and improve the light-output calibration accuracy for applied detection systems. Additionally, Chapter 6 will demonstrate how improvements to the energy resolution also greatly assist with particle-type identification for PSD-capable organic scintillators. In the process of conducting the energy resolution experiments, it was found that mating a highly polished cone to a PMT of equal base diameter produced an unusable energy spectrum, referred to as spectrum smearing in this work, and is shown below. Figure 4.1 shows a pulse-integral spectrum of the polished cone on a PMT of equal base diameter, where the cesium-137 Compton edge is not well defined. Spectrum smearing was not observed when larger PMT sizes and ground (i.e., unpolished) surface conditions were tested for the cone. The cause of spectrum smearing and how to avoid it is discussed in this chapter. In Chapter 7, simulations help to visualize optical-photon transport contributions to spectrum smearing.

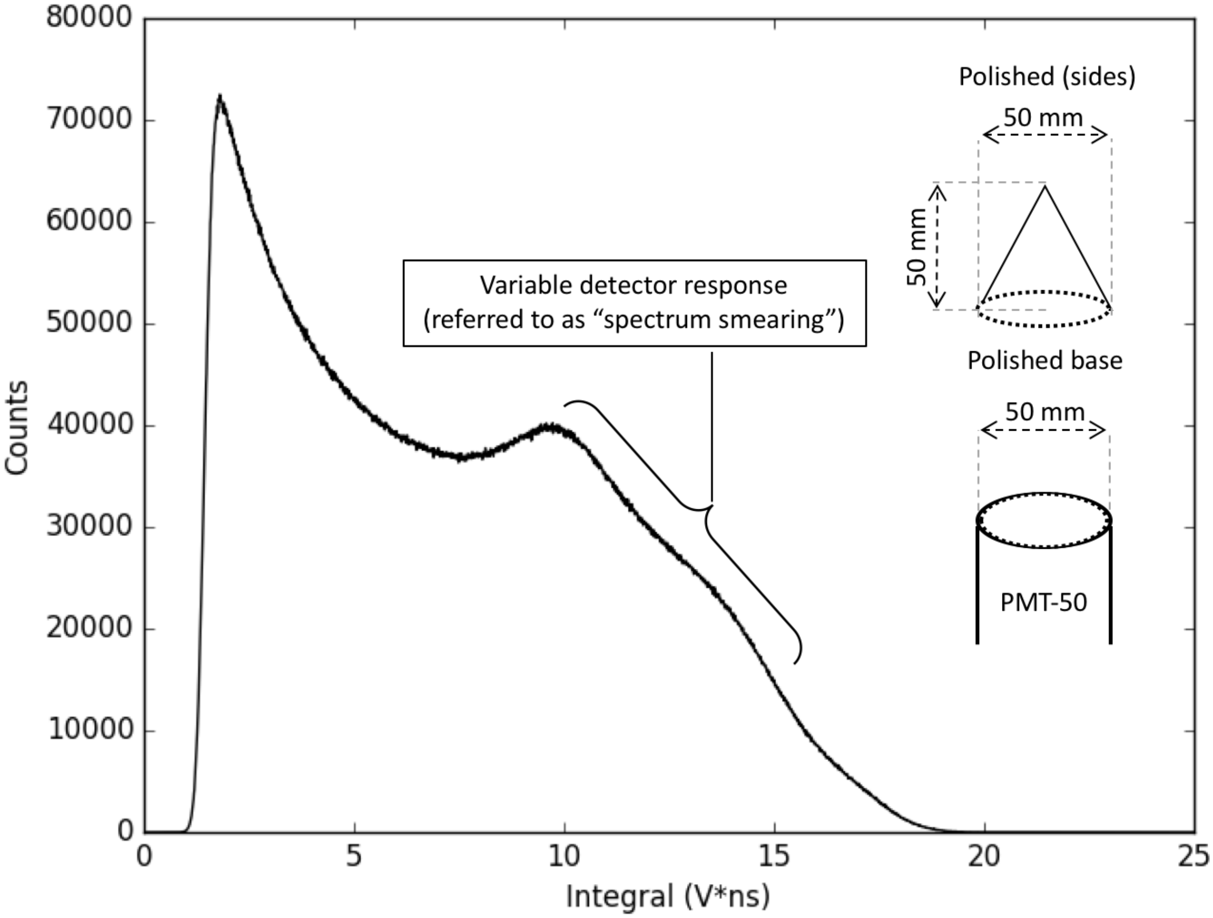


Figure 4.1: Spectrum smearing is shown. This phenomenon occurs when a machine-polished conical scintillator is mated to a PMT of equal base diameter.

4.2 Methods

4.2.1 Equipment

Four machine-polished organic scintillators (EJ200), two cylinders and two cones, were purchased from Eljen, and were compared using four PMT configurations (see Figure 4.3). Optical-grade silicone grease (EJ550) was used for scintillator-to-PMT coupling. Figure 4.2 shows the four scintillators. Each shape had two configurations: one scintillator remained polished and the other was lightly ground by hand using sand paper (150 grit). The bases of all four scintillators remained polished to ensure the highest transmission probability to the PMT.

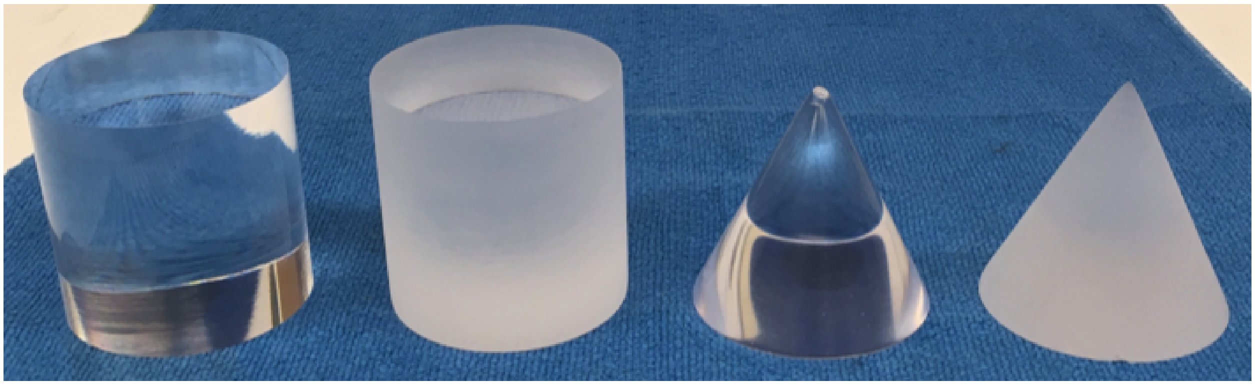


Figure 4.2: A picture of the four organic scintillators (EJ200) used in this chapter.

Three PMTs were used: an ETL-9214B (PMT-50), an ETL-9821B (PMT-76), and a Photonis XP4512B (PMT-127), where the number after the “PMT" label refers to its optical-window diameter in millimeters. Note that “edge-blinding" in Figure 4.3 refers to a fourth PMT configuration where a thin specular reflector (3M D50) ring with an adhesive backside was applied directly to the window of PMT-50. It is worth noting that although PMT-50 had an optical window with an outer diameter of 50 mm, the data sheet provided by the manufacturer listed an effective-photocathode diameter of approximately 46 mm. In other words, optical photons that enter the PMT window outside of the effective-photocathode region will have no chance of producing a photoelectron, thus decreasing LCE. Recall that photocathode non-uniformity contributions decrease the localized QE near its outer perimeter. To understand how photocathode non-uniformity (see Section 2.3.3) could impact the results, efforts were made to characterize the spatially-dependent response of each PMT and are further discussed in Section 4.2.2. Edge blinding on PMT-50 was an attempt to make the effective photocathode response more uniform to see what effect, if any, this would have on spectrum smearing for the polished cone and the energy resolution of the three other scintillators.

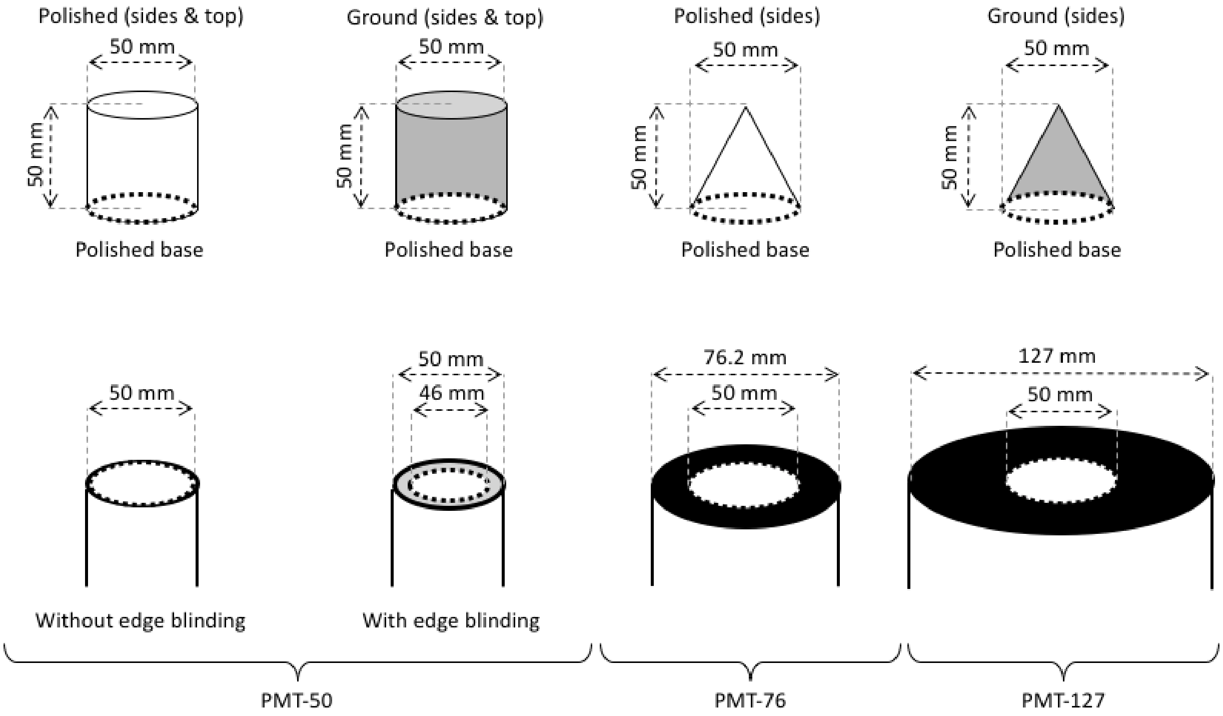


Figure 4.3: Four scintillators were individually coupled to four PMTs (a total of 16 configurations). PMT-50, PMT-76, and PMT-127, was operated at a gain of -1425 V, -1830 V, and -1700 V, respectively.

4.2.2 Compton coincidence

Compton-coincidence [43–45] was used to calibrate the light output and measure the energy resolution at the Compton edge (478 keV) of cesium-137 for the various organic scintillator and PMT configurations shown in Figure 4.3. Figure 4.4 shows a picture of the experimental setup where a point isotropic cesium-137 source (97.61 μCi) was placed in-between an organic and inorganic scintillator. Each measurement lasted 5 hours. The inorganic scintillator was a cylindrical (50 mm diameter and height) lanthanum bromide ($LaBr_3$) manufactured by Saint Gobain. The exact dimensions of the setup (see Figure 3.5), as well as a detailed explanation of Compton coincidence, how it works, and how it was implemented, can be found in Section 3.3.1.

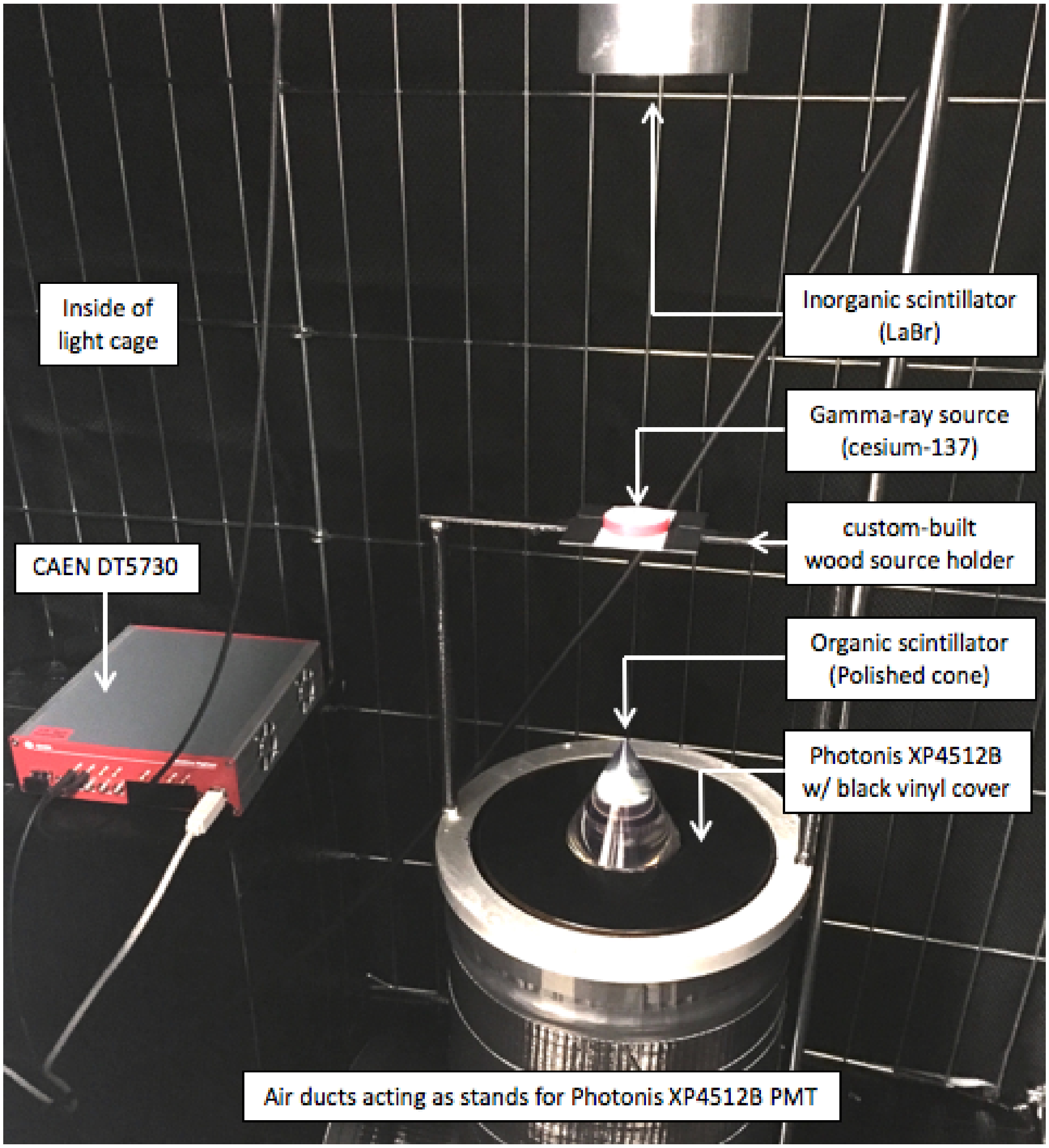

Inside of light cage
Inorganic scintillator (LaBr)
Gamma-ray source (cesium-137)
custom-built wood source holder
CAEN DT5730
Organic scintillator (Polished cone)
Photonis XP4512B w/ black vinyl cover
Air ducts acting as stands for Photonis XP4512B PMT

Figure 4.4: Custom built light-tight compartment. The highly polished conical scintillator (without its specular reflector) is shown mated to the largest-sized PMT configuration.

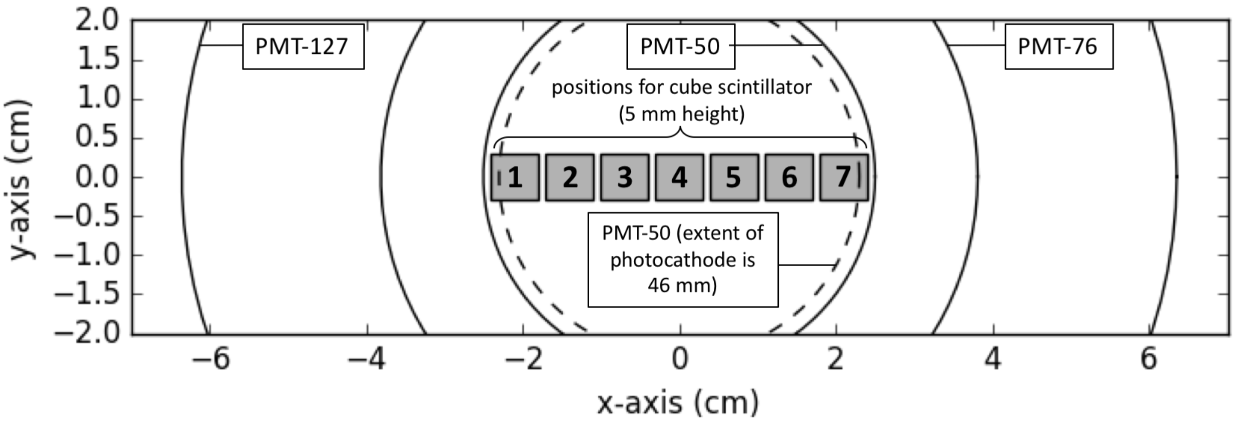


Figure 4.5: The mapping process used to assess the energy resolution as a function of position for each PMT, where the energy resolution is proportional to the localized QE.

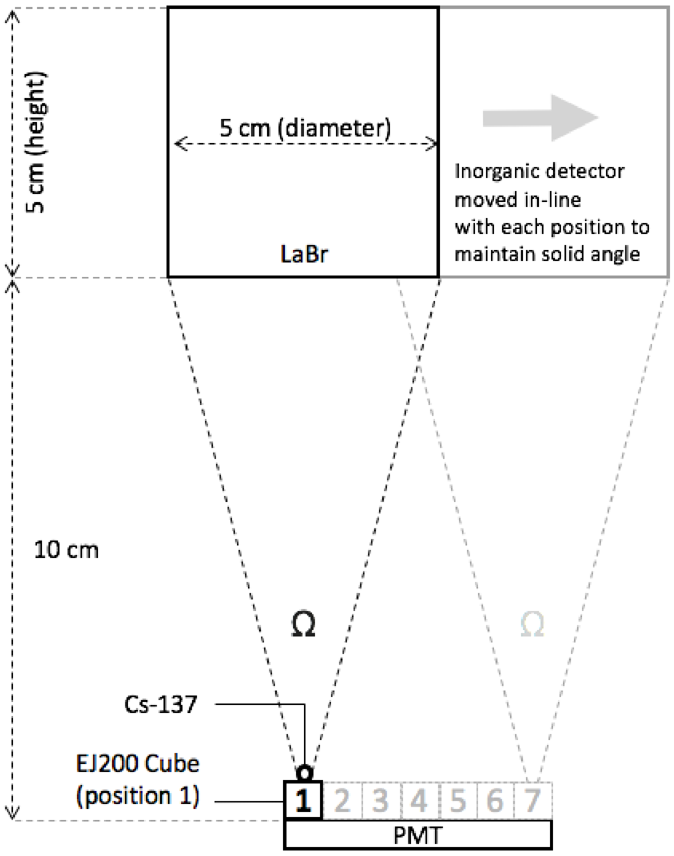


Figure 4.6: Dimensions for the Compton-coincidence experiment performed during PMT mapping process. The $LaBr_3$ detector was moved along the central axis of the scintillator in order to maintain the solid angle between experiments.

Compton coincidence was also used to assess the response uniformity of all three PMTs. A small cube (5 mm height) of EJ200 was placed in seven positions shown in Figure 4.5. All sides of the cube were ground with the exception of the surface coupled to the PMT with optical grease (EJ500). The cube was covered by a specular reflector encasement in a similar fashion to the cylinder and cone. From Compton coincidence, the calibration position and energy resolution were obtained at each of the seven positions. Each experiment lasted 30 minutes. The same cesium-137 source used previously was placed on top of the cube. The front plane of the $LaBr_3$ detector was placed 10 cm above the window of the PMT that was being characterized. The $LaBr_3$ detector was moved with the center line of each position to maintain the same solid angle between experiments. An illustration of the Compton-coincidence setup is shown in Figure 4.6.

4.3 Results

4.3.1 Photocathode non-uniformity

Figure 4.7 shows the energy resolution at 478 keV of the EJ200 cube as a function of position for each PMT. The greatest variability in the energy resolution exists in PMT-50. As expected, the highest QE exists near the center and degrades towards the outer edge due to photocathode non-uniformity effects [18–21]. The more uniform response of PMT-76 and PMT-127 across the 50 mm test diameter explains why the energy resolution of all scintillators improved when coupled to those PMTs (see Table 4.1).

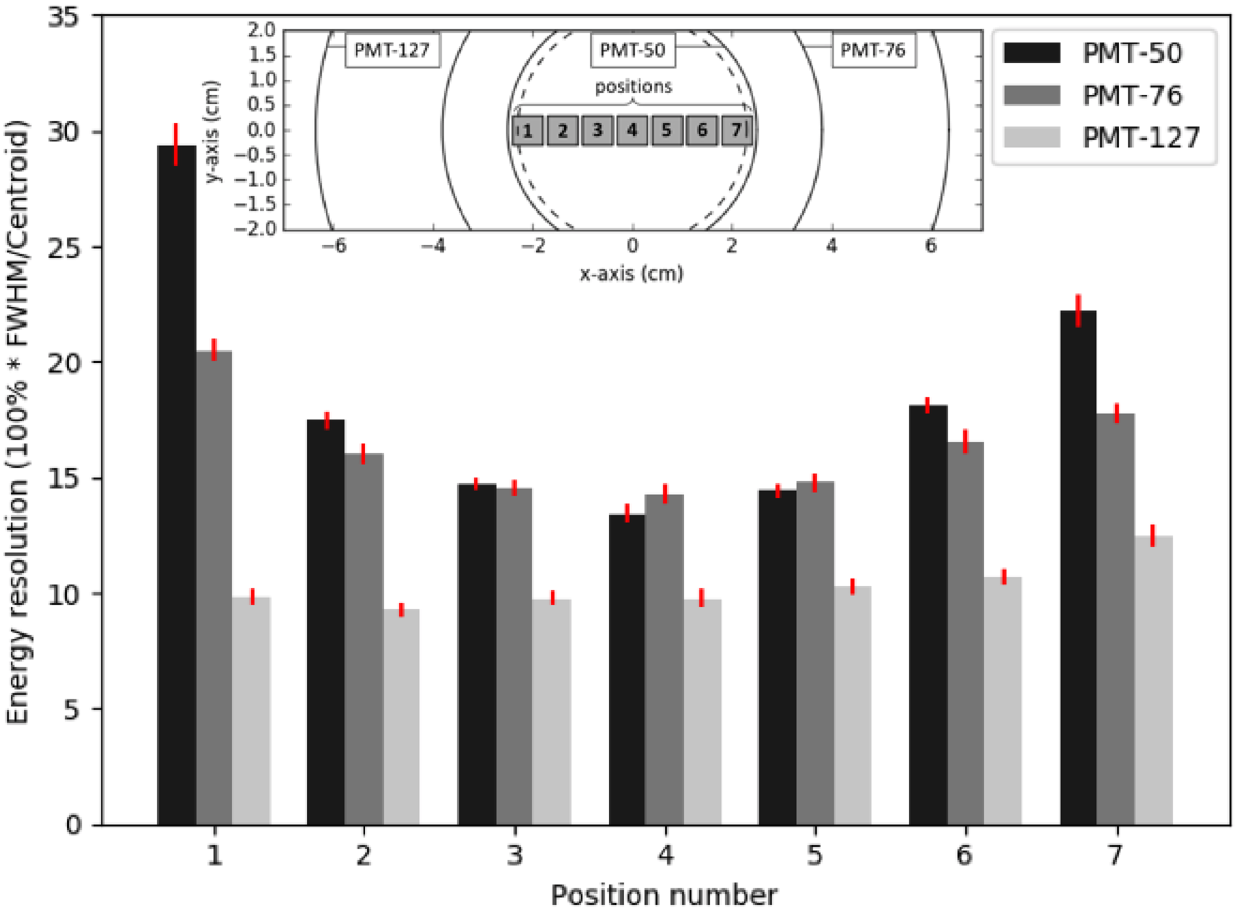


Figure 4.7: Energy resolution as a function of position on the surface of the PMT.

4.3.2 Calibration and energy resolution

Figures 4.8 and 4.9 show the cesium-137 standard (i.e., uncorrelated) and correlated light-output distributions of the ground cone and ground cylinder, respectively. The correlated spectra were obtained through Compton coincidence. Additionally, Figures 4.10 and 4.11 show the cesium-137 standard and correlated light-output distributions of the polished cone and polished cylinder, respectively. Figures 4.8 - 4.11 illustrate the effect of photocathode uniformity on energy resolution by showing how sharpness of peak at the Compton edge changes with PMT sizes. As the effect of photocathode non-uniformity is decreased (see Figure 4.7) with PMT size, so does the resolution. All energy resolution values were measured at 478 keV and are shown in Table 4.1. PMT-127 produced the best energy resolution values for all scintillators and this result is directly attributed to its more uniform photocathode response across the 50 mm test diameter. The ground cone outperformed the energy resolution of the ground cylinder for all PMT configurations. Although the polished surface treatment produced slightly higher calibration points (see Table 4.2), it generally resulted in poorer energy resolution values in comparison to the ground surface treatment for all scintillators. Moreover, the polished cone mated to PMT-50 produced a smeared energy spectrum.

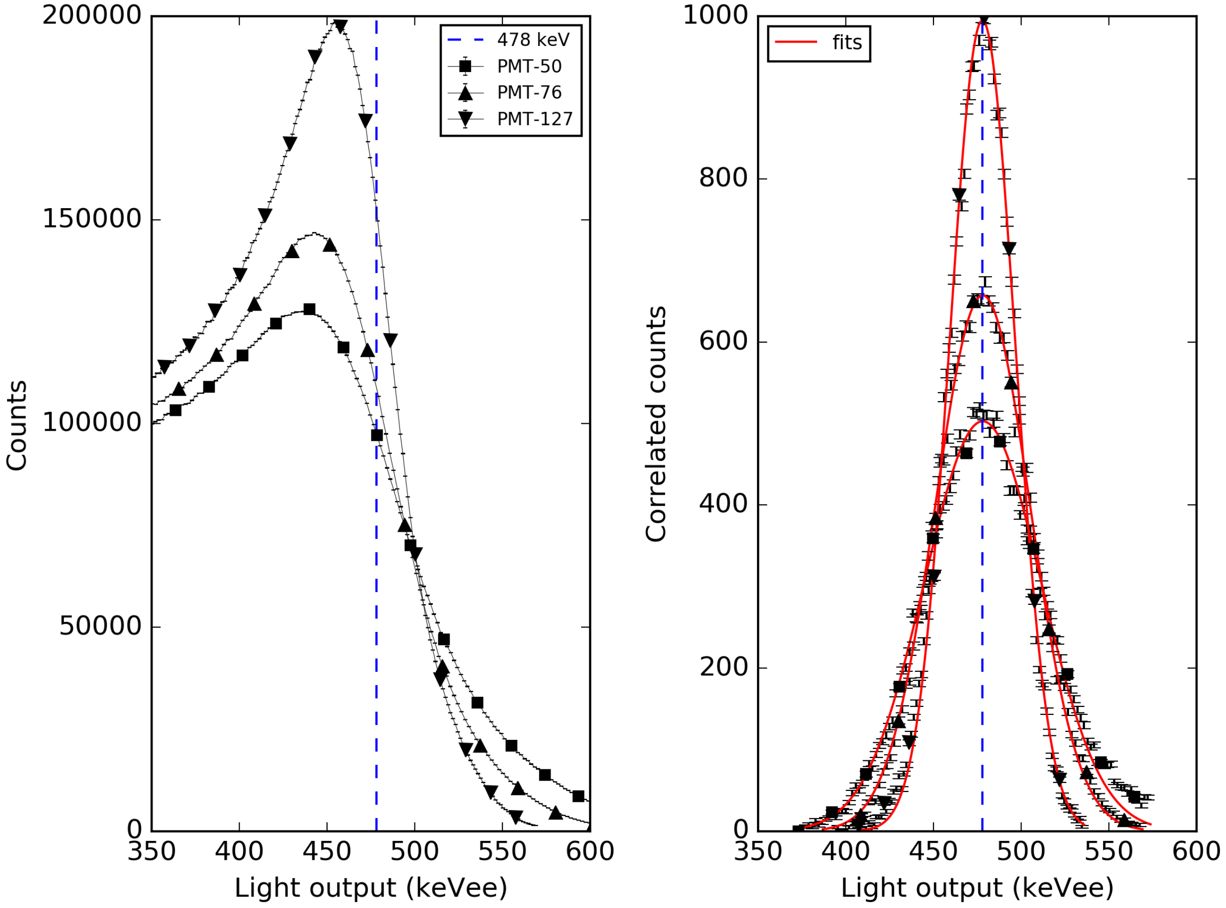


Figure 4.8: Standard (left) and correlated (right) light output spectra using the ground cone.

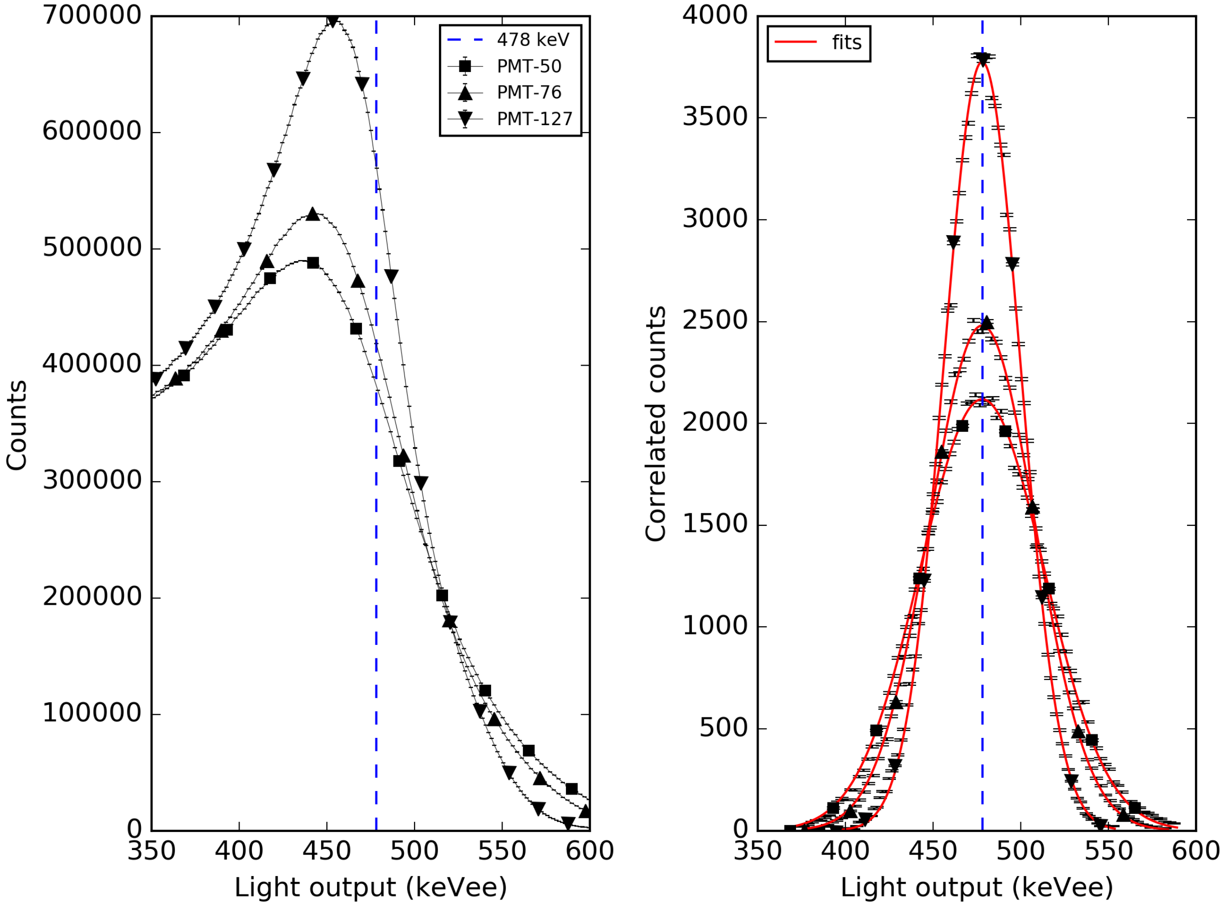


Figure 4.9: Standard (left) and correlated (right) light output spectra using the ground cylinder.

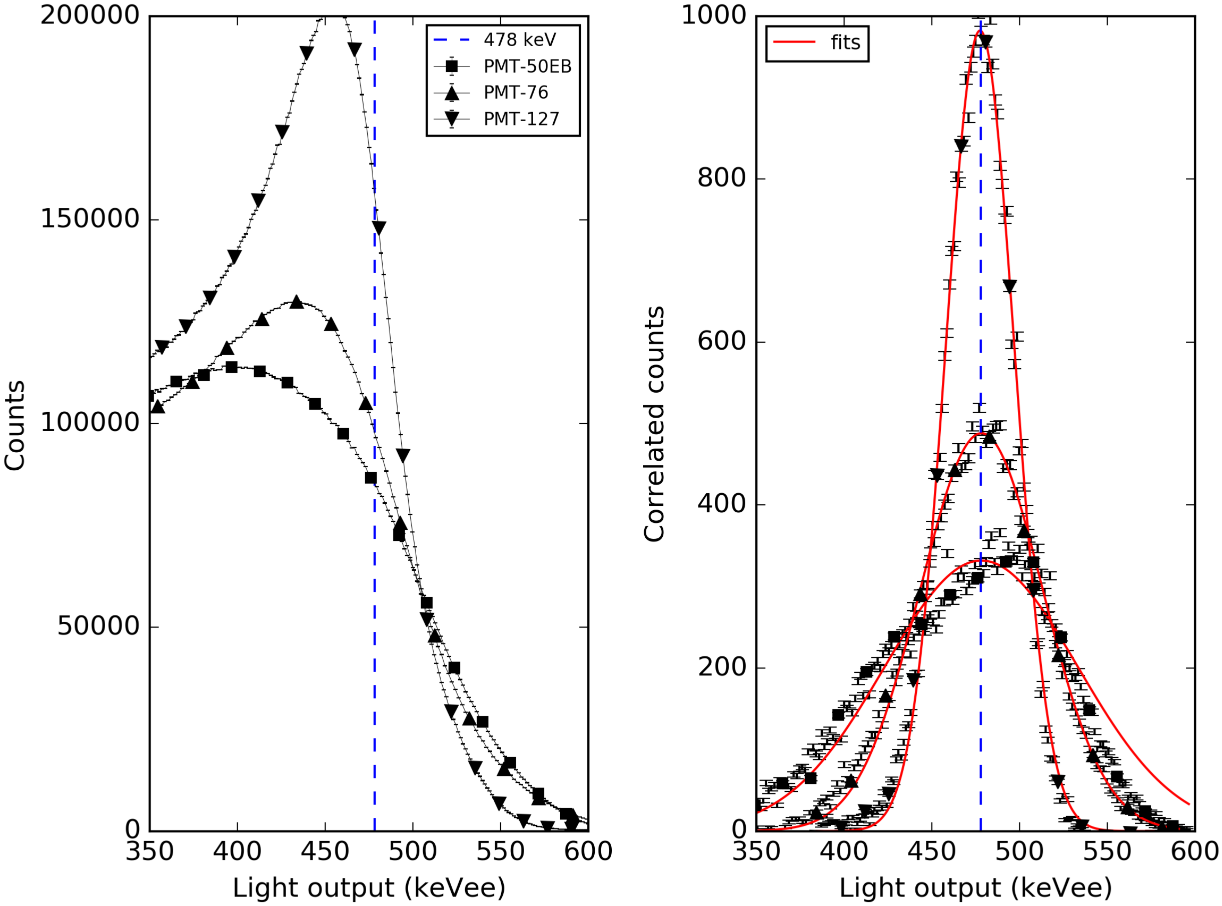


Figure 4.10: Standard (left) and correlated (right) light output spectra using the polished cone.

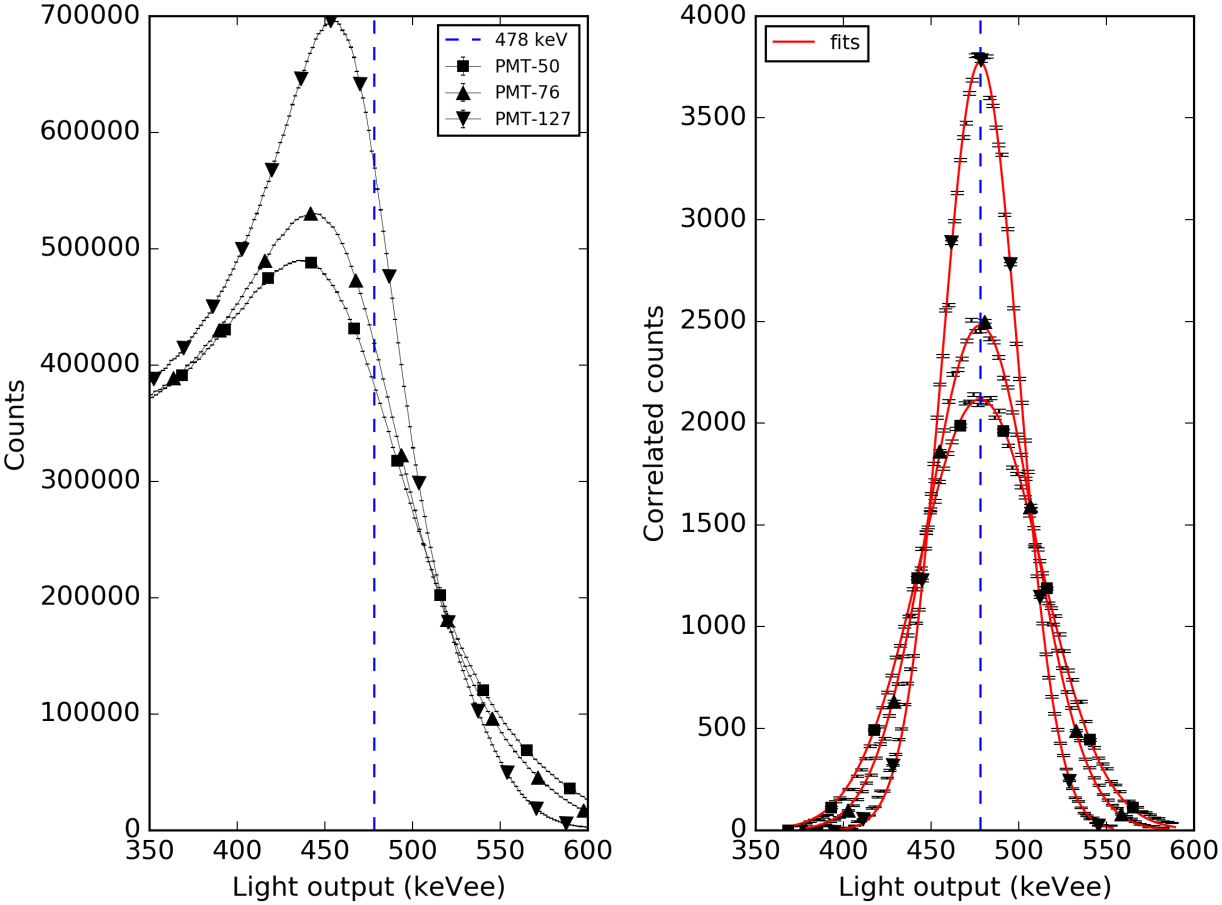
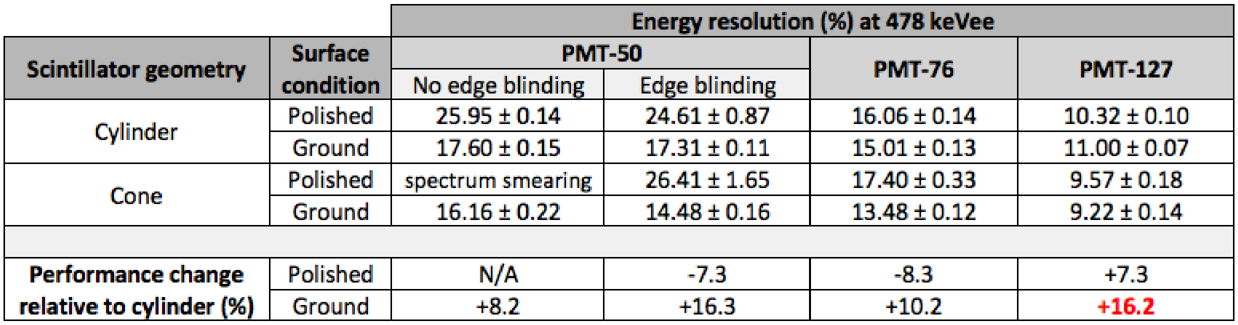


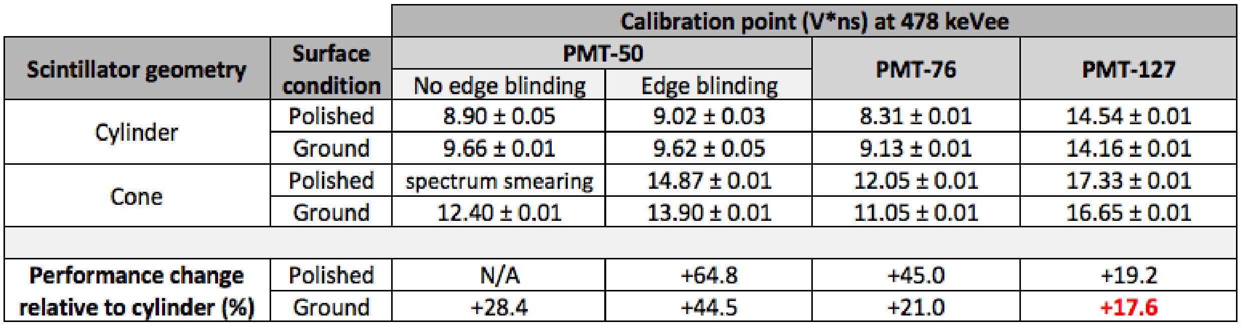
Figure 4.11: Standard (left) and correlated (right) light output spectra using the polished cylinder.

Table 4.1: Energy resolution values.



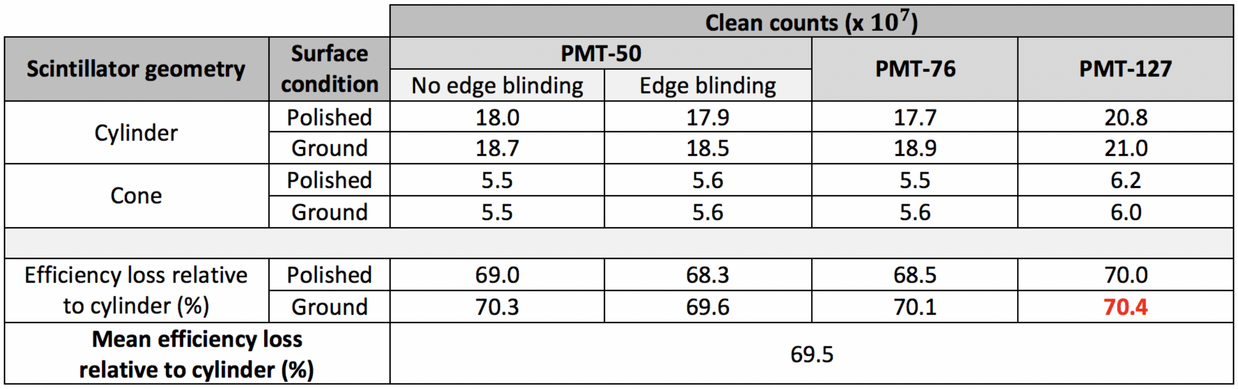
		Energy resolution (%) at 478 keVee			
Scintillator geometry	Surface condition	PMT-50		PMT-76	PMT-127
		No edge blinding	Edge blinding		
Cylinder	Polished	25.95 ± 0.14	24.61 ± 0.87	16.06 ± 0.14	10.32 ± 0.10
	Ground	17.60 ± 0.15	17.31 ± 0.11	15.01 ± 0.13	11.00 ± 0.07
Cone	Polished	spectrum smearing	26.41 ± 1.65	17.40 ± 0.33	9.57 ± 0.18
	Ground	16.16 ± 0.22	14.48 ± 0.16	13.48 ± 0.12	9.22 ± 0.14
Performance change relative to cylinder (%)	Polished	N/A	-7.3	-8.3	+7.3
	Ground	+8.2	+16.3	+10.2	+16.2

Table 4.2: Energy calibration values.



		Calibration point (V*ns) at 478 keVee			
Scintillator geometry	Surface condition	PMT-50		PMT-76	PMT-127
		No edge blinding	Edge blinding		
Cylinder	Polished	8.90 ± 0.05	9.02 ± 0.03	8.31 ± 0.01	14.54 ± 0.01
	Ground	9.66 ± 0.01	9.62 ± 0.05	9.13 ± 0.01	14.16 ± 0.01
Cone	Polished	spectrum smearing	14.87 ± 0.01	12.05 ± 0.01	17.33 ± 0.01
	Ground	12.40 ± 0.01	13.90 ± 0.01	11.05 ± 0.01	16.65 ± 0.01
Performance change relative to cylinder (%)	Polished	N/A	+64.8	+45.0	+19.2
	Ground	+28.4	+44.5	+21.0	+17.6

Table 4.3: Detected counts for all energy-resolution experiments.



		Clean counts (x 10^7)			
Scintillator geometry	Surface condition	PMT-50		PMT-76	PMT-127
		No edge blinding	Edge blinding		
Cylinder	Polished	18.0	17.9	17.7	20.8
	Ground	18.7	18.5	18.9	21.0
Cone	Polished	5.5	5.6	5.5	6.2
	Ground	5.5	5.6	5.6	6.0
Efficiency loss relative to cylinder (%)	Polished	69.0	68.3	68.5	70.0
	Ground	70.3	69.6	70.1	70.4
Mean efficiency loss relative to cylinder (%)		69.5			

Figure 4.12 shows the uncalibrated pulse-integral distributions of the polished and ground conical scintillators on PMT-50 with and without edge blinding. Spectrum smearing was partially alleviated with edge blinded. Although the detector response of the polished cone appeared more uniform with edge blinding, its energy resolution was poor. A ground cone on PMT-50 without edge blinding also alleviated spectrum smearing, however LCE was decreased and its improvement in energy resolution over the cylinder in the same configuration was marginal. A ground cone on PMT-50 with edge blinding on the other hand, not only produced a more uniform detector response, but also had a significant improvement in energy resolution over the cylinder in the same configuration.

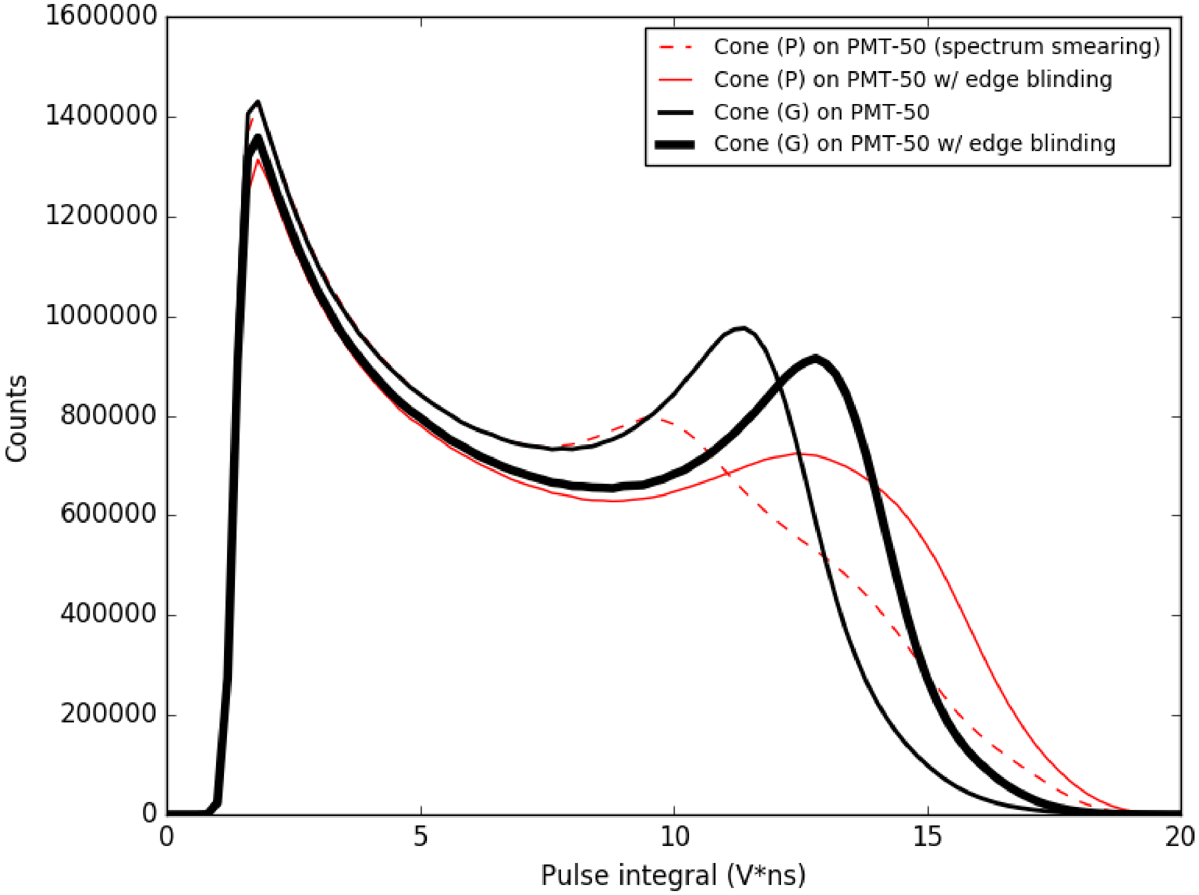


Figure 4.12: A comparison of four standard pulse-integral distributions for the cone on PMT-50, where letters “P" and “G" refer to a polished and ground surface, respectively.

