SIMULATION AND IDENTIFICATION OF NON-POISSONIAN NOISE TRIGGERS IN THE ICECUBE NEUTRINO DETECTOR

by

MICHAEL JAMES LARSON

DAWN WILLIAMS, COMMITTEE CHAIR
PATRICK TOALE
CONOR HENDERSON
PATRICK KUNG

A THESIS

Submitted in partial fulfilment of the requirements
for the degree of Master of Science
in the Department of Physics
in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2013

Abstract

The IceCube neutrino detector, located in the clear glacial ice at the South Pole, completed construction in 2011. The low-energy infill extension, DeepCore, forms a denser sub-detector using higher quantum efficiency photosensors. DeepCore has been taking data since May 2010 and lowers IceCube's energy threshold to about 10 GeV. These low-energy events are dim compared to higher energy events, necessitating the study of low-light backgrounds.

While Monte Carlo predictions give an expected rate of approximately 6 Hz due to atmospheric muons, DeepCore records a significantly higher rate of 13.5 Hz with most of the discrepancy due to unsimulated noise events. Much of the rate difference may be resolved by rejecting especially dim events by counting the number of locally coincident hits, retaining 55% of the ν_e and 62% of the ν_μ events sampled with energies of 10-300 GeV while rejecting 96% of noise events. However, differences in the timing distributions of noise hits indicates a need for further study. A new source of correlated noise has been discovered, necessitating an updated noise simulation model. IceCube's new noise generator is able to reproduce the correlated noise in both IceCube and DeepCore sensors. A Metropolis-Hastings algorithm has been used to identify relevant parameters for nearly all of the 5160 sensors that make up the IceCube detector. Initial low quality fits reduce the rate discrepancy between data and simulation from 32% using a Poissonian noise model to 20% using the updated noise model with additional reduction possible.

Noise which triggers the DeepCore detector is evaluated and rejected using the NoiseEngine filtering module. Minimal cleaning removes 94% of noise triggers while retaining 85% of ν_e and 87% of ν_μ events with energies of 10-300 GeV. Stringent cleaning removes 99.9% of noise triggers while retaining only 55% of ν_e and 60% of ν_μ signal events in the same energy range.

Acknowledgements

I would like to thank Dr. Dawn Williams for her help and suggestions during my time at the University of Alabama. Her fastidious attention to detail and thorough understanding of the detector has provided invaluable insights throughout both my research and writing. By providing me the freedom to choose my own topics and the guidance necessary for each study, Dawn has helped me to complete a project far beyond what I had expected. Dawn's willingness to support me throughout this process has been humbling, particularly over the last few months. I have enjoyed working with her during these years and look forward to collaborating in the future.

I would also like to thank Dr. Patrick Toale, who I have known since I started working with the IceCube detector in 2009. Pat was one of my first mentors in IceCube and it seems fitting that he helped guide me through my Master's degree. His advice on classwork and research has always been welcome. Pat's ability to explain challenging concepts has helped give me a far better understanding of the research process. Our conversations have ranged from focused and insightful to far-flung and entertaining. I can only hope that I'll be as good of a mentor someday.

Dr. Jason Koskinen has been both a friend and mentor over the years. His readiness to trust my judgement and his critical eye have molded me into the confident person I am. Between Jason, Dawn, and Pat, I couldn't have asked for a better set of scientific role models. As I continue my career, I can only hope that working together in Copenhagen will help Jason and I both grow as researchers.

My committee also deserves my deepest gratitude. Pat, Dawn, Dr. Conor Henderson, and Dr. Patrick Kung have all been nothing but helpful, particularly on such short notice. Their willingness to see me through this process has been astonishing and I owe each of them my thanks.

Many others have helped me through this degree. Without the help of my fellow students, I could not have reached this point. Broxton Miles, Digesh Rout, Josh Jones, Jamileh Beik Mohamadi, Donglian Xu, James Pepper and Rose Loucks have all helped me through this process. We've all shared the struggles and joys of graduate school, but the support of so many good friends makes life much more enjoyable. I hope to keep in touch with each of them as I continue on my career path.

My friends who are not students at the University of Alabama have been a special help to me throughout the last few years. There are many people to name and I'm grateful to all of them. I'd like to specifically thank Ashley Schneider and Laura Blain, whose late night messages have given

me more joy than I can explain. Both have always been around through the most trying of times and I'll never be able to thank them enough for their support.

Finally, I'd like to thank my family. I've been through quite a bit over the years, but their love and support has been unceasing. I know that my work has always been esoteric and remote, but your willingness to support all of my career choices has been inspiring. Thank you for putting up with my flaws and helping me to build a wonderful life. I only hope that I've made you proud.

Contents

List of Tables

List of Figures

Chapter 1: Introduction

IceCube is a neutrino telescope deployed within the glacial ice at the South Pole (Achterberg et al., 2006). The IceCube detector, shown in Figure 1.1, was built to detect neutrinos - elementary neutrally charged leptons - of all three known flavors. The detector consists of 5160 **D**igital **O**ptical **M**odules (**DOM**s) deployed on 86 cables known as strings. Each DOM houses a 10 inch **p**hoto**m**ultiplier **t**ube (**PMT**), a mainboard containing the digitization and communication electronics, 12 LEDs used for calibration purposes, and a high-voltage generator to power the PMT stored within a glass pressure vessel (Abassi & et al., 2010).

The IceCube detector is split into two regions with differing PMT densities. In the outer IceCube region, strings are spaced approximately 125 meters apart horizontally and DOMs are spaced 17 meters apart vertically on each string. (Abassi & et al., 2010) The inner **DeepCore** region consists of **H**igh **Q**uantum **E**fficiency (**HQE**) DOMs arranged in a denser configuration with strings spaced 72 meters apart horizontally and DOMs are separated by 7 meters vertically on each string (Abbasi et al., 2012). Because the region has a higher PMT density, DeepCore increases IceCube's sensitivity to lower energies than would be possible with the standard IceCube detector (Abbasi et al., 2012).

The low energy threshold for neutrino interactions in DeepCore is about 10 GeV (Abbasi et al., 2012). These events produce relatively little light via Cherenkov emission in the detector and therefore weak signals are recorded from their interaction. Because DOMs may register events from both particle interactions as well as detector noise, the low brightness of these events provide a particular challenge for analysis. Even a small amount of spurious detector noise can mask an interaction, pulling particle trajectory reconstructions away from their true values (Aartsen & et al., 2013).

The work within this thesis focuses on modelling, simulating, and removing events caused purely by detector noise. Some background information is necessary for a thorough explanation. To this end, Chapter 2 explains the expected signals for IceCube analysis of neutrinos (2.2) and muons

(2.1).

Chapter 3 consists of a summary of the IceCube hardware, describing the photomultiplier tube (3.2), the digital optical module (3.1), and local coincidence conditions (3.3).

Chapter 4 begins a discussion of the IceCube software environment known as IceTray. Following a brief introduction to IceTray (4.1), the simulation of events will be described in detail (4.2), including detector effects (4.2.4) and triggering (4.2.5).

Chapter 5 will introduce the empirically observed correlated noise and explain the problems underlying the standard Poissonian noise model (5.1). Section 5.3 introduces an empirical model for the correlated noise component.The new Vuvuzela noise module is discussed in detail in Section 5.4 with the new parameter fitting procedure used to identify model parameters given in Section 5.5. The results obtained with the new noise model are shown in Section 5.6.

Chapter 6 will explain the characteristics of noise triggers in DeepCore (6.1). NoiseEngine, a filtering algorithm to reject noise triggers, is described (6.2) and optimized (6.3), with the final results presented (6.4).