4.4 Discussion

Due to a detection efficiency loss of approximately 70%, a cone may not be the ideal choice for some applications. Results indicate that a conical geometry provides enhanced LCE over a standard cylindrical shape. Although LCE was not directly measured, the gain in LCE was interpreted using the calibration position of both geometries at the Compton edge (478 keVee) of cesium-137. A 16.2% improvement in energy resolution at 478 keVee was observed when using a conical scintillator in a place of a cylindrical scintillator of identical base diameter and height (50 mm), surface treatment (un-polished), and material (EJ200). These conditions were observed when the two scintillators were compared using a PMT (Photonis XP4512B) with a base diameter of 127 mm. However, improvements in energy resolution were strongly dependent on the light collection and conversion process. By using a smaller-sized PMT, the percent-increase in energy resolution over a cylinder decreased, and in the case of a machine-polished cone mated to a PMT of equal base diameter, a smeared energy spectrum was produced (see Figure 4.12). An increase in LCE can only benefit the user if the detector response remains uniform. It was demonstrated that the spatial uniformity of the photocathode response played a crucial role in the usability of the cone, in addition to the surface treatment of the cone.

To avoid spectrum smearing in a detection system that uses conical scintillators, the following is recommended: (1) a slightly ground surface instead of a machine-polished surface, and (2) the use of a light-sensing device with a spatially-uniform detection efficiency. The second point can be addressed in a few ways. If an application has size constraints, a PMT with an equal base diameter to the cone can be used if the perimeter of the PMT window is blinded by a highly reflective material, for example a specular reflector cut into the shape of a ring, ideally with an adhesive backside. Edge blinding may require some optimization, specifically with the inner diameter of the reflective ring. If size constraints eliminate the use of a PMT altogether, a silicone-photomultiplier array [46] may provide favorable results. However, if a detector application is not constrained by size, the simplest solution is to couple the ground cone to the center of a large PMT, which would also enable the use of a polished cone.

CHAPTER 5

Experiment: Time resolution

5.1 Motivation

The purpose of this work is to measure and compare the time properties of conical and cylindrical organic-scintillator detectors of equal base diameter, height, and material (EJ200). The time property measured for each geometry included the time resolution and pulse rise-time distribution. The time it takes a scintillator-based detection system to respond to a radiation event is a function of the scintillation-emission process, the light-collection process, and the PMT response. A reduction in the response time to any one of these three processes will improve the time resolution of the system. Chapter 4 showed that a cone had better energy resolution than the cylinder, and this result was attributed to enhancements in the light-collection process. Specifically, the cone exhibited greater LCE due to a reduction in optical-photon reflections when compared to the cylinder. These result are significant for their implications on the time properties of a detector system. If optical photons reflect less in the cone than in the cylinder, then it is expected that their average path length and time for detection is also reduced. In addition, the faster light-collection process in a conical scintillator also suggests that the pulse shape will change, and is expected to produce faster pulse rise times than the cylinder. The additional contribution of optical-photons towards the front end of the pulse could also reduce uncertainties in pulse-start time calculations for nuclear-safeguard and detector characterization experiments that rely on cross correlation and neutron time-of-flight techniques, where fast timing and excellent time determination is essential to their operation.

5.2 Methods

5.2.1 Equipment

Two conical and two cylindrical organic scintillators produced by Eljen with identical base diameter (50 mm), height (50 mm), material (EJ200), and surface roughness (machine polished), were used in the experiments described in this chapter. Figure 5.1 shows the two cones and two cylinders. Scintillators of same geometry were compared using a cross-correlation technique on two identical large-diameter PMTs (Photonis XP4512B). Channel 0 used PMT-127 from the energy-resolution experiments described in Chapter 4 and channel 1 used the same model PMT as channel 0 and was given the name PMT-127v. The v designation was used to differentiate the two PMTs, where v stands for "varied." In other words, the gain on channel 0 was fixed (-1700 V), and the gain on channel 1 was not. The reason for varying the gain on channel 1 is discussed in Section 5.2.4, where the importance of gain-matching two detectors in a cross-correlation experiment is further explained.

Polished instead of ground scintillators were compared to avoid unequal hand-grinding with sand paper of the scintillator surfaces. A change in the surface roughness would alter the total-reflectivity coefficient (TRC) [22, 23]. It is possible that grinding a machine-polished surface may not significantly bias results if hand grinding was carefully, lightly, and evenly applied to the surface as it was in Chapter 4 to one cone and one cylinder. However, since hand grinding would need to be repeated four times, it was decided that the process would be too tedious and that the potential for error was too high. Moreover, since this chapter focused on time resolution, fast response times are expected when applying a specular reflector to a polished surface. Therefore, a ground surface was not chosen, however future experiments may consider the ground surface if certain methods are in place, for example: (1) a machine surface grinder to ensure more uniform grinding and (2) a method for measuring the optical topography of the scintillator surface for estimating surface roughness. Since these methods were not accessible, it was decided to compare the polished scintillators for easily repeatable measurements in the future. It is worth mentioning that one conclusion of Chapter 4 was that the energy resolution of a polished-conical scintillator is strongly susceptible to photocathode non-uniformity effects. Therefore, two identical large-diameter PMTs were used to mitigate this effect (PMT-127 and PMT-127v). Optical grease (EJ500) was used to couple the scintillator bases to the center of the two PMTs.

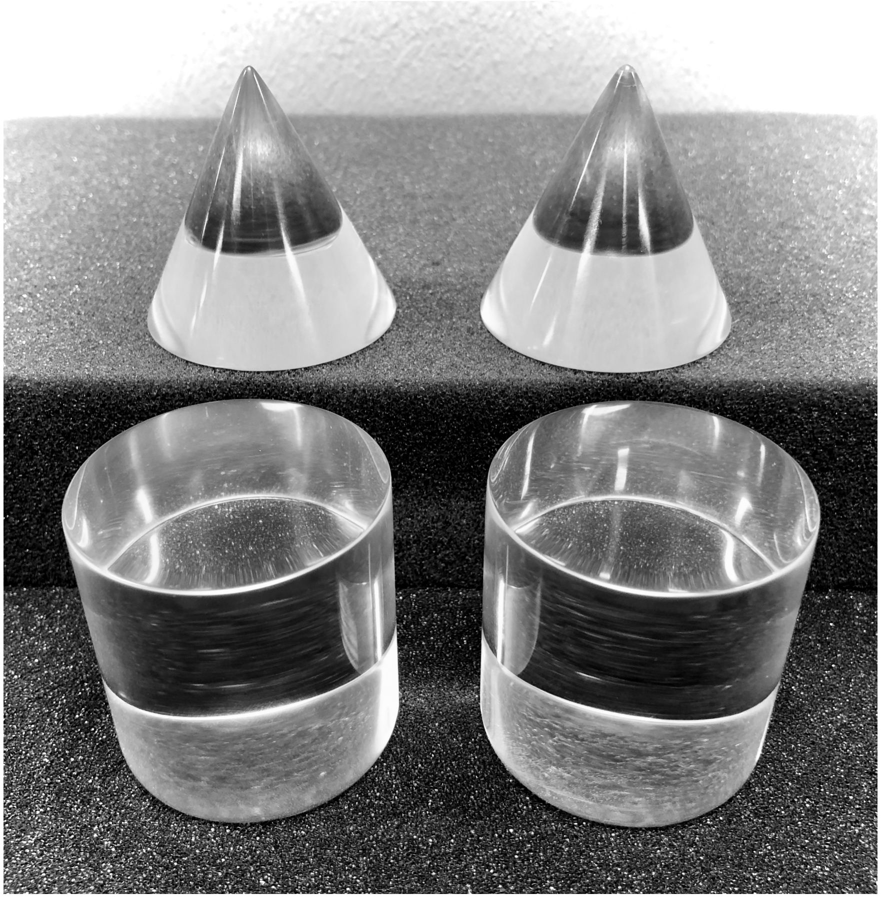


Figure 5.1: A picture of the four polished organic scintillators (EJ200) used in this chapter.

5.2.2 Time pick-off

Organic-scintillators produce a very fast scintillation pulse and therefore digitization error can be a concern if the sampling rate of the DAQ system is too low. The sampling rate of the digitzer is known to affect the quality of any pulse-shape dependent calculation, for example the rise time, the energy deposited, and the particle identity when using charge integration and a PSD-capable organic scintillator. Although the DAQ used in this work had a very fast sampling rate (2 ns steps) for a system with such a high voltage resolution across 2 V (16,384 channels), it was only able sample approximately 5 data points (a range of 10 ns) along the leading edge of any pulse. Therefore, the leading edge of the pulse was linearly interpolate between each point to more accurately calculate the time pick-off. The pulse rise time can also be found by calculating the time interval between two positions along the leading edge; 10% and 90% of the pulse amplitude [5]. This process is illustrated in Figure 5.2. A histogram of these time intervals produces a rise-time distribution. Rise-time distributions were calculated from pulses in channel 0 and 1 for both cross-correlation experiments (cylinder and cone). The experiments are described below in Section 5.2.3.

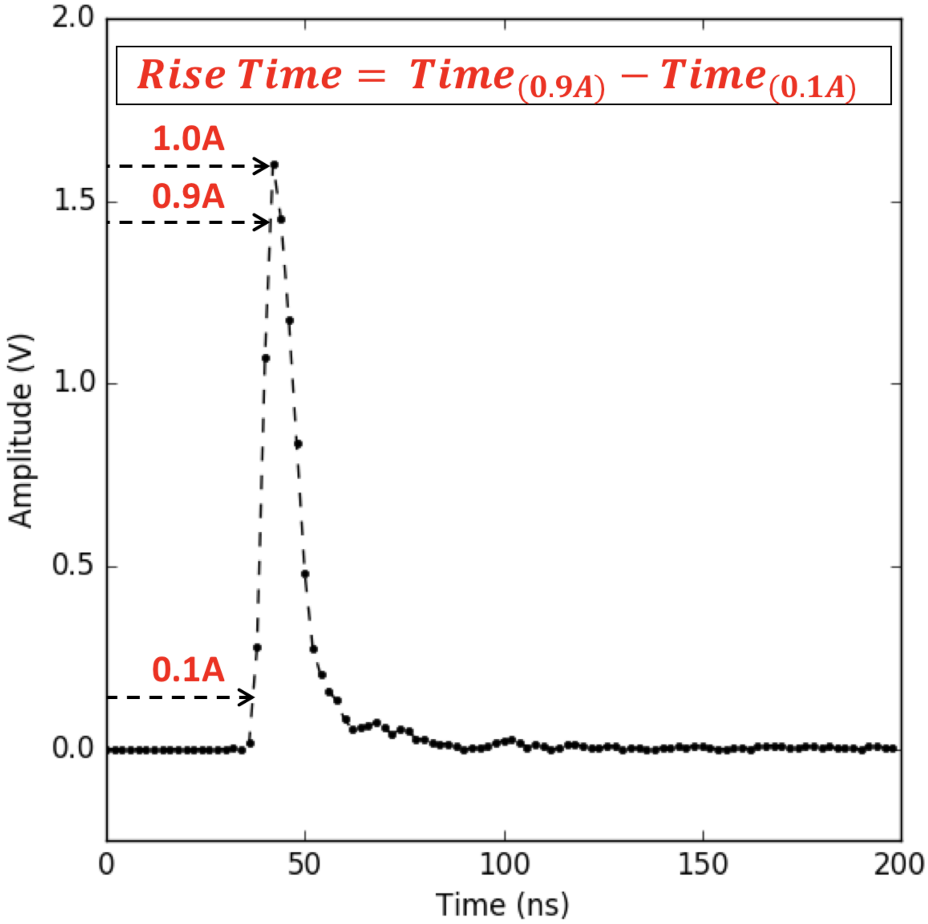


Figure 5.2: An illustration of time-pick off for measuring the rise time of a scintillator pulse.

5.2.3 Cross correlation setup

A graphical representation of how cross correlation works along with the dimensions of the experiment setup are described in Section 3.3.2. A picture of the setup for both cross correlation experiments is shown in Figure 5.3. PMT-127 was used for channel 0 and operated at a voltage gain of -1700 V. PMT-127v was used for channel 1. Many uncorrelated experiments of a sodium-22 source were performed to gain-match the detector response of channel 1 to channel 0, and this process was repeated for both geometry pairs. Gain matching is further discussed in Section 5.2.4. After successfully gain matching channel 1 to channel 0, a 5 minute cross-correlation experiment was performed for the two geometry pairs. A geometry pair is defined as the two identically scintillators used in a cross correlation experiment; for example, the two polished cylinders (one pair) and the two polished cones (another pair). The calibration procedure for each channel is described in Section 5.2.5. After both cross-correlation experiments, pulses from channel 0 and 1 were correlated using the processing algorithm described earlier in Section 3.3.2, which calculated the time resolution as a function of CFD. It is worth mentioning that since count times were identical between geometry pairs (5 minutes) and because the volume of the cone is less than the cylinder by 66.67%, all time-resolution analyses of the cylinder were forced to use the same number correlated events as the cone (29,604) to fairly compare statistical uncertainty.

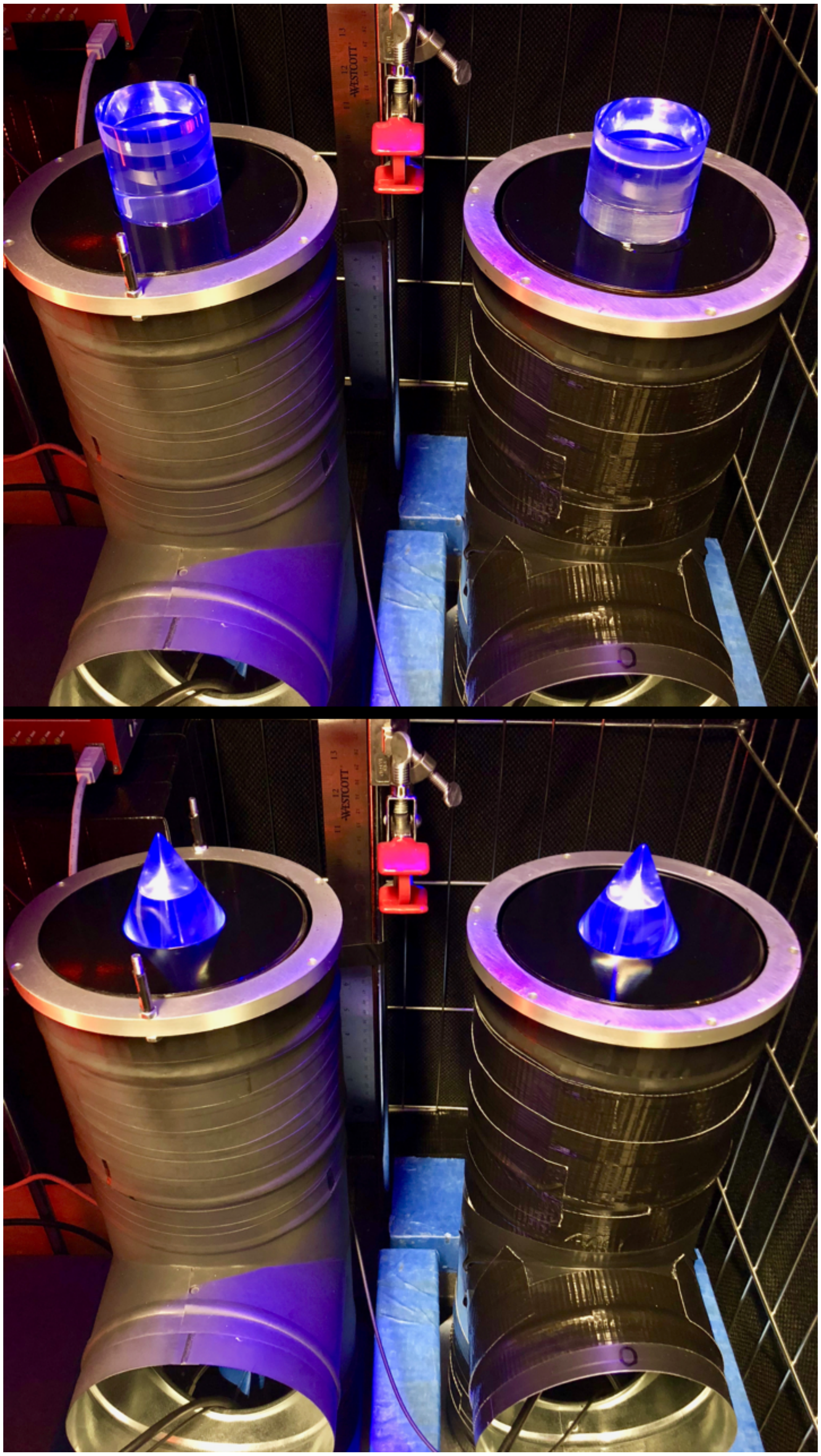


Figure 5.3: Cross-correlation experiments of the cone and cylinder shown without their reflector. Specular reflector encasements were used during the experiment.

5.2.4 Gain matching

For each geometry pair, numerous gain-matching experiments were conducted using the same sodium-22 source and experiment setup shown in Figure 5.3. All gain-matching experiments lasted 5 minutes. In each experiment, both channels simultaneously collected data. The trigger logic of the DAQ system for each gain-matching experiment was identical to the cross-correlation experiment, where each channel was allowed to trigger the other. By allowing one channel to trigger the other, both channels could simultaneously measure an uncorrelated energy spectrum of sodium-22 (in units of pulse integral [V*ns]). After each experiment, the Compton edge in the pulse-integral distribution of channel 0 and 1 were compared. The goal of each experiment was to find a gain for channel 1 that produced a calibration point in close proximity to the calibration point for channel 0. The calibration methodology is further discussed in Section 5.2.5. Channel 0 was therefore used as a detector-response benchmark that channel 1 had to match closely to. The gain of channel 0 was set to -1700 V and never changed for any iteration. Channel 0 was used as a benchmark because its exact configuration (the same polished-scintillator geometries, reflector, and PMT model), had already been accurately calibrated by a Compton-coincidence experiment described in Chapter 4. Gain matching was repeated for both geometry pairs and the best results are shown below in Figure 5.4. After successfully gain matching channel 1 to channel 0, a cross-correlation algorithm sorted through all pulses, in both channels, for both geometry pairs. The algorithm search for correlated 511 keV Compton-scatter interactions from the same annihilation event, within a fixed light-output range (200 keVee to 370 keVee). Recall that the Compton edge for a 511 keV gamma ray is 341 keV. The algorithm was previously described in Section 3.3.2.

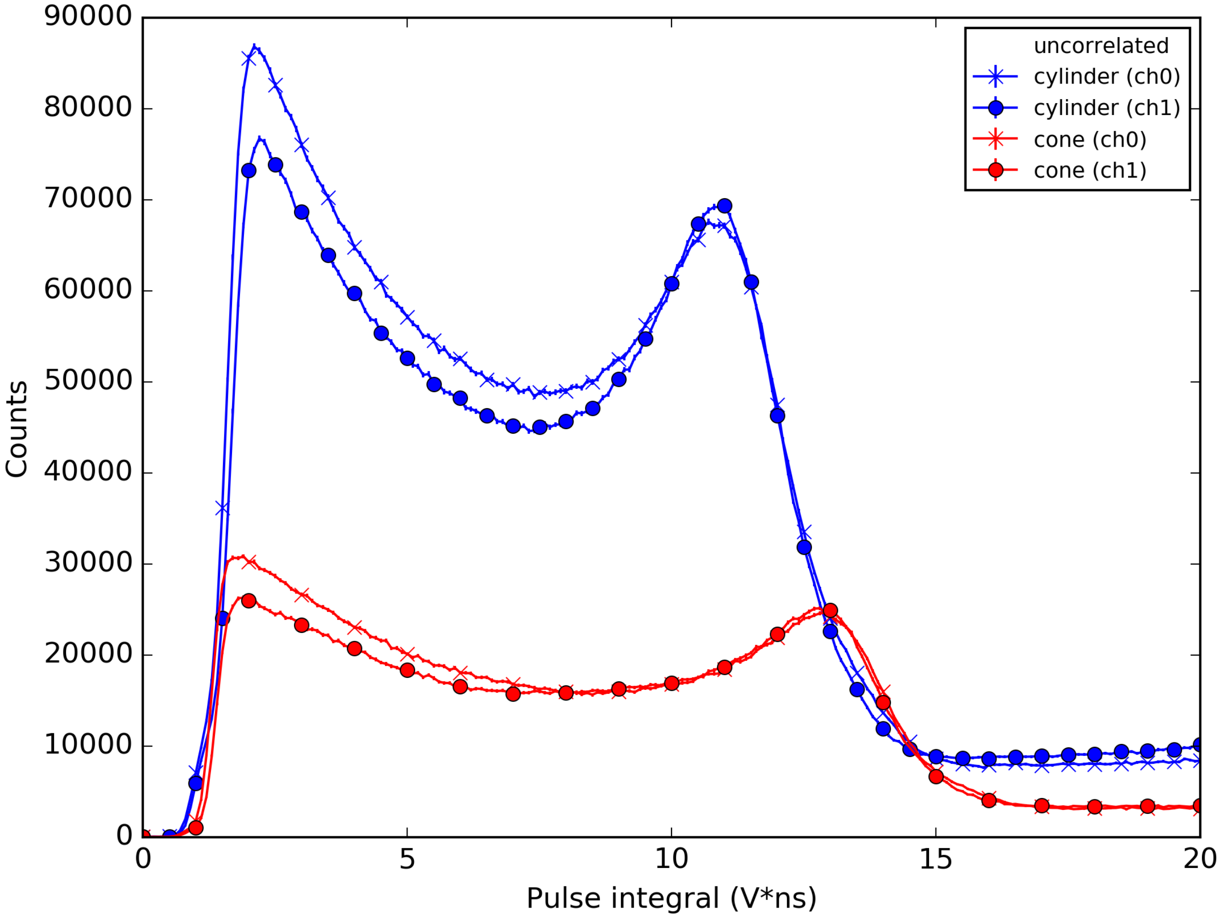


Figure 5.4: Standard (uncorrelated) pulse-integral spectra of both geometry pairs after successfully gain matching channel 1 to channel 0.

5.2.5 Calibration

A fractional Compton-edge approach was used to calibrate all pulse-integral distributions to light output for the gain-matching and cross-correlation experiments. The pulse-integral distribution of the two cylinders and cones were calibrated to 81.5% of the Compton-edge peak for sodium-22. Both cones were calibrated to 77.7%. The percentage was used to link the pulse-integral value of an uncalibrated spectrum with the maximum-energy deposited (341 keV) by a 511 keV gamma ray in a single Compton scatter.

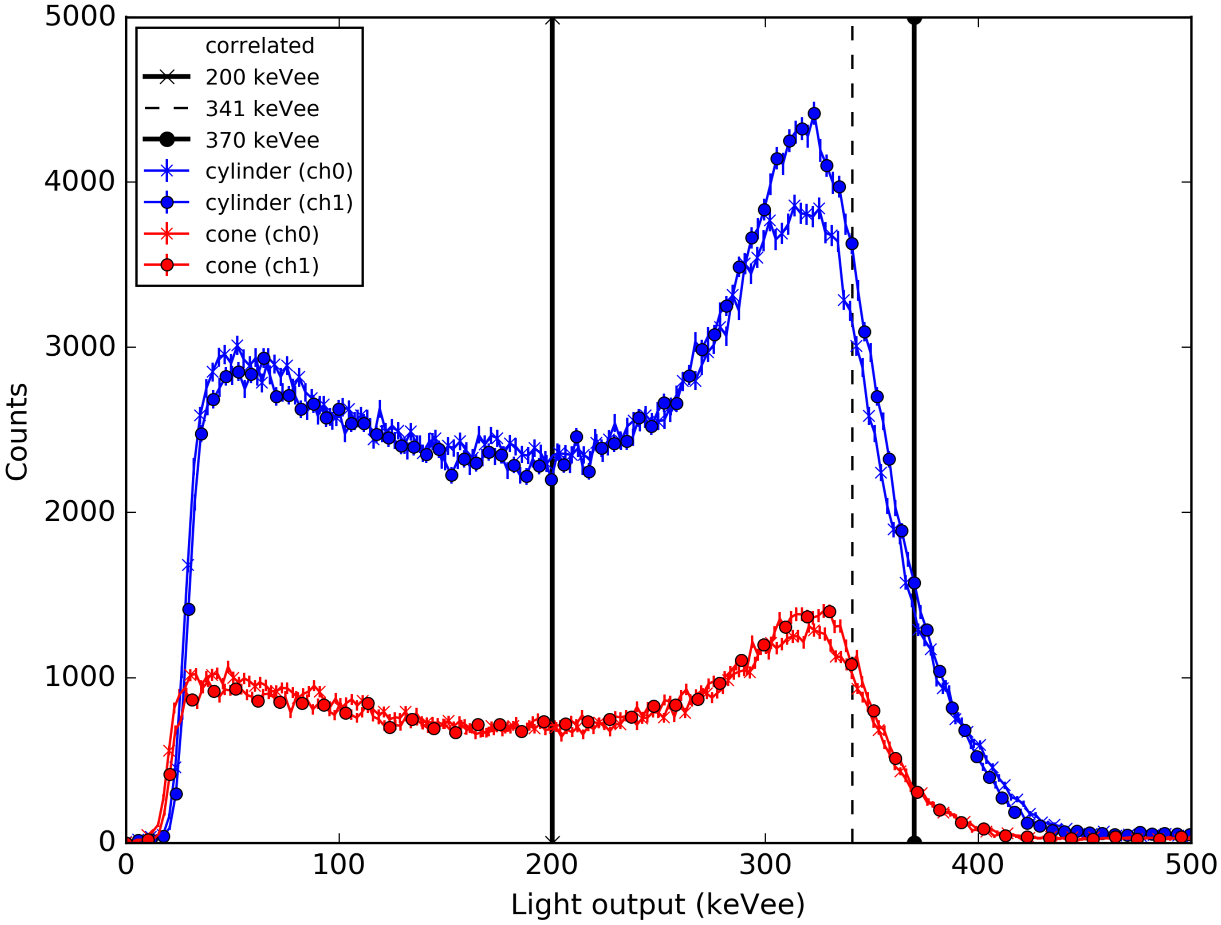


Figure 5.5: Light-output spectra after gain matching, correlation, and calibration.

These percentages were interpreted from the results presented in Chapter 4, specifically from the data shown in Figures 4.10 and 4.11. Both figures show uncorrelated and correlated (by Compton coincidence) light-output spectra of cesium-137, as measured by the same detector configurations for channel 0. Although these percentages were determined from the Compton edge of cesium-137, they can still be applied to the Compton edge of the 511 keV gamma ray from sodium-22. It is well documented that the light output from electron recoils in organic scintillators is linear above 100 keV [2]. The calibrated light-output spectra for channel 0 and 1 for both geometry pairs are shown in Figure 5.5.

5.2.6 CFD analysis

A pulse pair is defined as a pulse in channel 0 that is correlated to a pulse in channel 1, and vice versa. An algorithm described in Section 3.3.2 correlates pulse pairs by using a time and energy gate on the calibrated light-output spectra (see Figure 5.5) of both geometry pairs. Pulse pairs must satisfy the requirements of the algorithm, namely that the two pulses must have the same time stamp, and that their light output be between a LLD (200 keVee) and a ULD (370 keVee). Pulse pairs that satisfy these requirements are flagged as “clean" and are processed further. The time of each pulse in a clean pair is found using a particular CFD to calculate their time interval (ΔT) and produce a cross-correlation distribution. This process was repeated as a function of CFD for all clean pairs. For each geometry pair, a total of 30 CFD values were used, each producing a cross-correlation distribution. The purpose for calculating the cross-correlation distribution as a function of CFD was to determine which CFD produced the best time-resolution results; useful for comparing the time-resolution of the two geometries. Section 5.3.1 compares all the cross-correlation distributions for the cylinder pair and cone pair as calculated by the 30 CFD values. Section 5.3.2 discusses how the CFD was assessed for quality, and which CFD values were used as the basis for a fair and unbiased comparison of the time resolution of the cone and cylinder. Recall that time resolution is defined as the FWHM of a Gaussian fit to a cross-correlation distribution.

5.3 Results

5.3.1 Cross-correlation distributions

Figures 5.6 - 5.10 show the cross-correlation distribution of the cylinder and cone overlaid in the same plot for each CFD. The purpose of plotting all of these distributions together was to determine which CFD produced the best results for each geometry as a point of comparison. The CFD was finely sampled from near zero (0.001) to 0.7. The figures show two very interesting trends that occur in both geometries as the CFD is reduced.

The first trend is a progressive shift in the centroid to the left of the plot. This shift occurs for both geometries but is more prevalent for the cylinder. Both distributions converge to a ΔT of approximately 4 ns; this position on the time spectrum is referred to in this work as the system-response delay. It was interesting to note that the cone converged to this value more readily than the cylinder. In other words, high CFDs for the cone produced a cross-correlation distribution with a centroid close to 4 ns; unlike the cylinder. Since identical cable lengths were used to connect PMT-127 and PMT-127v to channels 0 and 1 respectively, the system-response delay is attributed to either DAQ channel being slower than the other. The second trend, namely a reduction in the integral of the distribution from high to low CFD, also occurs more readily in the cylinder. Conceptually, as the CFD is decreased, the time pick-off gets very close to the first digitized point of the pulse. Therefore, the cross-correlation distribution is independent of rise time and reveals the true system-response delay (4 ns). A low CFD also implies that the time pick-off is in close proximity to the baseline, where noise and digitization error of the first data point in the pulse can strongly influence the number of events in the cross-correlation peak. Recall that all distributions shown in Figures 5.6 - 5.10 are produced from the same number of correlated events (29,604) in the cylinder and cone, however as the CFD is reduced, the number of counts in the peak area leak to the right and left of the distribution as the time pick-off method begins to struggle with accurately assessing the start times for correlated-pulse pairs. This phenomena worsens at very low CFDs where the time pick-off is closer to baseline. A cross-correlation distribution that is truly representative of a detector systems time-interval between correlated-pulse pairs will produce a distribution where the majority of correlated counts contribute to the peak area. Using a low CFD to compare both geometries where some loss in useful correlated-peak events will not necessarily produce inaccurate results of the systems time resolution. However, a low CFD gives an isolated assessment of the time resolution by forcing reliance on pulses from channel 0 and 1 that both produced a high light output close to the ULD (370 keVee). A more holistic assessment of a systems time resolution will use a CFD where the total number of correlated events used to process the cross-correlation distribution, contribute to its peak area.

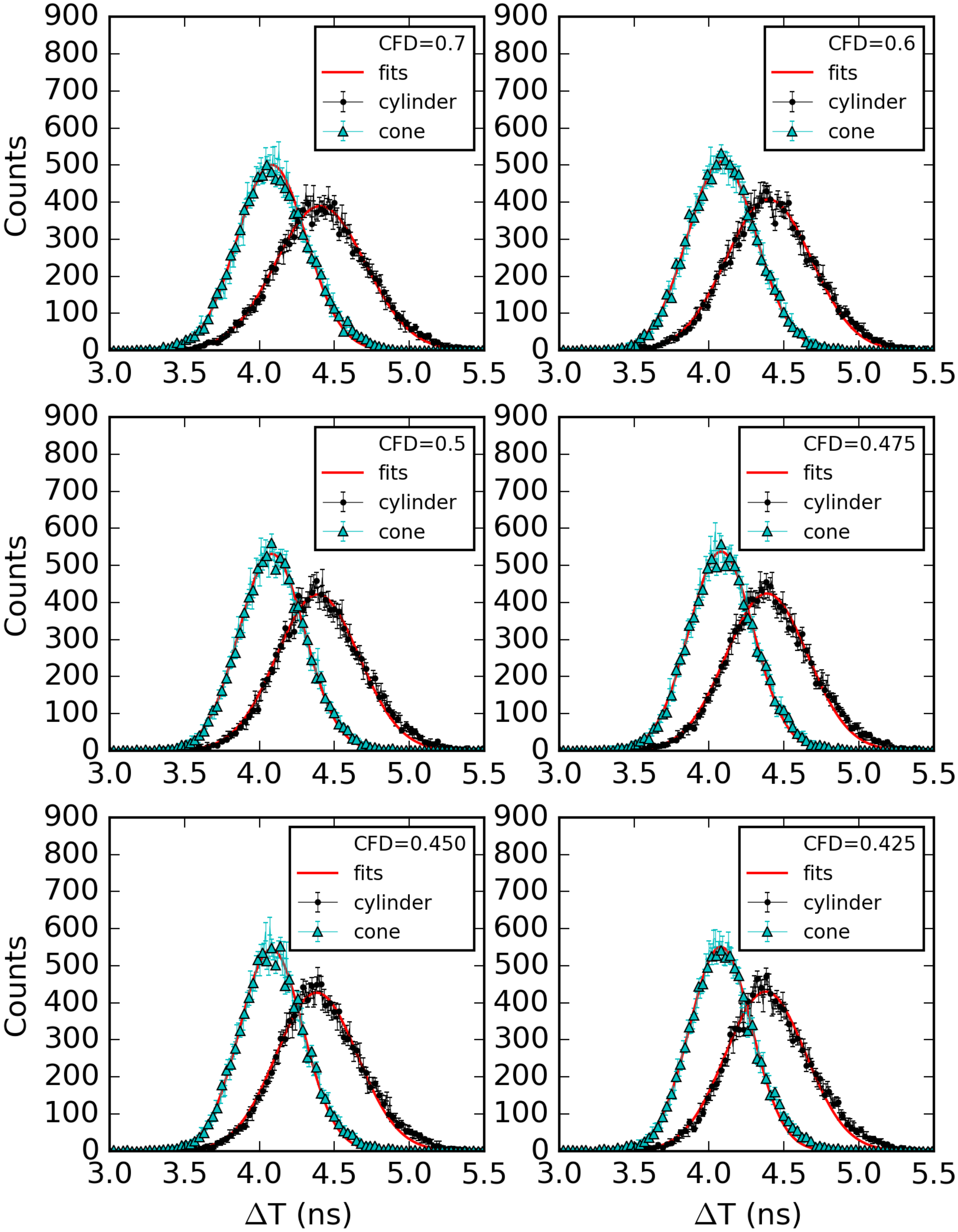


Figure 5.6: Cross-correlation distributions (set 1).

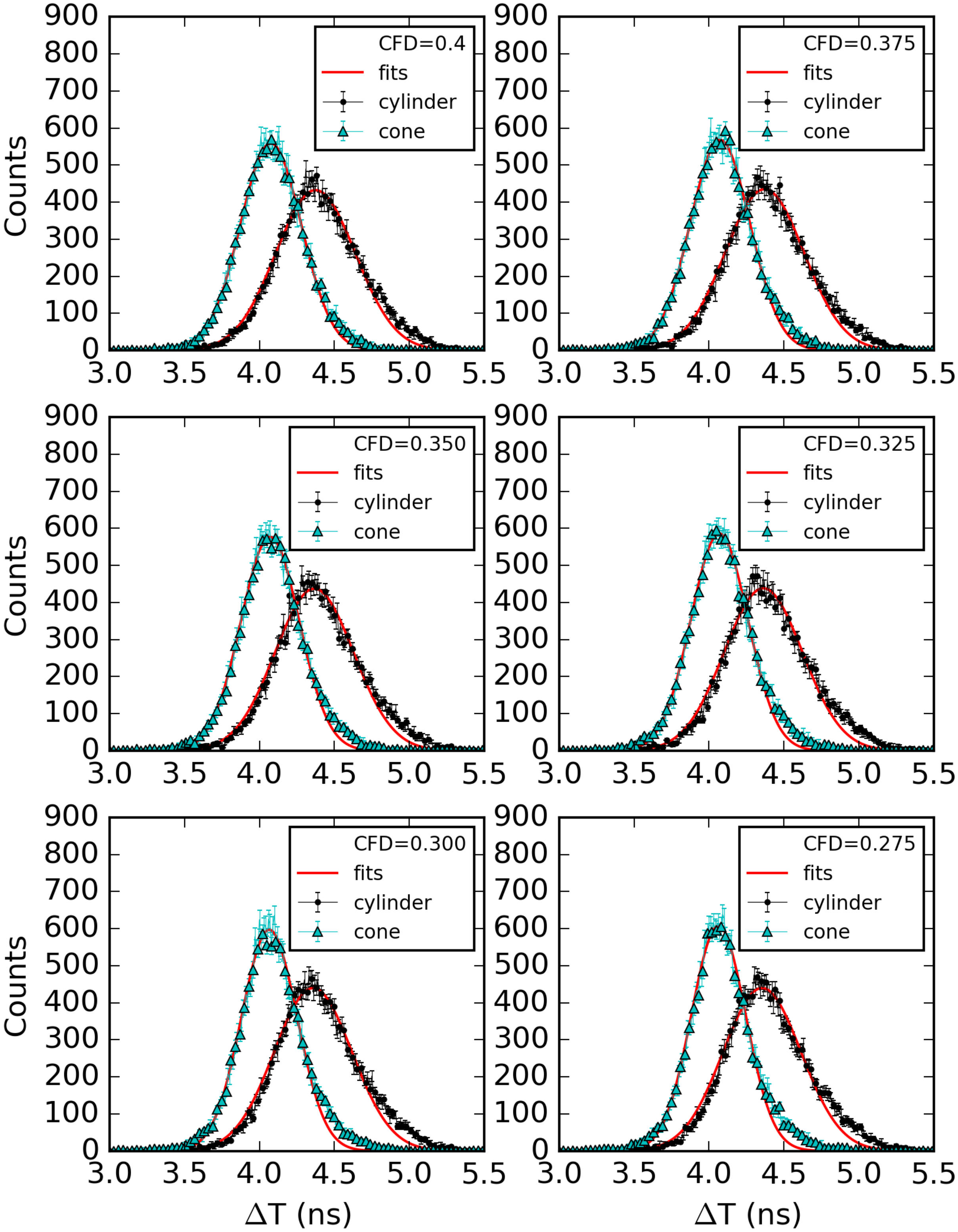


Figure 5.7: Cross-correlation distributions (set 2).

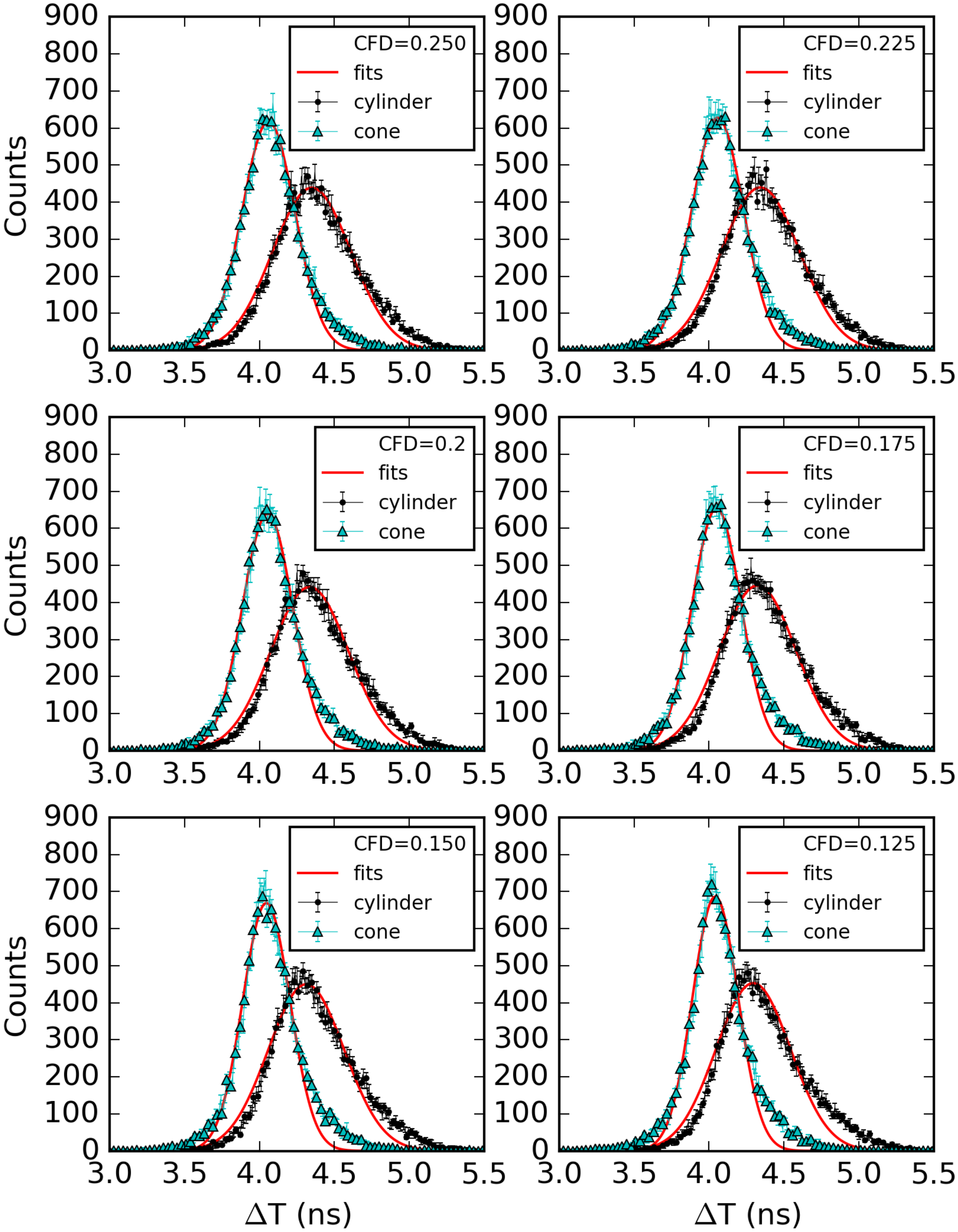


Figure 5.8: Cross-correlation distributions (set 3).

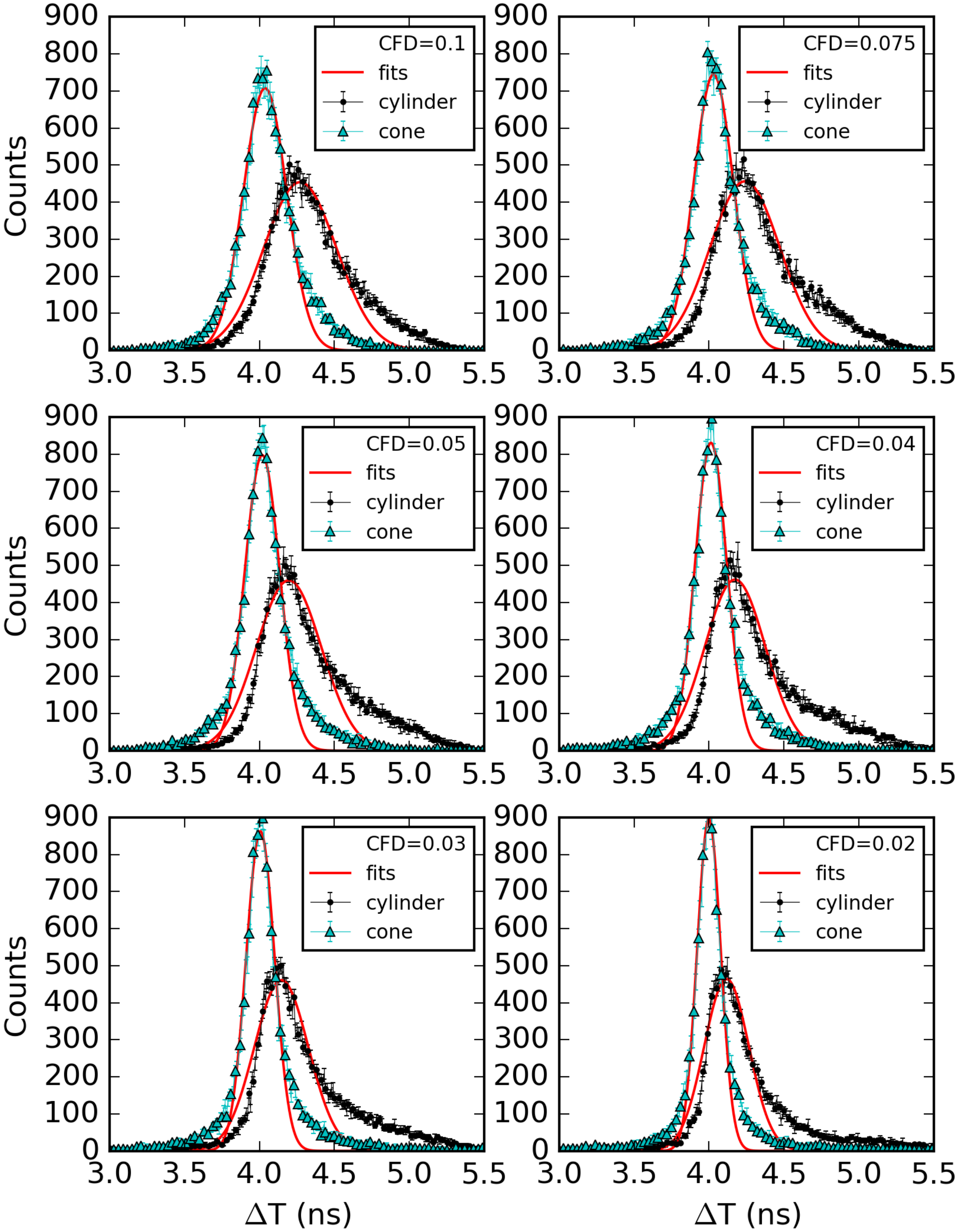


Figure 5.9: Cross-correlation distributions (set 4).

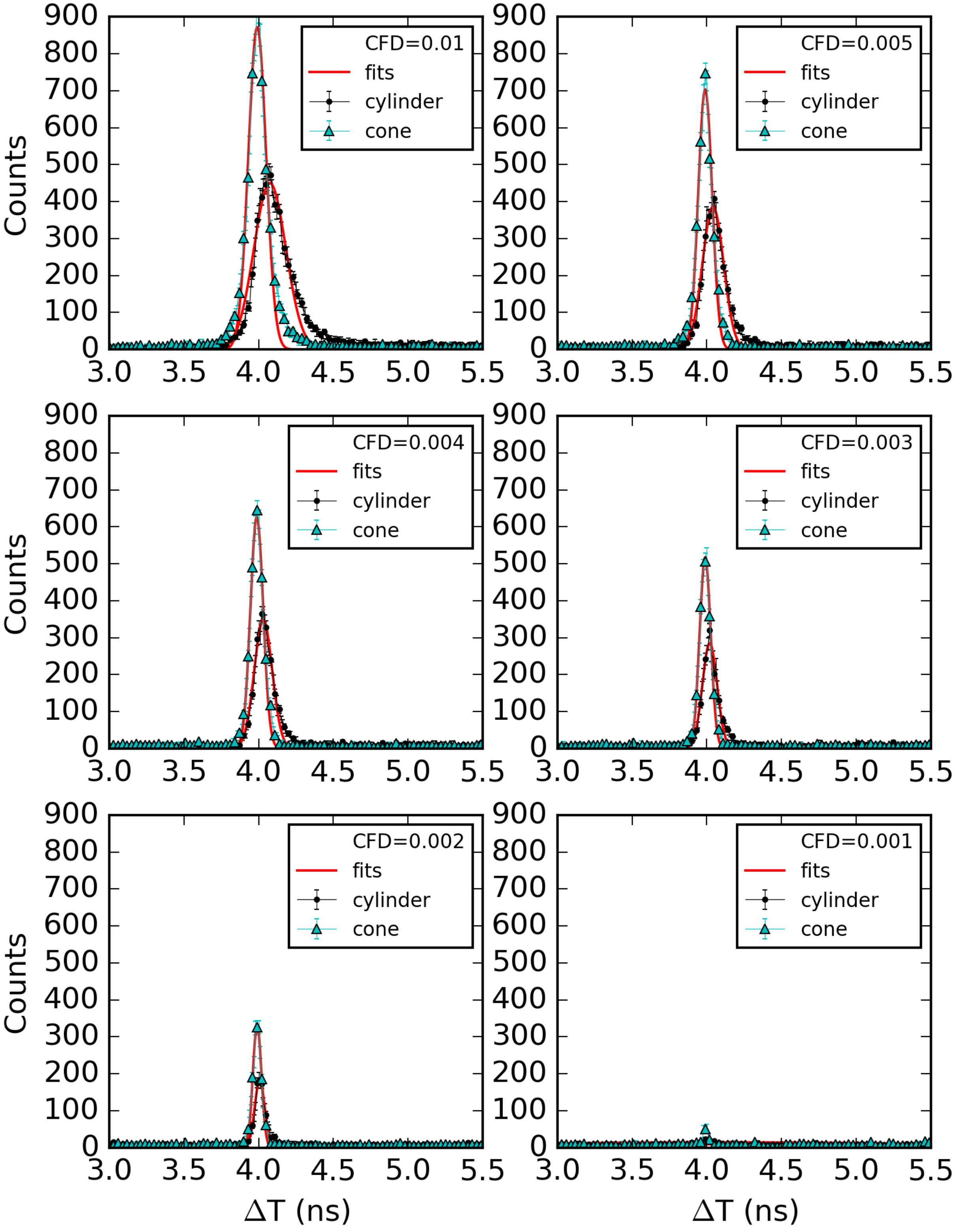


Figure 5.10: Cross-correlation distributions (set 5).