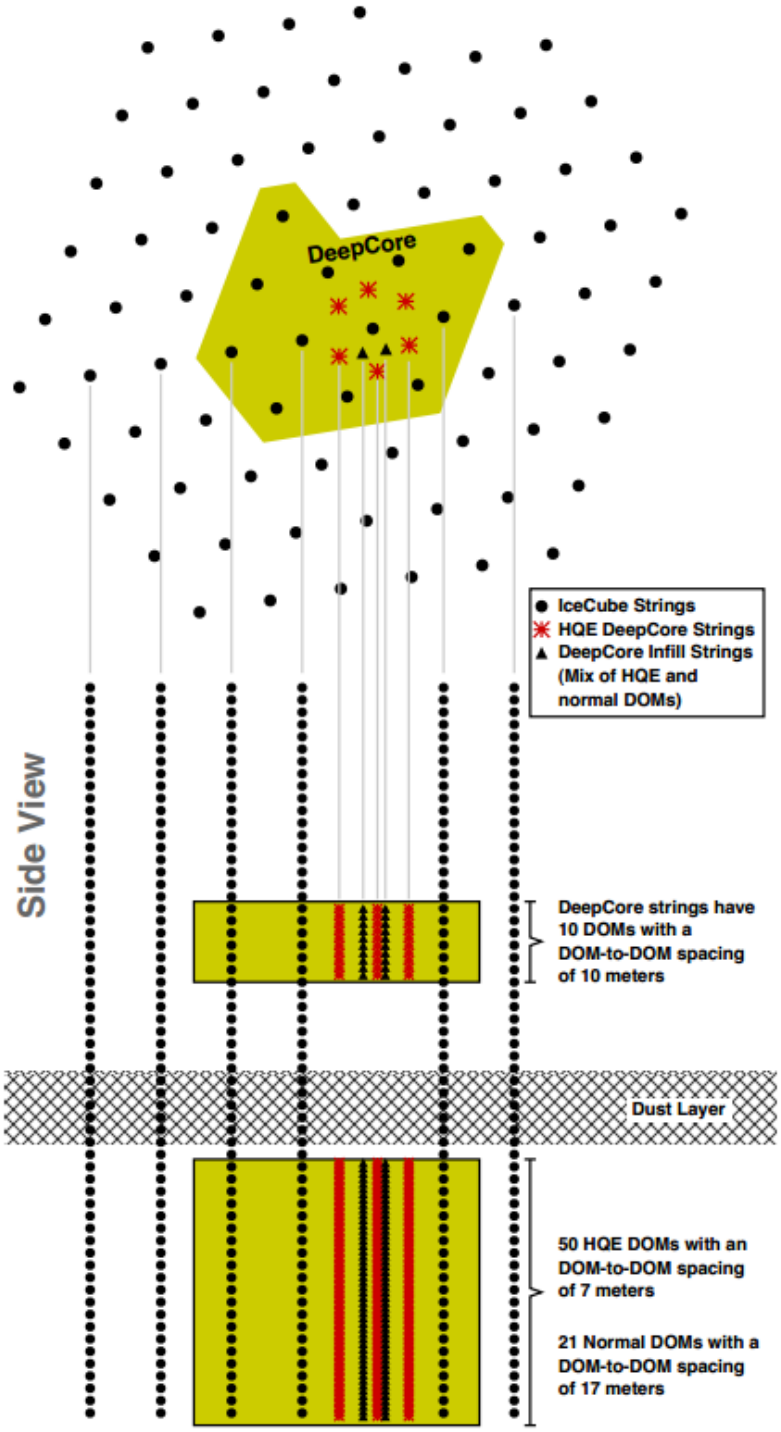


Fig. 1.1.: The geometry of the IceCube detector. The IceCube array consists of 86 strings of 60 optical sensors each. Strings are separated by 125 m in the standard IceCube array and 72 m in the DeepCore extension. DeepCore consists of the innermost strings instrumented with high quantum efficiency DOMs and is colored yellow. A "cap" of DOMs above DeepCore provides better identification of muons.

Chapter 2: Optical Signals Visible in DeepCore

The IceCube detector is designed to observe neutrinos of any type. The primary background for neutrino interactions is muons produced in cosmic ray interactions with the atmosphere (Aartsen et al., 2013b; Abbasi et al., 2012; Achterberg et al., 2006). Each type of signals has a unique topology that may be used for partial particle identification.

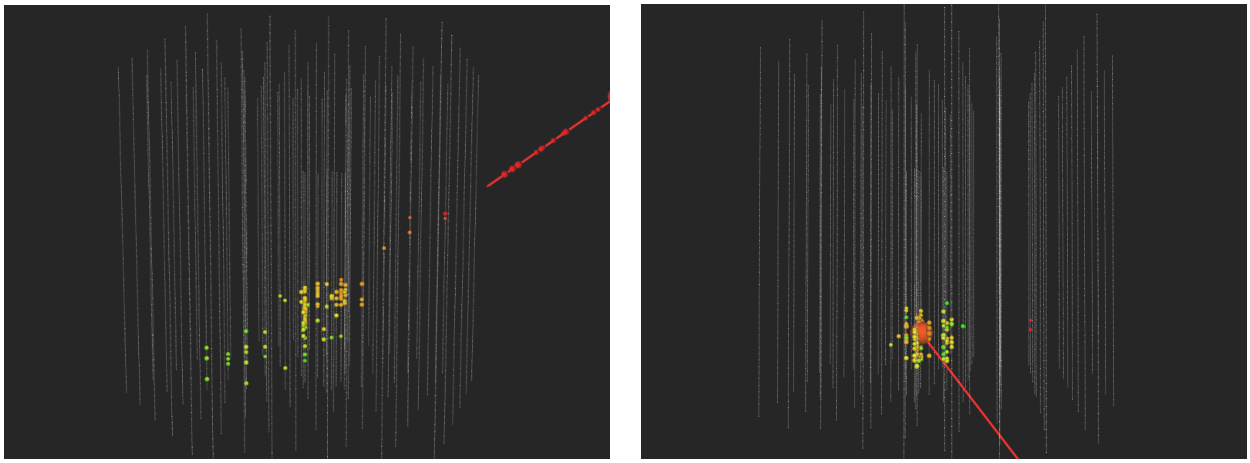
2.1 Atmospheric Muon Signals

Muons are stopped by the dense rock of the Earth, but may travel significant distances through the glacial ice at the South Pole (Achterberg et al., 2006; Beatty et al., 2011). Muon signals are therefore only observed from the Southern Hemisphere sky and are considered "down-going" in IceCube.

Muons interact in the ice through various processes, producing long tracks from continuous energy loss as well as discrete energy loss in the form of stochastic processes. The "track-like" topology shown in Figure 2.1 is used for vetoing purposes in the IceCube detector (Abbasi et al., 2012).

2.2 Neutrino Signals

High energy neutrinos have two possible interactions. The first, known as a "neutral current" interaction, involves the exchange of a neutral Z boson. The incident neutrino interacts, producing an outgoing neutrino carrying away some of the incident energy. The remaining energy is deposited in the target nucleus, disrupting it. The relativistic daughter particles then interacts with the medium via Cherenkov radiation to produce an approximately isotropic light distribution known as



(a) μ Track-like Event (b) ν Cascade-like Event

Fig. 2.1.: Events in IceCube are identified as track-like and cascade-like. DOMs not recording hits are in gray while those showing hits are colored. Red indicates hits earlier in time while blue represents hits occurring later in time.

a "cascade" (Bichsel et al., 2011). The signature of a neutral current interaction is common to all neutrino flavors.

The other potential interaction is the "charged current" interaction. This involves the exchange of a charged $W^{\pm}$ boson with a nucleus. Satisfying charge conservation, the charged current interaction must also transform the uncharged neutrino into a charged lepton. All energy is deposited into the ice, producing Cherenkov radiation both due to the hadronic cascade from the initial interaction as well as due to the outgoing charged lepton.

The topology of a charged current interaction depends on both the incident neutrino energy as well as the flavor. Electron neutrinos produce electrons via the charged current interaction. These electrons have a short interaction length in ice and are typically indistinguishable from the initial cascade. Muon neutrinos, on the other hand, will produce muons which may travel significantly beyond the initial cascade, depositing energy in a track-like pattern. Tau neutrinos with energies less than 100 GeV should be visible to DeepCore and will produce a cascade-like signature. Both track-like and cascade-like event topologies are shown in Figure 2.1.

Unlike muons, neutrinos have no continuous energy losses and may travel through the Earth (Beatty et al., 2011). Neutrinos may therefore be "down-going" or "up-going". Neutrino interactions may begin in the IceCube detector, producing events fully contained in the detector, or part of the event may occur outside of the detector.

In the DeepCore detector, the primary neutrino source is neutrinos produced in the atmosphere

following cosmic ray interactions (Abbasi et al., 2012). These interactions produce muon and electron neutrinos at low energies in a 2:1 ratio with the spectrum shown in Figure 2.2. The atmospheric neutrinos may oscillate, changing from one flavor to another during transit. This effect is particularly important for energies visible to the DeepCore detector. This oscillation is included for all weighting effects of DeepCore analyses.

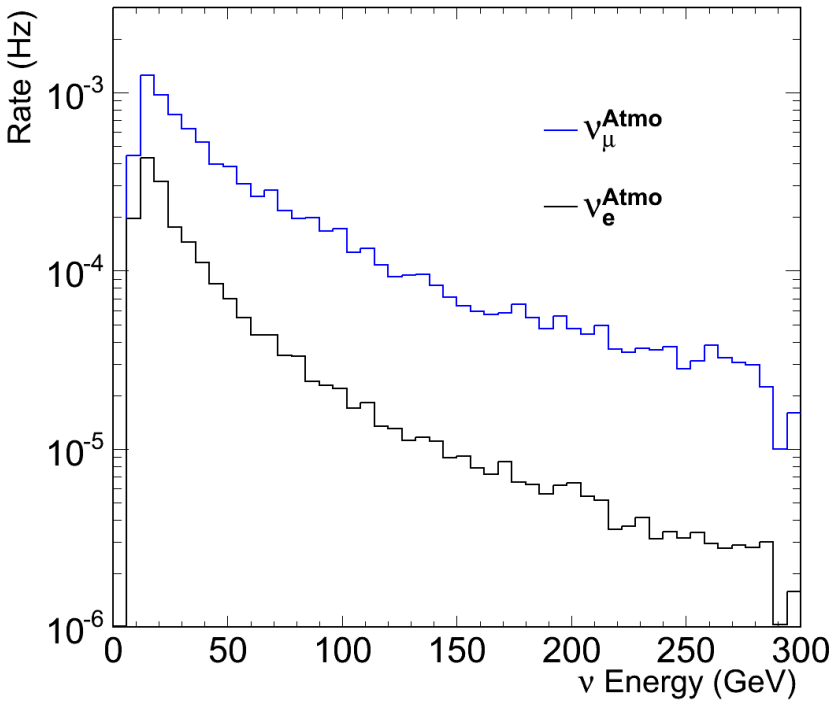


Fig. 2.2.: The spectrum of atmospheric neutrinos interacting in DeepCore.

Chapter 3: Hardware of the IceCube Detector

3.1 The Digital Optical Module

The basic sensor of IceCube is the **D**igital **O**ptical **M**odule (**DOM**) which includes a 10 inch diameter R7081-02 photomultiplier tube (PMT) produced by Hamamatsu Photonics (Achterberg et al., 2006). The PMT is optically coupled to a borosilicate glass pressure housing. Each DOM also includes a DOM mainboard containing digitization electronics, 12 LEDs used for *in situ* calibration, and a high voltage generator. DOMs identified as **H**igh **Q**uantum **E**fficiency (HQE) use a "super bialkali" photocathode which improves the quantum efficiency by about 35% (Abbasi et al., 2012).

Also included are two types of digitizers to record PMT signals (Abassi & et al., 2010). The first, the **A**nalog **T**ransient **W**aveform **D**igitizer chip (**ATWD**) possesses four analog inputs. Three of these are operated with different signal amplification levels, reducing the chance of saturation for bright signals, while the fourth is used for calibration. The ATWD stores 128 samples with a bin size of 3.3 ns each, providing 422.4 ns of information about PMT signals. The ATWD digitizes the samples when a PMT signal reaches the discriminator threshold and satisfies local coincidence conditions (see Section 3.3). The ATWD digitization process can produce an effective deadtime of 30 μs. Two chips are included to limit the effect of this deadtime.

The second type of digitizer is the **F**ast **A**nalog to **D**igital **C**onverter (**FADC**). The FADC stores 256 samples of 25 ns each, providing waveform information for 6.4 μs. The FADC has no deadtime and charge information may be stored for any hit possessing at least 0.25 photoelectrons of charge.