5.3.2 CFD determination

Figure 5.11 shows the percentage of correlated counts that contribute to the peak area, and the time resolution, as a function of CFD, for both geometries. As expected, when the CFD value is decreased, so is the time resolution. However, for a particular CFD, it is important to compare the time resolution with the correlated-count contribution to peak area to determine the usefulness of the time-resolution. In this work, efforts were made to ensure that the CFD chosen to compare the time resolution of the cylinder and cone maintained a minimum of 90% correlated-counts contributions to the peak area. In the end, two CFDs were chosen (0.2 and 0.5) to compare the time resolution of both geometries that met this peak-area criteria.

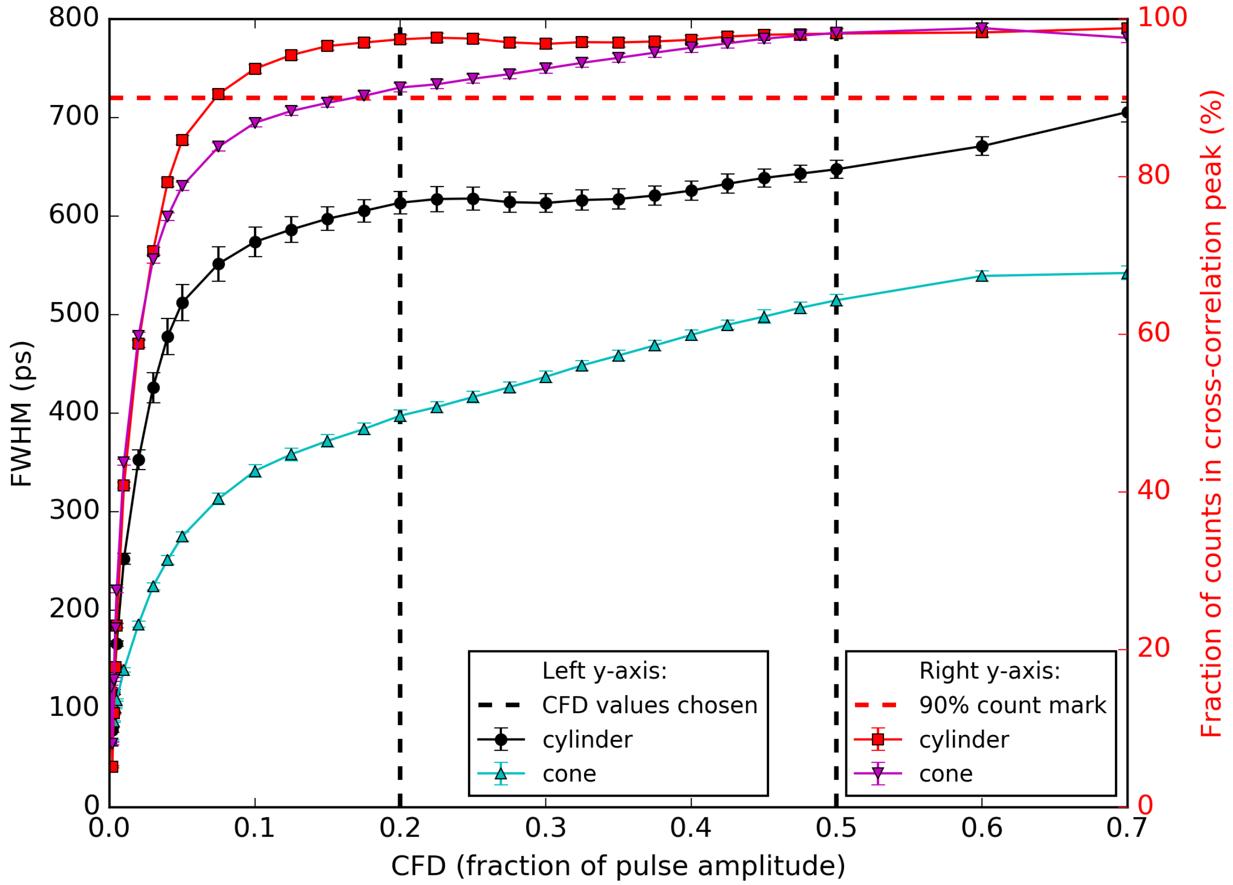


Figure 5.11: CFD determination methodology. The two top-most curves in the plot correspond to the right y-axis and the bottom two curves correspond to the left y-axis.

5.3.3 Time resolution

Figure 5.12 shows the cross-correlation distributions of both geometries as determined using both CFDs (0.2 and 0.5). Table 5.1 compares the time resolutions for both geometries and relative-percent increased achieved by the cone over the cylinder. For a CFD of 0.2 and 0.5, the cone demonstrates a gain in time resolution of approximately 35% and 21%, respectively, when compared to the cylinder.

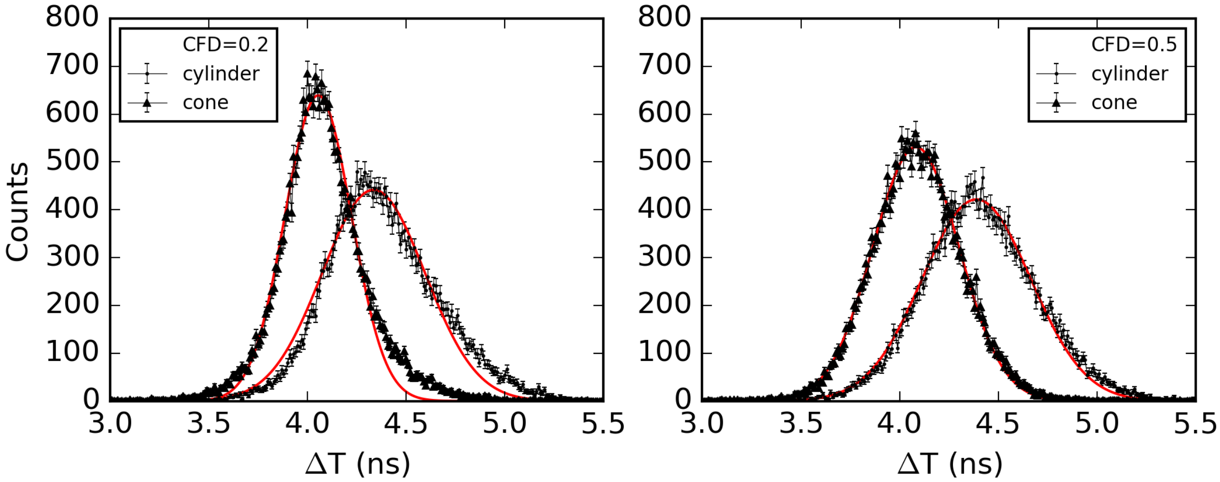
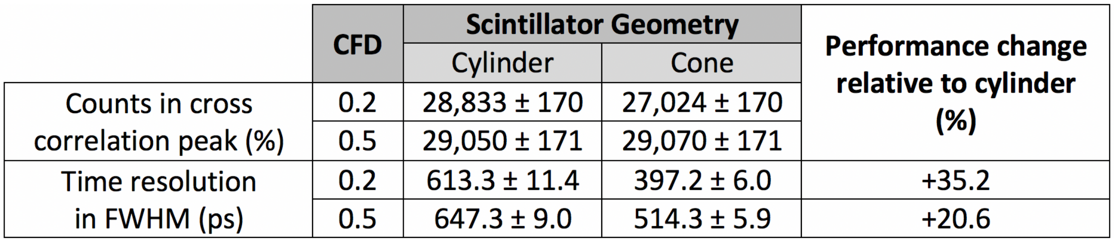


Figure 5.12: Cross-correlation distributions for the cylinder and cone.

Table 5.1: Time resolution values.



	CFD	Scintillator Geometry		Performance change relative to cylinder (%)
		Cylinder	Cone	
Counts in cross correlation peak (%)	0.2	28,833 ± 170	27,024 ± 170	
	0.5	29,050 ± 171	29,070 ± 171	
Time resolution in FWHM (ps)	0.2	613.3 ± 11.4	397.2 ± 6.0	+35.2
	0.5	647.3 ± 9.0	514.3 ± 5.9	+20.6

5.3.4 Rise-time distributions

Figure 5.13 shows the rise-time distributions for both geometries on channel 0 (top) and 1 (bottom). It is interesting to note that the width for all distributions shown (from leading edge to falling edge) is approximately 1.5 ns, which is very similar to the time-interval between sampled data points (2 ns). Therefore, the rise-time distributions are strongly impacted by digitization error of the DAQ. Recall that linear interpolation was used to assess time pick-off between sampled data points. Thus, some features of the rise-time distribution could be reconstructed despite the less than ideal sampling rate for rise-time estimation.

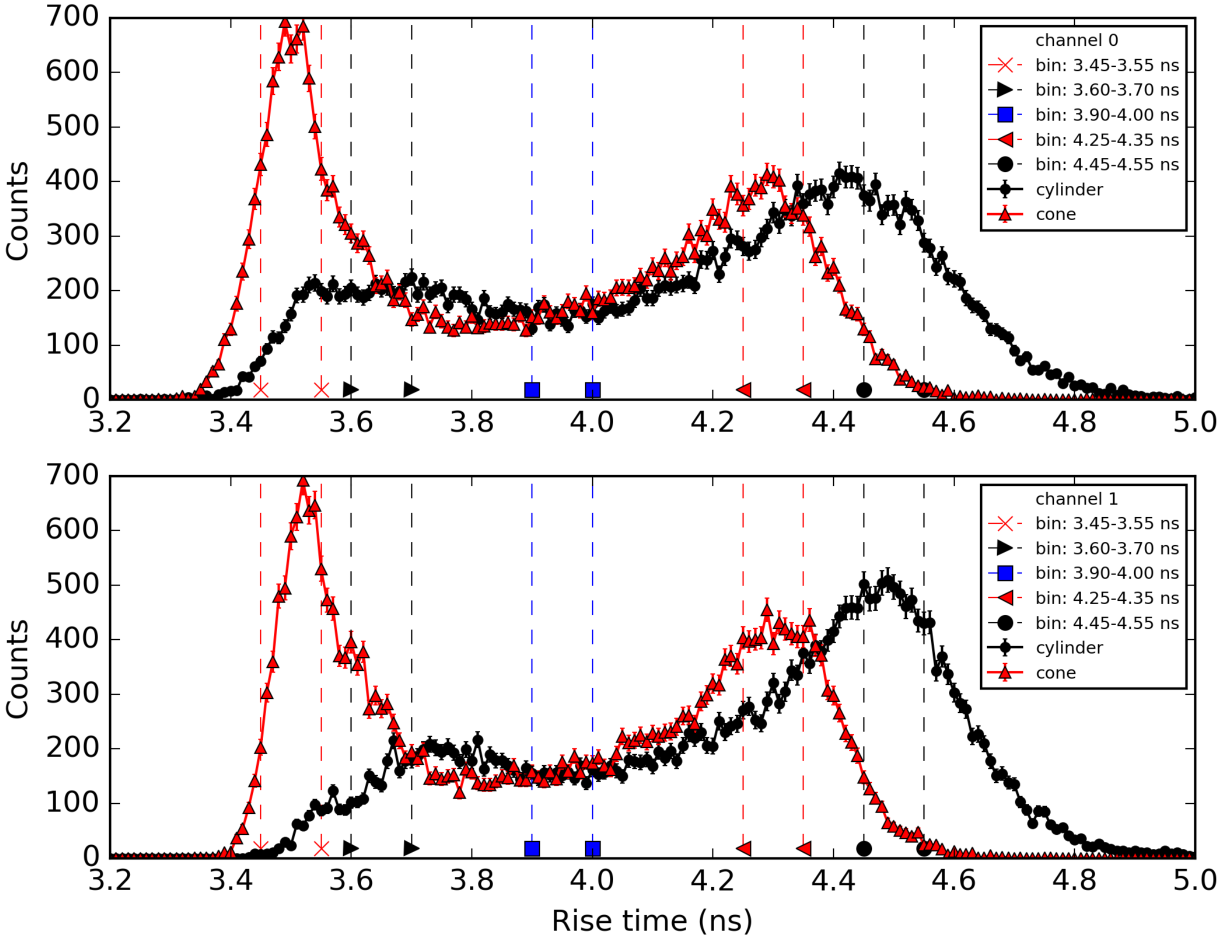


Figure 5.13: Rise-time distributions.

This work hypothesizes that the dual-peak shape of the rise-time distributions, particularly evident for both cones, are somehow related to the time-interval calculations determined from the time pick-off at 10% and 90% along the leading edge of the pulse. These results are interesting and should be investigated further. Initial attempts were made by comparing the shape of all pulses that contributed counts to three distinct rise-time intervals, each 0.10 ns in width, named LOW, $MIDDLE$, and $HIGH$. For both cones, the LOW, $MIDDLE$, and $HIGH$ regions corresponded to rise-time intervals between [3.45-3.44 ns], [3.90-4.00 ns], and [4.25-4.35 ns], respectively. For both cylinders, the LOW, $MIDDLE$, and $HIGH$ regions corresponded to rise-time intervals between [3.60-3.70 ns], [3.90-4.00 ns], and [4.45-4.55 ns], respectively.

5.3.5 Pulse-shape comparison

Exactly 500 pulses from the same channel, geometry, and rise-time region were extracted and compared in Figure 5.14 for the same light-output range (200 - 370 keVee). The pulses were initially misaligned because each channel was allowed to trigger the other. A post-processing script was written to force the first data point along the leading edge of the pulse above its baseline to begin at 43 ns. Once all pulses were normalized to the same starting position, the pulses for both geometries were overlaid in the same plot as a function of channel number and rise-time region.

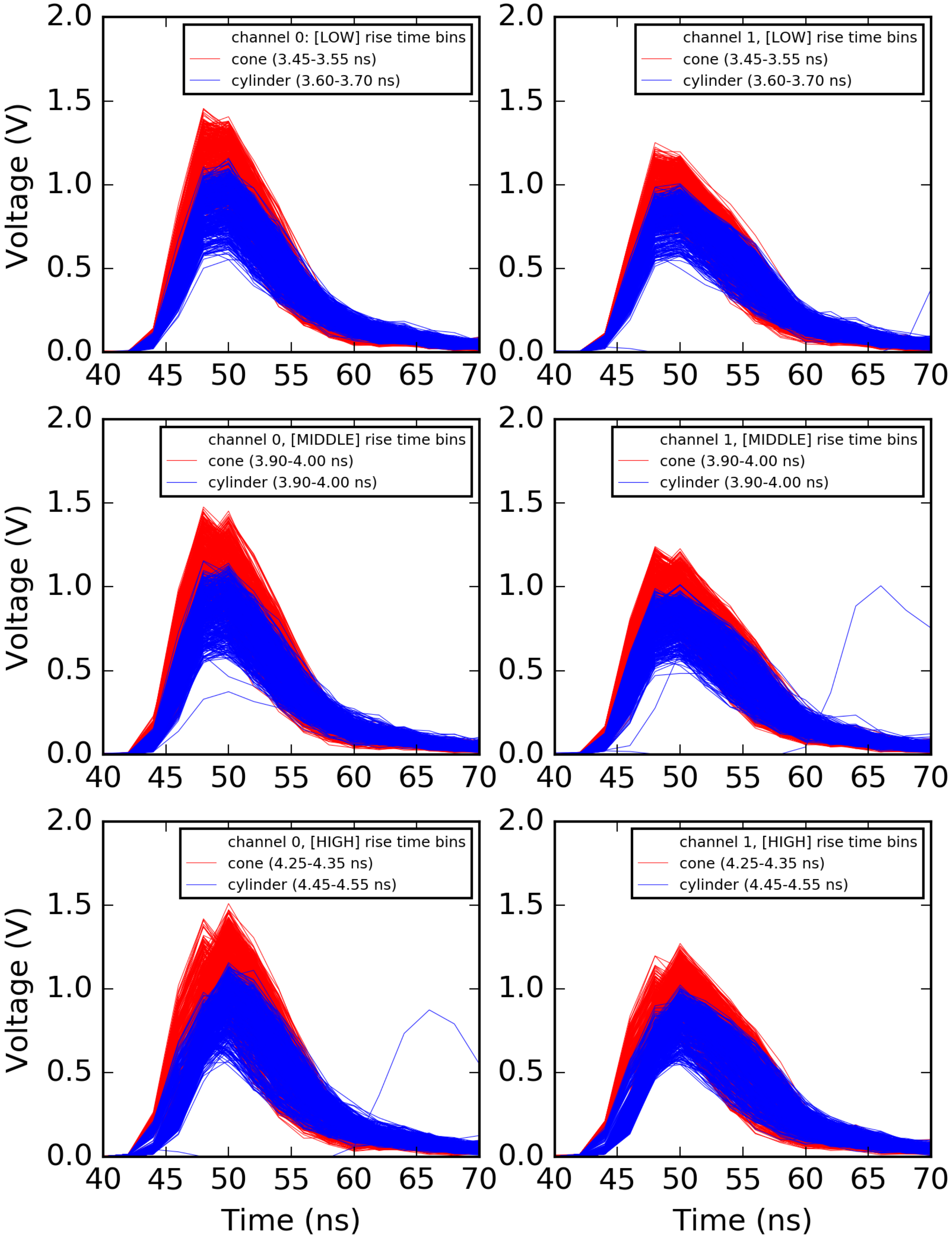


Figure 5.14: Pulses extract from rise-time distributions.

5.4 Discussion

Recall that a scintillator pulse is a convolution of many time-dependent processes including but not limited to the scintillation-emission time, the light-collection process to the PMT, as well as the PMT response. The results presented in this chapter demonstrate that the light-collection process is improved by using a conical organic scintillator instead of a cylinder of the identical base diameter (50 mm), height (50 mm), and material (EJ200). The light-collection process in both geometries were compared using the time resolution, pulse rise time, and pulse shapes. The time resolution of both geometries were obtained through a cross-correlation experiment and assessed at two CFDs (0.2 and 0.5). Both CFDs ensured a useful contribution of correlated counts to the peak area (> 90%) in the cross-correlation distribution. The cone was shown to have improved time resolution over the cylinder at both a CFD of 0.2 and 0.5 by approximately 35% and 21%, respectively. Theses results agree closely with the measured-percent increase (20.6%) reported in previous work that used an analog CFD of 0.4 [31]. The results in Figure 5.14 help to draw two important conclusions related to the scintillator-pulse shape and digitization error discussed below.

Figure 5.14 shows that the cone consistently produced faster rise times than the cylinder. Moreover, pulses from the cone appeared shorter in time and taller in amplitude than pulses from the cylinder. Therefore, the majority of scintillation-light in a cone is collected on a faster time scale than the cylinder, and this result confirms that optical-photon reflections in a cone are less when compared to a cylinder. On the other hand, a cylinder experiences multiple optical-photon reflections and that is why the scintillator-pulse tail appears extended when compared to the pulse tails of a cone. It is interesting to observe how the pulse tail of a cone returns to baseline faster than the cylinder (see Figure 5.14). A decrease in the average number of reflections reduces the average path-length traveled by optical photons in the scintillator prior to their detection. The faster light-collection process in a cone as compared to a cylinder can benefit many systems that aim to minimize propagated uncertainty from time pick-off. The increased presence of statistical information along the leading edge of the pulse for a cone may significantly reduce propagated errors when calculating the time pick-off or its derived quantities.

Figure 5.14 also demonstrates an important trend that occurs for pulses at the starting position of the rising edge in addition to the position of the peak amplitude. Note how both positions appear spread out, even though all pulses were accurately forced to begin at 43 ns. The apparent spread in the starting and peak-amplitude positions are due to digitization error between two adjacent data points at those positions. Digitization error is more obvious at the peak, where it appears as if its location undergoes a random and dichotomous shift. The same digitization error responsible for shifting the peak location also occurs throughout the entire pulse, including the leading edge; it just so happens to be most obvious at the peak.

Recall that the time intervals used to produce the rise-time distributions (see Figure 5.13) were found by subtracting time pick-off values along the leading edge that correspond to 90% and 10% of the peak amplitude. Note how these calculations are strongly dependent on the peak amplitude and two fractional amplitudes (90% and 10%) along the leading edge. Initially, the position of peak appears insignificant since because the two time pick-off fractions are strictly a function of amplitude. However, digitization error can influence the slope of the rising edge, and the effect is particularly more intense near the peak of the pulse; particularly at the point along the leading edge that corresponds to 90% of peak amplitude. Therefore, time pick-off at the 90% mark may greatly impact the rise-time calculations and as a result, the shape of the rise-time distributions. The slope variations near the 90% mark shifts in the peak-location between two adjacent digitized points, resulting from digitization error are believed to contribute to the binary-like rise-time distributions (see Figure 5.13). The binary-like effect is more pronounced in the distribution of the cone than it is for the cylinder. A very likely explanation is the increased presence of statistical information near the front-end of a scintillator pulse produced in a cone. Although a cone collects light faster and contributes additional counts along the leading edge, which helps time pick-off, the limiting factor to the rise-time assessment appears to be the sampling rate of the DAQ.

CHAPTER 6

Experiment: Pulse shape discrimination

6.1 Motivation

The purpose of this work is to measure and compare the PSD performance of conical and cylindrical PSD-capable organic scintillator detectors of equal base diameter, height, and material ($trans$-stilbene). Previous work has shown that conical scintillators, both inorganic and non-PSD capable organic, have greater LCE than their cylindrical counterparts [29, 30], and is further supported by the results shown in Chapter 4. Therefore, it is expected that if LCE is increased in a PSD-capable organic scintillator, then PSD performance will also improve, notably in low-light output events. Recall that gamma ray and fast neutron interactions with an organic scintillator will produce electron and proton recoils, respectively [5]. Due to the unique scintillation properties of stilbene, a greater fraction of delayed light is produced from proton recoils than from electron recoils [1, 2, 4, 5]. Therefore, when stilbene is coupled to a light-sensing device, such as a PMT, the shape of the pulse produced by the PMT will vary according to the particle type. As a result, a PSD technique, such as charge integration [12, 13], can be used to identify the particle based upon the shape of the pulse produced by particle it recoils. However, it is well known that PSD performance degrades at low-light output events (<100 keVee) [14, 15] where gamma-ray and neutron pulses appear similar. To counter the decrease in PSD performance for low-light output events, LCE can be increased by modifying the scintillator geometry to reduce the likelihood of optical-photon losses due to multiple reflections at the scintillator-reflector boundary. An increase in particle-discrimination at low-light output events can benefit many applied systems in nuclear-security research where gamma-ray and neutron identification is required [11].

6.2 Methods

6.2.1 Equipment

Two stilbene crystals of different shapes (a cylinder and a cone) produced by Inrad Optics with identical base diameter (50 mm) and height (50 mm) were used in the experiments described in this chapter. Both stilbene crystals are shown in Figure 6.1. For each crystal, all surfaces remained unpolished with the exception of the base, where a thin (1 mm) optical window made from fused silica was permanently adhered. Optical grease (EJ500) was used to couple the scintillator base to the center of PMT-127 that was operated at a gain of -1700 V for all experiments discussed in this chapter; the same PMT (Photonis XP4512B) and gain were used for some experiments described in Chapter 4. It has been shown in literature that the spatial distribution of optical photons entering the PMT from a cone and cylinder are different [29]. Thus, a large-diameter PMT was chosen so that stilbene samples were small in size compared to the PMT diameter in order to minimize photocathode non-uniformity effects. Experiments in Chapter 4 and literature [19, 29] have characterized the photocathode response for the Photonis XP4512B as a function of position and showed that the response at its center, where the stilbene samples were placed, was approximately uniform. The same conical and cylindrical specular-reflector encasements (see Figure 3.3) used in Chapter 4 were also used in the experiments described in this chapter.

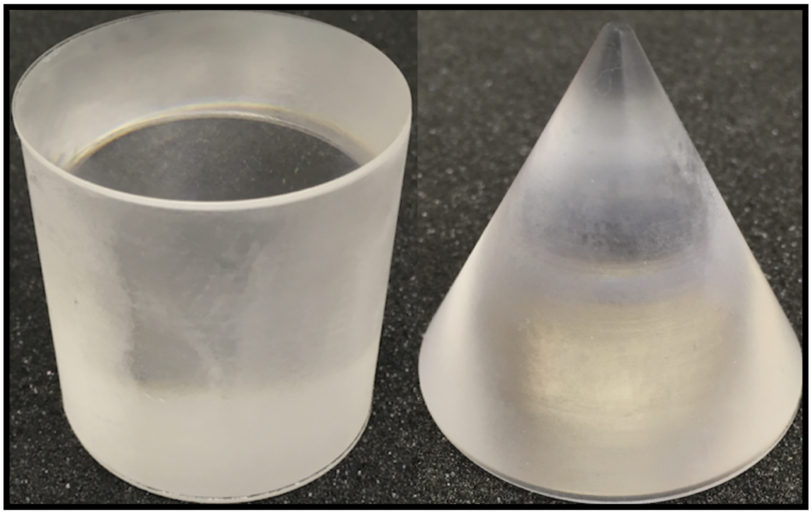


Figure 6.1: Cylindrical and conical *trans*-stilbene from Inrad Optics. Both geometries have an equal base diameter and height of 50 mm.

6.2.2 Compton coincidence

Two calibration experiments were conducted for each stilbene sample using cesium-137. Compton coincidence [43–45] was used to calibrate the pulse-integral distribution and obtain the energy resolution at the Compton edge (478 keV). The experimental setup is identical to the arrangement described in Section 3.3.1. The energy resolution was found by applying a Gaussian fit to the correlated count distribution (Compton-edge events) in stilbene obtained with Compton coincidence. The statistical uncertainty in the calibration point and the propagated uncertainty in the energy resolution were obtained from fit parameters using 95% confidence intervals.

6.2.3 Charge integration

Two PSD experiments were conducted for each stilbene sample using californium-252. The optimal start time for the tail integral for both scintillators between 25 keVee and 400 keVee was found to be 18 ns and 20 ns after the peak of the pulse for the cone and cylinder, respectively. The shorter start time in the cone demonstrates that the light collection process occurs on a shorter time scale in the cone vs. the cylinder, which supports the claim made earlier, namely that optical-photons reflect on average less in the cone than in the cylinder; thus, their detection occurs faster. The end points for the tail and total integrals for both scintillators were fixed to the maximum-window length in the data-acquisition system (200 ns). This end point roughly corresponds to 180 ns after the peak of the pulse. It is worth emphasizing that the data-acquisition window was sufficiently large to capture the entire pulse for both geometries. Lastly, a FOM was calculated as a function of four distinct light output regions: Region 1 (25-100 keVee), Region 2 (100-200 keVee), Region 3 (200-300 keVee), and Region 4 (300-400 keVee). These regions were chosen to determine how the quality of neutron/gamma-ray discrimination changed with light output. Lastly, the uncertainty in the FOM was propagated from the fit-function parameters with 95% confidence intervals.

6.3 Results

6.3.1 Calibration and energy resolution

Table 6.1 compares the calibration point in units of pulse integral, the energy resolution, and the total number of correlated counts at 478 keVee. Figure 6.2 shows the cesium-137 calibrated pulse-integral spectra for both geometries. Their respective correlated-count distributions as identified through Compton coincidence were also included in the same figure. These results show that the performance of the cone is better than that of the cylinder, as demonstrated by improvement in energy resolution (10.05%) and light output (18.01%). These improvements are attributed to a higher LCE of the cone. The decrease in detection efficiency (69%) of the cone relative to the cylinder is expected given the (67%) smaller volume of the cone. The loss in efficiency is slightly higher than the loss in volume due to a solid angle affect that is explained in previous work where monte carlo simulations of a conical and cylindrical organic scintillator calculated the efficiency loss to be approximately 70% [29]. The difference in the measured and simulated detection efficiency loss can be attributed to slight alterations to the crystal volumes during manufacturing.

Table 6.1: A comparison of cylindrical and conical stilbene performance metrics at 478 keVee.

Scintillator geometry	Energy resolution (%) at 478 keVee	Calibration point (V*ns) at 478 keVee	Correlated counts at 478 keVee
Cylinder	9.64 ± 0.06	19.83 ± 0.01	278,421 ± 1,034
Cone	8.76 ± 0.09	23.39 ± 0.01	86,228 ± 576
Performance change of cone relative to cylinder	+ 10.05 ± 0.12 %	+ 17.95 ± 0.01 %	- 69.03 ± 0.53 %

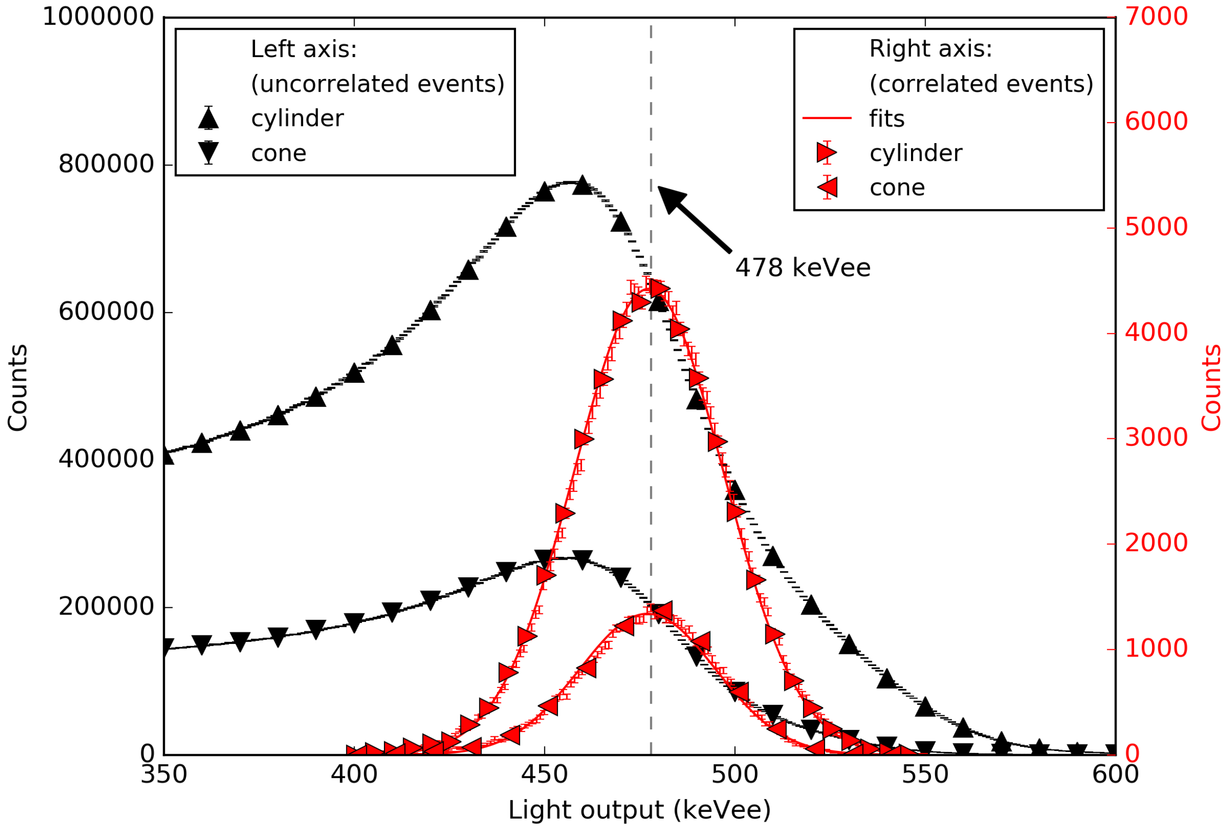


Figure 6.2: Uncorrelated (left axis) and correlated (right axis) pulse-integral distributions calibrated to light output for both geometries.

6.3.2 PSD heat maps

PSD heat maps for the cylinder and cone are shown together in Figure 6.3, found by plotting the S parameter (see Section 3.3.3) as a function of light output (i.e., calibrated pulse integral). Heat maps were processed using the same number of pulses (2.7e7). Figure 6.3 shows fewer events in the region between the neutron and gamma-ray bands for the cone versus the cylinder, providing visual evidence that the cone exhibits better separation between the gamma-ray and neutron clusters. It is also interesting to note that although the same digitizer voltage-amplitude threshold (0.1 V) was used for both PSD measurements, the effective light-output threshold for the cone is lower than the cylinder. The lower effective threshold of the cone is attributed to its greater LCE in contrast to the cylinder, allowing for more low-energy events to exceed the energy threshold. It is worth noting the diagonal component to the threshold in Figure 6.3. This is not a feature of the data but rather an effect from the pulse-integral technique caused by the change in neutron-pulse shape as a function of increasing energy deposition [14].

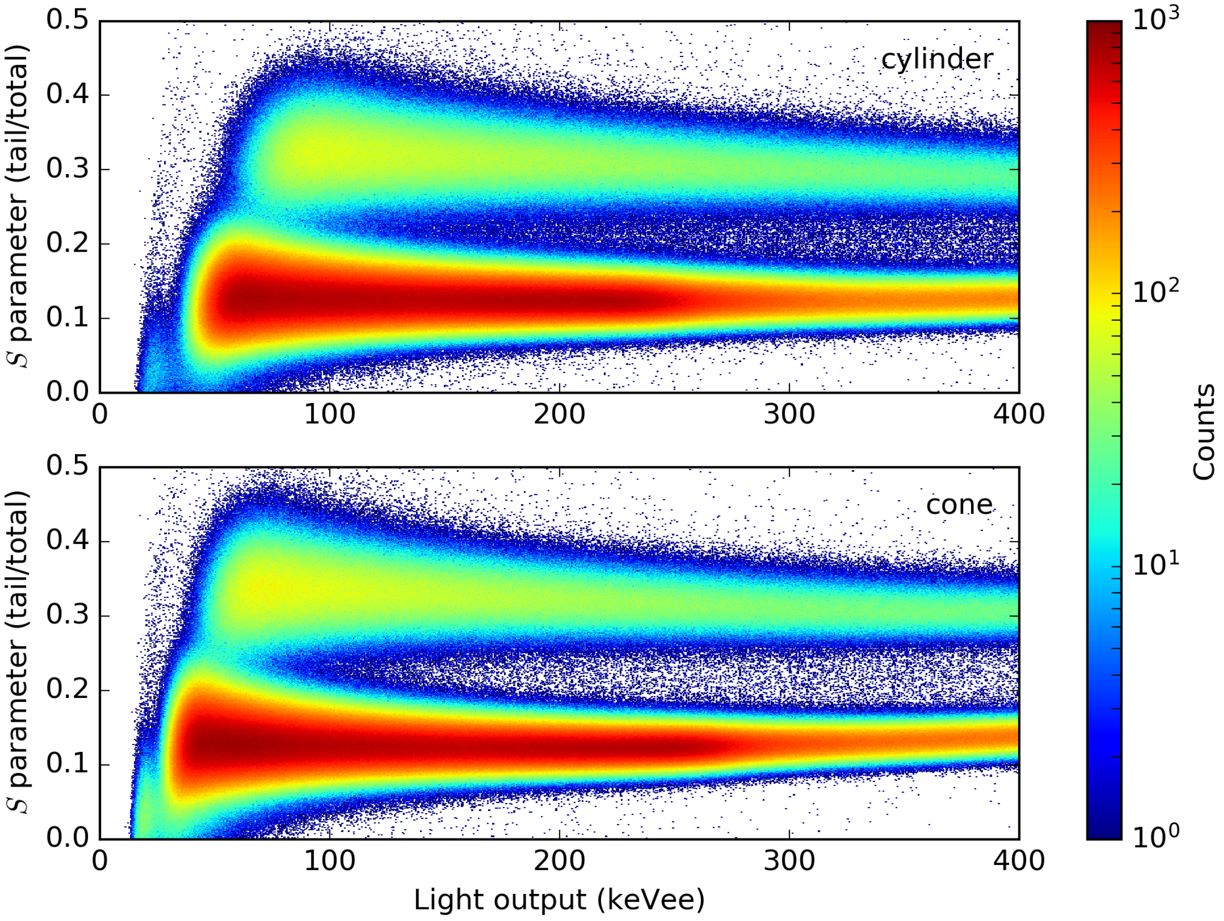


Figure 6.3: PSD heat maps for the cylinder (top) and cone (bottom) were processed using the same number of pulses (2.7e7) with identical bin widths in the x-axis (0.4 keVee), y-axis (6e-4), and color bar (1 count) in log scale.

6.3.3 FOM distributions

Heat maps were subdivided into four light-output regions, from which S-parameters were extracted for each region and used to calculate a FOM. The S distributions for the cylinder and cone are shown together in Figure 6.4. Table 6.2 lists the four light output regions and compares their FOM values for both geometries. Although the percent improvement in energy resolution offered by the cone appears low (10.05%), the impact it has on PSD to accurately resolve neutrons at low-light output events is more significant, as demonstrated by the PSD performance improvements in Table 6.2, particularly the result from 25-100 keVee (22.58%).

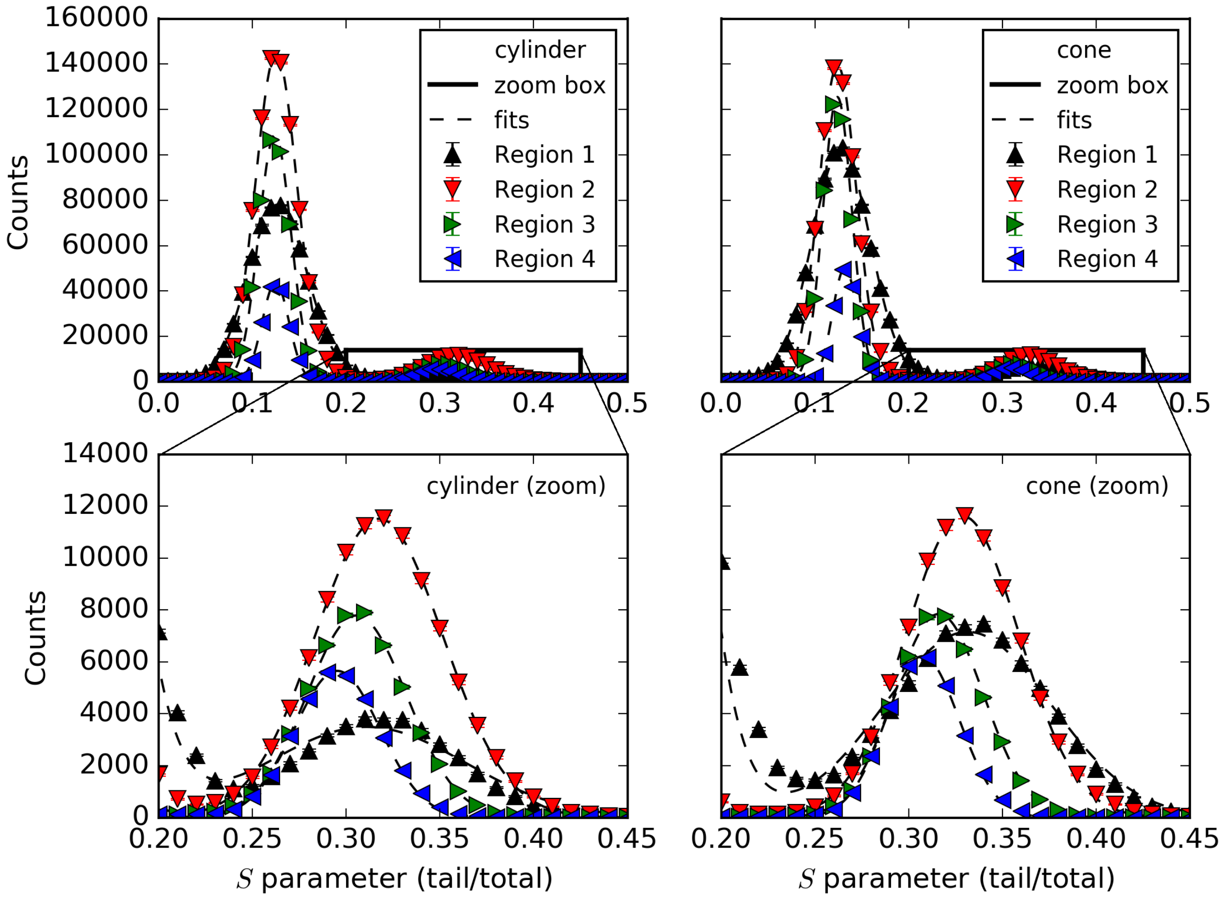


Figure 6.4: S distributions for the cylinder (left) and cone (right). The expanded plots show neutron counts (bottom).

Table 6.2: Performance change for californium-252 measurements across four light-output regions.

		Region 1	Region 2	Region 3	Region 4
Light output range (keVee)		25 to 100	100 to 200	200 to 300	300 to 400
FOM	Cylinder	0.93 ± 0.03	1.45 ± 0.03	1.76 ± 0.04	1.95 ± 0.02
	Cone	1.14 ± 0.03	1.73 ± 0.04	2.09 ± 0.04	2.27 ± 0.02
Performance change of cone relative to cylinder		+ 22.58 ± 0.94 %	+ 19.31 ± 0.60 %	+ 18.75 ± 0.56 %	+ 16.41 ± 0.22 %

It is well known that energy resolution worsens for lower energy events. In contrast to a cylinder, conical stilbene increases LCE throughout the entire pulse [29], thus increasing the amount of statistical information available for the charge integration calculation and producing a substantial improvement in PSD performance. It is also worth mentioning that the percent gain (10.05%) in energy resolution observed using conical stilbene as compared to a cylinder is less than the percent gain (16.20%) in energy resolution reported for EJ200 in Chapter 4. This discrepancy is caused by two reasons: (1) stilbene has a greater light output (approx. 14,000 photons/MeV) [10] in comparison to EJ200 (approx. 10,000 photons/MeV) and (2) the spectral-response of the PMT matches the wavelength emission of stilbene slightly better than the emission of EJ200. Both factors combine to diminish the gain in energy resolution due to the increased detection efficiency per photon in stilbene as compared to EJ200.

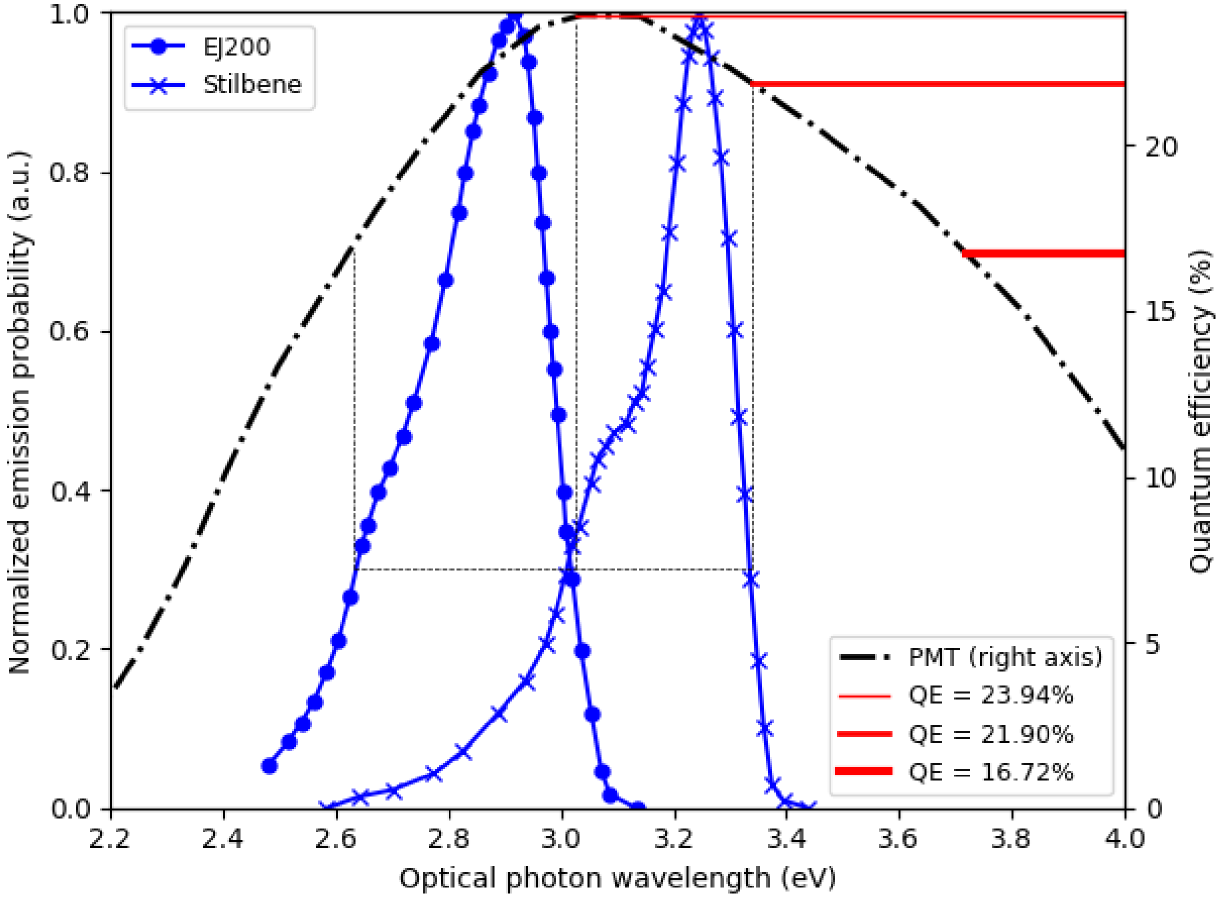


Figure 6.5: Spectral-emission comparison of *trans*-stilbene and EJ200 coupled to PMT-127.

6.4 Discussion

It was interesting to note how the gain in energy resolution (16.20%) of a cone made of EJ200 was greater than the gain (10.05%) of a cone made of *trans*-stilbene. Both cones had the same dimensions, used the same reflector, and were coupled to the same PMT (Photonis XP4512B), operated at a gain of -1700 V. Even more intriguing was the fact that the gain in LCE (17.60%) for conical EJ200 was similar to stilbene (17.95%). These results suggest that any gain in LCE achieved through geometric alterations is largely dependent on the TRC. However, the gain in energy resolution achieved through geometric alterations is shown to be strongly dependent on the scintillator material, particularly its light output and its distribution of wavelengths.

A material with a brighter light output and better spectral match to a PMT will not show as much of an improvement in energy resolution than a material with a dimmer light output with a spectral match that is less ideal (see Figure 6.5). Regardless, even if the gain in energy resolution achieved using a cone appeared marginal at a Compton edge (e.g., 478 keVee), it is still very significant for events that produce a low light-output (e.g., < 100 keVee), where the energy resolution rapidly deteriorates from decreased counting statistics (see Figure 7.5).

The FOM results presented in Chapter 6 demonstrate this trend as a function of light output and its significance at low-light output. Recall that the PSD performance of conical and cylindrical stilbene were compared as a function of four distinct light-output regions. In the lowest region of light output from 25-100 keVee, the cone achieved a vast improvement in particle discrimination, as demonstrated by a 22.6% higher FOM. The gain in FOM progressively reduced as the light output for radiation interactions increased: 19.31% (100-200 keVee) to 18.75% (200-300 keVee) to 16.41% (303-400 keVee). The trend shown by the decrease in PSD performance gained by the cone as a function of increased light output is analogous to the decrease in energy resolution gained at 478 keVee by stilbene than EJ200.

CHAPTER 7

Simulation: Optical-photon transport

7.1 Motivation

Geant4, which stands for **GE**ometry **AN**d **T**racking, is an open-source simulation toolkit developed at CERN, written in the C++ programming language, and used to model the transport and interaction of particles in matter [33], including optical photons. Geant4 can track secondary radiation from ionization particles, for example electron recoils from gamma rays that Compton scatter in a material. This capability is similar to other Monte carlo based radiation transport codes, for example MCNPX. What makes Geant4 especially useful for this work is its unique capability to produce and track optical photons from scintillation, which can provide further insight into how scintillator material, geometry, reflector conditions, and other optical parameters that affect LCE and by extension detector performance. Moreover, Geant4 can be compiled with a computer program called QT to support simulation visualization. Therefore, a model of conical and cylindrical EJ200 coupled to a PMT was eagerly developed in Geant4 to study and visualize the light-collection process in both geometries. To trust the quantities derived from the model, validation experiments were conducted to compare against simulated data. Model validation is further discussed in Section . Through validation, the model demonstrated that it could accurately represent a calibrated light-output spectrum from optical-photon transport. Note that the version of Geant4 used for all simulation results presented in this work was 10.3 (patch 3).

7.2 Geant4

7.2.1 Overview

Although the user is free to modify the source code at their own risk, no modifications were made to the version of Geant4 used in this work (10.3 patch 3). When the Geant4 source code is compiled and installed, numerous functions and libraries become available to the user for them to call on in a C++ program of their own. However, any Geant4 program must meet certain basic requirements in code structure and function calls, in order to run the most basic task. The simplest program consists of five source files. One of the five essential files is the main source file, which calls on four other required source files (enumerated below), where each source file has an associated header file. It is conventional to call on Geant4 libraries and functions in a header file for use in its source file.

1. Detector Construction

   The associated header file calls on functions that may include, for example, a system of units, geometry definitions, geometry placement in three dimensional space, and material definitions, just to name a few. Within the source file, the user provides specific values for the functions they called on in the header file. These values could include, for example, the chemical composition and density of a cylinder placed at some position in space.