The R7081-02 PMT has 10 dynode stages operated at a gain of 10^7. The peak quantum efficiency as measured by Hamamatsu in laboratory settings is 25% at 390 nm. The effective efficiency of

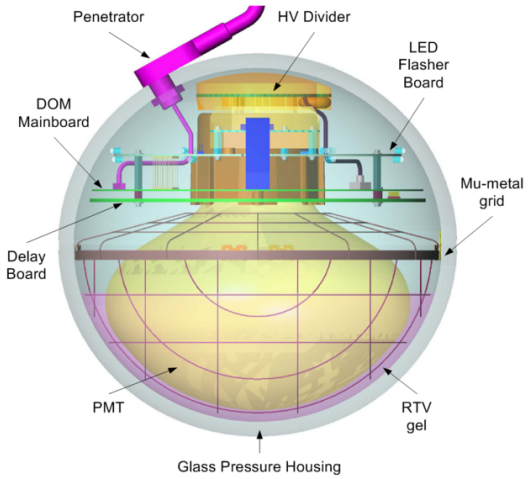


Fig. 3.1.: A schematic view of the IceCube DOM containing a 10 inch photomultiplier tube, a mainboard, 12 LEDs for calibration, a high voltage generator and the borosilicate pressure housing.

DOMs is modified in the ice. Absolute calibration of the total efficiency of deployed DOMs is currently ongoing.

3.2 PMT Calibration

3.2.1 Dark Rate of IceCube PMTs

At room temperature, the dark current - defined as the amount of current measured by the PMT in absence of light - follows Richardson's law (Richardson, 2003). In this model, thermal electrons are emitted from a high temperature surface spontaneously as they overcome the metal's work function, W. This "thermionic" dark current rate, J, is a Poisson process strongly sensitive to the temperature of the metal, T, and falls rapidly with decreasing temperature according to Equation 3.1.

$$J(T) \propto T^2 e^{-\frac{W}{K_b T}} \tag{3.1}$$

The IceCube DOMs are operated in the temperature range of -40° C to -20° C. At these temperatures, the thermal dark count rate is suppressed relative to the room temperature rate. The dark count rate reaches a minimum near -10° C, but increases at lower temperatures (Abassi & et al., 2010). This increase is a well-known property of photomultiplier tubes at low temperatures, although no consensus yet exists on the source. The base rate is believed to be due to radioac-

tive decays in the glass of both the PMT and pressure sphere while the low-temperature increase is thought to be caused by an enhancement of low-quality scintillation and Cherenkov light production in the glass. For more information, see Section 5.1.

The R7081-02 PMT is specified to have a typical dark current rate of 500 Hz when operated at -40° C. Measurement of the noise rate of individual DOMs was accomplished by collecting data for 10 minutes in Abassi & et al. (2010). Measurement of IceCube noise rates assumes an uncorrelated Poisson process model typical of thermionic emission.

3.2.2 Pre-pulsing and After-pulsing in IceCube DOMs

Pre-pulses occur when light bypasses the first dynode, creating a pulse that precedes the main signal pulse. After-pulsing generally occurs due to the ionization of residual gases within the vacuum of the PMT. These ions are then accelerated toward the photocathode and cause secondary electron emission. The timing of the after-pulses occurs as a function of both the mass of the ionized particle and the geometry of the PMT.

Laboratory measurements of a PMT were performed using an LED source. Initial tests at 25° C showed after-pulsing peaks in the range of 300 ns to 11 μs with 40 ns widths (Abassi & et al., 2010). Gaussian fits to each of the after-pulsing peaks measured show the most prominent after-pulsing peaks observed at 600 ns, 2 μs, and 8 μs after the main peak for the tested PMT.

3.3 Local Coincidence between DOMs

Once a DOM observes a signal passing the discriminator threshold of 0.25 photoelectrons, the DOM produces a "launch". Local coincidence is checked by listening for coincident signals on nearby DOMs on the same string. If another launch is registered within two DOMs and within ± 1 μs of the first, then the DOM identifies the launch as a **H**ard **L**ocal **C**oincidence (**HLC**) hit. Otherwise, if the launch is located within a trigger readout window, the launch is identified as a **S**oft **L**ocal **C**oincidence (**SLC**) hit. Only HLC hits have complete waveforms digitized by the DOM, although SLC hits record a charge stamp containing the bin with highest amplitude and two nearest bins from the FADC. Local coincidence allows the bandwidth-limited IceCube data cables to handle the communication demands of the detector.

Chapter 4: The IceTray Software Environment

4.1 The IceTray Software Framework

The IceCube software environment, known as **IceTray**, is a modular base consisting of C++ code (**modules**) accessed via a Python interface (**scripts**) through Boost C++ libraries. Modules may be added to scripts with their own specific configuration options to allow limited runtime customization. Common options allow conditional execution, changes to the names of objects to be processed, and settings specific to the given module. During processing, each event is processed sequentially through the the series of modules specified in the given Python script.