2. Physics List

   The associated header file can call on functions that may include pre-packaged physics libraries, for example, the standard electromagnetic (EM) physics library. Within the source file, the user can add or subtract features from pre-packaged libraries. For example, the EM library does not include physical processes associated with the generation and transport of optical photons (e.g., Scintillation, Cerenkov, optical-photon elastic scattering, absorption, etc). The user can include the particle definitions and associated processes for optical photons in the source file.

3. Action Initialization

   The associated header file calls on the action-class headers defined by the user. An action class is a list of tasks to perform during specific stages of the Monte-carlo process. The stage hierarchy is as followed: Run, Event, Track, Step, and Step Point. A Primary-Generation Action class controls the Run, and is required for the most basic program. A Run is a collection of events produced by a source particle. All action classes are compiled through the Action Initialization.

4. Primary Generator Action

   The associated header file calls on functions that define how source particles are sampled for a single Run by prescribing initial conditions. Within the source file, the user can define the type of source particle, its energy, position, and direction.

A Geant4 model was developed to simulate the transport of optical photons from scintillation events inside a conical and cylindrical non-PSD capable plastic scintillator coupled to a PMT. Figure 7.1 shows a flow chart of the code structure and important input parameters for each source file used to construct the model. A visual representation of the model is found in Section 7.2.2. A description of the material properties for each component in the model, the optical boundaries, and potential sources of error, are found in Sections 7.2.3, 7.2.4, and 7.2.6, respectively. The validation methodology and a summary of the simulations performed in this work are found in Sections 7.3.1 and 7.3.2, respectively.

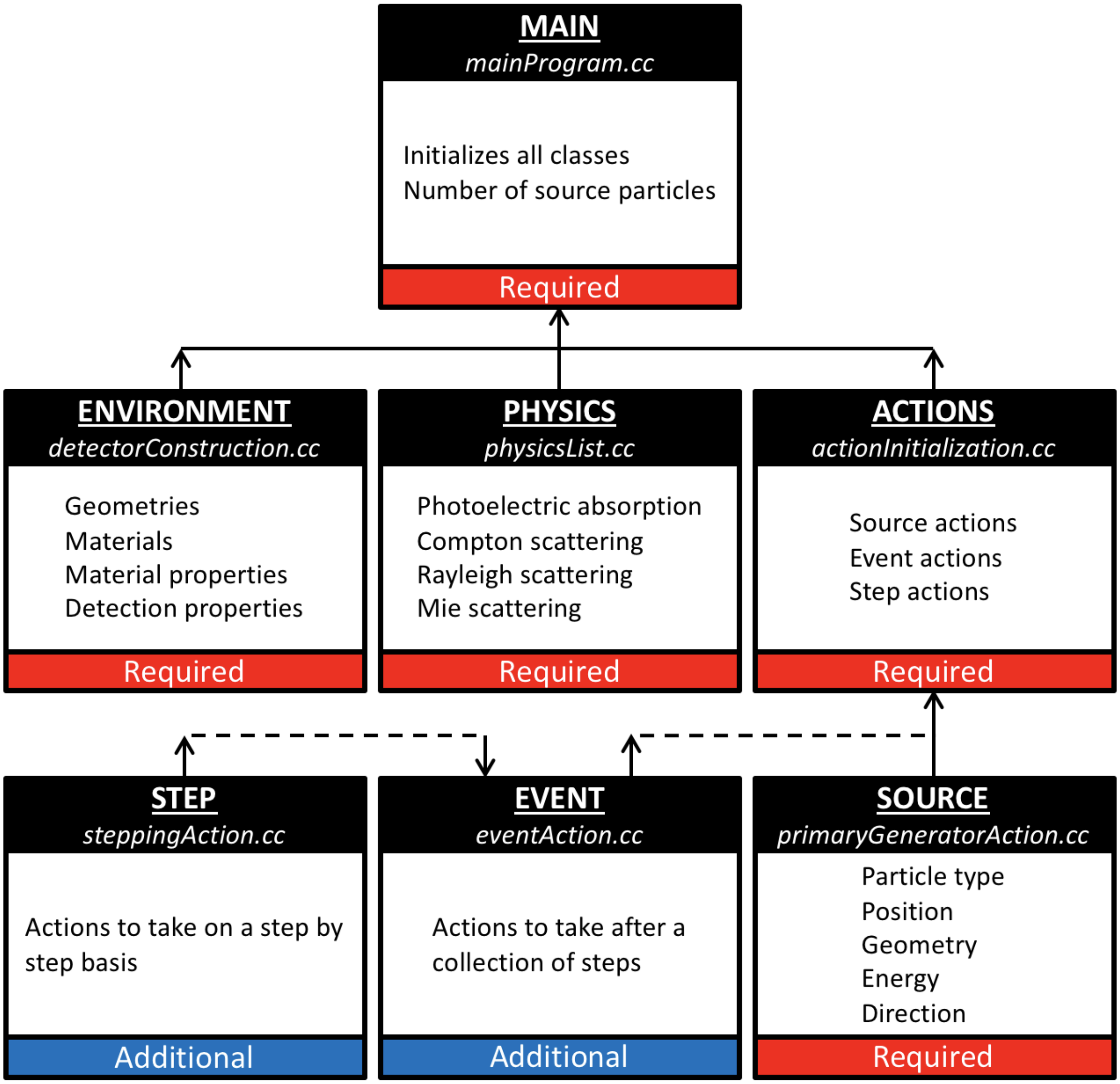


Figure 7.1: Simplified representation of a Geant4 code structure. Note that although Rayleigh and Mie scattering are included in the Physics box, these processes were not considered in the model.

7.2.2 Model

Figure 7.2 shows a cross-section of a cylindrical organic scintillator, the optical-coupling grease, the optical window of the PMT, and the photocathode, which was assumed to be an arbitrarily thin circular disc of negligible material. Physical processes that can occur with optical photons are shown overlaid in the same figure and also in Figure 7.3.

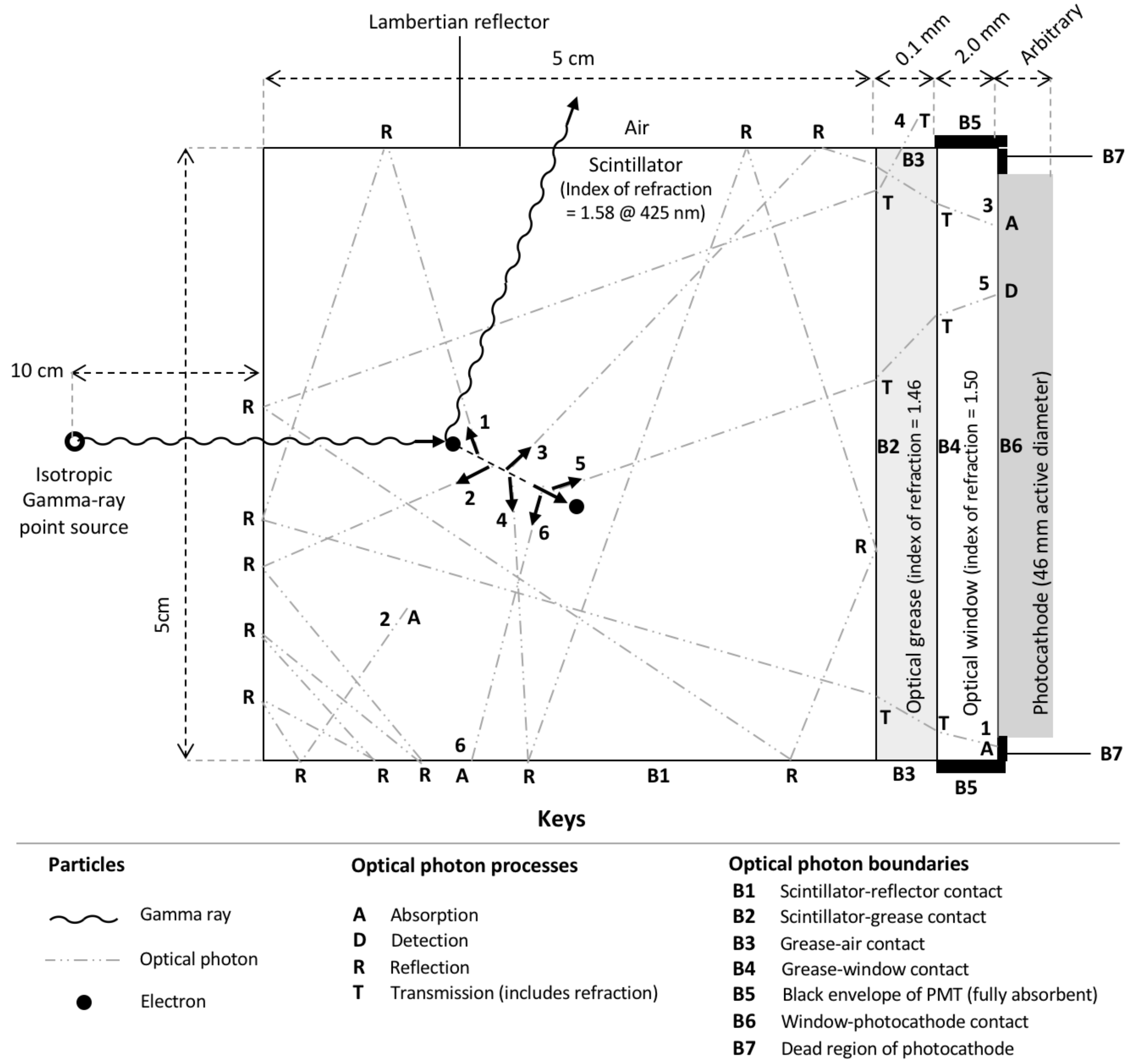

Lambertian reflector
5 cm
0.1 mm
2.0 mm
Arbitrary
Air
Scintillator
(Index of refraction
= 1.58 @ 425 nm)
10 cm
Isotropic
Gamma-ray
point source
5cm
R
T
A
D
1
2
3
4
5
6
B1
B2
B3
B4
B5
B6
B7
Optical grease (index of refraction = 1.46
Optical window (index of refraction = 1.50
Photocathode (46 mm active diameter)
Keys
Particles
Gamma ray
Optical photon
Electron
Optical photon processes
A Absorption
D Detection
R Reflection
T Transmission (includes refraction)
Optical photon boundaries
B1 Scintillator-reflector contact
B2 Scintillator-grease contact
B3 Grease-air contact
B4 Grease-window contact
B5 Black envelope of PMT (fully absorbent)
B6 Window-photocathode contact
B7 Dead region of photocathode

Figure 7.2: Simplified representation of the Geant4 model.

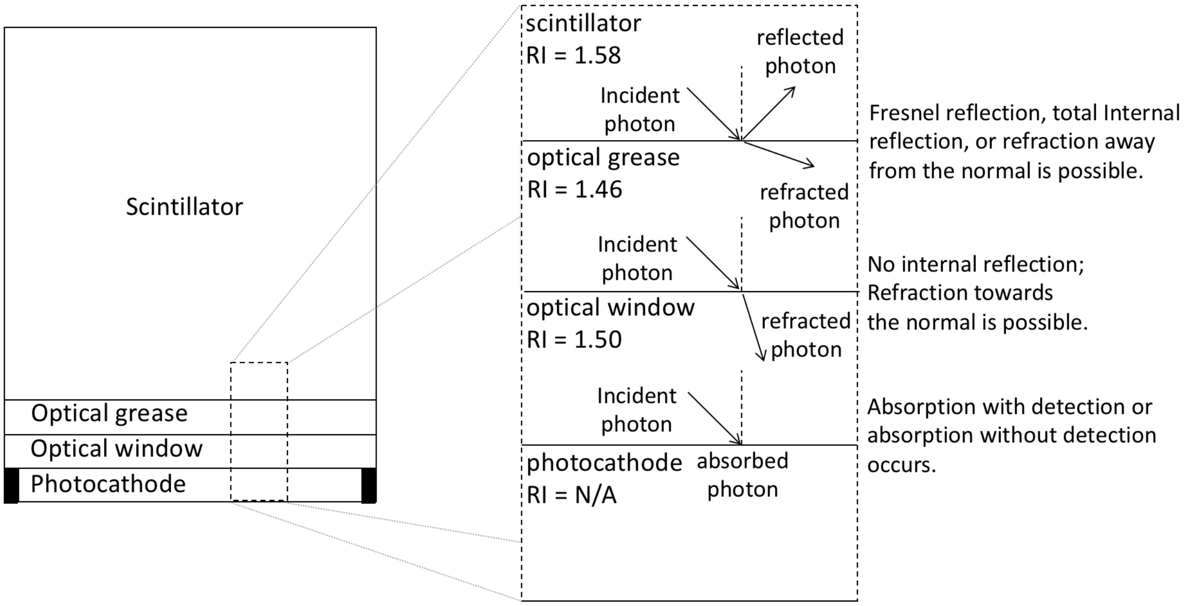


Figure 7.3: Simplified representation of optical-photon processes at the PMT window.

Each simulation began by sampling the birth of a gamma ray from a cesium-137 point source. If the gamma ray for that event produced a Compton interaction in the scintillator, then the Compton electron was free to cause further ionization and excitation. Along the excitation track of the electron, optical photons are equally distributed and randomly sampled from a distribution of possible wavelengths. These photons are tracked within the scintillator volume, where the majority will reflect at the scintillator-reflector boundary multiple times before one of three likely events occur: (1) reflector absorption, (2) photocathode absorption without detection, or (3) photocathode absorption with detection, where the detection probability varies as a function of optical-photon wavelength. The model also considers optical-photon attenuation within the scintillator volume.

7.2.3 Materials

The material properties of the scintillator were taken from the data sheet of EJ200. EJ200 has a density of 1.023 g/cm^3, a light output of 10,000 photons/MeV, and a peak scintillation emission of 425 nm. At peak emission, its RI and optical-photon attenuation length is 1.58 and 380 cm, respectively. The material properties of the optical-coupling grease (EJ550) were taken from data sheets provided by Eljen. EJ550 has a density of 1.06 g/cm^3 and a RI of 1.46. The grease was evenly distributed between the scintillator base and PMT window. Two photocathode options were included where both were assumed to be perfectly uniform, arbitrarily thin, and of negligible material. The photocathode detection efficiency was modeled from the QE curve of either PMT-127 (Photonis XP4512B) or PMT-50 (ETL 9214B). The QE at 425 nm of PMT-50 and PMT-127 is approximately 23% and 25%, respectively. Both PMTs used the same optical window material (borosilicate glass), thickness (2 mm), outer diameter (50 mm), and constant RI (1.50) for all scintillation wavelengths (400 - 500 nm). The EJ200 spectral emission curve and both PMT-response curves are shown below in Figure 7.4.

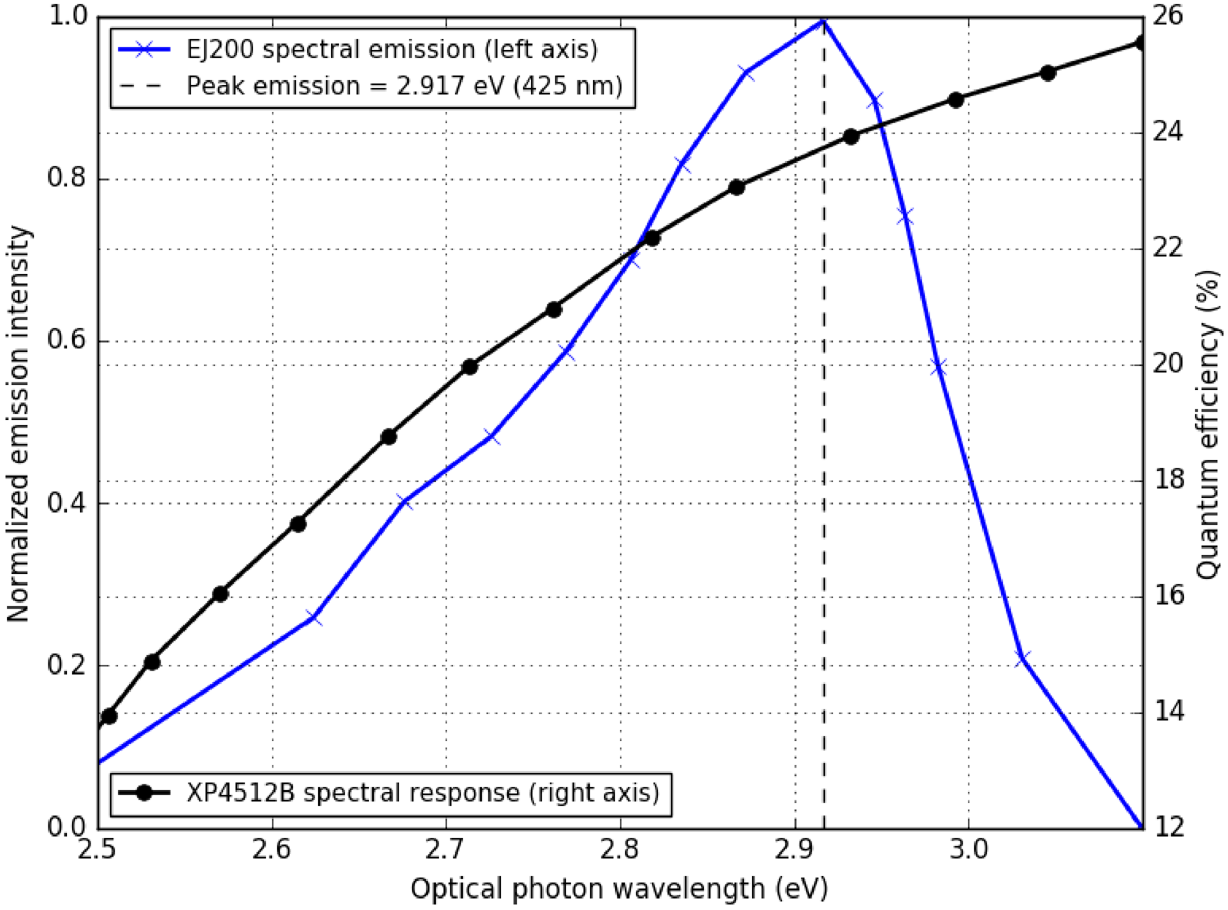


Figure 7.4: The spectral emission function of EJ200 and response function of PMT-127 were modeled in Geant4. The energy domain was set by the range of possible optical-photon wavelength emissions from EJ200.

7.2.4 Optical boundaries

The optical-photon model selected for all Geant4 simulations performed in this work was the UNIFIED model [47]. The Geant4 surface types [42] (i.e., boundaries between different volumes) were classified as either a “Dielectric Metal" (DM) or a “Dielectric Dielectric” (DD). In short, a DM boundary only allows that boundary to absorb, detect, or reflect a photon. A DD boundary on the other hand, allows for Fresnel reflection, refraction, and total internal reflection to occur. The surface finish of the scintillator was designated as either “Ground" (G) or “Polished Back Painted" (PBP). A G-finish on a DM boundary simulates a perfectly diffuse reflector. A PBP-finish on a DD boundary can simulate a specular reflector with a thin air gap between the reflector and scintillator surface.

At the scintillator-reflector boundary, two reflector configurations were considered: (1) a perfectly Lambertian (i.e., diffuse) reflector and (2) a perfectly specular reflector with a thin air gap between the reflector and scintillator surface. Although transmission is more likely than absorption at the reflector (if reflection does not occur) it was decided to absorb the particle to eliminate unnecessary computational run-time by tracking optical photons that leaked through the reflector, which can no longer contribute to detections on the photocathode surface used to model a PMT response. The same reflection and absorption probabilities were used for both reflector configurations (95.25% and 4.75%, respectively). Since the transmission probability was assumed to be zero, the absorption probability was inferred from the reflection probability, given that all three probabilities sum to 100%. The determination of the reflection probability is further discussed in Section 7.3.2.

7.2.5 Geant4 iterations

The purpose of simulation numbers 1 and 2 was to compare their agreement with the experimental increase in energy resolution, calibration position, and decrease in detection efficiency of the ground cone relative to the ground cylinder when using PMT-127. The iteration process began by searching for the optimal reflectivity value, which influenced the separation between the calibration positions of the cylinder and cone. Theoretically, a reflectivity value of 100% in Geant4 should produce the same optical photon collection distribution, regardless of scintillator geometry, especially in this particular comparison where the attenuation length is very long. Once the optimal reflectivity value was found, the resolution scale was iterated to converge with the measured percent increase in energy resolution.

7.2.6 Potential sources of error

The Rayleigh and Mie scattering lengths could not be found for EJ200 and therefore these processes were omitted from the Geant4 physics list. Generally speaking, Rayleigh and Mie scattering have low interaction probabilities in most organic scintillators. Additionally, the coefficients for the semi-empirical “Birks" model [2] were not found for EJ200 and thus this light-output reduction feature was removed from the physics list as well. The Birks’ option in Geant4 is used to reduce the light output at low-energy depositions in order to model quenching and recombination effects. If the option is turned off, then Geant4 assumes that the light output is perfectly linear with energy, which is a fair assumption for energy depositions above 100 keV for EJ200. Recall that the Compton edge energy used to assess changes in the energy resolution as a function of geometry is at 478 keV.

Additionally, even though the range of wavelength emission from EJ200 is tightly bound to the blue and violet portion of the visible spectrum, the energy dependence of optical-photon dispersion, responsible for the refraction angle, is expected to influence the RI. If a large range of optical-photon wavelengths are tracked, for example all photons in the visible-spectrum, setting a constant RI for any material would not accurately represent refraction or total internal reflection. Since the wavelength emission of EJ200 is tightly bound to violet and blue light, a constant RI is a more reasonable estimation, although it is possible that this uniform condition may create an unintentional source of error. To more appropriately model refraction, a Cauchy equation should be used to generate a distribution of wavelength-dependent RI values. The Cauchy equation provides an empirical relationship between the wavelength of visible light to its RI in a material. A

Sellmier-dispersion function could also be used to generate the same distribution. The Sellmier function has been shown in optics research to provide more a more accurate representation of RI near ultraviolet light. Since the coefficients required in both functions were not found for any of the exact material used in the model, a constant RI was set.

7.3 Methods

7.3.1 Validation techniques

The two methods used to validate the model are listed below. Both validation techniques compared simulated and measured data of EJ200. A detailed discussion of all Geant4 simulations performed in this work is found in Section 7.3.2.

1. Energy-resolution gain

   Simulations of a cylinder and cone were compared with the experimental increase in energy resolution, calibration position, and decrease in detection efficiency of the ground cone relative to the ground cylinder, discussed in Chapter 4. The photocathode detection efficiency was modeled from the QE curve of PMT-127 (peak QE $\approx 23\%$ at 425 nm).

2. Energy-resolution function

   Simulations of a cylinder were compared with the experimental light-output spectrum of cesium-137. The simulated spectrum was broadened using an energy-resolution function (see Figure 7.5). The function was obtained for the same simulated detector configuration through Compton coincidence at three angles (180°, 90°, and 45°). The peak photocathode detection efficiency was modeled off the QE curve of PMT-50 (peak QE $\approx 25\%$ at 425 nm).

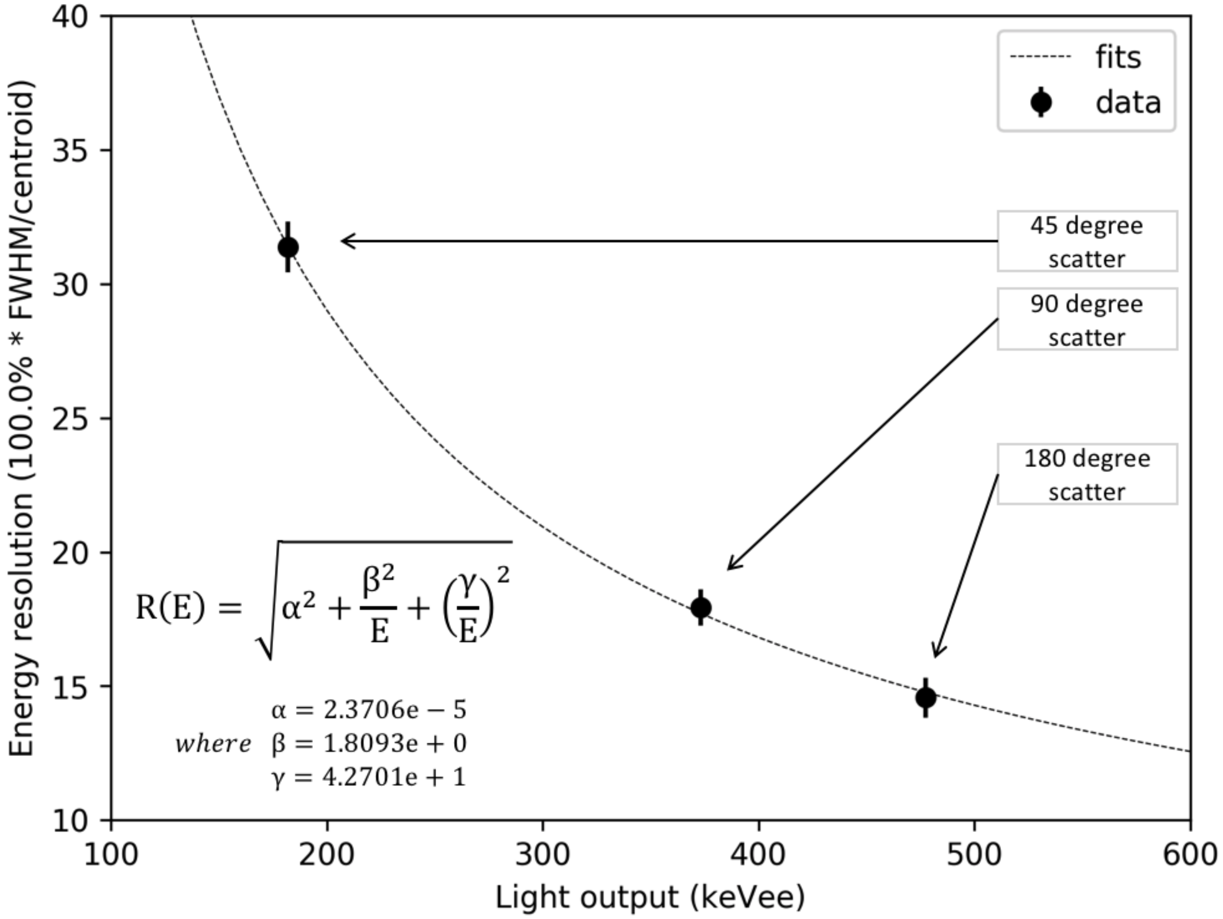


Figure 7.5: Energy-resolution function for EJ200 (cylinder) obtained using Compton coincidence at three different angles to isolate three different energies along the Compton continuum.

7.3.2 Simulations: EJ200

A total of eighteen simulations were performed. Table 7.1 illustrates the first six simulations and Figure 7.6 in Section 7.3.3 describes the twelve other simulations. For each of the first six simulations, a cesium-137 point source sampled in three fashions: isotropically and linearly in two unique pathways. One line source was directed towards the center of the scintillator (referred to as “down axis") and the other was directed towards the outer edge of the photocathode (referred to as “off-axis"). The direction of these line sources were specifically chosen to highlight the root cause of spectrum smearing in a highly polished cone.

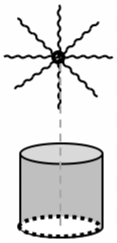
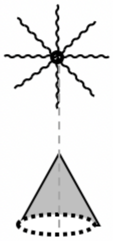
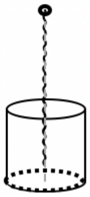
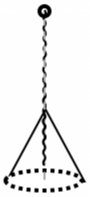
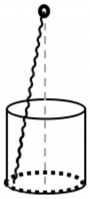
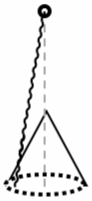
	Isotropic source		Mono-directional			
			Down axis: <0, 0, -1>		Off axis: <0, -0.1366, -0.9906>	
Simulation No.	1	2	3	4	5	6
Source particles	5e7		1e5			
Visual representation of simulation geometry and source (15 cm away from base)						
Reflector properties						
Optical model	Unified					
Optical type	Dielectric-Metal		Dielectric-Dielectric			
Optical finish	Ground		Polished Back Painted			
Sigma Alpha (roughness)	0					
Specular lobe probability	0		1.0			
Specular spike probability			0			
Backscatter probability						
Lambertian probability	1.0					
Refraction	No		Yes (internal reflection possible)			
Reflection probability	0.9525					
Absorption probability	0.0475					
Resolution scale	0					
Summary	Perfectly diffuse reflector w/o air gap between reflector and scintillator		Perfectly polished scintillator surface w/ air gap between specular reflector and scintillator			
Purpose	Compare w/ energy-resolution and validation (cylinder only) experiment		Assist w/ spectrum smearing discussion			

Table 7.1: A simplified description of all Geant4 simulations performed.

For each gamma-ray event that caused scintillation, the following tally information was available: (1) the energy deposited per source particle (2) the number of optical photons detected per source particle, (3) the location of each Compton-electron birth, (4) the location of each optical photon detection on the photocathode, (5) the number of reflections (NOR) an optical photon underwent prior to detection, and lastly (6) the time of arrival (TOA) of a detected optical photon to the photocathode surface. To simulate the correlated-count distribution shown in Chapter 4, a narrow gate width of 1 keV below and including the Compton-edge energy was used to flag back-scatter events and record the number of optical photons detected.

The purpose of simulation numbers 1 and 2 was to compare their agreement with the experimental increase in energy resolution, calibration position, and decrease in detection efficiency of the ground cone relative to the ground cylinder, when using PMT-127. Both simulations assumed a perfectly diffuse reflector with no air gap. An iterative process was used to search for the optimal reflectivity value that matched closest to the gain in calibration position (17.6%) achieved by the cone when compared to the cylinder on PMT-127 (see Chapter 4). The reflectivity influences the separation between the calibration positions of the cylinder and cone. Theoretically, a reflectivity value of 100% in Geant4 should produce the same optical photon collection distribution, regardless of scintillator geometry, especially in this particular comparison where the attenuation length is very long (380 cm) relative to the scintillator size. Once the optimal reflectivity value was found, another iterative process was used to determine the optimal resolution scale that best agreed with the measured percent increase in energy resolution. A reflectivity value of 95.25% and resolution scale of 0 were found to yield the closest agreement to the experimental results.

The purpose of simulation numbers 3 through 6 was to highlight a contributing factor to spectrum smearing in a highly polished cone, and to demonstrate why the same effect does not appear in a highly polished cylinder. In the case of a highly polished cone with a specular reflector, the presence of an air gap causes a sharp change in the index of refraction, creating the potential for optical photons to undergo total internal reflection. The closer a scintillation events occurs to the polished surface, the greater the likelihood for optical photons to reflect below the critical angle, and thus total internal reflection occurs. To visualize this effect in Geant4, two lines sources were simulated: one directed towards the center (“down axis"), and the other towards the outer edge of the photocathode (“off-axis").

7.3.3 Simulations: EJ276

The twelve remaining simulations consisted of an efficiency trade-off study where various shapes (see Figure 7.6) were compared of a new PSD-capable organic scintillator (EJ276). According to the manufacturer (Eljen), EJ276 has a lower light yield (8,600 photons/MeV) than EJ200 (10,000 photons/MeV), but otherwise has almost identical physical characteristics to EJ200. The plastic material of EJ276 means that it could be machined into any shape, where for example *trans*-stilbene could not (see Section 6). The added PSD capability of EJ276 made this trade-off efficiency study particularly intriguing in case a shape was found with excellent performance characteristics. The goal was to find a shape that performed similar to the cone (shape A) in terms of LCE, while demonstrating better gamma-ray detection efficiency, similar to that of the standard cylinder (shape K). Shapes B through J were tapered cones, each with identical height (50 mm) and base diameter (50 mm) to the cone (A) and standard cylinder (K), and unique front-plane diameters. It was theorized that a tapered cone might provide an ideal trade-off between the achievable LCE and gamma-ray detection efficiency of the cone (A) and standard cylinder (K), respectively. Also, it is conceivable that the most ideal shape for maximizing LCE is a cylinder with a small height relative to its base diameter, which could minimize optical-photon reflections and transit times to the PMT. Therefore, a truncated cylinder (shape L) was also included in this study. The height (16.7 mm) of the truncated cylinder was found by constraining its base diameter and volume to match the cone (A), which is used as a benchmark for improved LCE relative to the standard cylinder (K). Each shape had identical reflectors to simulations 1 and 2 (see Section 7.3.2). Each simulation sampled 2.5 billion particles from a isotropic cesium-137 (661.67 keV) point source placed 10 cm from the front plane of each shape (see Figure 7.6). The LCE, energy resolution, and gamma-ray detection efficiency, were calculated for each shape, and the results are discussed in Section 7.4.5.

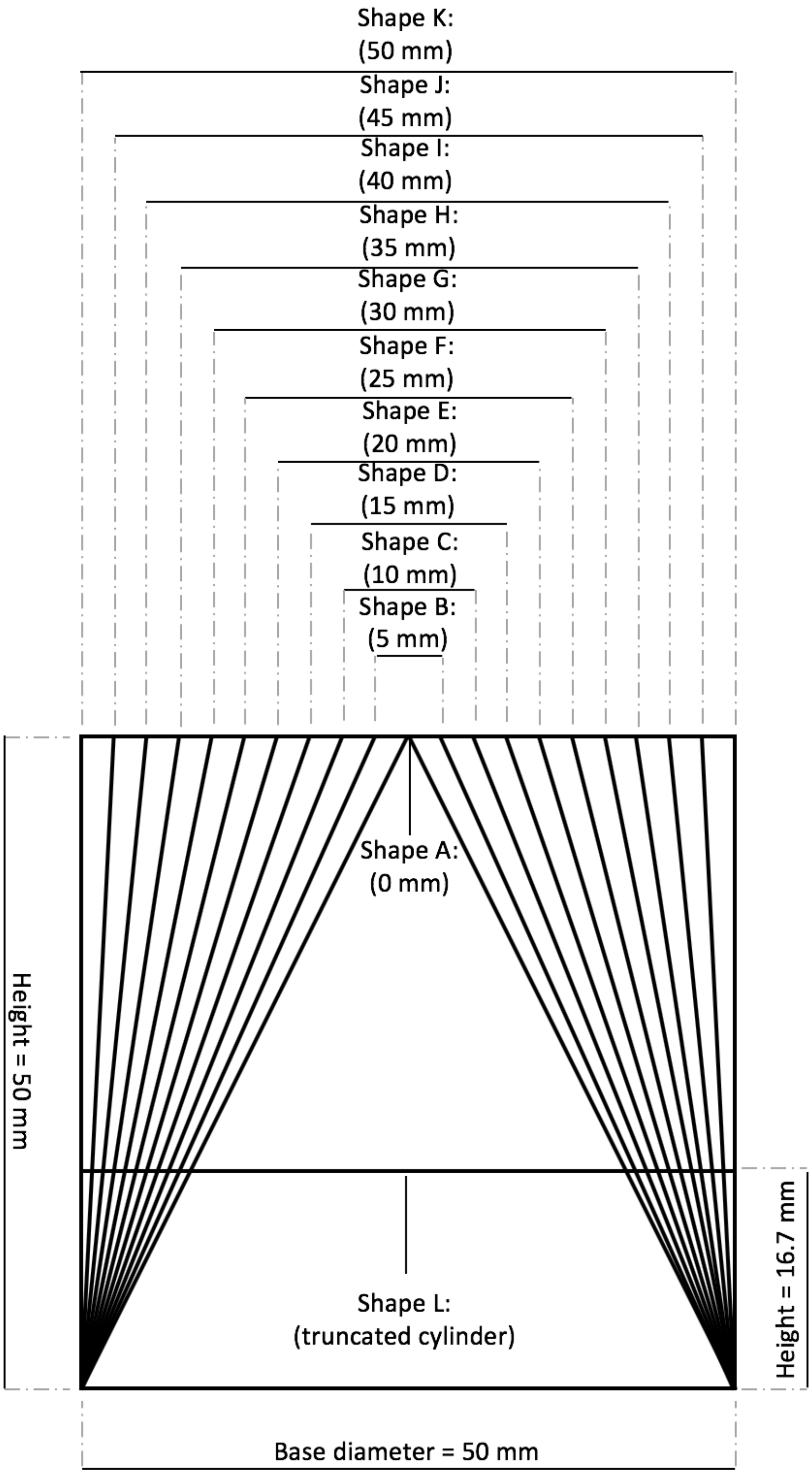


Figure 7.6: A cross-section diagram of the various geometries used in the efficiency-trade off study. The light-collection and detection efficiencies were compared for each geometry.

7.4 Results

7.4.1 Comparison with experiment

The standard and correlated cesium-137 spectra for both geometries are shown in Figures 7.7 and 7.8, respectively. Table 7.2 shows good agreement between simulation and experiment.

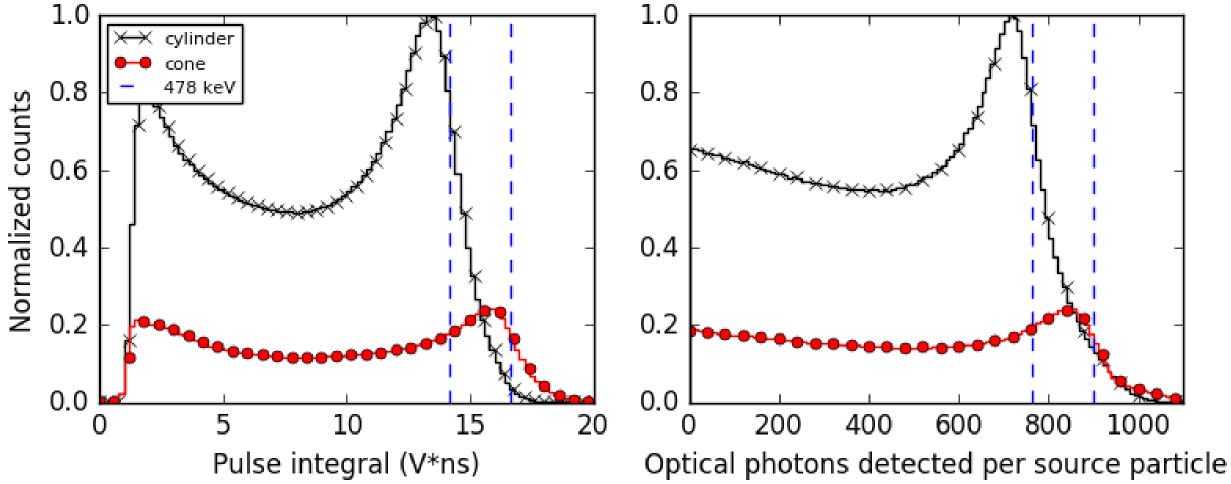


Figure 7.7: Standard spectrum comparison of experiment (left) and simulation (right).

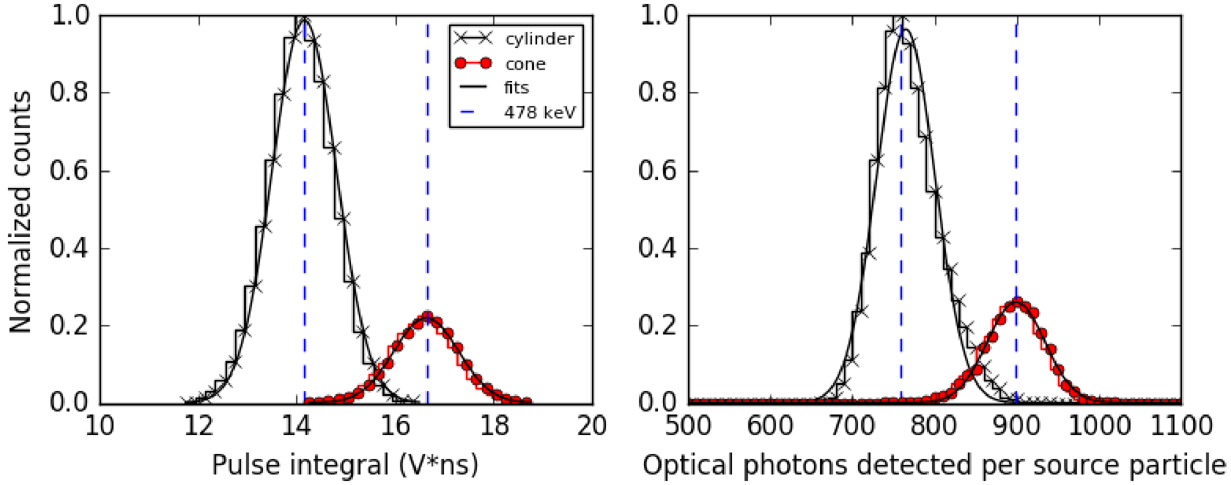
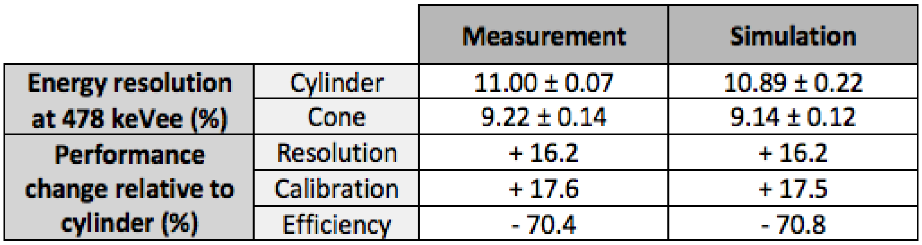


Figure 7.8: Correlated spectrum comparison of experiment (left) and simulation (right).

Table 7.2: Comparison of experiment and simulation results.



		Measurement	Simulation
Energy resolution at 478 keVee (%)	Cylinder	11.00 ± 0.07	10.89 ± 0.22
	Cone	9.22 ± 0.14	9.14 ± 0.12
Performance change relative to cylinder (%)	Resolution	+ 16.2	+ 16.2
	Calibration	+ 17.6	+ 17.5
	Efficiency	- 70.4	- 70.8

The simulated energy resolution uncertainty was assessed using the fit parameters (95% confidence) from the correlated distribution. Note that the overestimation of counts above the simulated Compton edge in Figure 7.7 is a result of omitting the Birks' light-output reduction feature for low-energy depositions (< 100 keV), where the light yield is non-linearly reduced [2]. Figure 7.9 shows four cesium-137 spectra calibrated to light output. Three were simulated and one was measured. The measured spectrum was taken with cylindrical EJ200 coupled to PMT-50, calibrated with Compton coincidence to 478 keVee, and normalized to source strength (97.61 uCi). The source was placed 10 cm above the front face of the cylinder. The model considered the same detector and source configuration as the measured spectrum and the three simulated curves were derived from the same simulation. The simulated curves shown in red, green, and blue represent the energy-deposition spectrum and both the un-broadened and broadened optical-photon detection spectra. Unlike the simulated spectrum of EJ200 couped to PMT-127, shown in Figure 7.7, the un-broadened spectrum (originally in units of optical photons detected) shown in Figure 7.9 was calibrated to light output (keVee). After calibration, the energy-resolution function (see Figure 7.5) was used to broaden the spectrum, shown in blue, and agrees well with the measured curve. The deviation at low-light output events (< 200 keVee) is expected due to room return and electronic noise not considered in the simulation.

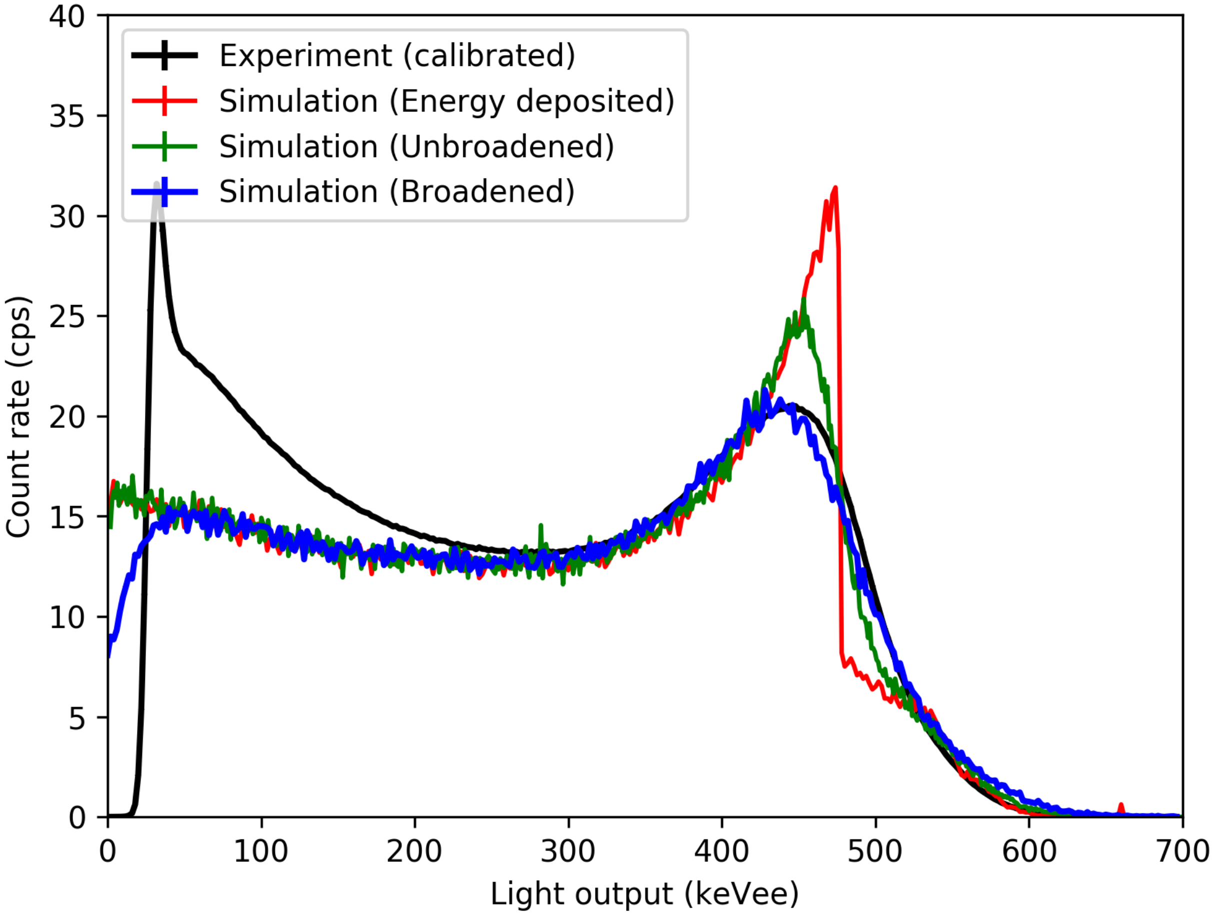


Figure 7.9: Energy-resolution function for EJ200 (cylinder) obtained using Compton coincidence.

Although a great deal of effort was placed in the construction of the model to ensure reproducibility and as much accuracy as possible, it is important to remember that certain assumptions were made that may not be completely accurate (see Section 7.2.6). However, both validation techniques demonstrate that the model can produce simulated spectra that agrees well with experimental data and trends. Therefore, the simulated results that follow can be interpreted with cautious confidence.

7.4.2 TOA and NOR distributions

Figure 7.10 shows that optical photons reflect on average less in the cone than in the cylinder prior to detection. Consequently, the average time it takes optical photons to get detected is faster in the cone than in the cylinder (see Figure 7.11), and this result is supported by a previous timing-resolution experiment [31].

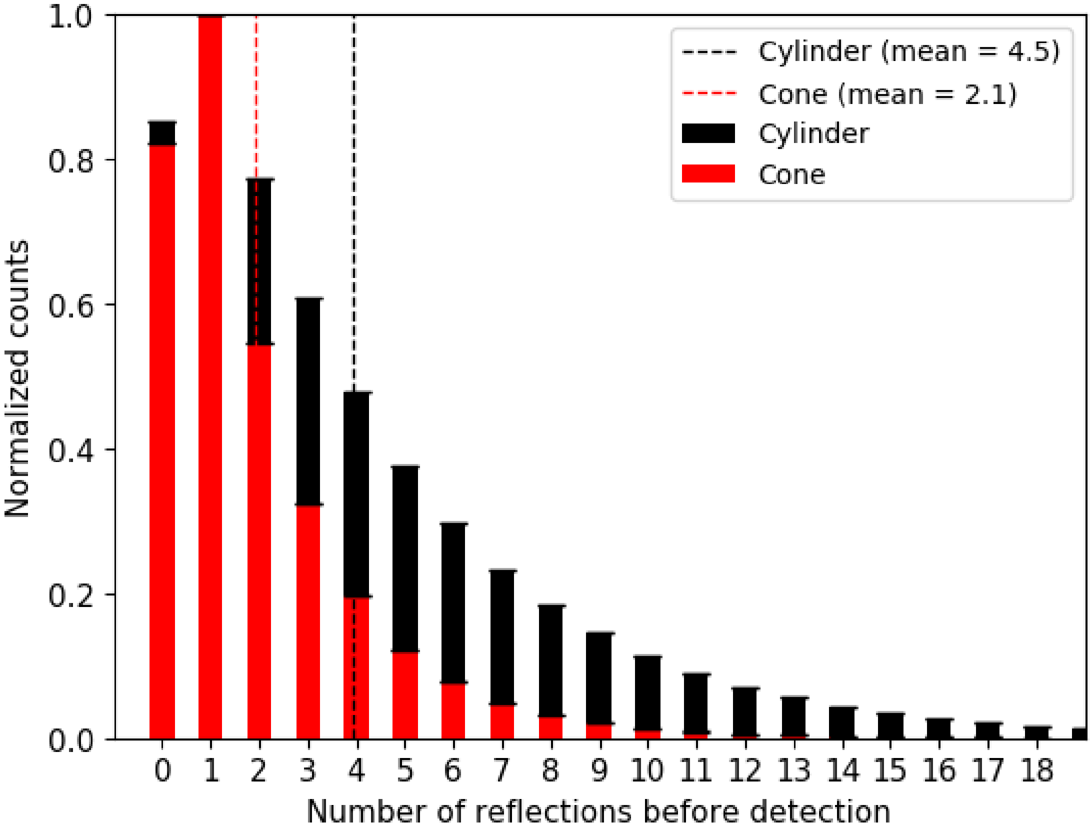


Figure 7.10: A comparison of the NOR distribution for the cone and standard cylinder.