IceCube events are stored in **frames**, which hold timing, charge, reconstructions, and other information used in analysis. Frames are typically preceded by files containing **G**eometry, **C**alibration, and **D**etector status information known as **GCD** files. These files store generic information regarding the detector such as noise rates, coincidence settings, temperature, etc.

Each type of frame can hold multiple objects. For the purposes of this work, the most important object types are **I**ce**C**ube **M**onte **C**arlo **P**hoto**e**lectrons (**I3MCPE**s), I3DOMLaunches, and I3RecoPulses described in Table 4.1. The first is a structure representing a single photoelectron impinging on the photocathode of a PMT before detector effects are applied and therefore only occurs as an intermediate step during simulation. These detector effects include pre-pulses, after-pulses,

Table 4.1.: The different types of hits used in IceCube. I3MCPE only occurs in simulated events while I3DOMLaunch and I3RecoPulse occur in both simulated events and data. Any of the three classes may be referred to as "hits".

Hit Type	**Description**
I3MCPE	Simulated photoelectrons before PMT effects are applied
I3DOMLaunch	Packaged HLC/SLC hit information before waveform extraction
I3RecoPulse	Hit information following waveform extraction

the discriminator, waveform shaping, and digitization.

The I3DOMLaunch object represents a single digitization event from a DOM and is used for low-level detector readout. The I3RecoPulse object is the software attempt at reconstruction of the original photoelectrons from the digitized waveform recorded by the DOM. In this sense, the I3RecoPulse is a representation of the I3MCPE after detector effects are applied.

4.2 The IceCube Simulation Chain

The IceCube simulation chain consists of modules that depend on the type of particle being simulated. In general, the simulation chain may be broken down into production, propagation, noise generation, detector effects, and triggering.

4.2.1 Atmospheric Muon and Neutrino Production

IceCube simulation of particles relies on various software systems. For atmospheric muon interactions, the **Co**smic **R**ay **Si**mulation for **KA**SCADE (**CORSIKA**) (Kampert & KASCADE-Collaboration, 2002) software environment developed by the KASCADE cosmic ray experiment is typically used. The software simulates atmospheric interactions and muon showers which are then propagated toward the IceCube detector. Various models are available, although this work uses the energy spectrum given by Hörandel (2005) and Beatty et al. (2011). The CORSIKA output may be read into the IceTray environment using the UCR-icetray module.

Neutrino simulation depends on the energy level desired. For energies greater than approximately 200 GeV, IceCube uses the I3NeutrinoGenerator module which implements high energy approximations to the neutrino interaction cross section. The I3NeutrinoGenerator module is based upon the earlier ANIS neutrino generator (Gazizov & Kowalski, 2005). For energies in the 1 GeV to 300 GeV range relevant to DeepCore, IceCube has adopted the **G**enerates **E**vents for **N**eutrino **I**nteraction **E**xperiments (**GENIE**) neutrino generation software (Andreopoulos et al., 2010).

4.2.2 Propagation of Light in IceCube

Once neutrinos and atmospheric muons are produced, they must be propagated through the ice and Earth in order to produce detectable photons in the form of I3MCPEs. Photon emission and

propagation is handled by the **P**hoton **P**ropagation **C**ode (**PPC**) module. The PPC module used to propagate photons through the ice relies on the capabilities of GPU processors and the Nvidia CUDA and OpenCL framework to simulate individual photon interactions in the ice and create I3MCPEs (Chirkin, 2013).

An ice model characterizing the scattering properties of the ice is required for propagation of light in the detector. The SPICE-Lea model adopted for this work also describes tilt in the ice and anisotropy in the scattering. The SPICE-Lea ice model is an updated model based on the SPICE-Mie model shown in Figure 4.1 and derived in Aartsen et al. (2013a).

Once propagated, events are passed through a filter that tests whether any I3MCPEs were created by the event. If no DOM records an I3MCPE, then the event is not processed further. All neutrino and atmospheric muon events are therefore guaranteed to contain at least one I3MCPE due to particle interactions.

4.2.3 Noise Simulation

In the past, IceCube has used a Poissonian thermionic noise model. The rates for each DOM were measured before deployment into the ice and periodically after freezing. Each DOM has an individually measured Poissonian noise rate stored in the GCD file that is used for simulation in the **noise-generator** module. The number of photoelectrons observed in a given window is drawn from a Poisson probability distribution following Equation 4.1 for observing x hits given a rate parameter λ and simulation time window length Δt.

$$f_{Uncorrelated}(x) = \frac{(\lambda \Delta t)^x}{x!} e^{(-\lambda \Delta t)} \tag{4.1}$$

IceCube typically simulates noise for 10 μs before and after existing I3MCPEs due to particle interactions in the ice. This allows later software to return a time window wider than the particle interaction time.