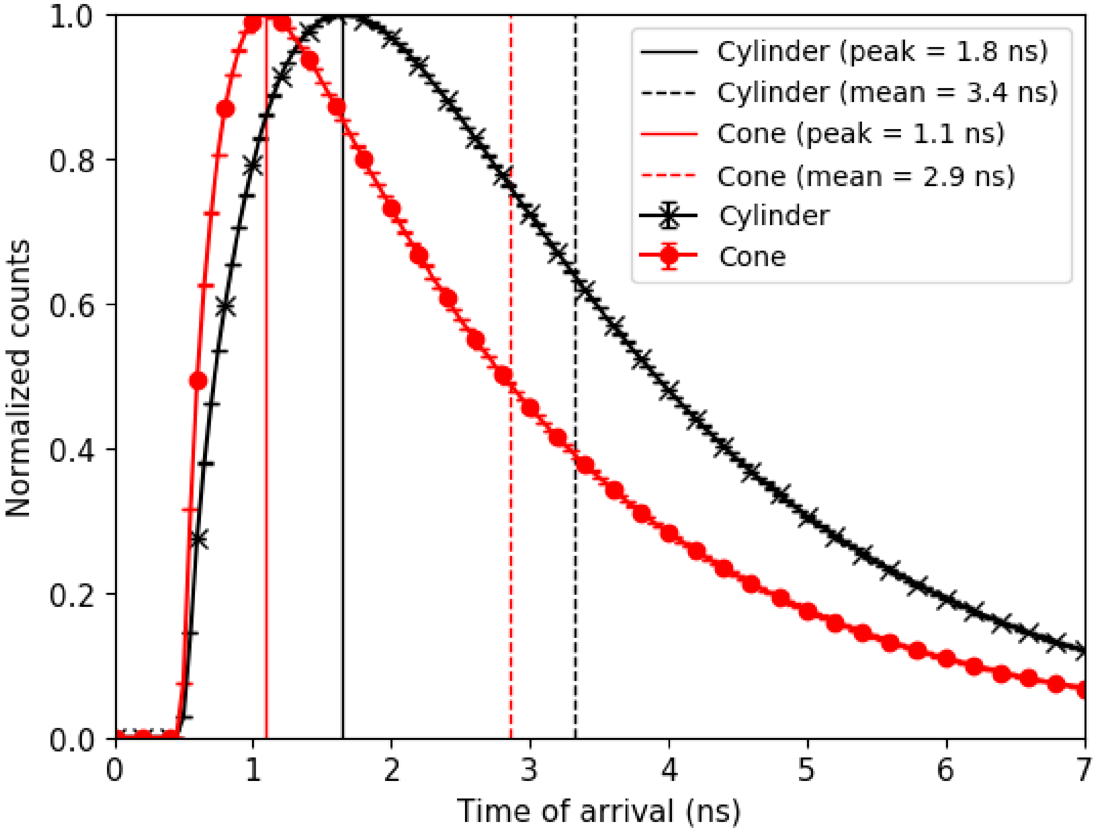


Figure 7.11: A comparison of the TOA distribution for the cone and standard cylinder. Note that the clock for the TOA calculation begins after the birth of the gamma ray.

7.4.3 Compton-scatter heat maps

Figure 7.12 shows a simulated heat map of Compton-scatter events integrated along the y-axis for both geometries. The cylinder experiences the greatest intensity of Compton-scatter events near its front face. The cone, on the other hand, offers little volume in its tip for gamma rays to interact with. Therefore, the majority of interactions occur near its base.

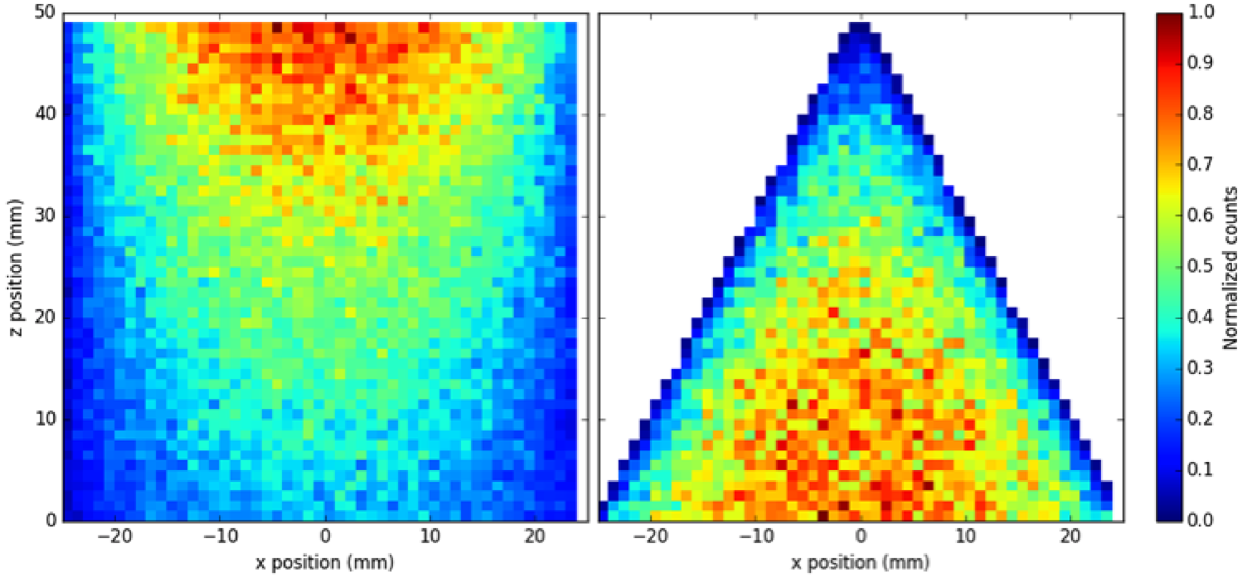
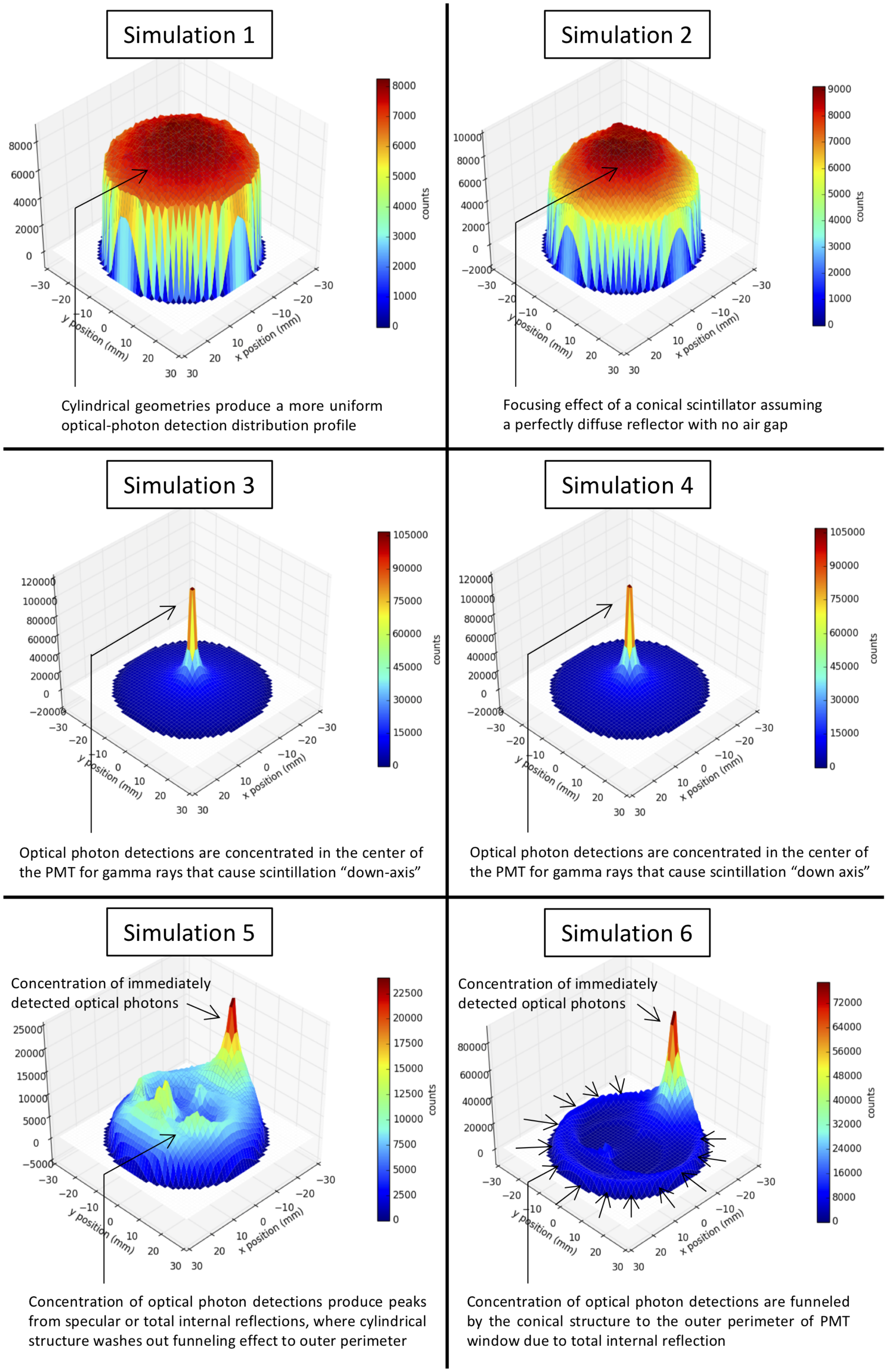


Figure 7.12: Volume-integrated heat map (along y-axis) of Compton-scatter locations in the cylinder and cone. Each map was normalized to its maximum value. Bin widths are 1 mm.

7.4.4 Photocathode hit distributions

Shown in Figure 7.13 are the heat map tallies of optical-photon detections on the photocathode. Note how the diffuse cylinder (simulation 1) has more uniform detections than the diffuse cone (simulation 2). Recall that in simulations 3 and 4, a highly polished surface and specular reflector was modeled, in addition to an air gap between the two surfaces. Simulations 3 and 4 show how down-axis gamma-rays produce the same localized collection distribution. However, off-axis gamma-rays with the same direction vector, produce a different collection distribution in the cone than in the cylinder. The heat map of the cylinder (simulation 5) shows a peak of immediately-detected optical photons, and so does the cone (simulation 6). However, the peak amplitude (25000 counts) in simulation 5 is less than in simulation 6 (80000 counts), due to the location of Compton-scatter interactions (see Figure 7.12). Furthermore, the distribution of counts in simulation 5 are more evenly spread out than simulation 6, which shows a ring of counts detected along the perimeter from total internal reflection. This process is referred to in this work as optical-photon funneling and can contribute to spectrum smearing (see Figure 4.1) for a polished cone.



Simulation 1

Cylindrical geometries produce a more uniform optical-photon detection distribution profile

Simulation 2

Focusing effect of a conical scintillator assuming a perfectly diffuse reflector with no air gap

Simulation 3

Optical photon detections are concentrated in the center of the PMT for gamma rays that cause scintillation “down-axis”

Simulation 4

Optical photon detections are concentrated in the center of the PMT for gamma rays that cause scintillation “down axis”

Simulation 5

Concentration of immediately detected optical photons

Concentration of optical photon detections produce peaks from specular or total internal reflections, where cylindrical structure washes out funneling effect to outer perimeter

Simulation 6

Concentration of immediately detected optical photons

Concentration of optical photon detections are funneled by the conical structure to the outer perimeter of PMT window due to total internal reflection

Figure 7.13: Photocathode hit distributions.

If a polished cone is coupled to a PMT that suffers from strong photocathode non-uniformity, then unreflected optical photons produced from scintillation events near the photocathode will create a detector response that is influenced by the localized QE. Crucially, if the surface is highly polished, scintillation events that occur near the edge can force optical photons to “funnel" and collect along the outer perimeter of the photocathode where QE suffers (see Figure 4.7), and spectrum smearing is produced. Optical-photon funneling is depicted in Figure 7.14. Both images show Geant4 visualizations of the same light-output event generated near a corner of a polished cone. In the image to the left, a specular reflector with an air gap is present, while the scintillator to the right does not have a reflector. The yellow dots on the surface of the base represent optical-photon detections on the photocathode. It is clear that whether the reflector is on or off, the predominant mode of surface reflection for these scintillation events is total internal reflection.

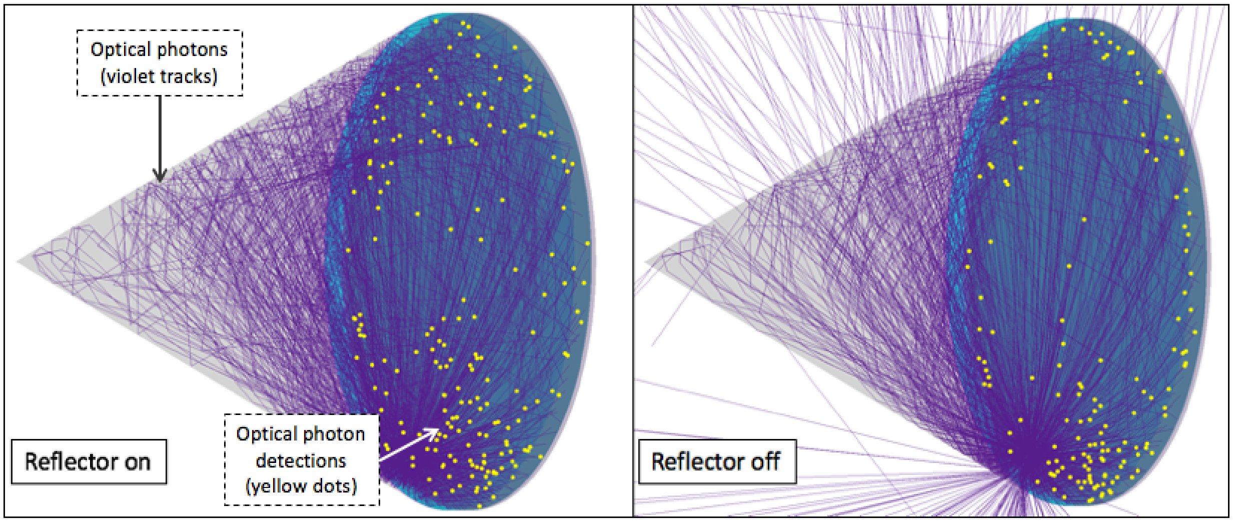


Figure 7.14: A Geant4 visualization of the same scintillation event occurring near the edge of a highly polished cone with a specular reflector.

7.4.5 Efficiency trade-off study

Figure 7.15 shows the energy-deposition and optical-photon detection spectra for all shapes (A-L), along with their respective calibration positions. Recall that the calibration position signifies the location of the Compton edge (477.33 keV). Detection events at the Compton edge where the total-energy deposited fell within a narrow energy gate (476.34 keV and 477.34 keV), were correlated with optical-photon detections. The calibration position for each shape was found by calculating the centroid of their correlated optical-photon detection spectra, shown in Figure 7.16. The number of gamma-ray detections, the calibration position, along with the FWHM and energy resolution of each correlated optical-photon detection spectra were compared for all shapes in Figure 7.17.

It is important to emphasize that the calibration position is an estimate of the average number of optical-photons detections per detection event at the Compton edge. Therefore, the LCE can be calculated by dividing the calibration point by the expected light yield for a particular energy deposition. For example, if EJ276 emits 8,600 optical photons per MeV-deposited, then the expected light yield at the Compton-edge is approximately 4,105 optical photons. As shown in Figure 7.17, the calibration point of the standard cylinder (K) is approximately 660, therefore its LCE is 16% at 477.33 keV. Figure 7.17 also shows that both the cone (A) and truncated cylinder (L) have approximately the same calibration point (780 optical photons) and therefore the same LCE (19% at 477.33 keV).

It was particularly exciting to observe that the loss in gamma-ray detection efficiency relative to the standard cylinder (K) for the truncated cylinder (L) was only 52% as opposed to 70% for the cone (A), even though both the cone (A) and truncated cylinder (L) reported the same LCE. The loss in gamma-ray detection efficiency was found by calculating the percent decrease in correlated optical-photon detections (see Figure 7.16) for the cone (A) and truncated cylinder (L) relative to the standard cylinder (K). Using the same methodology, the truncated cylinder (L) reported a 41% increase in gamma-ray detection efficiency relative to the cone (A).

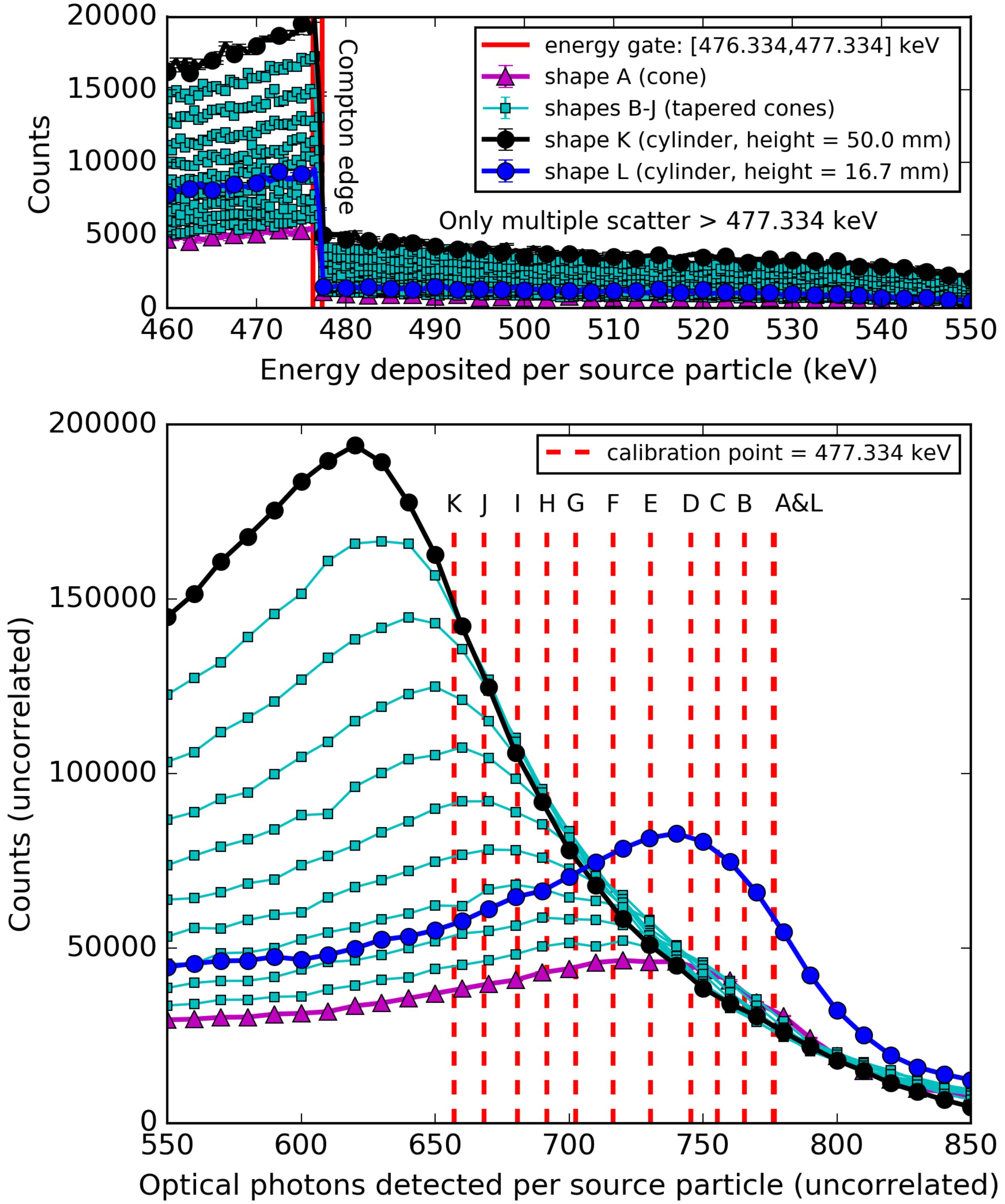


Figure 7.15: Energy deposition (top) and optical-photon detection (bottom) spectra for all shapes. The dotted red line in the bottom plot shows the location of the Compton edge (i.e., calibration point) for each shape.

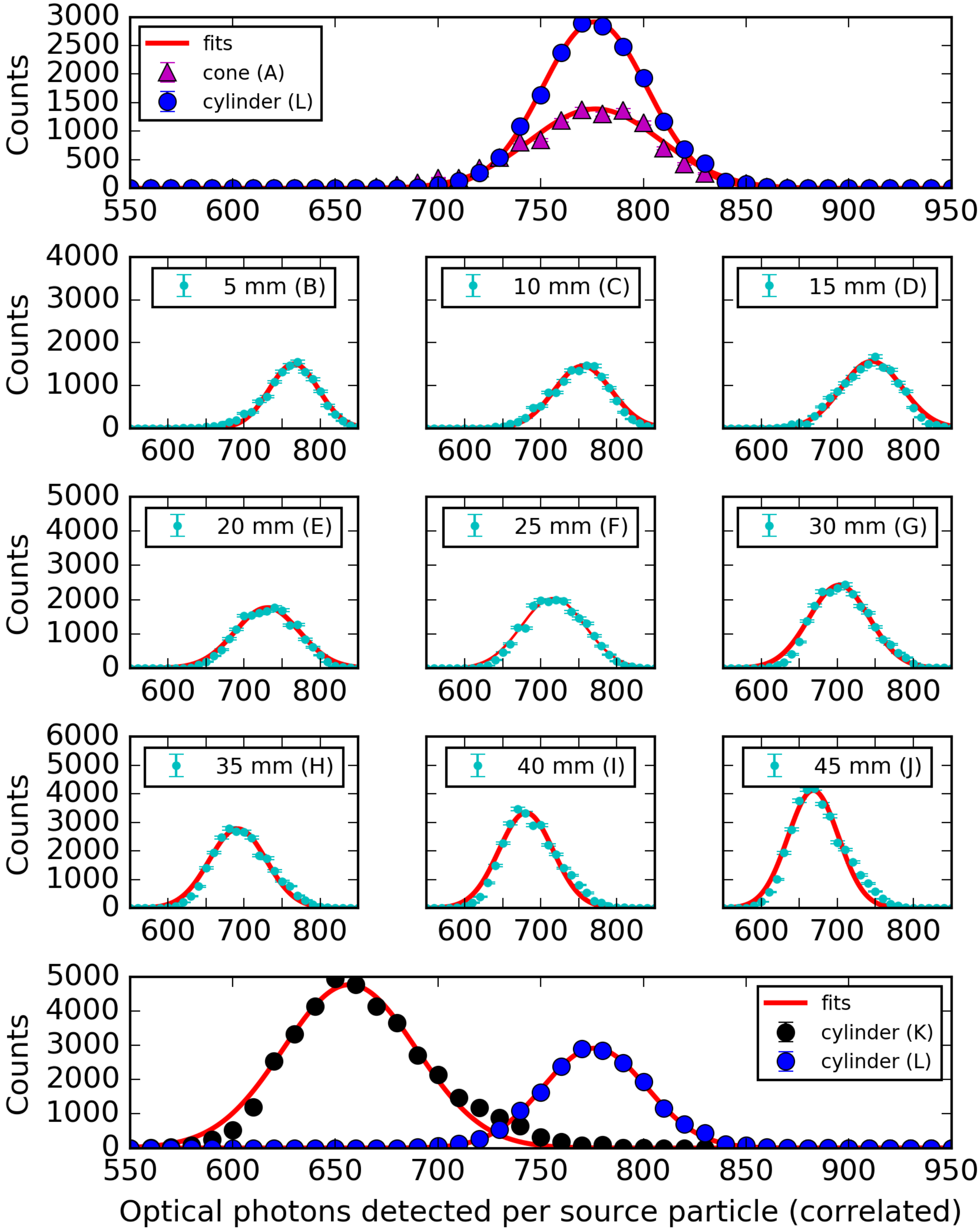


Figure 7.16: Correlated optical-photon detection spectra for all shapes.

It was interesting to note in Figure 7.17 how the FWHM for both the cone (A) and standard cylinder (K) were similar (approximately 75 optical photons). Although the FWHM of both shapes were essentially identical, the cone demonstrated greater LCE (expressed by the higher calibration point), than the cylinder. Dividing the same FWHM (75 optical photons) by the higher calibration point of the cone produced a better (i.e., lower) energy resolution (9.62%) than the standard cylinder (11.71%) for the same gamma-ray energy deposition (477.33 keV).

In the upper right plot of the Figure 7.17, the calibration position and gamma-ray detections were normalized and overlaid together as a function of shape ($A-L$). As expected, when the front-plane diameter of a tapered cone (shapes B-J) was reduced (45 mm to 5 mm), an increase in LCE and decrease in gamma-ray detections was observed. The intersection of both curves suggested that the ideal tapered cone for providing the optimal trade off in LCE and gamma-ray detection efficiency was shape G, however, the FWHM and energy resolution tell another story.

Figure 7.17 revealed an unexpected trend in FWHM as a function of shape. As the front-plane diameter for each tapered cone increased, it was expected that the energy resolution would progressively worsen (where the cylinder would produce the poorest energy resolution). Instead, the FWHM of tapered cones C through H peaked above the standard cylinder (K) and cone (A), thus producing even worse energy resolution than the standard cylinder (K). For example, Figure 7.17 shows how the FWHM of tapered cones E (20 mm) and F (25 mm) both produced a similar FWHM (approximately 100 optical photons) and energy resolution (approximately 13.7%); the worst values for all cases.

It was especially intriguing to observe how the truncated cylinder (L) outperformed the energy resolution of the cone (A), despite showing nearly identical calibration positions. It is evident that the driving factor behind the improved energy resolution of the truncated cylinder was its smaller FWHM (60 optical photons) as compared to the cone (75 optical photons).

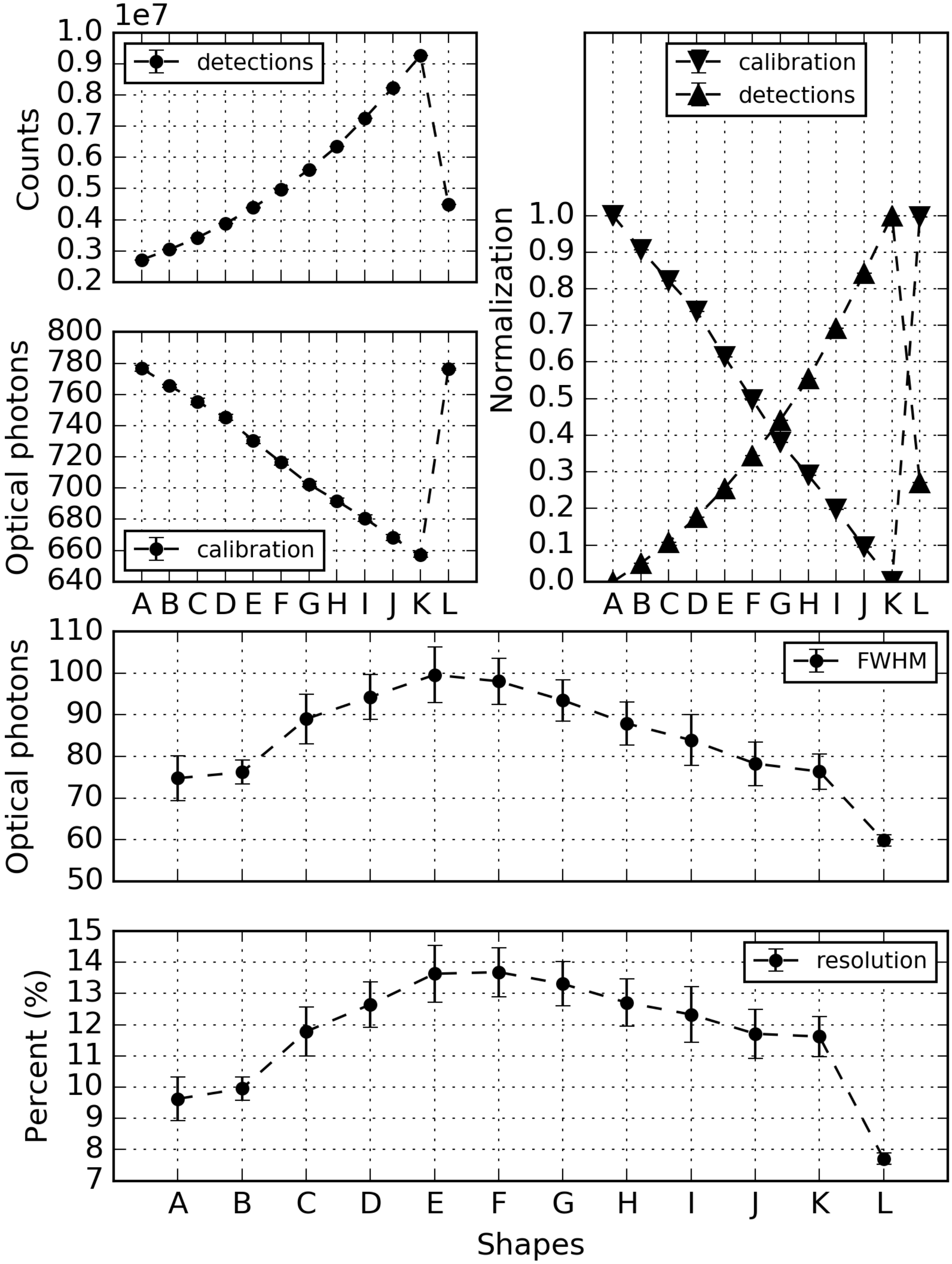


Figure 7.17: Efficiency trade-off results showing the calibration position, FWHM, energy resolution, and quantity of gamma-ray detections as a function of shape.

It is strongly suspected that the crest-like feature in the FWHM for the majority of tapered cones is due to an increase in multiple scatters per detection event. As the front-plane of a tapered cone is widened to a particular range of diameters (C through H), incident gamma rays see a more uniform and longer-average path length for shapes, which increases the likelihood for multiple-scatter per detection event.

Moreover, recall that for Compton scatters that produce low-light (below 100 keVee), singlet quenching non-linearly reduces the light yield [2]. However, this process was included in the Geant4 model (see 7.2.6). Therefore, multiple-scatter events contribute more light to the total light yield for a detection event than is realistically possible. In other words, if a gamma ray Compton scatters two or three times, and the total energy deposition for that detection event sums to a quantity that falls within the energy gate (476.34 to 477.34 keV) used for correlating optical-photon detections at the Compton edge, then a fraction of detected optical photons from each multiple scatter will sum together and contribute to a single quantity of total optical-photons detected, and crucially, so will their variance.

The light yield from any single-scatter event for any quantity of energy deposited will have some associated statistical variance in optical-photon production from scintillation. This variance is expected to increase for the same energy deposited (e.g., 477.33 keV) when produced by multiple-scatter events, where the variance in light yield for each multiple-scatter site is propagated to the summed quantity of optical-photon detections. Additionally, both single scatter and multiple-scatter light yields are further broadened by physical processes that reduce the optical-photon detection probability; for example the number of optical-photon reflections and the quantum efficiency of the PMT. In summary, multiple scatter is expected to broaden the optical-photon detection spectrum, particularly in simulations where the light yield for low-energy deposition events is overestimated and for scintillator shapes with greater likelihood for multiple-scatter contributions per detection event.

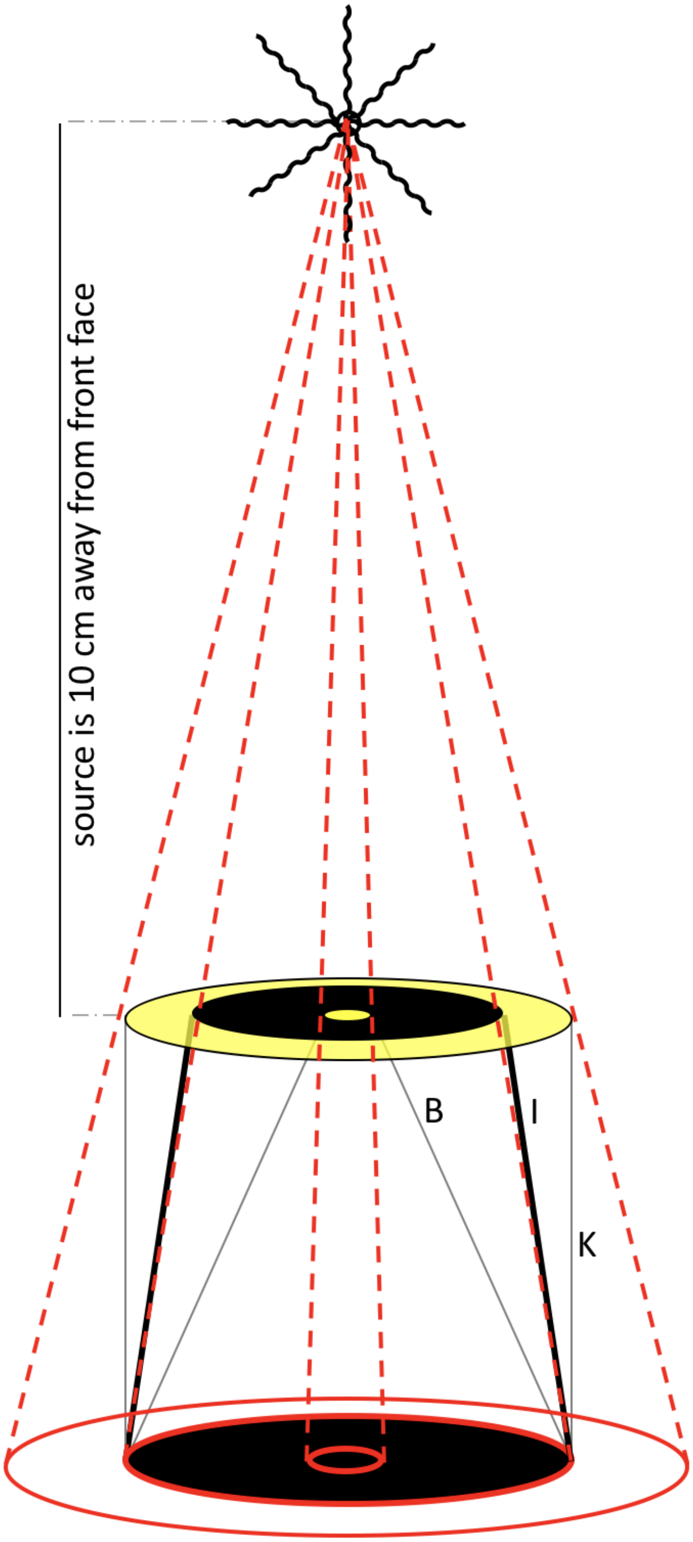


Figure 7.18: Shapes with the same height but longer average path length per incident gamma ray experience an increase multiple-scatter likelihood. The black and yellow regions represents front-plane diameters of tapered cones with poor and good energy resolution, respectively.

Figure 7.18 attempts to illustrates the solid-angle and geometry combinations that impact the FWHM of the correlated light-output spectra. The black region in Figure 7.18 show where the FWHM is increased due to light-output contributions from multiple scatter, resulting from an increase in the average path length traversed by a gamma ray in that material. If the front-plane diameter for a tapered cone falls within the black region shown in Figure 7.18, then these shapes are expected to experienced larger FWHM values due to the percent increase in multiple-scatter per detection event. The FWHM is reduced as the percent of single scatters are increased, particularly for shapes with lower-average path lengths, for example the cone and especially the truncated cylinder.

Figure 7.19 shows the percent contribution of single or multiple Compton scatters per detected event in the cone (A), standard cylinder (K), and standard cylinder (L). The plot on the left shows the scatter-quantity contributions per detection event for all energy depositions, whereas the plot on the right strictly compares the scatter-quantity contributions for energy depositions that fall within the energy gate (476.34 to 477.34 keV) used to correlate Compton-edge events (477.33 keV) to the total number of optical photon detected. What is immediately clear from both plots in Figure 7.19 is that the standard cylinder (K) has a higher fraction of double and triple scatters for all energies, which is expected due to its longer average path length when compared to the truncated cylinder (L) and cone (A). More importantly, the plot on the right shows that the cone (A) has a higher fraction of double and triple scatters per detection events at the Compton edge than does the truncated cylinder (L). The combined percent-increase in double and triple scatter likelihood per incident gamma ray produces a larger FWHM (75 optical photons) for the cone than the truncated cylinder (60 optical photons).

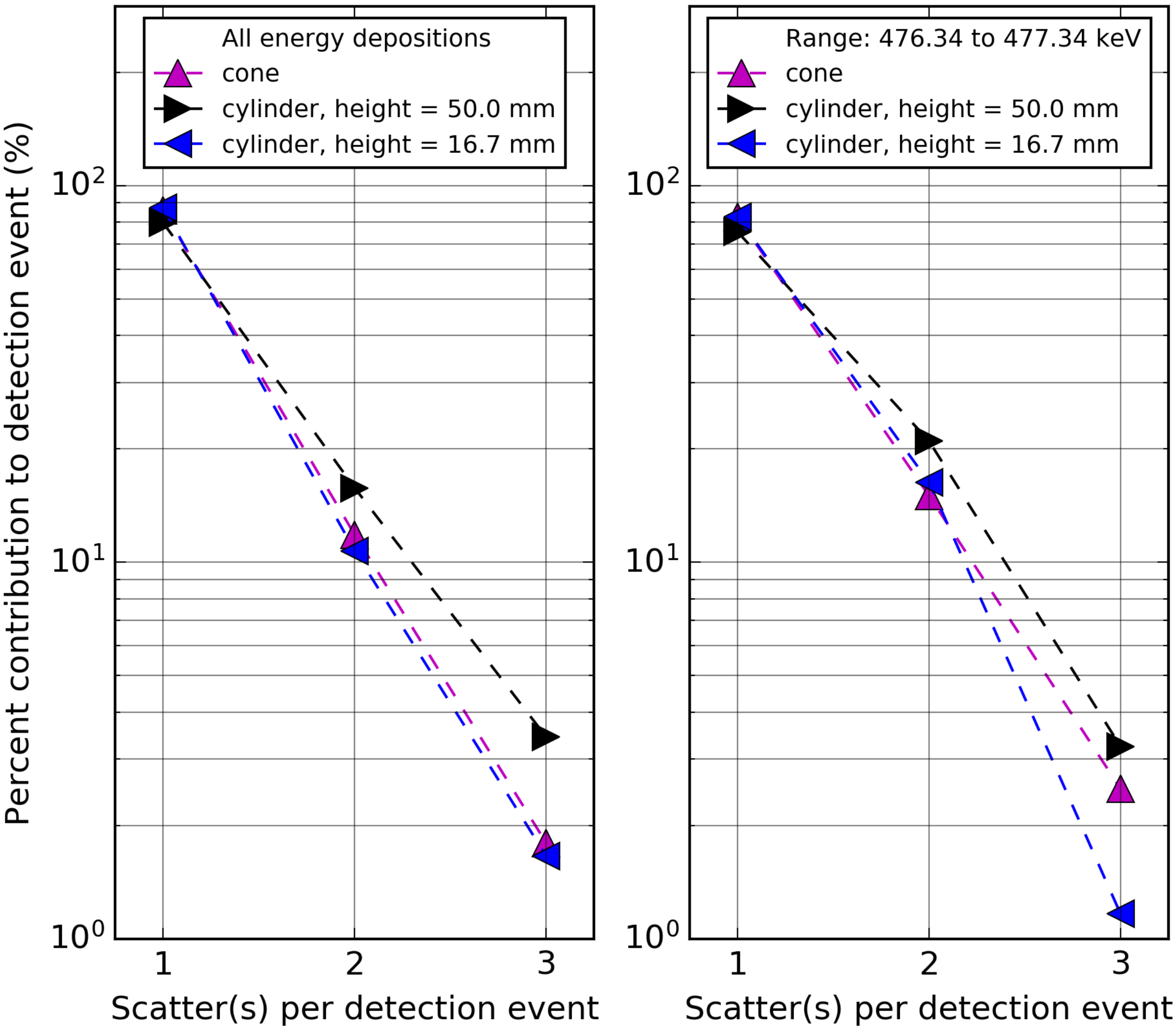


Figure 7.19: Percent contribution of single and multiple scatters per detected gamma ray.

Additionally, it is strongly suspected that scintillation events occurring near the tip of the cone, although rare (see Figure 7.22), further contribute to broadening the FWHM due to light trapping at the cone tip, which increases the number of optical-photon reflections required to reach the PMT surface; as opposed to scintillation events that occur near center or base of the cone. The simulated optical-photon reflection distributions for the cone (A), standard cylinder (K), and truncated cylinder (K) are shown together in Figure 7.20 and support the claim for light trapping near the cone tip. At first glance both reflection distributions of the cone (A) and truncated cylinder (L) look similar, however upon closer inspection an additional presence of counts is evident for the cone (A) along the trailing edge of its reflection distribution.

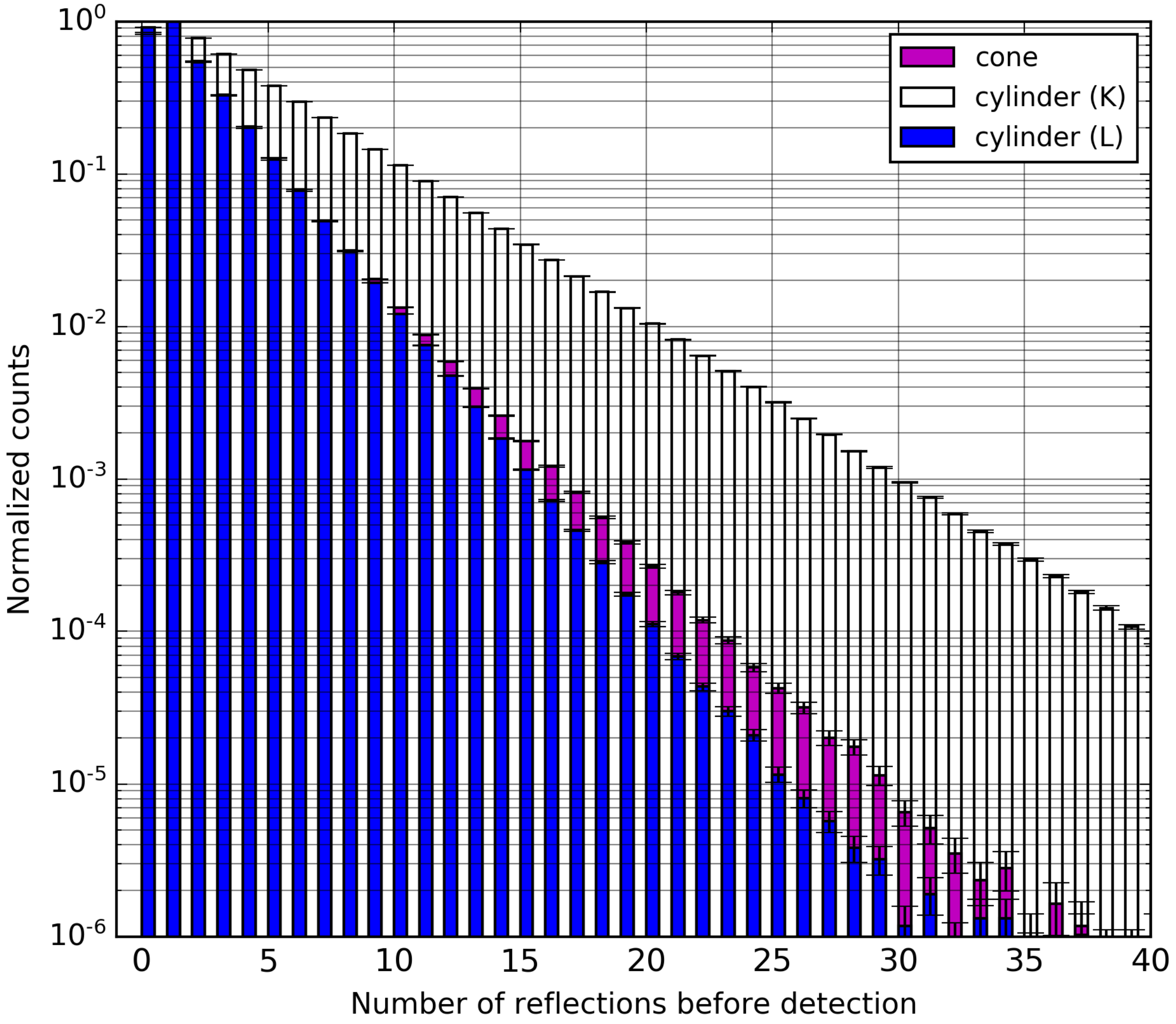


Figure 7.20: NOR distributions for the cone, truncated cylinder, and standard cylinder.

It is suspected that these additional reflections occur from Compton scatters near the tip of the cone, and thus offers support for the claim that light trapping could contribute to FWHM broadening. As mentioned, although the interactions near the tip of the cone are rare, they still carry the potential to broaden the mean quantity of optical-photons detections per gamma-ray detection event at the Compton edge. Therefore, the truncated cylinder shows the most promising results of all the other shapes tested (A through K). Moreover, the simulated light-collection process in the truncated cylinder was shown to be faster than cone, as expected given the short travel length to the PMT window (see Figure 7.21).

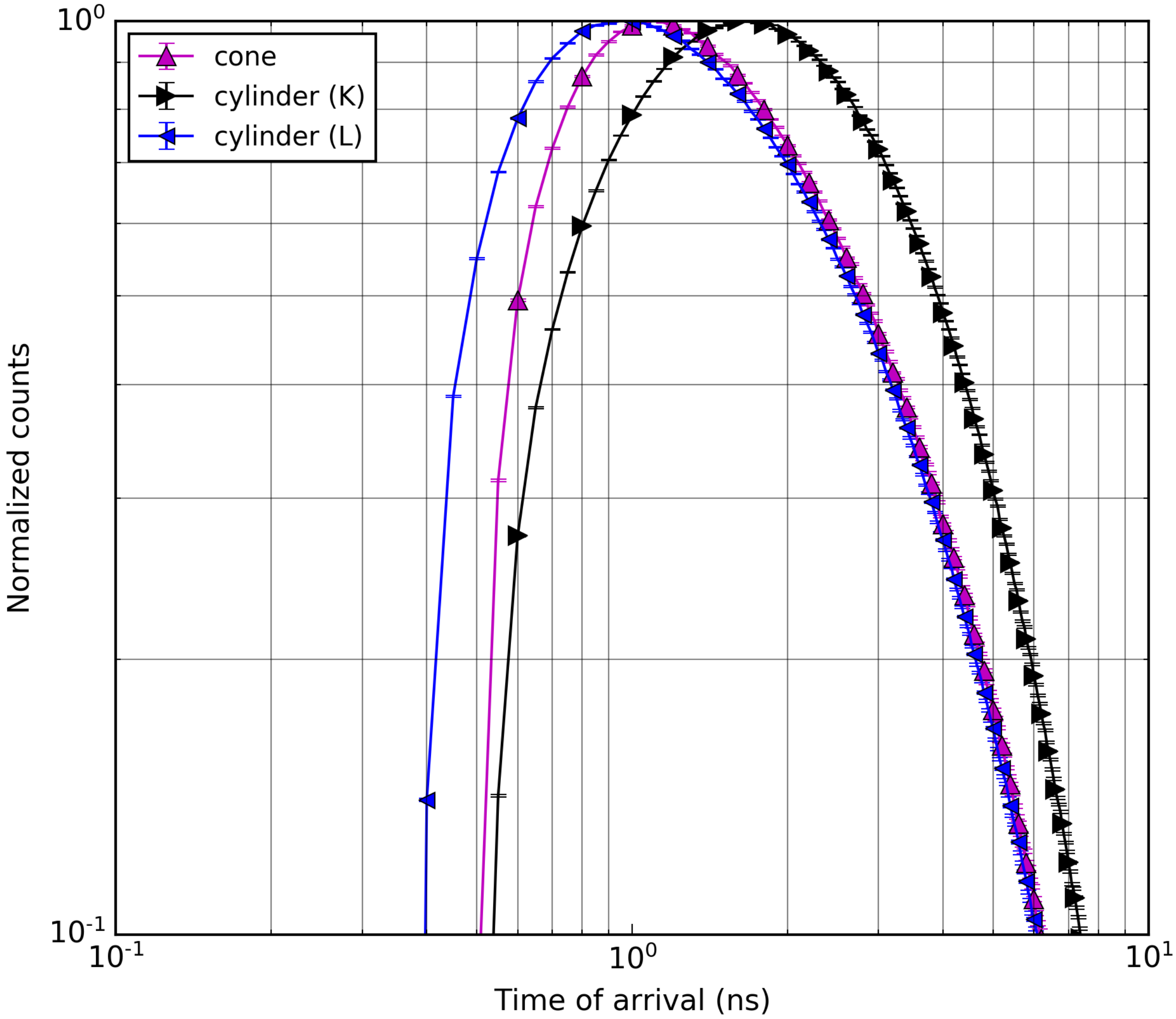


Figure 7.21: TOA distributions for the cone, truncated cylinder, and standard cylinder.