4.2.4 Detector and DOM Effects

After the noise is added to the photoelectrons produced by particle interactions in the ice, all photoelectrons are stored as I3MCPEs. For each existing I3MCPE, pre-pulses, after-pulses,

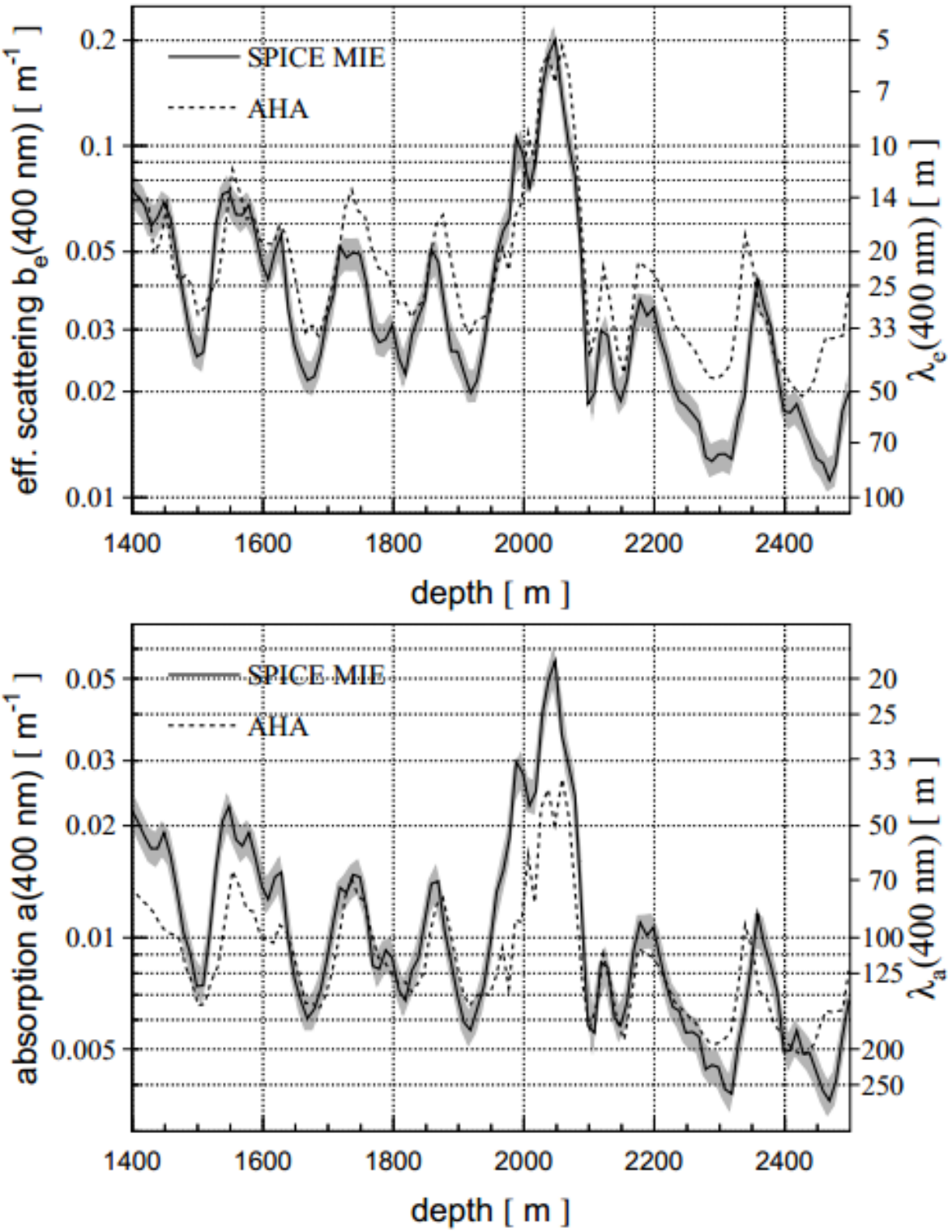


Fig. 4.1.: The scattering and absorption properties of the ice described by the SPICE-Mie ice model (Aartsen et al., 2013a) and the AHA ice model (Ackermann et al., 2006). The depth is measured from the surface. A dust layer is visible from 2000-2100 m below the surface. DeepCore DOMs are below this level. For this work, an updated version of the SPICE-Mie ice model is adopted.

and late pulses are simulated recursively with probability 0.7%, 3.5%, and 5.93% respectively. The DOMLauncher module simulates the DOM, processing these photoelectron hits through the

discriminator, coincidence, and digitizer logic to produce I3DOMLaunches.

4.2.5 Triggering of IceCube Data

The I3DOMLaunches are processed by various triggers in order to build "events". The trigger used by the IceCube detector is the **S**imple **M**ultiplicity **T**rigger (**SMT**) with a multiplicity of 8 (Achterberg et al., 2006). The SMT8 trigger searches the detector for 8 HLC hits within 5 μs. These hits may occur on multiple strings in any combination, but each must satisfy the HLC conditions.

For the DeepCore detector, a modified version of the SMT8 trigger is used. Introduced in Abbasi et al. (2012), the DeepCore SMT3 trigger searches the DeepCore fiducial volume for at least three HLC hits within a window of 2.5 μs. These hits may also be arranged on a single string or across multiple strings.

Once triggered, the IceCube detector records data for all I3DOMLaunches within 10 μs of the triggering HLC hits. All triggers with overlapping time windows are merged, forming discrete events. The stream of events undergoes calibration and simple reconstructions before being transferred to the IceCube data center in Madison, Wisconsin.