Lastly, the Compton-scatter heat map for both the truncated cylinder and cone were plotted together in Figure 7.22. The proximity for the predominant Compton-scatter locations in the truncated cylinder to the PMT surface is of concern. Similar to the cone, it is likely that the truncated cylinder may also experience degraded energy-resolution performance in experiments where a PMT of equal-base diameter is used due the strong influence from photocathode non-uniformity as a function of scintillation position.

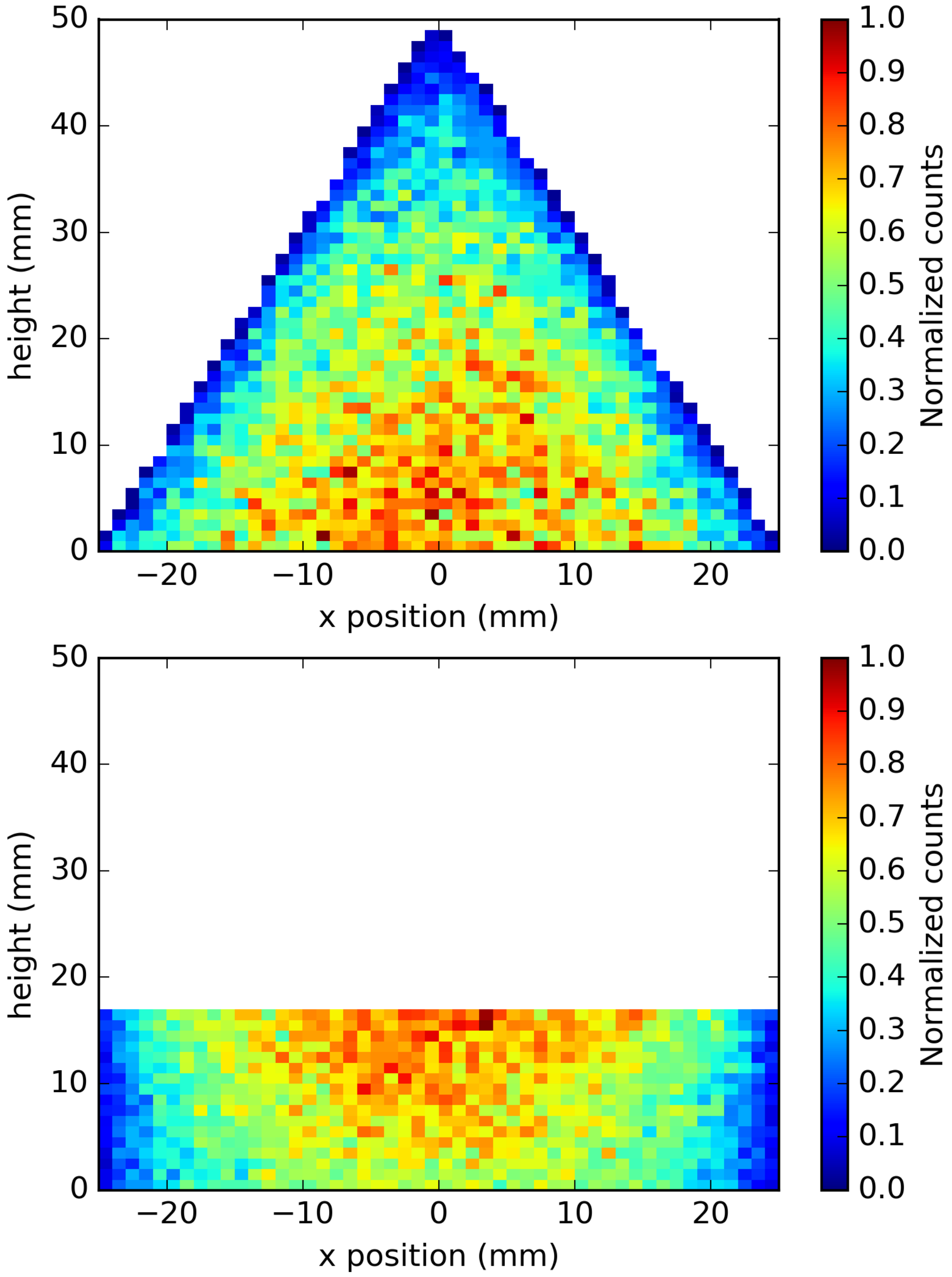


Figure 7.22: Compton-scatter heat maps for the cone, truncated cylinder, and standard cylinder.

7.5 Discussion

A Geant4 model of an EJ200 scintillator was developed and validated using experimental data that agreed well with simulated light-output spectra of a cesium-137 source (see Table 7.2). The model was used to study optical-photon behavior in an organic scintillator and helped to visualize how spectrum smearing can occur from optical-photon funneling along the surface of a highly-polished conical scintillator. It was also shown that on average, optical photons reflect in a cylinder and cone approximately 5 and 2 times, respectively (see Figure 7.4.2). These results confirm the hypothesis that a reduction in optical-photon reflections would translate to an increase in LCE, and moreover, reduced transit time prior to detection. The faster TOA distribution of the cone as compared to the cylinder also support trends observed in the time-resolution experiments (see Chapter 5).

Additionally, simulations were performed that compared the gamma-ray detection efficiency and LCE for various shapes: the cone, nine tapered cones with different front-plane diameters, the standard cylinder, and a truncated cylinder. The study revealed that in theory, a tapered cone with a front-surface diameter of 30 mm should produce the “best of both worlds" between the standard cylinder and cone. However, the tapered cone reported worse energy resolution than the standard cylinder. This unexpected result was attributed to an increase in multiple-scatter contributions per detection event at the Compton edge, which broadened the quantity of optical-photon detections for the tapered cone as compared to the standard cylinder.

The study also revealed that a truncated cylinder with a height of 16.7 mm and a base diameter of 50 mm outperformed all shapes tested, including the cone, both in energy resolution and gamma-ray detection efficiency. The poorer energy resolution of the cone as compared to the truncated cylinder is a consequence of the cone tip acting as a light trap, which increases the average number of reflections and thus reduces the detection probability for optical photons generated in that region. Additionally, when compared to the tapered cylinder, the cone exhibited a greater percent contribution of multiple scatters per detection event at the Compton edge, which combines with light-trapping effects to further broaden its FWHM to a value larger than the truncated cylinder (see Figures 7.17 and 7.19); despite both shapes having nearly identical LCE (19% at 477.33 keVee).

Recall that experiments in Chapter 4 demonstrated that a ground cone could still be coupled to a PMT of equal base diameter to the cone without producing spectrum smearing (see Figures 4.12

and 4.1). However, the gain in performance of the cone over the standard cylinder was reduced due to strong variations in the localized QE of the PMT (see Table 4.1). It is suspected that the truncated cylinder may be equally susceptible to photocathode non-uniformity effects as the cone if it coupled to a PMT with similar base diameter. Therefore, it is suspected that the truncated cylinder would performed best in experiment if its surface were ground rather than polished, and also if it were coupled to a large-diameter PMT where the photocathode response is relatively uniform across the scintillator base (see Figure 4.7).

Up until this point, experimental results have shown that the cone outperformed the standard cylinder in all three primary performance metrics (energy resolution, time resolution, and PSD). However, it is important to emphasize that the cone was not presented in this work as the optimal scintillator shape. Rather, the cone was shown as an alternative shape to the standard cylinder, commonly used in many nuclear nonproliferation and safeguard research equipment [16, 17, 35, 36], that could provide measurable gains in the three detector performance metrics explored in this work (see Tables 4.1, 5.1, and 6.2). It is with certainty that other shapes can provide even better overall performance than the cone; as evident by the simulated truncated cylinder, which showed better gamma-ray detection efficiency and energy resolution as compared to the cone. It is likely that a scintillator in the shape of a parabolic lens with a focal point that lands on the optical-window surface of a PMT may even outperform the energy resolution of the truncated cylinder, however the gamma-ray detection efficiency would likely be less. Detection efficiency, however, may not be a criteria that requires maximization for certain radiation detection systems. The choice of scintillator shape for a system is therefore strongly dependent on its application and the performance metrics chosen for maximization.

Lastly, it is worth empathizing that some result bias in the model may exist for the simulated FWHM of the optical-photon detection spectra correlated with detection events at the Compton-edge. Although the variance in light yield for single scatters is less than the propagated variance in total light yield from multiple scatters for the same energy deposited, it is unclear from simulation how much of an impact multiple scatters (where the energy deposited is less than 100 keV), contribute to broadening the FWHM.

Since the model did not simulate prompt-singlet quenching from low-energy deposition events (<100 keV), it is possible that the light-yield linearity of the model for all gamma-ray energy depositions could produce an unrealistic quantity of optical-photon emissions from multiple-scatter events. Moreover, double and triple scatter contributions to detection events at the Compton-edge are rare in comparison to single scatters. Therefore, it is possible that the simulated FWHM of the correlated optical-photon detection spectra may be over empathized. Experiments may show that the truncated cylinder has only a marginal improvement (or possibly similar) energy resolution to the cone. It is still suspected, however, that the energy resolution of the truncated cylinder should be better than the cone due to light-trapping effects in the cone. Regardless, even if both shapes are shown to produce a similar energy resolution in experiment, it is clear that the truncated cylinder would be more useful the cone due to the higher gamma-ray detection efficiency.

CHAPTER 8

Summary, Conclusions and Future Work

8.1 Summary

This thesis demonstrated the importance of LCE in radiation-detection systems that use organic scintillators, particularly for applications within nuclear nonproliferation and international safeguards. A new method was demonstrated for simultaneously improving three metrics of system performance: the energy resolution, time resolution, and particle identification. Specifically, this thesis has shown that there are better scintillator shapes than the commonly used right-circular cylinder for improving each of these three system performance metrics. Although these systems can range in size, complexity, and purpose, they all typically have the same fundamental configuration; a cylindrical organic scintillator coupled to a light-sensing device, for example a PMT. Factors that perpetuate optical-photon losses, for example the QE of a PMT or transmission through a reflector, negatively impact detector performance. Historically, efforts to improve detector performance have largely focused on the organic scintillator material to produce a greater light yield per energy deposited, rather than using the scintillator shape to return greater LCE. Most organic scintillators used in nuclear-security applications, especially PSD-capable *trans*-stilbene due to its manufacturing process, are commonly found in cylindrical form. This work has shown that improvements in LCE were achievable by using a shape better suited than the cylinder for reducing optical-photon reflections and thus their transmission likelihood through the reflector. Specifically, the three performance metrics mentioned earlier were experimentally obtained and compared for conical and cylindrical organic scintillators of equal base diameter, height, and material. The detector efficiency was also compared; an important consideration for certain applications that require a fast measurement time. Although the lower detection efficiency of the cone was unavoidable due to its decrease in volume (66.67%) when compared to the standard cylinder, the cone showed significant enhancements in detector performance criteria that were light-collection dependent. Therefore, a cone would be a useful alternative to the standard cylinder in systems that prioritize energy resolution, time resolution, and PSD performance, above all other considerations. A short description that explains why a cone improve these metrics in contrast to a standard cylinder, and by how much, is listed below. Included in the list is a short summary of the Geant4 simulation results.

1. Energy resolution

   The increase in LCE provides additional statistical information per scintillation event, which assists the PMT to generate a pulse that more accurately represents the energy deposited; thus improving energy resolution. A 16.2% improvement in energy resolution at 478 keVee was observed when using a conical scintillator in a place of a cylindrical scintillator of identical base diameter (50 mm), height (50 mm), surface treatment (un-polished), reflector (3M D2000), and material (EJ200). These conditions were observed when the two scintillators were compared using a PMT (Photonis XP4512B) with a base diameter of 127 mm.

2. Time resolution

   The increase in LCE is caused by a decrease in optical-photon reflections, which reduces their reflector-transmission likelihood, and as a consequence, also shortens the average distance traveled before detection. Therefore, the average transit time for optical photons to reach the photocathode is shortened, which improves the time resolution. A 35.2% improvement in time resolution at a CFD of 0.2 was observed when using a conical scintillator in place of a cylindrical scintillator of identical base diameter (50 mm), height (50 mm), surface treatment (polished), reflector (3M D2000), and material (EJ200).

3. PSD

   The increase in LCE provides additional pulse clarity (i.e., statistical information per pulse) at low-light output events, which helps the digital charge-integration technique to more accurately assess delayed-florescence contributions in the pulse tail; thus improving PSD. A 22.6% improvement in neutron/gamma-ray discrimination between 25-100 keVee was observed when using a conical scintillator in a place of a cylindrical scintillator of identical base diameter (50 mm), height (50 mm), surface treatment (un-polished), and material (*trans*-stilbene). These conditions were observed using a cross correlation experiment described in Section 3.3.2.

4. Geant4

   The simulated energy-resolution values of the cone and standard cylinder agreed well with measurement (see Table 7.2). Also, the simulated TOA distributions (see Figure 7.11) confirmed that optical photons were detected faster (on average) in the cone. The leading edge of the TOA distribution for the cone was also noticeably narrower in time than the cylinder. These results support the trends observed in the time resolution experiments (see Figures 5.13 - 5.14), where rise-time distributions and scintillator-pulse shapes of the cone and standard cylinder were compared. Additionally, efficiency trade-off simulations showed that although a tapered cone demonstrated better LCE than then standard cylinder, its energy resolution was worse; caused by a percent increase in multiple-scatter versus single scatter contributions to detection events where the total-energy deposited were equal. Multiple-scatter likelihood is directly correlated with the average path length in a material that is visible to an incident gamma ray. A truncated cylinder with a base diameter of 50 mm and height of 16.7 mm, was shown to outperform the energy resolution of the cone for the reason just described. In other words, the truncated cylinder exhibited less multiple scatters per incident gamma ray than the cone. More importantly, despite having equal volume to the cone, the truncated cylinder returned greater gamma-ray detection efficiency than the cone by approximately 41%.

8.2 Conclusions

Organic scintillators are used in a number of applied systems within the field of nuclear security, and new system technology is constantly being developed to either replace, enhance, or add capabilities to the field. This thesis demonstrated that organic scintillators with different geometries other than the standard right circular cylinder, for example a cone, or potentially a truncated cylinder, would substantially improve important light-collection dependent performance metrics (e.g., energy resolution, time resolution, and PSD). Through experiments, a conical scintillator was shown to be more useful than the standard cylinder in systems where high gamma-ray or neutron detection efficiency is not the primary objective. Due to the loss in detection efficiency (69%) as compared to the standard cylinder, a cone may not be the ideal scintillator shape for certain applications despite its improvement in other performance metrics.

For example, a cone would not be a useful replacement to the standard cylinder in a dual-particle imaging system, which is already count starved [36], a neutron-scatter camera used for long stand-off detection [35,38], or a hand held inorganic-scintillator based system used for gamma-ray source identification in emergency-response scenarios. However, simulations do suggest that a truncated cylinder may outperform the cone in energy resolution gains as compared to the standard cylinder, while at the same time providing better gamma-ray detection efficiency than the cone by approximately 41%. In scenarios where high counts are demanded but system size is not important to optimize, results demonstrated that any scintillator geometry, including the standard cylinder, would benefit from using a PMT with a larger diameter. Improved energy resolution was observed in experiment for both the cylinder and cone when the PMT size was increased. This result emphasizes the importance for future PMT design that mitigates photocathode non-uniformity effects from negatively affecting energy resolution.

On the other hand, the compactness of the cone, its excellent PSD performance, improved time resolution (35%), and faster rise times, may benefit novel hand held neutron-scatter cameras [37, 39] intended for portability and shorter stand-off detection. Fast response times and narrow time-resolution are essential for such compact devices that require the time resolution of each detector in its system to be narrower than the flight time between single scatters in adjacent scintillator volumes for reasonably accurate neutron-source localization and imaging. The excellent time resolution may also benefit gamma-ray sequencing methods to further improve the systems capability to image gamma-ray sources, as well. Additionally, the boosted discrimination capability (23%) of neutrons and gamma-rays below 100 keVee may significantly help fast-neutron multiplicity counters [16, 17] to more accurately identify neutron multiples. And although the defection efficiency of the multiplicity counter is reduced, the conical volumes can be brought in closer to the source to return greater efficiency.

The improved time resolution of a cone may also improve the quality of the neutron light-output function obtained through a TOF experiment. In short, TOF can be used to measure incident-neutron energy and unfold the neutron-light output response from proton and carbon recoils in a PSD-capable organic scintillator. Unlike electron recoils, the neutron response function is non-linear and the energy deposited are not directly equivalent to light output. By collecting optical photons on a shorter time scale, the cone effectively reduces the bin width in the TOF spectrum as compared to a cylinder, where each bin corresponds to a specific neutron-energy deposition through the kinetic-energy relationship. Smaller time bins improve the sampling of all incident-neutron energies and thus improve neutron-response unfolding calculations. Furthermore, by using pulse integrals rather than pulse heights to calculate the light output, the unfolding process becomes independent of pulse shape and rise time. Therefore, a conical PSD-capable organic scintillator can be used to measure the neutron-response function for any geometry of the same material, while simultaneously improving its sampling and reducing its uncertainty due to a decrease in propagated error from improved energy resolution and time pick-off assessment. These improvements would particularly benefit neutron-TOF experiments used to characterize new PSD-capable organic scintillator materials. The neutron-response function can be an important quantity to implement in experiments and simulations, for example Monte-carlo based radiation-transport codes, such as MCNPX-PoLiMi. Improving the data fidelity of these packages makes the simulated light-output results more realistic.

The increase of LCE through geometric means for the purpose of improving detector performance metrics has been demonstrated. A couple applications have already been introduced that may benefit from a conical scintillator. Additional and exciting ideas that expands on work presented in this thesis are discussed in the section below.

8.3 Future work

A number of interesting ideas can be explored in future studies. For example, a conical scintillator equal in volume to the standard cylinder can be compared both in simulation and experiment as an effort to increase gamma-ray detection efficiency. Additionally, Geant4 simulations suggested that a truncated cylinder could outperform the cone both in energy resolution and gamma-ray detection efficiency. Therefore, another idea would be confirm these results experimentally. One particularly exciting idea is to measure and compare the neutron-light output function of a PSD-capable organic scintillator in the shape of the cone, the standard cylinder, and the truncated cylinder. As demonstrated by experimental results in this work, a cone is expected to yield narrower time intervals in a TOF spectrum, and therefore improve the neutron-spectrum unfolding.

Another exciting idea would be to use a conical scintillator to expand upon the charge-integration technique for PSD by using the pulse rise time, in addition to ratio between the tail and total integrals, to better identify particle types. By combining both quantities, a new FOM can be created that may potentially further improve neutron and gamma-ray pulse identification in mixed-radiation fields. Such a study may not be as feasible with the standard cylinder, which may not have sufficient statistical information along the leading edge of the pulse so that the rise time can be used as an additional component to particle-discrimination. Another related experiment would be to repeat the rise-time analysis for both the cone and standard cylinder by using a faster-sampling

(> 500 MHz) DAQ system to more accurately resolve the rise-time distribution.

The Geant4 model developed for this work could also be improved upon by adding physical processes in organic scintillators that were not considered. For example, light yield non-linearity for low-energy deposition events (<100 keV) can be modeled by including the Birks' light-output reduction feature, available in Geant4. Once this feature is included, a sensitivity study on the variance in total light-yield from single and multiple scatters for the same detection event (i.e., energy deposited) should be performed. Additionally, Rayleigh scatter can also be added to the model and may improve its capability to more accurately represent optical-photon transport in large scintillator volumes, for example long rectangular plastic-scintillator prisms that are currently used in radiation-portal monitors.

Lastly, more Geant4 simulations can be performed to estimate gains in energy resolution over the standard cylinder for other unique and exciting shapes not explored in this work. One option is a paraboloid with a lens-like structure where its focal point lands directly on the center of the PMT window. Although this configuration may further reduce gamma-ray detection efficiency, it may further improve energy resolution, time resolution, and PSD performance beyond that of the cone and truncated cylinder. Since stilbene crystals must be grown as a cylinder and are extremely difficult to machine-cut into a paraboloid or hemisphere without cracking, a plastic material should be used to assess the PSD performance of the paraboloid lens. Some potential options include EJ299 and the newly released EJ276.The possibilities are endless!

APPENDIX A

Source code: Data Processing

A.1 Library of functions

```
from mpl_toolkits.axes_grid1.inset_locator import zoomed_inset_axes
from mpl_toolkits.axes_grid1.inset_locator import mark_inset
from matplotlib import pyplot as plt
from scipy.signal import find_peaks_cwt
from scipy.interpolate import interp1d
from scipy.optimize import curve_fit
from scipy import exp
from scipy import asarray as ar
from scipy import stats
from numpy import loadtxt
from shutil import copyfile
from shutil import rmtree
import matplotlib as mpl
from matplotlib import cm
from matplotlib.colors import LogNorm
from matplotlib.ticker import LogFormatter
from mpl_toolkits.mplot3d import Axes3D
mpl.style.use('classic')
import numpy as np
import math as m
import random as r
import shutil
import time
import os

dpiRes = 300.0

directoryPSD = [
'/Volumes/EX1/data/experiments/journal3/psd/
    ISBCONGS_PMT5_N1700V_CF252m_10CM_15HR',
'/Volumes/EX1/data/experiments/journal3/psd/
    ISBCYLGS_PMT5_N1700V_CF252m_10CM_15HR',
'/Volumes/EX1/data/experiments/journal3/psd/archive/
    EJ299CONGS_PMT5_N1700V_CF252m_10CM_15HR',
'/Volumes/EX1/data/experiments/journal3/psd/archive/
    EJ299CYLGS_PMT5_N1700V_CF252m_10CM_15HR']
```

```
directoryResolution = [
'/Volumes/EX1/data/experiments/journal2/pmt2/
    EJ200CONGS_PMT2_N1425V_CS137s_10CM_15CM_5HR_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt2/
    EJ200CONPS_PMT2_N1425V_CS137s_10CM_15CM_5HR_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt2/
    EJ200CYLGS_PMT2_N1425V_CS137s_10CM_15CM_5HR_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt2/
    EJ200CYLPS_PMT2_N1425V_CS137s_10CM_15CM_5HR_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt2/map/
    EJ200CUBGS_PMT2_POS1_N1425V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt2/map/
    EJ200CUBGS_PMT2_POS2_N1425V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt2/map/
    EJ200CUBGS_PMT2_POS3_N1425V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt2/map/
    EJ200CUBGS_PMT2_POS4_N1425V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt2/map/
    EJ200CUBGS_PMT2_POS5_N1425V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt2/map/
    EJ200CUBGS_PMT2_POS6_N1425V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt2/map/
    EJ200CUBGS_PMT2_POS7_N1425V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt2/EB/
    EJ200CONGS_PMT2EB_N1425V_CS137s_10CM_15CM_5HR_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt2/EB/
    EJ200CONPS_PMT2EB_N1425V_CS137s_10CM_15CM_5HR_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt2/EB/
    EJ200CYLGS_PMT2EB_N1425V_CS137s_10CM_15CM_5HR_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt2/EB/
    EJ200CYLPS_PMT2EB_N1425V_CS137s_10CM_15CM_5HR_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt3/
    EJ200CONGS_PMT3_N1830V_CS137s_10CM_15CM_5HR_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt3/
    EJ200CONPS_PMT3_N1830V_CS137s_10CM_15CM_5HR_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt3/
    EJ200CYLGS_PMT3_N1830V_CS137s_10CM_15CM_5HR_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt3/
    EJ200CYLPS_PMT3_N1830V_CS137s_10CM_15CM_5HR_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt3/map/
    EJ200CUBGS_PMT3_POS1_N1830V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt3/map/
    EJ200CUBGS_PMT3_POS2_N1830V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt3/map/
    EJ200CUBGS_PMT3_POS3_N1830V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt3/map/
    EJ200CUBGS_PMT3_POS4_N1830V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt3/map/
    EJ200CUBGS_PMT3_POS5_N1830V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt3/map/
    EJ200CUBGS_PMT3_POS6_N1830V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt3/map/
    EJ200CUBGS_PMT3_POS7_N1830V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt5/
    EJ200CONGS_PMT5_N1700V_CS137s_10CM_15CM_5HR_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt5/
    EJ200CONPS_PMT5_N1700V_CS137s_10CM_15CM_5HR_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt5/
```

```
    EJ200CYLGS_PMT5_N1700V_CS137s_10CM_15CM_5HR_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt5/
    EJ200CYLPS_PMT5_N1700V_CS137s_10CM_15CM_5HR_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt5/map/
    EJ200CUBGS_POS1_PMT5_N1700V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt5/map/
    EJ200CUBGS_POS2_PMT5_N1700V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt5/map/
    EJ200CUBGS_POS3_PMT5_N1700V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt5/map/
    EJ200CUBGS_POS4_PMT5_N1700V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt5/map/
    EJ200CUBGS_POS5_PMT5_N1700V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt5/map/
    EJ200CUBGS_POS6_PMT5_N1700V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal2/pmt5/map/
    EJ200CUBGS_POS7_PMT5_N1700V_CS137s_0CM_15CM_30MIN_180DEG',
'/Volumes/EX1/data/experiments/journal3/energy/
    ISBCONGS_PMT5_N1700V_CS137s_10CM_15CM_5HR_180DEG',
'/Volumes/EX1/data/experiments/journal3/energy/
    ISBCYLGS_PMT5_N1700V_CS137s_10CM_15CM_5HR_180DEG',
'/Volumes/EX1/data/experiments/supplemental/042217
    _EJ200CYL0D_ETLPMT0_N1150V_CS137s_30CM_2HR_180DEG',
'/Volumes/EX1/data/experiments/supplemental/042217
    _EJ200CYL0D_ETLPMT0_N1150V_CS137s_30CM_2HR_90DEG',
'/Volumes/EX1/data/experiments/supplemental/042217
    _EJ200CYL0D_ETLPMT0_N1150V_CS137s_30CM_2HR_45DEG']

directoryTIME = [
'/Users/cssosa/Box Sync/work/data/experiments/journal4/010418
    _EJ200CONPS_PMTC1_N1700V_PMTC2_N1310V_NA22s_22CM_5MIN',
'/Users/cssosa/Box Sync/work/data/experiments/journal4/010418
    _EJ200CYLPS_PMTC1_N1700V_PMTC2_N1298V_NA22s_22CM_5MIN']

def findEnergyFinal(energyInitial,theta):
A = 511.0; # keV
B = 1.0 + energyInitial*(1.0-m.cos(m.radians(theta)))/A
energyFinal = energyInitial/B
return energyFinal

def findEnergyDepositied(energyInitial,theta):
A = 511.0; # keV
B = 1.0 + energyInitial*(1.0-m.cos(m.radians(theta)))/A
energyDep = energyInitial/B
return energyInitial - energyDep

def findThetaScatter(energyInitial,energyFinal):
A = 1.0 - 511.0/energyFinal + 511.0/energyInitial
print(A)
if A > -1.0:
print('cond1')
return m.degrees(m.acos(A))
if A < -1.0:
print('cond2')
B = m.degrees(m.acos(-1.0))
C = m.degrees(m.acos(1 - (-1 - A)))
return B
```

```
def hist(data,binRange):
counts, bins = np.histogram(data,binRange)
bins   = bins[0:counts.size]
counts = counts
return counts, bins

def singleGauss_func(x,a1,b1,sigma1):
return a1*exp(-(x-b1)**2/(2.0*(sigma1**2)))

def doubleGauss_func(x,a1,b1,sigma1,a2,b2,sigma2):
return a1*exp(-(x-b1)**2/(2.0*(sigma1**2))) + a2*exp(-(x-b2)**2/(2.0*(sigma2
    **2)))

def singleGauss_guess(xData,yData):
x = ar(xData)
y = ar(yData)
maxima = max(y)
mean   = x[yData.argmax(axis=0)]
sigma  = np.sqrt((1./(len(x)-1.))*sum(np.power((x-mean),2)))/2.0
guess  = [maxima,mean,sigma]
return guess[0], guess[1], guess[2]

def doubleGauss_guess(xData,yData,cut):
x  = ar(xData)
y  = ar(yData)
i1 = np.where(x<cut)
x1 = ar(x[i1])
y1 = ar(y[i1])
i2 = np.where(x>cut)
x2 = ar(x[i2])
y2 = ar(y[i2])
maxima1 = max(y1)
mean1   = x1[y1.argmax(axis=0)]
sigma1  = np.sqrt((1./(len(x1)-1.))*sum(np.power((x1-mean1),2)))/2.0
guess1  = [maxima1,mean1,sigma1]
#plt.plot(x2,y2);plt.show()
maxima2 = max(y2)
mean2   = x2[y2.argmax(axis=0)]
sigma2  = np.sqrt((1./(len(x2)-1.))*sum(np.power((x2-mean2),2)))/2.0
guess2  = [maxima2,mean2,sigma2]
return guess1[0], guess1[1], guess1[2], guess2[0], guess2[1], guess2[2]

def singleGauss_fit(x,y,g):
w = np.where(y > 0)
popt, pcov = curve_fit(singleGauss_func,x[w],y[w],sigma=1.0/np.sqrt(y[w]),p0=g
    ,absolute_sigma=False,method='lm')
perr       = np.sqrt(np.diag(pcov))
return popt[0], perr[0], popt[1], perr[1], popt[2], perr[2]

def doubleGauss_fit(x,y,g):
#w = np.where(y > 0)
popt, pcov = curve_fit(doubleGauss_func,x,y,p0=g,method='lm')
perr       = np.sqrt(np.diag(pcov))
return popt[0], perr[0], popt[1], perr[1], popt[2], perr[2], popt[3], perr[3],
     popt[4], perr[4], popt[5], perr[5]

def doubleGauss_findCentroids(cent1,eCent1,cent2,eCent2):
ci               = 0.95
```

```
pp                = (1.0 + ci)/2.0
nstd              = stats.norm.ppf(pp)
centroid1         = cent1
centroid2         = cent2
sigmaCentroid1    = nstd*eCent1
sigmaCentroid2    = nstd*eCent1
return centroid1, sigmaCentroid1, centroid2, sigmaCentroid2

def doubleGauss_findWidths(sigma1,errorSigma1,sigma2,errorSigma2):
ci                          = 0.95
pp                          = (1.0+ci)/2.0
nstd                        = stats.norm.ppf(pp)
k                           = 2.0*m.sqrt(2.0*m.log(2.0))
fwhm1             = k*abs(sigma1)
fwhm2             = k*abs(sigma2)
sigmaFWHM1        = nstd*k*errorSigma1
sigmaFWHM2        = nstd*k*errorSigma2
return fwhm1, sigmaFWHM1, fwhm2, sigmaFWHM2

def findFOMerror1(centroid1,sigmaCentroid1,centroid2,sigmaCentroid2,FWHM1,
    sigmaFWHM1,FWHM2,sigmaFWHM2):
C1        = centroid1
C2        = centroid2
F1        = FWHM1
F2        = FWHM2
sigmaC1   = sigmaCentroid1
sigmaC2   = sigmaCentroid2
sigmaF1   = sigmaFWHM1
sigmaF2   = sigmaFWHM2
M                 = (C2-C1)/(F1+F2)
a                 = (sigmaCentroid1**2) + (sigmaCentroid2**2)
b                 = F1+F2
c                 = C2-C1
d                 = (sigmaFWHM1**2) + (sigmaFWHM2**2)
sigmaM    = M*np.sqrt((a/(b*c)) + (c*d/(b**3)))
return M, sigmaM

def findFOMerror2(centroid1,sigmaCentroid1,centroid2,sigmaCentroid2,FWHM1,
    sigmaFWHM1,FWHM2,sigmaFWHM2):
C1        = centroid1
C2        = centroid2
F1        = FWHM1
F2        = FWHM2
sigmaC1   = sigmaCentroid1
sigmaC2   = sigmaCentroid2
sigmaF1   = sigmaFWHM1
sigmaF2   = sigmaFWHM2
M                 = (C2-C1)/np.sqrt(F1+F2)
a                 = sigmaCentroid1/(C2-C1)
b                 = sigmaCentroid2/(C2-C1)
c                 = sigmaFWHM1/(2.0*(F1+F2))
d                 = sigmaFWHM2/(2.0*(F1+F2))
sigmaM    = M*np.sqrt((a**2)+(b**2)+(c**2)+(d**2))
return M, sigmaM

def findResolution(centroid,errorCentroid,sigma,errorSigma):
ci                          = 0.95
pp                          = (1.0+ci)/2.0
```

```
nstd                    = stats.norm.ppf(pp)
k                       = 2.0*m.sqrt(2.0*m.log(2.0))
fwhm                    = k*abs(sigma)
resolution              = 100.0*fwhm/centroid
sigmaFWHM       = nstd*k*errorSigma
sigmaCentroid   = nstd*errorCentroid
sigmaResolution = resolution*np.sqrt(((sigmaCentroid/centroid)**2.0)+((
    sigmaFWHM/fwhm)**2.0))
return fwhm, sigmaFWHM, resolution, sigmaResolution

def findCalibration(cent,eCent):
ci              = 0.95
pp              = (1.0+ci)/2.0
nstd            = stats.norm.ppf(pp)
centroid        = cent
sigmaCentroid   = nstd*eCent
return centroid, sigmaCentroid

def writeWaves(nGEN,sGEN,nCH0,sCH0,nCH1,sCH1):

numbersGEN  = nGEN
stringsGEN  = sGEN
numbersCH0  = nCH0
stringsCH0  = sCH0
numbersCH1  = nCH1
stringsCH1  = sCH1

text_file  = open('waves.txt','w')
text_file.write('//' + '\n')
text_file.write('// GENERAL SETTINGS' + '\n')
text_file.write('// ________________' + '\n')
text_file.write('%d\t\t// number of folders\n' % numbersGEN[0])
text_file.write('%d\t\t// number of headers\n' % numbersGEN[1])
text_file.write('%d\t\t// number of points per pulse\n' % numbersGEN[2])
text_file.write('%d\t// number of pulses\n' % numbersGEN[3])
text_file.write('%d\t\t// method: 0 (read numPulses) 1 (get numPulses)\n' %
    numbersGEN[4])
text_file.write('%d\t\t// bin width (ns)\n' % numbersGEN[5])
text_file.write('%s\t\t// data file extension\n' % stringsGEN)
text_file.write('%d\t\t// number of vertical bits\n' % numbersGEN[6])
text_file.write('%d\t\t// dynamic range (V)\n' % numbersGEN[7])
text_file.write('%d\t\t// write clean pulses file\n' % numbersGEN[8])
text_file.write('%d\t\t// write clipped pulses file\n' % numbersGEN[9])
text_file.write('%d\t\t// write double pulses file\n' % numbersGEN[10])
text_file.write('%d\t\t// write pulses below threshold file\n' % numbersGEN
    [11])
text_file.write('%d\t\t// write results\n' % numbersGEN[12])
text_file.write('//' + '\n')
text_file.write('// CHANNEL 0 SETTINGS' + '\n')
text_file.write('// __________________' + '\n')
text_file.write('%d\t\t// amplitude setting 1\n' % numbersCH0[0])
text_file.write('%d\t\t// amplitude setting 2\n' % numbersCH0[1])
text_file.write('%d\t\t// double pulse setting 1\n' % numbersCH0[2])
text_file.write('%d\t\t// double pulse setting 2\n' % numbersCH0[3])
text_file.write('%s\t\t// pulse type (pos or neg)\n' % stringsCH0)
text_file.write('%0.1f\t\t// threshold (V)\n' % numbersCH0[4])
text_file.write('%d\t\t// number of points in baseline\n' % numbersCH0[5])
text_file.write('%d\t\t// double pulse cleaning\n' % numbersCH0[6])
text_file.write('%0.1f\t\t// double pulse window (left) threshold fraction\n'
   % numbersCH0[7])
text_file.write('%0.1f\t\t// double pulse window (right) threshold fraction\n'
    % numbersCH0[8])
text_file.write('%d\t\t// start total integral\n' % numbersCH0[9])
text_file.write('%d\t\t// start tail integral\n' % numbersCH0[10])
text_file.write('//' + '\n')
text_file.write('// CHANNEL 1 SETTINGS' + '\n')
text_file.write('// __________________' + '\n')
text_file.write('%d\t\t// amplitude setting 1\n' % numbersCH1[0])
text_file.write('%d\t\t// amplitude setting 2\n' % numbersCH1[1])
text_file.write('%d\t\t// double pulse setting 1\n' % numbersCH1[2])
text_file.write('%d\t\t// double pulse setting 2\n' % numbersCH1[3])
text_file.write('%s\t\t// pulse type (pos or neg)\n' % stringsCH1)
text_file.write('%0.1f\t\t// threshold (V)\n' % numbersCH1[4])
text_file.write('%d\t\t// number of points in baseline\n' % numbersCH1[5])
text_file.write('%d\t\t// double pulse cleaning\n' % numbersCH1[6])
text_file.write('%0.1f\t\t// double pulse window (left) threshold fraction\n'
   % numbersCH1[7])
text_file.write('%0.1f\t\t// double pulse window (right) threshold fraction\n'
    % numbersCH1[8])
text_file.write('%d\t\t// start total integral\n' % numbersCH1[9])
text_file.write('%d\t\t// start tail integral\n' % numbersCH1[10])
text_file.close()
```

A.2 Energy resolution

```
# Charles Sosa
# University of Michigan
# Created 04-03-2018
# Updated 05-20-2018

import sys
#sys.path.insert(0, 'path')
from library import *
t0 = time.clock()
src = os.getcwd()

# ------------------------------------
# data directories
# ------------------------------------

# 0  journal2/pmt2/EJ200CONGS_PMT2_N1425V_CS137s_10CM_15CM_5HR_180DEG
# 1  journal2/pmt2/EJ200CONPS_PMT2_N1425V_CS137s_10CM_15CM_5HR_180DEG
# 2  journal2/pmt2/EJ200CYLGS_PMT2_N1425V_CS137s_10CM_15CM_5HR_180DEG
# 3  journal2/pmt2/EJ200CYLPS_PMT2_N1425V_CS137s_10CM_15CM_5HR_180DEG
# 4  journal2/pmt2/map/
   EJ200CUBGS_PMT2_POS1_N1425V_CS137s_0CM_15CM_30MIN_180DEG
# 5  journal2/pmt2/map/
   EJ200CUBGS_PMT2_POS2_N1425V_CS137s_0CM_15CM_30MIN_180DEG
# 6  journal2/pmt2/map/
   EJ200CUBGS_PMT2_POS3_N1425V_CS137s_0CM_15CM_30MIN_180DEG
# 7  journal2/pmt2/map/
   EJ200CUBGS_PMT2_POS4_N1425V_CS137s_0CM_15CM_30MIN_180DEG
# 8  journal2/pmt2/map/
   EJ200CUBGS_PMT2_POS5_N1425V_CS137s_0CM_15CM_30MIN_180DEG
```

```
# 9   journal2/pmt2/map/
    EJ200CUBGS_PMT2_POS6_N1425V_CS137s_0CM_15CM_30MIN_180DEG
# 10  journal2/pmt2/map/
    EJ200CUBGS_PMT2_POS7_N1425V_CS137s_0CM_15CM_30MIN_180DEG
# 11  journal2/pmt2/EB/EJ200CONGS_PMT2EB_N1425V_CS137s_10CM_15CM_5HR_180DEG
# 12  journal2/pmt2/EB/EJ200CONPS_PMT2EB_N1425V_CS137s_10CM_15CM_5HR_180DEG
# 13  journal2/pmt2/EB/EJ200CYLGS_PMT2EB_N1425V_CS137s_10CM_15CM_5HR_180DEG
# 14  journal2/pmt2/EB/EJ200CYLPS_PMT2EB_N1425V_CS137s_10CM_15CM_5HR_180DEG
# 15  journal2/pmt3/EJ200CONGS_PMT3_N1830V_CS137s_10CM_15CM_5HR_180DEG
# 16  journal2/pmt3/EJ200CONPS_PMT3_N1830V_CS137s_10CM_15CM_5HR_180DEG
# 17  journal2/pmt3/EJ200CYLGS_PMT3_N1830V_CS137s_10CM_15CM_5HR_180DEG
# 18  journal2/pmt3/EJ200CYLPS_PMT3_N1830V_CS137s_10CM_15CM_5HR_180DEG
# 19  journal2/pmt3/map/
    EJ200CUBGS_PMT3_POS1_N1830V_CS137s_0CM_15CM_30MIN_180DEG
# 20  journal2/pmt3/map/
    EJ200CUBGS_PMT3_POS2_N1830V_CS137s_0CM_15CM_30MIN_180DEG
# 21  journal2/pmt3/map/
    EJ200CUBGS_PMT3_POS3_N1830V_CS137s_0CM_15CM_30MIN_180DEG
# 22  journal2/pmt3/map/
    EJ200CUBGS_PMT3_POS4_N1830V_CS137s_0CM_15CM_30MIN_180DEG
# 23  journal2/pmt3/map/
    EJ200CUBGS_PMT3_POS5_N1830V_CS137s_0CM_15CM_30MIN_180DEG
# 24  journal2/pmt3/map/
    EJ200CUBGS_PMT3_POS6_N1830V_CS137s_0CM_15CM_30MIN_180DEG
# 25  journal2/pmt3/map/
    EJ200CUBGS_PMT3_POS7_N1830V_CS137s_0CM_15CM_30MIN_180DEG
# 26  journal2/pmt5/EJ200CONGS_PMT5_N1700V_CS137s_10CM_15CM_5HR_180DEG
# 27  journal2/pmt5/EJ200CONPS_PMT5_N1700V_CS137s_10CM_15CM_5HR_180DEG
# 28  journal2/pmt5/EJ200CYLGS_PMT5_N1700V_CS137s_10CM_15CM_5HR_180DEG
# 29  journal2/pmt5/EJ200CYLPS_PMT5_N1700V_CS137s_10CM_15CM_5HR_180DEG
# 30  journal2/pmt5/map/
    EJ200CUBGS_POS1_PMT5_N1700V_CS137s_0CM_15CM_30MIN_180DEG
# 31  journal2/pmt5/map/
    EJ200CUBGS_POS2_PMT5_N1700V_CS137s_0CM_15CM_30MIN_180DEG
# 32  journal2/pmt5/map/
    EJ200CUBGS_POS3_PMT5_N1700V_CS137s_0CM_15CM_30MIN_180DEG
# 33  journal2/pmt5/map/
    EJ200CUBGS_POS4_PMT5_N1700V_CS137s_0CM_15CM_30MIN_180DEG
# 34  journal2/pmt5/map/
    EJ200CUBGS_POS5_PMT5_N1700V_CS137s_0CM_15CM_30MIN_180DEG
# 35  journal2/pmt5/map/
    EJ200CUBGS_POS6_PMT5_N1700V_CS137s_0CM_15CM_30MIN_180DEG
# 36  journal2/pmt5/map/
    EJ200CUBGS_POS7_PMT5_N1700V_CS137s_0CM_15CM_30MIN_180DEG
# 37  journal3/energy/ISBCONGS_PMT5_N1700V_CS137s_10CM_15CM_5HR_180DEG
# 38  journal3/energy/ISBCYLGS_PMT5_N1700V_CS137s_10CM_15CM_5HR_180DEG
# 39  supplemental/042217_EJ200CYL0D_ETLPMT0_N1150V_CS137s_30CM_2HR_180DEG
# 40  supplemental/042217_EJ200CYL0D_ETLPMT0_N1150V_CS137s_30CM_2HR_90DEG
# 41  supplemental/042217_EJ200CYL0D_ETLPMT0_N1150V_CS137s_30CM_2HR_45DEG

# ---------------------------------------
# choose directories
# ---------------------------------------

chooseDirectory = np.array([1,3,11,12,13,14,16,18,27,29])

for d in range(0,len(chooseDirectory)):
```

```
print(directoryResolution[chooseDirectory[d]])
os.chdir(directoryResolution[chooseDirectory[d]])

# ------------------------------------------------------------
# load data
# ------------------------------------------------------------

print('loading data...')

organicPathPulse    = 'data/pid0.dat'
organicPathTime     = 'data/gtime0new.dat'
organicFilePulse    = open(organicPathPulse,'rb')
organicFileTime     = open(organicPathTime,'rb')
organicPulse        = np.fromfile(organicFilePulse, dtype = np.float)
organicTime         = np.fromfile(organicFileTime,  dtype = np.uint64)

inorganicPathPulse  = 'data/pid1.dat'
inorganicPathTime   = 'data/gtime1new.dat'
inorganicFilePulse  = open(inorganicPathPulse,'rb')
inorganicFileTime   = open(inorganicPathTime,'rb')
inorganicPulse      = np.fromfile(inorganicFilePulse, dtype = np.float)
inorganicTime       = np.fromfile(inorganicFileTime,  dtype = np.uint64)

# ------------------------------------------------------------
# settings
# ------------------------------------------------------------

print('loading settings...')

settings                                    = loadtxt("resolution.txt",
    delimiter="#", unpack=False)
organicPulseBinStart        = settings[0]
organicPulseBinEnd          = settings[1]
organicPulseBinWidth        = settings[2]
organicPulseGateStart       = settings[3]
organicPulseGateEnd         = settings[4]
organicPulseContinuumStart  = settings[5]
organicPulseContinuumEnd    = settings[6]
inorganicPulseGatePeak      = settings[7]
inorganicPulseGateWidth     = settings[8]
inorganicPulseBinStart      = settings[9]
inorganicPulseBinEnd        = settings[10]
inorganicPulseBinWidth      = settings[11]
inorganicPulseGateLow       = inorganicPulseGatePeak - inorganicPulseGateWidth
    /2.0
inorganicPulseGateHigh      = inorganicPulseGatePeak + inorganicPulseGateWidth
    /2.0

# ------------------------------------------------------------
# process
# ------------------------------------------------------------

print('processing data...')

# histogram (organic)
organicPulseBinRange = np.arange(organicPulseBinStart,organicPulseBinEnd,
    organicPulseBinWidth)
organicPulseCounts, organicPulseBins = hist(organicPulse,organicPulseBinRange)
```

```
# histogram (inorganic)
inorganicPulseBinRange = np.arange(inorganicPulseBinStart,inorganicPulseBinEnd
    ,inorganicPulseBinWidth)
inorganicPulseCounts, inorganicPulseBins = hist(inorganicPulse,
    inorganicPulseBinRange)

# process time gate
organicTimeCheckBool  = np.in1d(organicTime,inorganicTime)
organicTimeCheckIndex = np.where(organicTimeCheckBool == True)[0]
organicTimeGateIndex  = np.array(organicTime[organicTimeCheckIndex])
organicTimeGateData   = np.array(organicPulse[organicTimeCheckIndex])

# process energy gate
inorganicEnergyCheckBool  = np.logical_and(inorganicPulseGateLow<
    inorganicPulse,inorganicPulse<inorganicPulseGateHigh)
inorganicEnergyCheckIndex = np.where(inorganicEnergyCheckBool == True)[0]
inorganicEnergyGateIndex  = np.array(inorganicTime[inorganicEnergyCheckIndex
    [0:inorganicEnergyCheckIndex.size-1]])
inorganicEnergyLinkBool   = np.in1d(organicTimeGateIndex,
    inorganicEnergyGateIndex)
inorganicEnergyLinkIndex  = np.where(inorganicEnergyLinkBool == True)[0]
organicPulseGate          = organicTimeGateData[inorganicEnergyLinkIndex]

# histogram gate
organicPulseGateCounts, organicPulseGateBins = hist(organicPulseGate,
    organicPulseBinRange)
organicPulseGateInterp         = interp1d(organicPulseGateBins,
    organicPulseGateCounts, kind = 'cubic')
organicPulseGateContinuum      = organicPulseGateInterp(np.arange(
    organicPulseContinuumStart,organicPulseContinuumEnd,organicPulseBinWidth))
organicPulseGateContinuumMean  = organicPulseGateContinuum.mean()
organicPulseContinuumX         = [organicPulseContinuumStart,
    organicPulseContinuumEnd]
organicPulseContinuumY         = [organicPulseGateContinuumMean,
    organicPulseGateContinuumMean]

# interpolation (accidentals)
organicPulseAccidentalsX      = [organicPulseGateStart, organicPulseGateEnd]
organicPulseAccidentalsY      = [organicPulseGateContinuumMean, 0.0]
organicPulseAccidentalsInterp = interp1d(organicPulseAccidentalsX,
    organicPulseAccidentalsY, kind = 'linear')

# accidentals subtraction
organicPulseGateRange  = np.arange(organicPulseGateStart,organicPulseGateEnd,
    organicPulseBinWidth)
organicPulseGateAccSub = organicPulseGateInterp(organicPulseGateRange) -
    organicPulseAccidentalsInterp(organicPulseGateRange)

# curve fit
gH,gC,gS          = singleGauss_guess(organicPulseGateRange,
    organicPulseGateAccSub)
H,eH,C,eC,S,eS = singleGauss_fit(organicPulseGateRange,organicPulseGateAccSub,
    g=[gH,gC,gS])

# get information
centroid,sigmaCentroid                      = findCalibration(C,eC)
fwhm,sigmaFWHM,resolution,sigmaResolution = findResolution(C,eC,S,eS)
```

```
# ------------------------------------------
# output
# ------------------------------------------

# print information
print('Total singles (organic)      = %5.1f' % organicPulse.size)
print('Total singles (inorganic)    = %5.1f' % inorganicPulse.size)
print('Total correlations           = %5.1f' % organicTimeGateIndex.size)
print('Clean correlations           = %5.1f' % inorganicEnergyGateIndex.size)
print('Calibration point (V*ns)     = %5.5f' % centroid)
print('Full width half max (V*ns)   = %5.5f' % fwhm)
print('Energy resolution (percent) = %5.5f' % resolution)
print('Centroid sigma (V*ns)        = %5.5f' % sigmaCentroid)
print('FWHM sigma (V*ns)            = %5.5f' % sigmaFWHM)
print('Resolution sigma (percent)   = %5.5f' % sigmaResolution)

# write information
result_file = open("output/outputResult.txt", "w")
result_file.write('Total singles (organic)      = %5.1f\n' % organicPulse.size)
result_file.write('Total singles (inorganic)    = %5.1f\n' % inorganicPulse.
    size)
result_file.write('Total correlations           = %5.1f\n' %
    organicTimeGateIndex.size)
result_file.write('Clean correlations           = %5.1f\n' %
    inorganicEnergyGateIndex.size)
result_file.write('Calibration point (V*ns)     = %5.5f\n' % centroid)
result_file.write('Full width half max (V*ns)   = %5.5f\n' % fwhm)
result_file.write('Energy resolution (percent) = %5.5f\n' % resolution)
result_file.write('Centroid sigma (V*ns)        = %5.5f\n' % sigmaCentroid)
result_file.write('FWHM sigma (V*ns)            = %5.5f\n' % sigmaFWHM)
result_file.write('Resolution sigma (percent)   = %5.5f\n' % sigmaResolution)
result_file.close()

# export data for uncertainty analysis
finerange = np.arange(organicPulseGateStart,organicPulseGateEnd,0.01)
np.savetxt('output/xPID.txt',            organicPulseBins,
                              delimiter=' ', fmt='%5.4f')
np.savetxt('output/yPID.txt',            organicPulseCounts,
                      delimiter=' ', fmt='%5.4f')
np.savetxt('output/xPIDgate.txt',    organicPulseGateBins,
                      delimiter=' ', fmt='%5.4f')
np.savetxt('output/yPIDgate.txt',        organicPulseGateCounts,
                      delimiter=' ', fmt='%5.4f')
np.savetxt('output/xPIDgateSub.txt', organicPulseGateRange,
              delimiter=' ', fmt='%5.4f')
np.savetxt('output/yPIDgateSub.txt', organicPulseGateAccSub,
              delimiter=' ', fmt='%5.4f')
np.savetxt('output/xPIDgateFit.txt', finerange,
                              delimiter=' ', fmt='%5.4f')
np.savetxt('output/yPIDgateFit.txt', singleGauss_func(finerange,*[H,C,S]),
    delimiter=' ', fmt='%5.4f')

print('Time passed = ', (time.clock() - t0)/60.0, " minutes")

# ------------------------------------------
# figures
# ------------------------------------------
# pulse integral spectrum (inorganicPulseGateWidthd)
plt.figure()
plt.errorbar(organicPulseGateBins , organicPulseGateCounts , yerr=np.sqrt(
    organicPulseGateCounts),fmt='-k',label='Gated PID')
plt.plot(organicPulseAccidentalsX , organicPulseAccidentalsY , '--ob',label='
    Accidentals')
plt.plot(finerange , singleGauss_func(finerange ,*[H,C,S]) , '-r',label='Gaussian
    fit',linewidth=3)
plt.xlim(0,organicPulseBinEnd)
plt.ylim(ymin=0)
plt.xticks()
plt.yticks()
plt.xlabel('Total integral (V*ns)')
plt.ylabel('Counts')
plt.legend(loc='upper left')
plt.savefig('images/resolution.png',bbox_inches='tight',dpi=dpiRes)
plt.clf()

# pulse integral spectrum (non-inorganicPulseGateWidthd)
plt.figure()
plt.errorbar(organicPulseBins , organicPulseCounts , yerr=np.sqrt(
    organicPulseCounts),fmt='-k',label='Non-gated PID')
plt.xlim(0,organicPulseBinEnd)
plt.ylim(ymin=0)
plt.xticks()
plt.yticks()
plt.xlabel('Total integral (V*ns)')
plt.ylabel('Counts')
plt.legend(loc='upper left')
plt.savefig('images/position.png',bbox_inches='tight',dpi=dpiRes)
plt.clf()

# pulse height spectrum (inorganic inorganicPulseGateWidthd)
plt.figure()
plt.errorbar(inorganicPulseBins,inorganicPulseCounts,yerr=np.sqrt(
    inorganicPulseCounts),fmt='-k',label='Time-correlated PHD')
plt.plot([inorganicPulseGatePeak-inorganicPulseGateWidth/2,
inorganicPulseGatePeak-inorganicPulseGateWidth/2],
[0, max(inorganicPulseCounts)],'--r',label='Enegy gate')
plt.plot([inorganicPulseGatePeak+inorganicPulseGateWidth/2,
inorganicPulseGatePeak+inorganicPulseGateWidth/2],
[0, max(inorganicPulseCounts)],'--r')
plt.xlim(0,inorganicPulseBinEnd)
plt.ylim(ymin=0)
plt.xlabel('Total integral (V*ns)')
plt.ylabel('Counts')
plt.legend(loc='upper right')
plt.savefig('images/gate.png',bbox_inches='tight',dpi=dpiRes)
plt.clf()

# go back to source
os.chdir(src)

# close all figures
plt.close('all')
```

A.3 Time resolution

```
# Charles Sosa
# University of Michigan
# Created 05-14-2018
# Updated 05-14-2018

import sys
from library import *
t0 = time.clock()

parent = os.getcwd()

# ---------------------------------------------
# data directories
# ---------------------------------------------

# 0 journal4/010418_EJ200CONPS_PMTC1_N1700V_PMTC2_N1310V_NA22s_22CM_5MIN
# 1 journal4/010418_EJ200CYLPS_PMTC1_N1700V_PMTC2_N1298V_NA22s_22CM_5MIN

# ---------------------------------------------
# choose directories
# ---------------------------------------------

t0 = time.clock()

# ---------------------------------------------
# functions
# ---------------------------------------------

def gaus(x,a,x0,sigma):
return a*exp(-(x-x0)**2/(2*sigma**2))

# ---------------------------------------------
# main
# ---------------------------------------------

chooseDirectory = np.array([0,1])

for d in range(0,len(chooseDirectory)):

print('DIRECTORY')
print(directoryTIME[chooseDirectory[d]])
os.chdir(directoryTIME[chooseDirectory[d]])
src = os.getcwd()

print('loading settings...')

# ---------------------------------------------
# main settings
# ---------------------------------------------
checkRise     = 1
DR            = 2.0
numBits       = 14
numPoints     = 100

numBase       = 10
numBoots      = 100
```

```
numVBins    = np.power(DR,numBits)
threshold   = 0.1  # V
timeBinWidth = 0.01 # ps
timeStep    = 2.0  # ns
timeRange   = np.arange(0,numPoints*timeStep,timeStep)
pointsDomain = np.arange(0,int(numPoints))

if(chooseDirectory[d]==0):

# -----------------------------------------
# settings (cone)
# -----------------------------------------
winLeft      = 15 # V*ns
winRight     = 30 # V*ns
winBottom0   = 13.51-(5.0*1.12) # V*ns
winTop0      = 13.51+(1.0*1.12) # V*ns
winBottom1   = 13.23-(5.0*1.14) # V*ns
winTop1      = 13.23+(1.0*1.14) # V*ns

if(chooseDirectory[d]==1):

# -----------------------------------------
# settings (cylinder)
# -----------------------------------------
winLeft      = 15 # V*ns
winRight     = 30 # V*ns
winBottom0         = 11.80-(5.0*0.97) #11.821 #10.66-1.12 V*ns new! 29 to - 5*29
    keVee
winTop0            = 11.80+(1.0*0.97) #11.821 #10.66+1.12 V*ns
winBottom1         = 11.67-(5.0*0.99) #11.698 #10.89-1.14 V*ns
winTop1            = 11.67+(1.0*0.99) #11.698 #10.89+1.14 V*ns

# -----------------------------------------
# load data
# -----------------------------------------

print('loading data...')

dataType = np.dtype([('header1','u4'),
('header2','u4'),
('header3','u4'),
('header4','u4'),
('header5','u4'),
('header6','u4'),
('pulse','u2',(numPoints,))])
dataRaw0 = np.fromfile(open('data/1/wave0.dat','rb'), dtype = dataType)
dataRaw1 = np.fromfile(open('data/1/wave1.dat','rb'), dtype = dataType)

# -----------------------------------------
# process
# -----------------------------------------

print('processing data (correlations)...')

#numPulses   = dataRaw1.size
numPulses   = 6449280 #3000000
targetClean = 29620   #11152
```

```
# ---------------------------------------------
# looping through CFD
# ---------------------------------------------

#cfdloop    = np.array([0.005, 0.010, 0.050, 0.100, 0.150, 0.200, 0.250,
    0.300, 0.350, 0.400, 0.450, 0.500])
#cfdloop    = np.array([0.005, 0.006, 0.007, 0.008, 0.009, 0.010])
#cfdloop    = np.array([0.050, 0.100, 0.150, 0.200, 0.250, 0.300, 0.350,
    0.400, 0.450, 0.500])
#cfdloop    = np.array([0.001, 0.002, 0.003, 0.004, 0.005, 0.010,
#0.020, 0.030, 0.040, 0.050, 0.075, 0.100,
#0.125, 0.150, 0.175, 0.200, 0.225, 0.250,
#0.275, 0.300, 0.325, 0.350, 0.375, 0.400,
#0.425, 0.450, 0.475, 0.500, 0.600, 0.700])
cfdloop    = np.array([0.500])

FWHMloop0  = np.array([])
FWHMloop1  = np.array([])
sFWHMloop0 = np.array([])
sFWHMloop1 = np.array([])

for c in range(0,len(cfdloop)):
# ---------------------------------------------
# re-initialize settings
# ---------------------------------------------
countPulse              = 0
countClipped            = 0
countBelow              = 0
countPH                 = 0
countClean              = 0
count_cleanLRT0             = 0
count_cleanMRT0             = 0
count_cleanHRT0             = 0
count_cleanLRT1             = 0
count_cleanMRT1             = 0
count_cleanHRT1             = 0
countOutsideTimeGate    = 0
countOutsideEnergyGate  = 0
clean0                              = np.array([])
clean1                              = np.array([])
cleanLRT0               = np.array([])
cleanHRT0               = np.array([])
cleanMRT0               = np.array([])
cleanMRT1               = np.array([])
cleanLRT1               = np.array([])
cleanHRT1               = np.array([])
timeDelta               = np.array([])
timeRise0               = np.array([])
timeRise1               = np.array([])
pid0                    = np.array([])
pid1                    = np.array([])
# ---------------------------------------------
# enter primary loop
# ---------------------------------------------
CFD = cfdloop[c]
for p in np.arange(0,numPulses):
countPulse = countPulse + 1
if (dataRaw0[p][6].min() == 0 or dataRaw1[p][6].min() == 0):
```

```
countClipped = countClipped + 1
else:
pulse0    = np.array((DR/numVBins)*(-1.0*dataRaw0[p][6] + numVBins))
pulse1    = np.array((DR/numVBins)*(-1.0*dataRaw1[p][6] + numVBins))
base0     = np.mean(pulse0[0:numBase])
base1     = np.mean(pulse1[0:numBase])
pulse0    = pulse0 - base0
pulse1    = pulse1 - base1
peakAmp0 = pulse0.max()
peakAmp1 = pulse1.max()
if (np.logical_or(peakAmp0 < threshold, peakAmp1 < threshold)==True):
countBelow = countBelow + 1
else:
peakLoc0         = pulse0.argmax(axis=0)
peakLoc1         = pulse1.argmax(axis=0)
pulseIntegral0  = timeStep*np.sum(pulse0[peakLoc0-4:100])
pulseIntegral1  = timeStep*np.sum(pulse1[peakLoc1-4:100])
pid0             = np.append(pid0, pulseIntegral0)
pid1             = np.append(pid1, pulseIntegral1)
if (np.logical_and(peakLoc0>winLeft,peakLoc0<winRight)==False or
np.logical_and(peakLoc1>winLeft,peakLoc1<winRight)==False):
countOutsideTimeGate = countOutsideTimeGate + 1
else:
if (np.logical_and(pulseIntegral0>winBottom0,pulseIntegral0<winTop0)==False or
np.logical_and(pulseIntegral1>winBottom1,pulseIntegral1<winTop1)==False):
countOutsideEnergyGate = countOutsideEnergyGate + 1
else:
countClean   = countClean + 1
cleanLoc0    = peakLoc0
cleanLoc1    = peakLoc1
cleanAmp0    = peakAmp0
cleanAmp1    = peakAmp1
cleanPulse0  = pulse0
cleanPulse1  = pulse1
clean0       = np.append(clean0, cleanPulse0)
clean1       = np.append(clean1, cleanPulse1)
peakTime0    = timeRange[cleanLoc0]
peakTime1    = timeRange[cleanLoc1]
# find the constant fraction amplitudes
startAmp0    = CFD*cleanAmp0
startAmp1    = CFD*cleanAmp1
# find the rise time amplitudes
ampBot0          = 0.10*cleanAmp0
ampTop0          = 0.90*cleanAmp0
ampBot1          = 0.10*cleanAmp1
ampTop1          = 0.90*cleanAmp1
# interpolate the inverted pulse to find the time from the CFD amplitudes
rise0ir      = interp1d(cleanPulse0[0:cleanLoc0+1],timeRange[0:cleanLoc0+1],
   kind = 'linear')
rise1ir      = interp1d(cleanPulse1[0:cleanLoc1+1],timeRange[0:cleanLoc1+1],
   kind = 'linear')
# find the times (CFD)
startLoc0    = rise0ir(startAmp0)
startLoc1    = rise1ir(startAmp1)
# find the change in time (delta T)
timeDelta    = np.append(timeDelta, startLoc1-startLoc0)
# find the times (rise time)
riseBot0         = rise0ir(ampBot0)
```

```
riseTop0        = rise0ir(ampTop0)
riseBot1        = rise1ir(ampBot1)
riseTop1        = rise1ir(ampTop1)
if (checkRise == 1):
# find the change in time (rise time)
timeRise0       = np.append(timeRise0, riseTop0 - riseBot0)
timeRise1       = np.append(timeRise1, riseTop1 - riseBot1)
# discriminate rise times
# GEOMETRY: CONE
if(chooseDirectory[d]==0):
# CHANNEL0 LOW
if (np.logical_and(riseTop0 - riseBot0 > 3.45,riseTop0 - riseBot0 < 3.55)==
    True):
count_cleanLRT0 = count_cleanLRT0 + 1
if(count_cleanLRT0<1001):
cleanLRT0 = np.append(cleanLRT0, cleanPulse0)
# CHANNEL0 MED
if (np.logical_and(riseTop0 - riseBot0 > 3.90,riseTop0 - riseBot0 < 4.00)==
    True):
count_cleanMRT0 = count_cleanMRT0 + 1
if(count_cleanMRT0<1001):
cleanMRT0 = np.append(cleanMRT0, cleanPulse0)
# CHANNEL0 HIGH
if (np.logical_and(riseTop0 - riseBot0 > 4.25,riseTop0 - riseBot0 < 4.35)==
    True):
count_cleanHRT0 = count_cleanHRT0 + 1
if(count_cleanHRT0<1001):
cleanHRT0 = np.append(cleanHRT0, cleanPulse0)
# CHANNEL1 LOW
if (np.logical_and(riseTop1 - riseBot1 > 3.45,riseTop1 - riseBot1 < 3.55)==
    True):
count_cleanLRT1 = count_cleanLRT1 + 1
if(count_cleanLRT1<1001):
cleanLRT1 = np.append(cleanLRT1, cleanPulse1)
# CHANNEL1 MED
if (np.logical_and(riseTop1 - riseBot1 > 3.90,riseTop1 - riseBot1 < 4.00)==
    True):
count_cleanMRT1 = count_cleanMRT1 + 1
if(count_cleanMRT1<1001):
cleanMRT1 = np.append(cleanMRT1, cleanPulse1)
# CHANNEL1 HIGH
if (np.logical_and(riseTop1 - riseBot1 > 4.25,riseTop1 - riseBot1 < 4.35)==
    True):
count_cleanHRT1 = count_cleanHRT1 + 1
if(count_cleanHRT1<1001):
cleanHRT1 = np.append(cleanHRT1, cleanPulse1)
# GEOMETRY: CYLINDER
if(chooseDirectory[d]==1):
# CHANNEL0 LOW
if (np.logical_and(riseTop0 - riseBot0 > 3.60,riseTop0 - riseBot0 < 3.70)==
    True):
count_cleanLRT0 = count_cleanLRT0 + 1
if(count_cleanLRT0<1001):
cleanLRT0 = np.append(cleanLRT0, cleanPulse0)
# CHANNEL0 MED
if (np.logical_and(riseTop0 - riseBot0 > 3.90,riseTop0 - riseBot0 < 4.00)==
    True):
count_cleanMRT0 = count_cleanMRT0 + 1
```

```
if(count_cleanMRT0<1001):
cleanMRT0 = np.append(cleanMRT0, cleanPulse0)
# CHANNEL0 HIGH
if (np.logical_and(riseTop0 - riseBot0 > 4.45,riseTop0 - riseBot0 < 4.55)==
    True):
count_cleanHRT0 = count_cleanHRT0 + 1
if(count_cleanHRT0<1001):
cleanHRT0 = np.append(cleanHRT0, cleanPulse0)
# CHANNEL1 LOW
if (np.logical_and(riseTop1 - riseBot1 > 3.60,riseTop1 - riseBot1 < 3.70)==
    True):
count_cleanLRT1 = count_cleanLRT1 + 1
if(count_cleanLRT1<1001):
cleanLRT1 = np.append(cleanLRT1, cleanPulse1)
# CHANNEL1 MED
if (np.logical_and(riseTop1 - riseBot1 > 3.90,riseTop1 - riseBot1 < 4.00)==
    True):
count_cleanMRT1 = count_cleanMRT1 + 1
if(count_cleanMRT1<1001):
cleanMRT1 = np.append(cleanMRT1, cleanPulse1)
# CHANNEL1 HIGH
if (np.logical_and(riseTop1 - riseBot1 > 4.45,riseTop1 - riseBot1 < 4.55)==
    True):
count_cleanHRT1 = count_cleanHRT1 + 1
if(count_cleanHRT1<1001):
cleanHRT1 = np.append(cleanHRT1, cleanPulse1)
#exit loop
if (np.logical_and(chooseDirectory[d]==1,countClean==targetClean)==True):
break
if (np.logical_and(chooseDirectory[d]==1,countClean==targetClean)==True):
break
if (np.logical_and(chooseDirectory[d]==1,countClean==targetClean)==True):
break
if (np.logical_and(chooseDirectory[d]==1,countClean==targetClean)==True):
break
if (np.logical_and(chooseDirectory[d]==1,countClean==targetClean)==True):
break

# ---------------------------------------------
# results
# ---------------------------------------------

timeDeltaCountsFirst, timeDeltaBinsFirst = np.histogram(timeDelta,np.arange
    (0,10,timeBinWidth))
timeDeltaBinsFirst                       = timeDeltaBinsFirst[0:
    timeDeltaCountsFirst.size]
timeDeltaCountsFirst                     = timeDeltaCountsFirst

timeRise0CountsFirst, timeRise0BinsFirst = np.histogram(timeRise0,np.arange
    (0,10,timeBinWidth))
timeRise0BinsFirst                       = timeRise0BinsFirst[0:
    timeRise0CountsFirst.size]
timeRise0CountsFirst                     = timeRise0CountsFirst

timeRise1CountsFirst, timeRise1BinsFirst = np.histogram(timeRise1,np.arange
    (0,10,timeBinWidth))
timeRise1BinsFirst                       = timeRise1BinsFirst[0:
    timeRise1CountsFirst.size]
```

```
timeRise1CountsFirst                    = timeRise1CountsFirst

pidBins             = np.arange(0,30,0.1)
pid0Counts, pid0Bins = np.histogram(pid0,pidBins)
pid0Bins            = pid0Bins[0:pid0Counts.size]
pid0Counts          = pid0Counts
pid1Counts, pid1Bins = np.histogram(pid1,pidBins)
pid1Bins            = pid1Bins[0:pid1Counts.size]
pid1Counts          = pid1Counts

# ----------------------------------
# print data
# ----------------------------------

if (np.logical_and(CFD>0.000,CFD<0.010)==True):
outputCFDPath = 'outputCFD' + format(cfdloop[c], '#.0')
if (np.logical_and(CFD>=0.010,CFD<0.100)==True):
outputCFDPath = 'outputCFD' + format(cfdloop[c], '#.2')
if (CFD>=0.10):
outputCFDPath = 'outputCFD' + format(cfdloop[c], '#.3')

if not os.path.exists(outputCFDPath):
os.mkdir(outputCFDPath)

os.chdir(outputCFDPath)
np.savetxt('pid0.txt', pid0, delimiter=' ', fmt='%5.4f')
np.savetxt('pid1.txt', pid1, delimiter=' ', fmt='%5.4f')
np.savetxt('pid0X.txt', pid0Bins, delimiter=' ', fmt='%5.4f')
np.savetxt('pid0Y.txt', pid0Counts, delimiter=' ', fmt='%5.4f')
np.savetxt('pid1X.txt', pid1Bins, delimiter=' ', fmt='%5.4f')
np.savetxt('pid1Y.txt', pid1Counts, delimiter=' ', fmt='%5.4f')
np.savetxt('clean0.txt', clean0, delimiter=' ', fmt='%5.4f')
np.savetxt('clean1.txt', clean1, delimiter=' ', fmt='%5.4f')
np.savetxt('cleanLRT0.txt', cleanLRT0, delimiter=' ', fmt='%5.4f')
np.savetxt('cleanMRT0.txt', cleanMRT0, delimiter=' ', fmt='%5.4f')
np.savetxt('cleanHRT0.txt', cleanHRT0, delimiter=' ', fmt='%5.4f')
np.savetxt('cleanLRT1.txt', cleanLRT1, delimiter=' ', fmt='%5.4f')
np.savetxt('cleanMRT1.txt', cleanMRT1, delimiter=' ', fmt='%5.4f')
np.savetxt('cleanHRT1.txt', cleanHRT1, delimiter=' ', fmt='%5.4f')
np.savetxt('deltaTimeX.txt', timeDeltaBinsFirst, delimiter=' ', fmt='%5.4f')
np.savetxt('deltaTimeY.txt', timeDeltaCountsFirst, delimiter=' ', fmt='%5.4f')

# ----------------------------------
# process
# ----------------------------------

# cross correlation distribution
timeDeltaCounts = timeDeltaCountsFirst
timeDeltaBins   = timeDeltaBinsFirst
timeDeltaMean   = timeDeltaBins[timeDeltaCounts.argmax(axis=0)]
timeDeltaStart  = timeDeltaMean -2.0
timeDeltaStop   = timeDeltaMean+2.0
timeDeltaRange  = np.arange(timeDeltaStart,timeDeltaStop,0.01)
timeDeltaInterp = interp1d(timeDeltaBins,timeDeltaCounts, kind = 'cubic')
finerange               = np.arange(timeDeltaStart,timeDeltaStop,0.005)

# curve fit
gH,gC,gS        = singleGauss_guess(timeDeltaRange,timeDeltaInterp(
    timeDeltaRange))
H,eH,C,eC,S,eS = singleGauss_fit(timeDeltaRange,timeDeltaInterp(timeDeltaRange
    ),g=[gH,gC,gS])

# get information
centroid,sigmaCentroid                     = findCalibration(C,eC)
fwhm,sigmaFWHM,resolution,sigmaResolution = findResolution(C,eC,S,eS)

# ----------------------------------------
# output
# ----------------------------------------

# print information
print('Constant fraction value     = %5.3f' % CFD)
print('Count clipped pulses        = %5.1f' % countClipped)
print('Below threshold count      = %5.1f' % countBelow)
print('Outside time gate count    = %3.1f' % countOutsideTimeGate)
print('Outside energy gate count  = %3.1f' % countOutsideEnergyGate)
print('Clean correlation count    = %5.1f' % countClean)
print('Calibration point (ns)     = %5.5f' % centroid)
print('Full width half max (ns)   = %5.5f' % fwhm)
print('Time resolution (percent)  = %5.5f' % resolution)
print('Centroid sigma (ns)        = %5.5f' % sigmaCentroid)
print('FWHM sigma (ns)            = %5.5f' % sigmaFWHM)
print('Resolution sigma (percent) = %5.5f' % sigmaResolution)
if (checkRise == 1):
print('Ch0 mean rise time (ns)    = %5.5f' % timeRise0.mean())
print('Ch1 mean rise time (ns)    = %5.5f' % timeRise1.mean())
print('Time passed = ', (time.clock() - t0)/60.0, " minutes")

if (chooseDirectory[d]==0):
FWHMloop0  = np.append(FWHMloop0, fwhm)
sFWHMloop0 = np.append(sFWHMloop0, sigmaFWHM)

if (chooseDirectory[d]==1):
FWHMloop1 = np.append(FWHMloop1, fwhm)
sFWHMloop1 = np.append(sFWHMloop1, sigmaFWHM)

# write information
result_file = open("outputResult.txt", "w")
result_file.write('Constant fraction value          = %5.5f\n' % CFD)
result_file.write('Count clipped pulses             = %5.1f\n' % countClipped)
result_file.write('Below threshold count        = %5.1f\n' % countBelow)
result_file.write('Outside time gate count      = %3.1f\n' %
    countOutsideTimeGate)
result_file.write('Outside energy gate count    = %3.1f\n' %
    countOutsideEnergyGate)
result_file.write('Clean correlation count      = %5.1f\n' % countClean)
result_file.write('Calibration point (ns)       = %5.5f\n' % centroid)
result_file.write('Full width half max (ns)     = %5.5f\n' % fwhm)
result_file.write('Energy resolution (percent) = %5.5f\n' % resolution)
result_file.write('Centroid sigma (ns)          = %5.5f\n' % sigmaCentroid)
result_file.write('FWHM sigma (ns)              = %5.5f\n' % sigmaFWHM)
result_file.write('Resolution sigma (percent)   = %5.5f\n' % sigmaResolution)
if (checkRise == 1):
result_file.write('Ch0 mean rise time (ns)      = %5.5f\n' % timeRise0.mean())
result_file.write('Ch1 mean rise time (ns)      = %5.5f\n' % timeRise1.mean())
result_file.close()
```

```
# ----------------------------------------------------
# figures
# ----------------------------------------------------

print('re-shaping pulses...')

data_clean0    = np.reshape(clean0,(int(countClean),int(numPoints)))
data_clean1    = np.reshape(clean1,(int(countClean),int(numPoints)))
data_clean0    = data_clean0.T
data_clean1    = data_clean1.T
#if (checkRise==1):
#       data_cleanLRT0 = np.reshape(cleanLRT0,(int(count_cleanLRT0),int(
    numPoints)))
#       data_cleanMRT0 = np.reshape(cleanMRT0,(int(count_cleanMRT0),int(
    numPoints)))
#       data_cleanHRT0 = np.reshape(cleanHRT0,(int(count_cleanHRT0),int(
    numPoints)))
#       data_cleanLRT0 = data_cleanLRT0.T
#       data_cleanMRT0 = data_cleanMRT0.T
#       data_cleanHRT0 = data_cleanHRT0.T
#       data_cleanLRT1 = np.reshape(cleanLRT1,(int(count_cleanLRT1),int(
    numPoints)))
#       data_cleanMRT1 = np.reshape(cleanMRT1,(int(count_cleanMRT1),int(
    numPoints)))
#       data_cleanHRT1 = np.reshape(cleanHRT1,(int(count_cleanHRT1),int(
    numPoints)))
#       data_cleanLRT1 = data_cleanLRT1.T
#       data_cleanMRT1 = data_cleanMRT1.T
#       data_cleanHRT1 = data_cleanHRT1.T

print('saving data...')

#if (data_clean0.size>0):
#       plt.figure()
#       plt.plot(pointsDomain,data_clean0)
#       plt.xlim(0,numPoints)
#       plt.ylim(0,DR)
#       plt.xlabel('Data point')
#       plt.ylabel('Amplitude (V)')
#       plt.savefig('clean0.png',bbox_inches='tight',dpi=300)
#       plt.clf()
#if (data_clean1.size>0):
#       plt.figure()
#       plt.plot(pointsDomain,data_clean1)
#       plt.xlim(0,numPoints)
#       plt.ylim(0,DR)
#       plt.xlabel('Data point')
#       plt.ylabel('Amplitude (V)')
#       plt.savefig('clean1.png',bbox_inches='tight',dpi=300)
#       plt.clf()

np.savetxt('ccdx.txt', timeDeltaBinsFirst, delimiter=' ', fmt='%5.4f')
np.savetxt('ccdy.txt', timeDeltaCountsFirst, delimiter=' ', fmt='%5.4f')
np.savetxt('ccde.txt', np.sqrt(timeDeltaCountsFirst), delimiter=' ', fmt='%5.4
    f')
np.savetxt('ccfx.txt', finerange, delimiter=' ', fmt='%5.4f')
np.savetxt('ccfy.txt', gaus(finerange,*[H,C,S]), delimiter=' ', fmt='%5.4f')
```

```
if (checkRise==1):
np.savetxt('rt0x.txt', timeRise0BinsFirst, delimiter=' ', fmt='%5.4f')
np.savetxt('rt0y.txt', timeRise0CountsFirst, delimiter=' ', fmt='%5.4f')
np.savetxt('rt0e.txt', np.sqrt(timeRise0CountsFirst), delimiter=' ', fmt='%5.4f')
np.savetxt('rt1x.txt', timeRise1BinsFirst, delimiter=' ', fmt='%5.4f')
np.savetxt('rt1y.txt', timeRise1CountsFirst, delimiter=' ', fmt='%5.4f')
np.savetxt('rt1e.txt', np.sqrt(timeRise1CountsFirst), delimiter=' ', fmt='%5.4f')

print('plotting data...')

plt.figure()
plt.errorbar(timeDeltaBinsFirst,timeDeltaCountsFirst,yerr=np.sqrt(timeDeltaCountsFirst),fmt='ok',label='data')
plt.plot(finerange,gaus(finerange,*[H,C,S]),'-r',linewidth=2,label='fit')
plt.xlim(2,6)
plt.xlabel('Time (ns)')
plt.ylabel('Counts')
plt.legend(loc='upper right',fontsize=10,numpoints=1)
plt.savefig('cc.png',bbox_inches='tight',dpi=300)
plt.clf()

plt.figure()
plt.errorbar(pid0Bins,pid0Counts,yerr=np.sqrt(pid0Counts),fmt='-r',label='Ch0')
plt.errorbar(pid1Bins,pid1Counts,yerr=np.sqrt(pid1Counts),fmt='-b',label='Ch1')
plt.xlim(0,20)
plt.xlabel('Integral (V*ns)')
plt.ylabel('Counts')
plt.legend(loc='upper right',fontsize=10,numpoints=1)
plt.savefig('pid.png',bbox_inches='tight',dpi=300)
plt.clf()

if (checkRise==1):
plt.figure()
plt.errorbar(timeRise0BinsFirst,timeRise0CountsFirst,yerr=np.sqrt(timeRise0CountsFirst),fmt='ok',label='ch0')
plt.errorbar(timeRise1BinsFirst,timeRise1CountsFirst,yerr=np.sqrt(timeRise1CountsFirst),fmt='or',label='ch1')
plt.xlim(2.5,5.5)
plt.xlabel('Rise time (ns)')
plt.ylabel('Counts')
plt.legend(loc='upper right',fontsize=10,numpoints=1)
plt.savefig('rt.png',bbox_inches='tight',dpi=300)
plt.clf()

# ---------------------------------------------------
# revert to source
# ---------------------------------------------------

os.chdir('..')

# ---------------------------------------------------
# CFD plot
# ---------------------------------------------------
```

```
if (chooseDirectory[d]==0):

np.savetxt('cfd0x.txt', cfdloop, delimiter=' ', fmt='%5.4f')
np.savetxt('cfd0y.txt', 1000.0*FWHMloop0, delimiter=' ', fmt='%5.4f')
np.savetxt('cfd0e.txt', 1000.0*sFWHMloop0, delimiter=' ', fmt='%5.4f')

plt.figure()
plt.errorbar(cfdloop,1000.0*FWHMloop0,yerr=1000.0*sFWHMloop0,fmt='-ok',
    markersize=4,label='0')
plt.xlabel('CFD (%)')
plt.ylabel('FWHM (ps)')
plt.xlim([0.0,0.60])
plt.ylim([0,900])
plt.legend(loc='upper right',fontsize=10,numpoints=1)
plt.savefig('cfd0.png',bbox_inches='tight',dpi=300)
plt.clf()

if (chooseDirectory[d]==1):

np.savetxt('cfd1x.txt', cfdloop, delimiter=' ', fmt='%5.4f')
np.savetxt('cfd1y.txt', 1000.0*FWHMloop1, delimiter=' ', fmt='%5.4f')
np.savetxt('cfd1e.txt', 1000.0*sFWHMloop1, delimiter=' ', fmt='%5.4f')

plt.figure()
plt.errorbar(cfdloop,1000.0*FWHMloop1,yerr=1000.0*sFWHMloop1,fmt='-ok',
    markersize=4,label='1')
plt.xlabel('CFD (%)')
plt.ylabel('FWHM (ps)')
plt.xlim([0.0,0.60])
plt.ylim([0,900])
plt.legend(loc='upper right',fontsize=10,numpoints=1)
plt.savefig('cfd1.png',bbox_inches='tight',dpi=300)
plt.clf()

os.chdir(parent)

print('done')

# close all figures
plt.close('all')

# ----------------------------------------
# End of program
# ----------------------------------------
```

APPENDIX B

Source code: Geant4 model

B.1 Action initialization

```
#include "lightTrackActionInitialization.hh"

//
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
// Initialize user actions
//
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

lightTrackActionInitialization::lightTrackActionInitialization():
   G4VUserActionInitialization() {}

lightTrackActionInitialization::~lightTrackActionInitialization() {}

void lightTrackActionInitialization::BuildForMaster() const {}

void lightTrackActionInitialization::Build() const {
    SetUserAction(new lightTrackPrimaryGeneratorAction());
    SetUserAction(new lightTrackEventAction());
    SetUserAction(new lightTrackSteppingAction());
}
```

B.2 Event action

```
#include "lightTrackEventAction.hh"

lightTrackEventAction::lightTrackEventAction():G4UserEventAction(),
organicHCID(-1),
photocathode1HCID(-1)
{
    dataFile5.open("HPE1.dat",          std::ios::binary|std::ios::ate);
    dataFile6.open("ERG.dat",           std::ios::binary|std::ios::ate);
```

```
}

lightTrackEventAction::~lightTrackEventAction() {
    dataFile5.close();
    dataFile6.close();
}

void lightTrackEventAction::BeginOfEventAction(const G4Event*) {
    G4SDManager* SDman = G4SDManager::GetSDMpointer();
    if (organicHCID<0)        {organicHCID        = SDman->GetCollectionID("
       organicSD/organicHitCollection");}
    if (photocathode1HCID<0) {photocathode1HCID = SDman->GetCollectionID("
       photocathode1SD/photocathode1HitCollection");}
}

void lightTrackEventAction::EndOfEventAction(const G4Event* event) {
    G4HCofThisEvent                      * hitsCE            = event->
       GetHCofThisEvent();
    lightTrackOrganicHitsCollection      * organicHC         = 0;
    lightTrackPhotocathode1HitsCollection * photocathode1HC = 0;
    if(hitsCE) {
        if(organicHCID>=0)organicHC                = (
           lightTrackOrganicHitsCollection*)(hitsCE->GetHC(organicHCID));
    }
    if(hitsCE != nullptr) {
        photocathode1HC = (lightTrackPhotocathode1HitsCollection*)(hitsCE->
           GetHC(photocathode1HCID));
    }
    struct photocathode1HITS {G4int a;} photocathode1Hits;
    photocathode1Hits.a = photocathode1HC->GetSize();
    if(organicHC) {
        int n_organicHit = organicHC->entries();
        if (n_organicHit > 0) {
            if(photocathode1HCID>0) {
                //G4cout << "HITS[organic] = " << photocathode1Hits.a <<
                   G4endl;
                dataFile5.write(reinterpret_cast<char *>(&photocathode1Hits),
                   sizeof(photocathode1Hits));
                G4double edep = 0;
                G4double etot = 0;
                for(int i=0;i<n_organicHit;i++){ //gather info on hits in
                   scintillator
                    edep=(*organicHC)[i]->GetEdep();
                    etot = etot + edep;
                }
                struct ERG {
                    G4double a;
                } erg;
                erg.a = etot;
                //G4cout << "EDEP[organic] = " << erg.a << G4endl;
                dataFile6.write(reinterpret_cast<char *>(&erg), sizeof(erg));
                //G4UImanager* UImanager = G4UImanager::GetUIpointer();
                //UImanager -> ApplyCommand("/vis/viewer/clearTransients");
            }
        }
    }
}
```

B.3 Stepping action

```
#include "lightTrackSteppingAction.hh"

lightTrackSteppingAction::lightTrackSteppingAction(): fOneStepPrimaries(false)
     {
     dataFile1.open("TOA.dat", std::ios::binary|std::ios::ate); // detected
         photon time of arrival
     dataFile2.open("NOR.dat", std::ios::binary|std::ios::ate); // detected
         photon reflections
     dataFile3.open("HLX.dat", std::ios::binary|std::ios::ate); // detected
         photon hit location on PMT
     dataFile4.open("HLY.dat", std::ios::binary|std::ios::ate); // detected
         photon hit location on PMT
     dataFile5.open("CSX.dat", std::ios::binary|std::ios::ate); // compton
         scatter location (x)
     dataFile6.open("CSY.dat", std::ios::binary|std::ios::ate); // compton
         scatter location (y)
     dataFile7.open("CSZ.dat", std::ios::binary|std::ios::ate); // compton
         scatter location (z)
     fExpectedNextStatus = Undefined;
     fOpProcess = NULL;
}

lightTrackSteppingAction::~lightTrackSteppingAction() {
     dataFile1.close();
     dataFile2.close();
     dataFile3.close();
     dataFile4.close();
     dataFile5.close();
     dataFile6.close();
     dataFile7.close();
}


void lightTrackSteppingAction::UserSteppingAction(const G4Step * theStep) {

     G4Track* theTrack = theStep->GetTrack();
     if ( theTrack->GetCurrentStepNumber() == 1 ) fExpectedNextStatus =
         Undefined;
     G4StepPoint* thePostPoint = theStep->GetPostStepPoint();
     G4OpBoundaryProcessStatus boundaryStatus=Undefined;
     static G4ThreadLocal G4OpBoundaryProcess* boundary=NULL;
     G4ProcessManager* pm = theStep->GetTrack()->GetDefinition()->
         GetProcessManager();
     G4int nprocesses = pm->GetProcessListLength();
     G4ProcessVector* pv = pm->GetProcessList();
     G4int i;
     for( i=0;i<nprocesses;i++){
          if((*pv)[i]->GetProcessName()=="OpBoundary"){
               boundary = (G4OpBoundaryProcess*)(*pv)[i];
               //G4cout << boundary->GetStatus() << G4endl;
               break;
          }
     }
     G4ParticleDefinition* particleType = theTrack->GetDefinition();
     if(particleType==G4Gamma::GammaDefinition()) {
```

```
    if(thePostPoint->GetProcessDefinedStep()->GetProcessName()=="compt") {
        if (thePostPoint->GetPhysicalVolume()->GetName()=="organic") {
            G4ThreeVector comptonPos = thePostPoint->GetPosition();
            struct CSX {
                G4double a;
            } csx;
            csx.a = comptonPos(0);
            dataFile5.write(reinterpret_cast<char *>(&csx), sizeof(csx));
            struct CSY {
                G4double a;
            } csy;
            csy.a = comptonPos(1);
            dataFile6.write(reinterpret_cast<char *>(&csy), sizeof(csy));
            struct CSZ {
                G4double a;
            } csz;
            csz.a = comptonPos(2);
            dataFile7.write(reinterpret_cast<char *>(&csz), sizeof(csz));
        }
    }
}
if(particleType==G4OpticalPhoton::OpticalPhotonDefinition()) {
    if(thePostPoint->GetProcessDefinedStep()->GetProcessName()=="
       OpAbsorption") {
        countR = 0;
    }
    boundaryStatus=boundary->GetStatus();
    if(thePostPoint->GetStepStatus()==fGeomBoundary) {
        switch(boundaryStatus) {
            case Absorption:
                //G4cout << "absorption" << G4endl;
                countR = 0;
                break;
            case Detection:
                //G4cout << "detection" << G4endl;
            {
                G4StepPoint* post = theStep->GetPostStepPoint();
                if (countR == 0) {
                    G4double       detectionTime  = post->GetGlobalTime();
                    G4ThreeVector  detectionPos   = post->GetPosition();
                    struct TOA {
                        G4double a;
                    } toa;
                    toa.a = detectionTime;
                    dataFile1.write(reinterpret_cast<char *>(&toa), sizeof
                        (toa));
                    struct NOR {
                        G4int a;
                    } nor;
                    nor.a = countR;
                    dataFile2.write(reinterpret_cast<char *>(&nor), sizeof
                        (nor));
                    struct HLX {
                        G4double a;
                    } hlx;
                    hlx.a = detectionPos(0);
                    dataFile3.write(reinterpret_cast<char *>(&hlx), sizeof
                        (hlx));
                        struct HLY {
                            G4double a;
                        } hly;
                        hly.a = detectionPos(1);
                        dataFile4.write(reinterpret_cast<char *>(&hly), sizeof
                            (hly));
                    } else {
                        G4double      detectionTime = post->GetGlobalTime();
                        G4ThreeVector detectionPos  = post->GetPosition();
                        struct TOA {
                            G4double a;
                        } toa;
                        toa.a = detectionTime;
                        dataFile1.write(reinterpret_cast<char *>(&toa), sizeof
                            (toa));
                        struct NOR {
                            G4int a;
                        } nor;
                        nor.a = countR;
                        dataFile2.write(reinterpret_cast<char *>(&nor), sizeof
                            (nor));
                        struct HLX {
                            G4double a;
                        } hlx;
                        hlx.a = detectionPos(0);
                        dataFile3.write(reinterpret_cast<char *>(&hlx), sizeof
                            (hlx));
                        struct HLY {
                            G4double a;
                        } hly;
                        hly.a = detectionPos(1);
                        dataFile4.write(reinterpret_cast<char *>(&hly), sizeof
                            (hly));
                        countR = 0;
                    }
                }
                    break;
                case FresnelReflection:
                case TotalInternalReflection:
                case LambertianReflection:
                case LobeReflection:
                case SpikeReflection:
                case BackScattering:
                    countR = countR + 1;
                    break;
                default:
                    break;
            }
        }
    }
}
```

B.4 Physics list

```
#include "lightTrackPhysicsList.hh"
```

```
lightTrackPhysicsList::lightTrackPhysicsList():G4VUserPhysicsList(),
   fVerboseLevel(0) {}

lightTrackPhysicsList::~lightTrackPhysicsList() {}

void lightTrackPhysicsList::ConstructParticle() {
    G4BosonConstructor  bConstructor; bConstructor.ConstructParticle();
    G4LeptonConstructor lConstructor; lConstructor.ConstructParticle();
    G4MesonConstructor  mConstructor; mConstructor.ConstructParticle();
    G4BaryonConstructor rConstructor; rConstructor.ConstructParticle();
    G4IonConstructor    iConstructor; iConstructor.ConstructParticle();
}

void lightTrackPhysicsList::ConstructProcess() {
    AddTransportation();
    ConstructDecay();
    ConstructEM();
    ConstructOp();
}

void lightTrackPhysicsList::ConstructDecay() {
    G4Decay* theDecayProcess = new G4Decay();
    theDecayProcess->SetVerboseLevel(0);
    GetParticleIterator() -> reset();
    while( (*GetParticleIterator())() ) {
        G4ParticleDefinition* particle = GetParticleIterator()->value();
        particle->SetVerboseLevel(0);
        G4ProcessManager* pmanager = particle->GetProcessManager();
        if (theDecayProcess->IsApplicable(*particle)) {
            pmanager ->AddProcess(theDecayProcess);
            pmanager ->SetProcessOrdering(theDecayProcess, idxPostStep);
            pmanager ->SetProcessOrdering(theDecayProcess, idxAtRest);
        }
    }
}

void lightTrackPhysicsList::ConstructEM() {
    GetParticleIterator() -> reset();
    while((*GetParticleIterator())()) {
        G4ParticleDefinition *particle     = GetParticleIterator() -> value();
        G4ProcessManager     *pmanager     = particle              ->
            GetProcessManager();
        G4String              particleName = particle              ->
            GetParticleName();
        if (particleName == "gamma") {
            G4GammaConversion* gammaConversion = new G4GammaConversion();
            G4ComptonScattering* comptonScattering = new G4ComptonScattering()
                ;
            G4PhotoElectricEffect* photoElectric = new G4PhotoElectricEffect()
                ;
            pmanager -> AddDiscreteProcess(gammaConversion);
            pmanager -> AddDiscreteProcess(comptonScattering);
            pmanager -> AddDiscreteProcess(photoElectric);
            gammaConversion->SetVerboseLevel(0);
            comptonScattering->SetVerboseLevel(0);
            photoElectric->SetVerboseLevel(0);
        } else if (particleName == "e-") {
            G4eMultipleScattering* eMultipleScattering = new
                G4eMultipleScattering();
            G4eIonisation* eIonisation = new G4eIonisation();
            G4eBremsstrahlung* eBremsstrahlung = new G4eBremsstrahlung();
            pmanager -> AddProcess(eMultipleScattering,  -1, 1, 1);
            pmanager -> AddProcess(eIonisation,          -1, 2, 2);
            pmanager -> AddProcess(eBremsstrahlung,      -1, 3, 3);

            eMultipleScattering -> SetVerboseLevel(0);
            eIonisation -> SetVerboseLevel(0);
            eBremsstrahlung -> SetVerboseLevel(0);

        } else if (particleName == "e+") {
            G4eMultipleScattering* eMultipleScattering = new
                G4eMultipleScattering();
            G4eIonisation* eIonisation = new G4eIonisation();
            G4eBremsstrahlung* eBremsstrahlung = new G4eBremsstrahlung();
            G4eplusAnnihilation* ePlusAnnihilation = new G4eplusAnnihilation()
                ;
            pmanager -> AddProcess(eMultipleScattering,  -1, 1, 1);
            pmanager -> AddProcess(eIonisation,          -1, 2, 2);
            pmanager -> AddProcess(eBremsstrahlung,      -1, 3, 3);
            pmanager -> AddProcess(ePlusAnnihilation,     0,-1, 4);

            eMultipleScattering -> SetVerboseLevel(0);
            eIonisation -> SetVerboseLevel(0);
            eBremsstrahlung -> SetVerboseLevel(0);
            ePlusAnnihilation -> SetVerboseLevel(0);

        } else if (particleName == "mu+" || particleName == "mu-") {
            G4MuMultipleScattering* muMultipleScattering = new
                G4MuMultipleScattering();
            G4MuIonisation* muIonisation = new G4MuIonisation();
            G4MuBremsstrahlung* muBremsstrahlung = new G4MuBremsstrahlung();
            G4MuPairProduction* muPairProduction = new G4MuPairProduction();

            pmanager -> AddProcess(muMultipleScattering, -1, 1, 1);
            pmanager -> AddProcess(muIonisation,         -1, 2, 2);
            pmanager -> AddProcess(muBremsstrahlung,     -1, 3, 3);
            pmanager -> AddProcess(muPairProduction,     -1, 4, 4);

            muMultipleScattering->SetVerboseLevel(0);
            muIonisation->SetVerboseLevel(0);
            muBremsstrahlung->SetVerboseLevel(0);
            muPairProduction->SetVerboseLevel(0);

        } else {
            if ((particle -> GetPDGCharge() != 0.0) &&
                (particle -> GetParticleName() != "chargedgeantino") &&
                !particle -> IsShortLived()) {
                G4hMultipleScattering* hMultipleScattering = new
                  G4hMultipleScattering();
                G4hIonisation* hIonisation = new G4hIonisation();
                pmanager  -> AddProcess(hMultipleScattering, -1, 1, 1);
                pmanager  -> AddProcess(hIonisation,         -1, 2, 2);

                hMultipleScattering->SetVerboseLevel(0);
                hIonisation->SetVerboseLevel(0);
            }
```

```
        }
    }
}

void lightTrackPhysicsList::ConstructOp() {
    G4OpticalPhoton::OpticalPhoton();
    G4Scintillation      *scintillationProcess      = new G4Scintillation("
        Scintillation");
    G4OpAbsorption       *absorptionProcess         = new G4OpAbsorption();
    //G4OpRayleigh         *rayleighScatteringProcess = new G4OpRayleigh();
    //G4OpMieHG            *mieHGScatteringProcess    = new G4OpMieHG();
    G4OpBoundaryProcess  *boundaryProcess           = new G4OpBoundaryProcess()
        ;
    scintillationProcess      -> SetTrackSecondariesFirst(true);
    scintillationProcess      -> SetScintillationYieldFactor(1.);
    //scintillationProcess      -> SetFiniteRiseTime(true);
    scintillationProcess      -> SetVerboseLevel(0);
    absorptionProcess         -> SetVerboseLevel(0);
    //rayleighScatteringProcess -> SetVerboseLevel(0);
    //mieHGScatteringProcess    -> SetVerboseLevel(0);
    boundaryProcess           -> SetVerboseLevel(0);
    /*
    // Birks correction
    if(G4Threading::IsMasterThread()) {
        G4EmSaturation* emSaturation = G4LossTableManager::Instance()->
            EmSaturation();
        scintillationProcess->AddSaturation(emSaturation);
    }
    */
    GetParticleIterator()     -> reset();
    while((*GetParticleIterator())()) {
        G4ParticleDefinition *particle     = GetParticleIterator() -> value();
        G4ProcessManager     *pmanager     = particle              ->
            GetProcessManager();
        G4String             particleName = particle              ->
            GetParticleName();
        particle->SetVerboseLevel(0);
        if (scintillationProcess -> IsApplicable(*particle)) {
            pmanager             -> AddProcess(scintillationProcess);
            pmanager             -> SetProcessOrderingToLast(
                scintillationProcess, idxAtRest);
            pmanager             -> SetProcessOrderingToLast(
                scintillationProcess, idxPostStep);
        }
        if (particleName == "opticalphoton") {
            pmanager -> AddDiscreteProcess(absorptionProcess);
            //pmanager -> AddDiscreteProcess(rayleighScatteringProcess);
            //pmanager -> AddDiscreteProcess(mieHGScatteringProcess);
            pmanager -> AddDiscreteProcess(boundaryProcess);
            absorptionProcess->SetVerboseLevel(0);
            //rayleighScatteringProcess->SetVerboseLevel(0);
            //mieHGScatteringProcess->SetVerboseLevel(0);
            boundaryProcess->SetVerboseLevel(0);
        }
    }
}

void lightTrackPhysicsList::SetVerbose(G4int) {fVerboseLevel = 0;}
void lightTrackPhysicsList::SetCuts() {
    //DumpCutValuesTable();
    SetCutsWithDefault();
    //defaultCutValue = 0.7*mm;
    //SetCutValue(30.0*mm, "gamma");
    //SetCutValue( 0.7*mm, "e-");
    //SetCutValue( 0.7*mm, "e+");
    //SetCutValue(defaultCutValue, "proton");
}
```

B.5 Detector construction

```
#include "lightTrackDetectorConstruction.hh"


lightTrackDetectorConstruction::lightTrackDetectorConstruction():
   G4VUserDetectorConstruction(),
lvWorld(0),
lvOrganic(0),
lvGrease(0),
lvWindow(0),
lvPhotocathode1(0){}


lightTrackDetectorConstruction::~lightTrackDetectorConstruction() {}


G4VPhysicalVolume *lightTrackDetectorConstruction::Construct() {

    //
       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    // LOAD SETTINGS
    //
       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~



    // File path
    fstream infile("detector.txt");

    // Check that the file exists and then load it
    if (!infile.is_open()) {
        G4cout << " " << G4endl;
        G4cout << "Could not open settings file!" << G4endl;
        G4cout << " " << G4endl;
    } else {
        G4cout << " " << G4endl;
        G4cout << "Settings opened successfully!" << G4endl;
        G4cout << " " << G4endl;
    }

    // Read each line from the file
    string skip;
```

```
getline(infile,skip,'\n');
getline(infile,skip,'\n');
getline(infile,skip,'\n');
string setVar[100];
for (int i = 0; i < 16; i++) {
    getline(infile,setVar[i],'\t');
    getline(infile,skip,'\n');
}
infile.close();

// Convert settings from strings to variables
string::size_type sz;
G4int    printSettings       = stoi (setVar[0],&sz);  // print settings to
   screen
G4int    geometry            = stoi (setVar[1],&sz);  // geometry type (1 =
    cylinder, 2 = cone)
G4double reflectivity        = stod (setVar[2],&sz);  // scintillation
   yield
G4double scintillationYield  = stod (setVar[3],&sz);  // scintillation
   yield
G4double resolutionScale     = stod (setVar[4],&sz);  // resolution scale
G4double constRiseTime       = stod (setVar[5],&sz);  // fast rise time
G4double constFastDecayTime  = stod (setVar[6],&sz);  // fast decay time
G4double yieldRatio          = stod (setVar[7],&sz);  // yield ratio (fast
   vs slow)
G4double absLengthOrg        = stod (setVar[8],&sz);  // absorption length
   (organic)
G4double absLengthGre        = stod (setVar[9],&sz);  // absorption length
   (grease)
G4double absLengthWin        = stod (setVar[10],&sz);  // absorption length
    (window)
G4double birksConstant       = stod (setVar[11],&sz); // birks constant
G4double rindexScint         = stod (setVar[12],&sz); // birks constant
G4double rindexGrease        = stod (setVar[13],&sz); // birks constant
G4double rindexWindow        = stod (setVar[14],&sz); // birks constant
G4double rindexWorld         = stod (setVar[15],&sz); // birks constant

if (printSettings==1) {
    cout << "geometry                = " << geometry           << endl;
    cout << "reflectivity (%)        = " << reflectivity       << endl;
    cout << "scintillation yield (%) = " << scintillationYield << endl;
    cout << "resolution scale (arb.) = " << resolutionScale    << endl;
    cout << "rise time (ns)          = " << constRiseTime      << endl;
    cout << "fast decay time (ns)    = " << constFastDecayTime << endl;
    cout << "yield ratio (%)         = " << yieldRatio         << endl;
    cout << "abs. len. organic (cm)  = " << absLengthOrg       << endl;
    cout << "abs. len. grease (cm)   = " << absLengthGre       << endl;
    cout << "abs. len. window (cm)   = " << absLengthWin       << endl;
    cout << "birks constant (mm/MeV) = " << birksConstant      << endl;
    cout << "ref. index (scint)      = " << rindexScint        << endl;
    cout << "ref. index (grease)     = " << rindexGrease       << endl;
    cout << "ref. index (window)     = " << rindexWindow       << endl;
    cout << "ref. index (world)      = " << rindexWorld        << endl;
}


//
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
// MATERIALS
//
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


// Call the material manager
G4NistManager* nistManager = G4NistManager::Instance();

// Pre-built material definitions
G4Material *materialWorld   = nistManager -> FindOrBuildMaterial("G4_AIR")
   ;
G4Material *materialOrganic = nistManager -> FindOrBuildMaterial("
   G4_PLASTIC_SC_VINYLTOLUENE");
G4Material *materialGrease  = nistManager -> FindOrBuildMaterial("
   G4_SILICON_DIOXIDE");

// Custom material definitions
// Declare elements and variables
G4double fractionMass, density; G4int ncomponents;
G4Element* Si = nistManager->FindOrBuildElement("Si");
G4Element* B  = nistManager->FindOrBuildElement("B");
G4Element* O  = nistManager->FindOrBuildElement("O");
G4Element* Na = nistManager->FindOrBuildElement("Na");
G4Element* Al = nistManager->FindOrBuildElement("Al");
G4Element* K  = nistManager->FindOrBuildElement("K");
G4Element* Cs = nistManager->FindOrBuildElement("Cs");
G4Element* Sb = nistManager->FindOrBuildElement("Sb");

// Custom material
// PMT Window (borosilicate)
G4Material* materialWindow = new G4Material("window", density= 2.23*g/cm3,
     ncomponents=6);
materialWindow->AddElement(B,  fractionMass=0.0400);
materialWindow->AddElement(O,  fractionMass=0.5395);
materialWindow->AddElement(Na, fractionMass=0.0281);
materialWindow->AddElement(Al, fractionMass=0.0116);
materialWindow->AddElement(Si, fractionMass=0.3772);
materialWindow->AddElement(K,  fractionMass=0.0033);

// Custom material
// Photocathode (bialkali)
G4Material* materialPhotocathode = new G4Material("photocathode", density=
     4.28*g/cm3, ncomponents=3);
materialPhotocathode->AddElement(K,  fractionMass=0.133);
materialPhotocathode->AddElement(Cs, fractionMass=0.452);
materialPhotocathode->AddElement(Sb, fractionMass=0.415);


//
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

// PROPERTIES
// http://www.eljentechnology.com/products/plastic-scintillators/ej-200-ej
   -204-ej-208-ej-212
// https://hallcweb.jlab.org/DocDB/0008/000809/001/PhotonisCatalog.pdf
// http://www.hzcphotonics.com/products/XP4512.pdf
```

```
// https://www.ncbi.nlm.nih.gov/pubmed/28976914
// https://refractiveindex.info/
//
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~



// Properties: organic (EJ200)
G4double refracticeIndexOrganic[14]  = {rindexScint,rindexScint,
    rindexScint,rindexScint,rindexScint,rindexScint,rindexScint,
                                              rindexScint,rindexScint,
                                                  rindexScint,rindexScint,
                                                  rindexScint,rindexScint,
                                                  rindexScint};
G4double absorptionLengthOrganic[14] = {absLengthOrg*cm,absLengthOrg*cm,
    absLengthOrg*cm,absLengthOrg*cm,
                                              absLengthOrg*cm,absLengthOrg*cm,
                                                  absLengthOrg*cm,absLengthOrg*
                                                  cm,
                                              absLengthOrg*cm,absLengthOrg*cm,
                                                  absLengthOrg*cm,absLengthOrg*
                                                  cm,
                                              absLengthOrg*cm,absLengthOrg*cm};
G4double absorptionLengthGrease[14]  = {absLengthGre*cm,absLengthGre*cm,
    absLengthGre*cm,absLengthGre*cm,
                                              absLengthGre*cm,absLengthGre*cm,
                                                  absLengthGre*cm,absLengthGre*
                                                  cm,
                                              absLengthGre*cm,absLengthGre*cm,
                                                  absLengthGre*cm,absLengthGre*
                                                  cm,
                                              absLengthGre*cm,absLengthGre*cm};
G4double absorptionLengthWindow[14]  = {absLengthWin*cm,absLengthWin*cm,
    absLengthWin*cm,absLengthWin*cm,
                                              absLengthWin*cm,absLengthWin*cm,
                                                  absLengthWin*cm,absLengthWin*
                                                  cm,
                                              absLengthWin*cm,absLengthWin*cm,
                                                  absLengthWin*cm,absLengthWin*
                                                  cm,
                                              absLengthWin*cm,absLengthWin*cm};
G4double energyDistributionOrganic[14] =
    {2.4797,2.6241,2.6764,2.7260,2.7689,2.8061,2.8351,
                                                2.8721,2.9164,2.9458,2.9635,2.9826,3.0307,3.0

G4double fastComponentOrganic[14]      =
    {0.0494,0.2589,0.4021,0.4820,0.5866,0.6994,0.8176,
                                                0.9304,0.9938,0.8977,0.7549,0.5682,0.2085,0.0

G4MaterialPropertiesTable* propertyOrganic = new G4MaterialPropertiesTable
    ();
propertyOrganic -> AddConstProperty("SCINTILLATIONYIELD",
    scintillationYield/MeV);
propertyOrganic -> AddConstProperty("RISETIMECONSTANT",    constRiseTime*ns
    );
propertyOrganic -> AddConstProperty("FASTTIMECONSTANT",
    constFastDecayTime*ns);
propertyOrganic -> AddConstProperty("YIELDRATIO",          yieldRatio);
```

```
propertyOrganic -> AddConstProperty("RESOLUTIONSCALE",    resolutionScale)
    ;
propertyOrganic -> AddProperty("SCINTILLATION", energyDistributionOrganic,
     fastComponentOrganic,     14);
propertyOrganic -> AddProperty("FASTCOMPONENT", energyDistributionOrganic,
     fastComponentOrganic,     14);
propertyOrganic -> AddProperty("RINDEX",         energyDistributionOrganic,
     refracticeIndexOrganic,  14);
propertyOrganic -> AddProperty("ABSLENGTH",      energyDistributionOrganic,
     absorptionLengthOrganic, 14);
materialOrganic -> SetMaterialPropertiesTable(propertyOrganic);
//materialOrganic -> GetIonisation()->SetBirksConstant(birksConstant*mm/
    MeV);

// Properties: grease (EJ550)
G4MaterialPropertiesTable* propertyGrease = new G4MaterialPropertiesTable
    ();
G4double energyDistributionGrease[2] = {2.000*eV, 4.000*eV};
G4double refractiveIndexGrease[2]    = {rindexGrease, rindexGrease};
propertyGrease -> AddProperty("RINDEX",    energyDistributionGrease,
    refractiveIndexGrease,  2);
propertyGrease -> AddProperty("ABSLENGTH", energyDistributionGrease,
    absorptionLengthGrease, 2);
materialGrease -> SetMaterialPropertiesTable(propertyGrease);

// Properties: window (borosilicate glass)
G4MaterialPropertiesTable* propertyWindow = new G4MaterialPropertiesTable
    ();
G4double energyDistributionWindow[2] = {2.00*eV, 4.00*eV};
G4double refractiveIndexWindow[2]    = {rindexWindow, rindexWindow};
propertyWindow -> AddProperty("RINDEX",    energyDistributionWindow,
    refractiveIndexWindow,  2);
propertyWindow -> AddProperty("ABSLENGTH", energyDistributionWindow,
    absorptionLengthWindow, 2);
materialWindow -> SetMaterialPropertiesTable(propertyWindow);

// Properties: world (air stp)
G4MaterialPropertiesTable* propertyWorld = new G4MaterialPropertiesTable()
    ;
G4double energyDistributionWorld[2] = {2.00*eV, 4.00*eV};
G4double refractiveIndexWorld[2]    = {rindexWorld, rindexWorld};
propertyWorld -> AddProperty("RINDEX", energyDistributionWorld,
    refractiveIndexWorld, 2);
materialWorld -> SetMaterialPropertiesTable(propertyWorld);


//
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
// DIMENSIONS
//
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~



// shift
G4double xs =  0.00*cm;
G4double ys =  0.00*cm;
```

```
G4double zs = −5.00*cm; // front face is −2.5cm away from z=0

// world
G4double        xlengthWorld        = 2.75*cm;
G4double        ylengthWorld        = 2.75*cm;
G4double        zlengthWorld        = 8.00*cm;
G4ThreeVector   positionWorld       = G4ThreeVector(0.0, 0.0, 0.0);

// organic scintillator
G4double diameterOrganic            = 50.0*mm;
G4double heightOrganic              = 50.0*mm;
G4ThreeVector positionOrganic       = G4ThreeVector(0.0+xs, 0.0+ys, 0.0+zs
   );

// grease
G4double diameterGrease             = 50.0*mm;
G4double heightGrease               = 0.50*mm;
G4ThreeVector positionGrease        = G4ThreeVector(0.0+xs,
                                                    0.0+ys,
                                                    −heightOrganic/2.0 −
                                                    heightGrease/2.0 +
                                                    zs);

// window
G4double diameterWindow             = 50.0*mm;
G4double heightWindow               = 0.50*mm;
G4ThreeVector positionWindow        = G4ThreeVector(0.0+xs,
                                                    0.0+ys,
                                                    −heightOrganic/2.0 −
                                                    heightGrease −
                                                    heightWindow/2.0 +
                                                    zs);

// photocathode
G4double diameterInPhotocathode1    = 0.00*mm;
G4double diameterOutPhotocathode1   = diameterWindow;
G4double heightPhotocathode1        = 0.50*mm;
G4ThreeVector positionPhotocathode1 = G4ThreeVector(0.0+xs,
                                                    0.0+ys,
                                                    −heightOrganic/2.0 −
                                                    heightGrease −
                                                    heightWindow −
                                                    heightPhotocathode1
                                                       /2.0 +
                                                    zs);


//
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

//
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

// SOLID VOLUMES (variable) * * * * * * * * * * * * * * * * * * * * * * *
   * * * * * * * * *
//
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
//
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


G4double rotate        = 0.0;
G4double diameterFront = 0.00*cm;

if (geometry==1) {
    zs                 = 5.0*cm/2.0-7.5*cm; // front face is -2.5cm away
        from z=0
    diameterFront      = 0.00*mm;
    diameterOrganic    = 50.0*mm;
    heightOrganic      = 50.0*mm;
    positionOrganic    = G4ThreeVector(0.0, 0.0, 0.0+zs);
    G4Cons *svOrganic  = new G4Cons("organic", 0.0, (diameterOrganic/2.0)
        , 0.0, (diameterFront/2.0), (heightOrganic/2.0), 0.0, twopi);
    lvOrganic          = new G4LogicalVolume(svOrganic, materialOrganic,
        "organic", 0,0,0);
    rotate             = 0.0;
}
if (geometry==2) {
    zs                 = 5.0*cm/2.0-7.5*cm; // front face is -2.5cm away
        from z=0
    diameterFront      = 5.00*mm;
    diameterOrganic    = 50.0*mm;
    heightOrganic      = 50.0*mm;
    positionOrganic    = G4ThreeVector(0.0, 0.0, 0.0+zs);
    G4Cons *svOrganic  = new G4Cons("organic", 0.0, (diameterOrganic/2.0)
        , 0.0, (diameterFront/2.0), (heightOrganic/2.0), 0.0, twopi);
    lvOrganic          = new G4LogicalVolume(svOrganic, materialOrganic,
        "organic", 0,0,0);
    rotate             = 0.0;
}
if (geometry==3) {
    zs                 = 5.0*cm/2.0-7.5*cm; // front face is -2.5cm away
        from z=0
    diameterFront      = 10.0*mm;
    diameterOrganic    = 50.0*mm;
    heightOrganic      = 50.0*mm;
    positionOrganic    = G4ThreeVector(0.0, 0.0, 0.0+zs);
    G4Cons *svOrganic  = new G4Cons("organic", 0.0, (diameterOrganic/2.0)
        , 0.0, (diameterFront/2.0), (heightOrganic/2.0), 0.0, twopi);
    lvOrganic          = new G4LogicalVolume(svOrganic, materialOrganic,
        "organic", 0,0,0);
    rotate             = 0.0;
}
if (geometry==4) {
    zs                 = 5.0*cm/2.0-7.5*cm; // front face is -2.5cm away
        from z=0
    diameterFront      = 15.0*mm;
    diameterOrganic    = 50.0*mm;
    heightOrganic      = 50.0*mm;
    positionOrganic    = G4ThreeVector(0.0, 0.0, 0.0+zs);
    G4Cons *svOrganic  = new G4Cons("organic", 0.0, (diameterOrganic/2.0)
        , 0.0, (diameterFront/2.0), (heightOrganic/2.0), 0.0, twopi);
    lvOrganic          = new G4LogicalVolume(svOrganic, materialOrganic,
        "organic", 0,0,0);
```

```
        rotate              = 0.0;
    }
    if (geometry==5) {
        zs                  = 5.0*cm/2.0-7.5*cm; // front face is -2.5cm away
            from z=0
        diameterFront       = 20.0*mm;
        diameterOrganic     = 50.0*mm;
        heightOrganic       = 50.0*mm;
        positionOrganic     = G4ThreeVector(0.0, 0.0, 0.0+zs);
        G4Cons *svOrganic   = new G4Cons("organic", 0.0, (diameterOrganic/2.0)
            , 0.0, (diameterFront/2.0), (heightOrganic/2.0), 0.0, twopi);
        lvOrganic           = new G4LogicalVolume(svOrganic, materialOrganic,
            "organic", 0,0,0);
        rotate              = 0.0;
    }
    if (geometry==6) {
        zs                  = 5.0*cm/2.0-7.5*cm; // front face is -2.5cm away
            from z=0
        diameterFront       = 25.0*mm;
        diameterOrganic     = 50.0*mm;
        heightOrganic       = 50.0*mm;
        positionOrganic     = G4ThreeVector(0.0, 0.0, 0.0+zs);
        G4Cons *svOrganic   = new G4Cons("organic", 0.0, (diameterOrganic/2.0)
            , 0.0, (diameterFront/2.0), (heightOrganic/2.0), 0.0, twopi);
        lvOrganic           = new G4LogicalVolume(svOrganic, materialOrganic,
            "organic", 0,0,0);
        rotate              = 0.0;
    }
    if (geometry==7) {
        zs                  = 5.0*cm/2.0-7.5*cm; // front face is -2.5cm away
            from z=0
        diameterFront       = 30.0*mm;
        diameterOrganic     = 50.0*mm;
        heightOrganic       = 50.0*mm;
        positionOrganic     = G4ThreeVector(0.0, 0.0, 0.0+zs);
        G4Cons *svOrganic   = new G4Cons("organic", 0.0, (diameterOrganic/2.0)
            , 0.0, (diameterFront/2.0), (heightOrganic/2.0), 0.0, twopi);
        lvOrganic           = new G4LogicalVolume(svOrganic, materialOrganic,
            "organic", 0,0,0);
        rotate              = 0.0;
    }
    if (geometry==8) {
        zs                  = 5.0*cm/2.0-7.5*cm; // front face is -2.5cm away
            from z=0
        diameterFront       = 35.0*mm;
        diameterOrganic     = 50.0*mm;
        heightOrganic       = 50.0*mm;
        positionOrganic     = G4ThreeVector(0.0, 0.0, 0.0+zs);
        G4Cons *svOrganic   = new G4Cons("organic", 0.0, (diameterOrganic/2.0)
            , 0.0, (diameterFront/2.0), (heightOrganic/2.0), 0.0, twopi);
        lvOrganic           = new G4LogicalVolume(svOrganic, materialOrganic,
            "organic", 0,0,0);
        rotate              = 0.0;
    }
    if (geometry==9) {
        zs                  = 5.0*cm/2.0-7.5*cm; // front face is -2.5cm away
            from z=0
        diameterFront       = 40.0*mm;
        diameterOrganic      = 50.0*mm;
        heightOrganic        = 50.0*mm;
        positionOrganic      = G4ThreeVector(0.0, 0.0, 0.0+zs);
        G4Cons *svOrganic    = new G4Cons("organic", 0.0, (diameterOrganic/2.0)
            , 0.0, (diameterFront/2.0), (heightOrganic/2.0), 0.0, twopi);
        lvOrganic            = new G4LogicalVolume(svOrganic, materialOrganic,
            "organic", 0,0,0);
        rotate               = 0.0;
    }
    if (geometry==10) {
        zs                   = 5.0*cm/2.0-7.5*cm; // front face is -2.5cm away
            from z=0
        diameterFront        = 45.0*mm;
        diameterOrganic      = 50.0*mm;
        heightOrganic        = 50.0*mm;
        positionOrganic      = G4ThreeVector(0.0, 0.0, 0.0+zs);
        G4Cons *svOrganic    = new G4Cons("organic", 0.0, (diameterOrganic/2.0)
            , 0.0, (diameterFront/2.0), (heightOrganic/2.0), 0.0, twopi);
        lvOrganic            = new G4LogicalVolume(svOrganic, materialOrganic,
            "organic", 0,0,0);
        rotate               = 0.0;
    }
    if (geometry==11) {
        zs                   = 5.0*cm/2.0-7.5*cm; // front face is -2.5cm away
            from z=0
        diameterOrganic      = 50.0*mm;
        heightOrganic        = 50.0*mm;
        positionOrganic      = G4ThreeVector(0.0, 0.0, 0.0+zs);
        G4Tubs *svOrganic    = new G4Tubs("organic", 0.0, (diameterOrganic/2.0)
            , (heightOrganic/2.0), 0.0, twopi);
        lvOrganic            = new G4LogicalVolume(svOrganic, materialOrganic,
            "organic", 0,0,0);
        rotate               = 0.0;
    }


    //
       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    // SOLID VOLUMES (fixed)
    //
       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~



    G4Box  *svWorld          = new G4Box("world",
                                      xlengthWorld,
                                      ylengthWorld,
                                      zlengthWorld);
    G4Tubs *svGrease         = new G4Tubs("grease",
                                       0.0,
                                       (diameterGrease/2.0),
                                       (heightGrease/2.0),
                                       0.0,
                                       twopi);
    G4Tubs *svWindow         = new G4Tubs("window",
                                       0.0,
                                       (diameterWindow/2.0),
```

```
                                     (heightWindow/2.0),
                                     0.0,
                                     twopi);
G4Tubs *svPhotocathode1 = new G4Tubs("photocathode1",
                                     diameterInPhotocathode1/2.0,
                                     diameterOutPhotocathode1/2.0,
                                     heightPhotocathode1/2.0,
                                     0.0,
                                     twopi);


//
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
// LOGICAL VOLUMES
//
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~



lvWorld         = new G4LogicalVolume(svWorld,          materialWorld,
          "world",          0,0,0);
lvGrease        = new G4LogicalVolume(svGrease,         materialGrease,
         "grease",          0,0,0);
lvWindow        = new G4LogicalVolume(svWindow,         materialWindow,
         "window",          0,0,0);
lvPhotocathode1 = new G4LogicalVolume(svPhotocathode1,
   materialPhotocathode, "photocathode1", 0,0,0);


//
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
// ROTATIONS
//
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~



G4RotationMatrix* rotateOrganic       = new G4RotationMatrix();
rotateOrganic       -> rotateX(0.*deg);
rotateOrganic       -> rotateY(0.*deg);
rotateOrganic       -> rotateZ(0.*deg);

G4RotationMatrix* rotateGrease        = new G4RotationMatrix();
rotateGrease        -> rotateX(0.*deg);
rotateGrease        -> rotateY(0.*deg);
rotateGrease        -> rotateZ(0.*deg);

G4RotationMatrix* rotateWindow        = new G4RotationMatrix();
rotateWindow        -> rotateX(0.*deg);
rotateWindow        -> rotateY(0.*deg);
rotateWindow        -> rotateZ(0.*deg);

G4RotationMatrix* rotatePhotocathode1 = new G4RotationMatrix();
rotatePhotocathode1 -> rotateX(0.*deg);
rotatePhotocathode1 -> rotateY(0.*deg);
rotatePhotocathode1 -> rotateZ(0.*deg);
```

```
//
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
// PHYSICAL VOLUMES
//
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


G4VPhysicalVolume *pvWorld         = new G4PVPlacement(0,
                                                     positionWorld,
                                                     lvWorld,
                                                     "world",
                                                     0,
                                                     false,
                                                     1);
G4VPhysicalVolume *pvOrganic       = new G4PVPlacement(rotateOrganic,
                                                     positionOrganic,
                                                     lvOrganic,
                                                     "organic",
                                                     lvWorld,
                                                     false,
                                                     1);
G4VPhysicalVolume *pvGrease        = new G4PVPlacement(rotateGrease,
                                                     positionGrease,
                                                     lvGrease,
                                                     "grease",
                                                     lvWorld,
                                                     false,
                                                     1);
G4VPhysicalVolume *pvWindow        = new G4PVPlacement(rotateWindow,
                                                     positionWindow,
                                                     lvWindow,
                                                     "window",
                                                     lvWorld,
                                                     false,
                                                     1);
new G4PVPlacement(rotatePhotocathode1,
                  positionPhotocathode1,
                  lvPhotocathode1,
                  "photocathode1",
                  lvWorld,
                  false,
                  1);


//
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
// OPTICAL SURFACE SUMMARY
//
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


G4MaterialPropertiesTable *reflectorProp     = new
```

```
    G4MaterialPropertiesTable();
G4MaterialPropertiesTable *greaseEdgeProp     = new
    G4MaterialPropertiesTable();
G4MaterialPropertiesTable *windowEdgeProp     = new
    G4MaterialPropertiesTable();
G4MaterialPropertiesTable *greaseFaceProp     = new
    G4MaterialPropertiesTable();
G4MaterialPropertiesTable *windowFaceProp     = new
    G4MaterialPropertiesTable();
G4OpticalSurface          *osReflector        = new G4OpticalSurface("
    reflector");
G4OpticalSurface          *greaseEdgeOptSurf = new G4OpticalSurface("
    grease_edge");
G4OpticalSurface          *windowEdgeOptSurf = new G4OpticalSurface("
    window_edge");
G4OpticalSurface          *greaseFaceOptSurf = new G4OpticalSurface("
    grease_face");
G4OpticalSurface          *windowFaceOptSurf = new G4OpticalSurface("
    window_face");


//
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
//
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
// OPTICAL SURFACE ATTRIBUTES (variable) * * * * * * * * * * * * * * * * *
     * * * * * * * * *
//
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
//
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~



//
// OPTICAL SURFACE: reflector
// ------------------------------------
G4double reflectorEnergy[2]        = {2.00*eV, 4.00*eV};
G4double reflectorReflectivity[2]  = {reflectivity, reflectivity}; //
    variable reflectivity
G4double reflectorTransmittance[2] = {0.0000,  0.0000};
G4double reflectorSpecularSpike[2] = {0.0000,  0.0000};
G4double reflectorSpecularLobe[2]  = {0.0000,  0.0000};
G4double reflectorBackscatter[2]   = {0.0000,  0.0000};
reflectorProp -> AddProperty("REFLECTIVITY",             reflectorEnergy,
    reflectorReflectivity,2);
reflectorProp -> AddProperty("TRANSMITTANCE",            reflectorEnergy,
    reflectorTransmittance,2);
reflectorProp -> AddProperty("SPECULARSPIKECONSTANT",    reflectorEnergy,
    reflectorSpecularSpike,2);
reflectorProp -> AddProperty("SPECULARLOBECONSTANT",     reflectorEnergy,
    reflectorSpecularLobe,2);
reflectorProp -> AddProperty("BACKSCATTERCONSTANT",      reflectorEnergy,
    reflectorBackscatter,2);
osReflector   -> SetMaterialPropertiesTable(             reflectorProp);
```

```
osReflector   -> SetModel(                          unified);
                 // optical model
osReflector   -> SetType(                           dielectric_metal
   );      // medium contact
osReflector   -> SetFinish(                         ground);
                  // polished reflector
osReflector   -> SetSigmaAlpha(0.0);
                                              // facet normal
new G4LogicalBorderSurface("reflector", pvOrganic, pvWorld, osReflector);
            // one direction


//
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
// OPTICAL SURFACE ATTRIBUTES (fixed)
//
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~



//
// OPTICAL SURFACE: grease (face)
// ----------------------------------
greaseFaceOptSurf -> SetMaterialPropertiesTable(          greaseFaceProp);
greaseFaceOptSurf -> SetModel(                            unified);
greaseFaceOptSurf -> SetType(
   dielectric_dielectric);
greaseFaceOptSurf -> SetFinish(                           polished);
new G4LogicalBorderSurface("grease_face", pvOrganic, pvGrease,
   greaseFaceOptSurf);  // both directions
new G4LogicalBorderSurface("grease_face", pvGrease,  pvOrganic,
   greaseFaceOptSurf); // both directions

//
// OPTICAL SURFACE: grease (edge)
// ----------------------------------
greaseEdgeOptSurf -> SetMaterialPropertiesTable(          greaseEdgeProp);
greaseEdgeOptSurf -> SetModel(                            unified);
greaseEdgeOptSurf -> SetType(
   dielectric_dielectric);
greaseEdgeOptSurf -> SetFinish(                           polished);
new G4LogicalBorderSurface("grease_edge", pvGrease, pvWorld,
   greaseEdgeOptSurf);    // one direction

//
// OPTICAL SURFACE: window (face)
// ----------------------------------
windowFaceOptSurf -> SetMaterialPropertiesTable(          windowFaceProp);
windowFaceOptSurf -> SetModel(                            unified);
windowFaceOptSurf -> SetType(
   dielectric_dielectric);
windowFaceOptSurf -> SetFinish(                           polished);
new G4LogicalBorderSurface("window_face", pvGrease, pvWindow,
   windowFaceOptSurf);   // both directions
new G4LogicalBorderSurface("window_face", pvWindow, pvGrease,
   windowFaceOptSurf);   // both directions
```

```
//
// OPTICAL SURFACE: window (edge)
// ——————————————————————————
G4double windowEdgeEnergy[2]        = {2.00*eV, 4.00*eV};
G4double windowEdgeReflectivity[2]  = {0.000,   0.000};
G4double windowEdgeTransmittance[2] = {0.000,   0.000};
windowEdgeProp     -> AddProperty("REFLECTIVITY",          windowEdgeEnergy
    ,windowEdgeReflectivity,2);
windowEdgeProp     -> AddProperty("TRANSMITTANCE",         windowEdgeEnergy
    ,windowEdgeTransmittance,2);
windowEdgeOptSurf -> SetMaterialPropertiesTable(          windowEdgeProp);
windowEdgeOptSurf -> SetModel(                            unified);
windowEdgeOptSurf -> SetType(                             dielectric_metal
    );
windowEdgeOptSurf -> SetFinish(                           ground);
new G4LogicalBorderSurface("window_edge", pvWindow, pvWorld,
    windowEdgeOptSurf);     // one direction


//
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
// DETECTION SURFACE SUMMARY
//
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~



G4MaterialPropertiesTable *photocathode1Prop     = new
    G4MaterialPropertiesTable();
G4OpticalSurface          *photocathode1OptSurf = new G4OpticalSurface("
    photocathode1");


//
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
// DETECTION ATTRIBUTES (PHOTOCATHODE SETTINGS)
//
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~



//
// DETECTION SURFACE: photocathode
// ——————————————————————————
G4double photocathode1Energy[14]        =
    {2.4800,2.5060,2.5306,2.5705,2.6146,2.6667,2.7132,
     2.7613,2.8182,2.8665,2.9318,2.9920,3.0451,3.0996};
G4double photocathode1Reflectivity[14] =
    {0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,
     0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000};
G4double photocathode1Efficiency[14]    = {12.9122/100.0,13.9361/100.0,
                                          14.8738/100.0,16.0520/100.0,
                                          17.2570/100.0,18.7498/100.0,
                                          19.9507/100.0,20.9569/100.0,
                                          22.1930/100.0,23.0541/100.0,
                                          23.9274/100.0,24.5701/100.0,
```

```
                                              25.0440/100.0,25.5543/100.0};
    photocathode1Prop     -> AddProperty("REFLECTIVITY", photocathode1Energy,
       photocathode1Reflectivity,14);
    photocathode1Prop     -> AddProperty("EFFICIENCY",   photocathode1Energy,
       photocathode1Efficiency,14);
    photocathode1OptSurf -> SetMaterialPropertiesTable( photocathode1Prop);
    photocathode1OptSurf -> SetModel(                    unified);
    photocathode1OptSurf -> SetType(                     dielectric_metal);
    photocathode1OptSurf -> SetFinish(                   polished);
    new G4LogicalSkinSurface("photocathode1", lvPhotocathode1,
       photocathode1OptSurf);


    //
       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    // VISUAL SETTINGS
    //
       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~



    // Attributes: organic detector
    G4VisAttributes* vaOrganic         = new G4VisAttributes(G4Colour
       (0.1,0.1,0.1,0.1));
    G4VisAttributes* vaGrease          = new G4VisAttributes(G4Colour
       (0.3,0.5,1.0,0.2));
    G4VisAttributes* vaWindow          = new G4VisAttributes(G4Colour
       (0.0,1.0,1.0,0.5));
    G4VisAttributes* vaPhotocathode1   = new G4VisAttributes(G4Colour
       (0.1,0.5,4.0,0.7));
    G4VisAttributes* vaWorld           = new G4VisAttributes(G4Colour
       (0.1,0.1,0.1,1.0));

    //lvWorld          -> SetVisAttributes(G4VisAttributes::Invisible);
    lvOrganic          -> SetVisAttributes(vaOrganic);
    lvGrease           -> SetVisAttributes(vaGrease);
    lvWindow           -> SetVisAttributes(vaWindow);
    lvPhotocathode1    -> SetVisAttributes(vaPhotocathode1);
    lvWorld            -> SetVisAttributes(vaWorld);
    vaPhotocathode1    -> SetForceSolid(true);
    vaOrganic          -> SetForceSolid(true);
    vaGrease           -> SetForceSolid(true);
    vaWindow           -> SetForceSolid(true);
    vaWorld            -> SetForceWireframe(true);
    //vaWorld            -> SetVisibility(false); // comment to see world
       boundary

    return pvWorld;


}

void lightTrackDetectorConstruction::ConstructSDandField() {


    //
       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
// SENSITIVE DETECTOR
//
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


/*
// manager pointers
G4String      SDname;
G4SDManager *SDman1 = G4SDManager::GetSDMpointer();
G4SDManager *SDman2 = G4SDManager::GetSDMpointer();

// sensitive detectors paths
G4VSensitiveDetector *organicSD       = new lightTrackOrganicSD(SDname="/
   BSG/organicSD");
G4VSensitiveDetector *photocathode1SD = new lightTrackPhotocathode1SD(
   SDname="/BSG/photocathode1SD");

// filters
G4SDParticleFilter *opticalFilter = new G4SDParticleFilter("opticalFilter
   ");
opticalFilter   -> add("opticalphoton");
photocathode1SD -> SetFilter(opticalFilter);

G4SDParticleFilter *electronFilter = new G4SDParticleFilter("
   electronFilter");
electronFilter -> add("e-");
organicSD      -> SetFilter(electronFilter);

// add detectors
SDman1 -> AddNewDetector(organicSD);
SDman2 -> AddNewDetector(photocathode1SD);

// set detectors
lvOrganic       -> SetSensitiveDetector(organicSD);
lvPhotocathode1 -> SetSensitiveDetector(photocathode1SD);
 */

// manager pointers
G4String      SDname;
G4SDManager *SDman1 = G4SDManager::GetSDMpointer();
G4SDManager *SDman2 = G4SDManager::GetSDMpointer();

// sensitive detectors paths
G4VSensitiveDetector *organicSD       = new lightTrackOrganicSD(SDname="/
   BSG/organicSD");
G4VSensitiveDetector *photocathode1SD = new lightTrackPhotocathode1SD(
   SDname="/BSG/photocathode1SD");

// filter
G4SDParticleFilter *particleFilter = new G4SDParticleFilter("
   particleFilter");
particleFilter -> add("e-");
particleFilter -> add("gamma");
organicSD      -> SetFilter(particleFilter);

// filter
G4SDParticleFilter *opticalFilter = new G4SDParticleFilter("opticalFilter"
```

```
        );
    opticalFilter   -> add("opticalphoton");
    photocathode1SD -> SetFilter(opticalFilter);

    // add detectors
    SDman1 -> AddNewDetector(organicSD);
    SDman2 -> AddNewDetector(photocathode1SD);

    // set detectors
    lvOrganic       -> SetSensitiveDetector(organicSD);
    lvPhotocathode1 -> SetSensitiveDetector(photocathode1SD);


}
```

BIBLIOGRAPHY

[1] F. Brooks, "A scintillation counter with neutron and gamma-ray discriminators," *Nuclear Instruments and Methods*, vol. 4, no. 3, pp. 151–163, 1959.

[2] J. B. Birks, *The Theory and Practice of Scintillation Counting*, vol. 148. Elsevier, 1964.

[3] P. E. Stanley, *Liquid scintillation counting: Recent developments*. Elsevier, 1974.

[4] F. Brooks, "Development of organic scintillators," *Nuclear Instruments and Methods*, vol. 162, no. 1-3, pp. 477–505, 1979.

[5] G. Knoll, *Radiation Detection and Measurement*. Wiley, 4th ed., 2010.

[6] I. V. Kilimchuk, V. A. Tarasov, and I. D. Vlasova, "Study of light collection as a function of scintillator surface roughness," in *Radiation Measurements*, 2010.

[7] B. K. Lubsandorzhiev, "On the history of photomultiplier tube invention," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 567, no. 1 SPEC. ISS., pp. 236–238, 2006.

[8] D. Aliaga-Kelly and D. R. Nicoll, "Recent developments in scintillation detectors," *Nuclear Instruments and Methods*, vol. 43, pp. 110–115, 1966.

[9] N. Zaitseva, A. Glenn, L. Carman, H. Paul Martinez, R. Hatarik, H. Klapper, and S. Payne, "Scintillation properties of solution-grown trans-stilbene single crystals," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 789, pp. 8–15, 2015.

[10] N. P. Zaitseva, A. M. Glenn, A. N. Mabe, M. L. Carman, C. R. Hurlbut, J. W. Inman, and S. A. Payne, "Recent developments in plastic scintillators with pulse shape discrimination," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 889, no. January, pp. 97–104, 2018.

[11] R. C. Runkle, D. L. Chichester, and S. J. Thompson, "Rattling nucleons: New developments in active interrogation of special nuclear material," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 663, no. 1, pp. 75–95, 2012.

[12] J. Polack, M. Flaska, A. Enqvist, C. Sosa, C. Lawrence, and S. Pozzi, "An algorithm for charge-integration, pulse-shape discrimination and estimation of neutron/photon misclassification in organic scintillators," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 795, 2015.

[13] P. Schuster and E. Brubaker, "Investigating the Anisotropic Scintillation Response in Anthracene through Neutron, Gamma-Ray, and Muon Measurements," *IEEE Transactions on Nuclear Science*, vol. 63, no. 3, pp. 1942–1954, 2016.

[14] C. S. Sosa, M. M. Bourne, S. D. Clarke, S. A. Pozzi, R. Sciences, and A. Arbor, "Optimization of PSD for Stilbene as a Function of Neutron Energy Deposition," *INMM Proceedings*, vol. 2, 2015.

[15] C. S. Sosa, M. Flaska, and S. A. Pozzi, "Comparison of analog and digital pulse-shape-discrimination systems," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 826, pp. 72–79, 2016.

[16] A. Di Fulvio, "Passive Assay of Plutonium Metal Plates using a Fast-Neutron Multiplicity Counter," *Nuclear Inst. and Methods in Physics Research, A*, vol. 855, no. February, pp. 92–101, 2017.

[17] A. Di Fulvio, T. H. Shin, A. Basley, C. Swenson, C. Sosa, S. D. Clarke, J. Sanders, S. Watson, D. L. Chichester, and S. A. Pozzi, "Fast-neutron multiplicity counter for active measurements of uranium oxide certified material," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, no. May, 2018.

[18] D. Sertore, P. Michelato, L. Monaco, and C. Pagani, "A study for the characterization of high QE photocathodes," in *Proceedings of the IEEE Particle Accelerator Conference*, pp. 2760–2762, 2007.

[19] M. Mottaghian, R. Koohi-Fayegh, N. Ghal-Eh, and G. R. Etaati, "Photocathode non-uniformity contribution to the energy resolution of scintillators," *Radiation Protection Dosimetry*, vol. 140, no. 1, pp. 16–24, 2010.

[20] J. Fraser-Mitchell and A. Wright, "Contribution of photocathode nonuniformity to energy resolution in NaI(TI) scintillation detectors," *Nuclear Instruments and Methods in Physics Research*, vol. 288, pp. 429–438, 1990.

[21] P. Takhar, "Resolution and cathode uniformity in scintillation counters," in *IEEE Transactions on Nuclear Science*, pp. 438–442, 1967.

[22] M. Janecek and W. W. Moses, "Measuring light reflectance of BGO crystal surfaces," in *IEEE Transactions on Nuclear Science*, pp. 2443–2449, 2008.

[23] M. Janecek, "Reflectivity spectra for commonly used reflectors," *IEEE Transactions on Nuclear Science*, 2012.

[24] F. A. Danevich, V. V. Kobychev, R. V. Kobychev, H. Kraus, V. B. Mikhailik, V. M. Mokina, and I. M. Solsky, "Impact of geometry on light collection efficiency of scintillation detectors for cryogenic rare event searches," *Nuclear Instruments and Methods in Physics Research, Section B: Beam Interactions with Materials and Atoms*, vol. 336, pp. 26–30, 2014.

[25] Y. Xiaoguang, "A study of light collection efficiency in scintillation detectors," *Nuclear Instruments and Methods in Physics Research*, vol. 228, pp. 101–104, 1984.

[26] J. S. Huber, W. W. Moses, M. S. Andreaco, M. Loope, C. L. Melcher, and R. Nutt, "Geometry and surface treatment dependence of the light collection from LSO crystals," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 437, no. 2-3, pp. 374–380, 1999.

[27] K. Pauwels, E. Auffray, S. Gundacker, A. Knapitsch, and P. Lecoq, "Effect of aspect ratio on the light output of scintillators," *IEEE Transactions on Nuclear Science*, 2012.

[28] H. Klein and H. Scholermann, "Improvement of the light collection in scintillation detectors," *IEEE Transactions on Nuclear Science*, 1979.

[29] C. S. Sosa, S. J. Thompson, D. L. Chichester, S. D. Clarke, A. D. Fulvio, and S. A. Pozzi, "Energy resolution experiments of conical organic scintillators and a comparison with Geant4 simulations," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 898, no. April, pp. 77–84, 2018.

[30] G. Hull, S. Du, T. Niedermayr, S. Payne, N. Cherepy, A. Drobshoff, and L. Fabris, "Light collection optimization in scintillator-based gamma-ray spectrometers," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 588, no. 3, pp. 384–388, 2008.

[31] J. D. McGervey, J. Vogel, P. Sen, and C. Knox, "Time resolution measurements with an improved discriminator and conical scintillators," *Nuclear Instruments and Methods*, vol. 143, no. 3, pp. 435–439, 1977.

[32] A. Di Fulvio, T. H. Shin, M. C. Hamel, and S. A. Pozzi, "Digital pulse processing for NaI(Tl) detectors," *Nuclear Inst. and Methods in Physics Research, A*, vol. 806, pp. 169–174, 2015.

[33] S. Agostinelli, J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G. Barrand, F. Behner, L. Bellagamba, J. Boudreau, L. Broglia, A. Brunengo, H. Burkhardt, S. Chauvie, J. Chuma, R. Chytracek, G. Cooperman, G. Cosmo, P. Degtyarenko, A. DellAcqua, G. Depaola, D. Dietrich, R. Enami, A. Feliciello, C. Ferguson, H. Fesefeldt, G. Folger, F. Foppiano, A. Forti, S. Garelli, S. Giani, R. Giannitrapani, D. Gibin, J. Gomez Cadenas, I. Gonzalez, G. Gracia Abril, G. Greeniaus, W. Greiner, V. Grichine, A. Grossheim, S. Guatelli, P. Gumplinger, R. Hamatsu, K. Hashimoto, H. Hasui, A. Heikkinen, A. Howard, V. Ivanchenko, A. Johnson, F. Jones, J. Kallenbach, N. Kanaya, M. Kawabata, Y. Kawabata, M. Kawaguti, S. Kelner, P. Kent, A. Kimura, T. Kodama, R. Kokoulin, M. Kossov, H. Kurashige, E. Lamanna, T. Lampen, V. Lara, V. Lefebure, F. Lei, M. Liendl, W. Lockman, F. Longo, S. Magni, M. Maire, E. Medernach, K. Minamimoto, P. Mora de

Freitas, Y. Morita, K. Murakami, M. Nagamatu, R. Nartallo, P. Nieminen, T. Nishimura, K. Ohtsubo, M. Okamura, S. O'Neale, Y. Oohata, K. Paech, J. Perl, A. Pfeiffer, M. Pia, F. Ranjard, A. Rybin, S. Sadilov, E. Di Salvo, G. Santin, T. Sasaki, N. Savvas, Y. Sawada, S. Scherer, S. Sei, V. Sirotenko, D. Smith, N. Starkov, H. Stoecker, J. Sulkimo, M. Takahata, S. Tanaka, E. Tcherniaev, E. Safai Tehrani, M. Tropeano, P. Truscott, H. Uno, L. Urban, P. Urban, M. Verderi, A. Walkden, W. Wander, H. Weber, J. Wellisch, T. Wenaus, D. Williams, D. Wright, T. Yamada, H. Yoshida, and D. Zschiesche, "Geant4–a simulation toolkit," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 506, no. 3, pp. 250–303, 2003.

[34] C. Burt and D. Ramsden, "The development of large-area plastic gamma-ray spectrometers," *IEEE Nuclear Science Symposium Conference Record*, no. 3, pp. 1186–1190, 2008.

[35] N. Mascarenhas, J. Brennan, K. Krenz, P. Marleau, and S. Mrowka, "Results with the neutron scatter camera," in *IEEE Nuclear Science Symposium Conference Record*, vol. 56, pp. 3368–3371, 2008.

[36] A. Poitrasson-Rivière, M. C. Hamel, J. K. Polack, M. Flaska, S. D. Clarke, and S. a. Pozzi, "Dual-particle imaging system based on simultaneous detection of photon and neutron collision events," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 760, pp. 40–45, 2014.

[37] M. Ruch, J. Nguyen, M. Flaska, and S. Pozzi, "Time Resolution of Stilbene Coupled to Silicon Photomultipliers for use in a Handheld Dual Particle Scatter Camera," in *2015 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, 2015.

[38] M. D. Gerling, J. E. Goldsmith, and J. S. Brennan, "MINER - A mobile imager of neutrons for emergency responders," *2014 IEEE Nuclear Science Symposium and Medical Imaging Conference, NSS/MIC 2014*, 2016.

[39] M. Ruch, W. Steinberger, N. Giha, M. Norsworthy, P. Marleau, and P. S.A., "Handheld Neutron Scatter Camera for Imaging Special Nuclear Material," in *2017 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, 2017.

[40] E. Hecht, *Optics*. Pearson Education, 5th ed., 2017.

[41] M. A. Norsworthy, M. L. Ruch, M. C. Hamel, S. D. Clarke, P. A. Hausladen, and S. A. Pozzi, "Light output response of EJ-309 liquid organic scintillator to 2.86–3.95 MeV carbon recoil ions due to neutron elastic and inelastic scatter," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 884, no. December 2017, pp. 82–91, 2018.

[42] J. Nilsson, V. Cuplov, and M. Isaksson, "Identifying key surface parameters for optical photon transport in GEANT4 GATE simulations," *Applied Radiation and Isotopes*, vol. 103, pp. 15–24, 2015.

[43] E. R. Siciliano, J. H. Ely, R. T. Kouzes, J. E. Schweppe, D. M. Strachan, and S. T. Yokuda, "Energy calibration of gamma spectra in plastic scintillators using Compton kinematics," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 594, no. 2, pp. 232–243, 2008.

[44] G. Pausch, K. Roemer, C. M. Herbach, Y. Kong, R. Lentering, C. Plettner, F. Scherwinski, and J. Stein, "Characterization and calibration of large-volume PVT detectors by backscatter gating," in *IEEE Nuclear Science Symposium Conference Record*, pp. 2214–2219, 2012.

[45] L. Swiderski, M. Moszyński, W. Czarnacki, J. Iwanowska, A. Syntfeld-Kauch, T. Szczśniak, G. Pausch, C. Plettner, and K. Roemer, "Measurement of Compton edge position in low-Z scintillators," *Radiation Measurements*, vol. 45, pp. 605–607, 2010.

[46] M. L. Ruch, M. Flaska, and S. A. Pozzi, "Pulse shape discrimination performance of stilbene coupled to low-noise silicon photomultipliers," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 793, pp. 1–5, sep 2015.

[47] A. Levin and C. Moisan, "A more physical approach to model the surface treatment of scintillation counters and its implementation into DETECT," in *1996 IEEE Nuclear Science Symposium. Conference Record*, 1996.