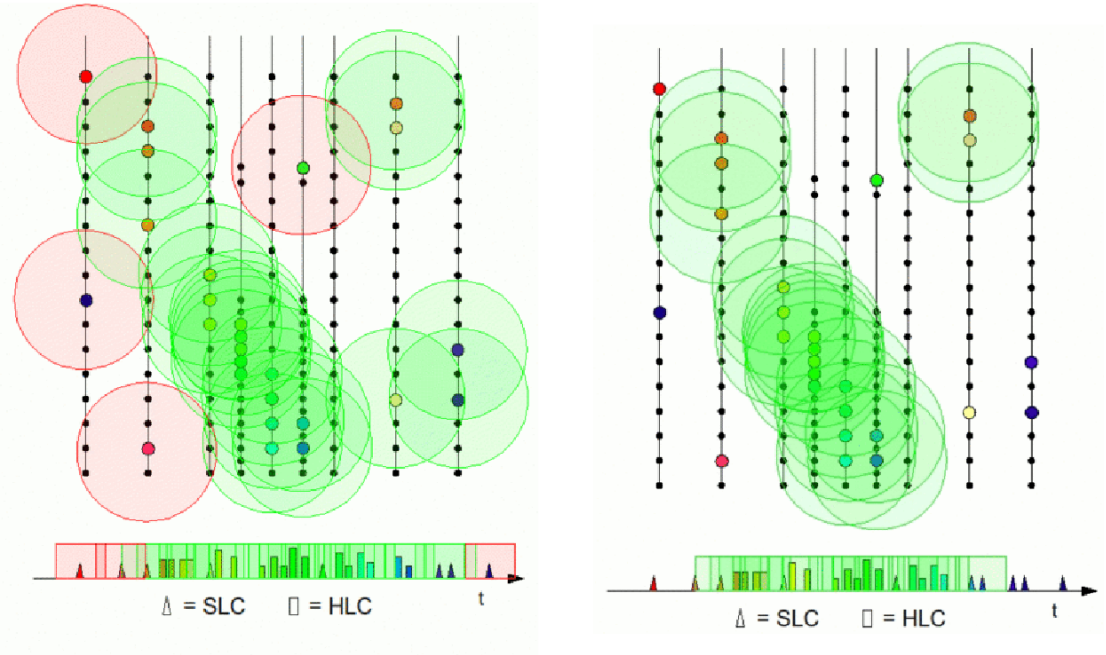
4.3 Pulse Extraction

After triggering, I3DOMLaunches for both data and simulation may be extracted following the prescription of Aartsen & et al. (2013). Waveforms digitized for HLC hits are fit to determine the charge and timing of each pulse. SLC hits, which do not have full waveforms, are assigned a charge and timing based on the three recorded bins of the waveform. Pulses are stored in the I3RecoPulse class and are used for most processing.

4.4 Hit Cleaning Methods

HLC hits rely on inter-DOM coincidence to identify hits likely to be caused by nearby light in the detector. Because of this, HLC hits are generally unlikely to be caused by noise in the detector. HLC hits are therefore a useful first-guess cleaning to eliminate noise, but the strict conditions required also remove some information useful for reconstruction.

A looser condition can be applied to regain some of the lost information. The **R**adius-**T**ime cleaning algorithm (**RTCleaning**) is one such module in the IceTray software environment and is shown schematically in Figure 4.2. Using the I3RecoPulses extracted, each pulse may be compared to one another to identify coincidence. Any pulse occurring within a given radius and time of one another are recorded. Clusters of pulses are assigned a multiplicity which may be used to modify the stringency of the cleaning algorithm. The empirically chosen radius and time used by the algorithm are $R = 150$ m and $T = 1\mu$s, effectively extending the HLC conditions to nearby standard IceCube strings. For comparison, the distance travelled by light in ice over 1 μs is about 200 m. The selected radius and time are therefore consistent with light travelling between neighboring strings in the IceCube detector.



(a) RTCleaning (b) SeededRT

Fig. 4.2.: Left:The RTCleaning module checks all DOMs for coincidence with any other DOMs within a radius R and time T. If the pulse is coincident with at least some threshold number of other pulses, then it is accepted into the cleaned pulse series. Right:The SeededRTCleaning module begins with all HLC hits. Hits satisfying the radius and timing requirements are added to the HLC hits and the combined hits are iteratively seeded until the accepted set of hits converges. Note that the three SLC hits on the bottom right are no longer accepted because they are isolated, but the HLC hits in the top right are retained.

Noise hits may cause clusters with some probability, however. To combat this, the **SeededRT** module, shown in Figure 4.2, is used. While the RTCleaning algorithm checks all possible DOMs for neighboring hits, the SeededRT module instead begins the search with a subset of DOMs,

typically those satisfying the stringent HLC condition. The module then checks nearby hits using the RTCleaning algorithm, adding SLC hits to the HLC hits as a cleaned hit map. The cleaned hits are then used for a new iteration of coincidence checking, iteratively expanding the search to include all hits that form clusters connected to HLC hits and rejecting SLC hits that are spatially or temporally isolated.

Finally, simple cleaning based on the trigger timing, known as **T**ime **W**indow **Cleaning** (**TWCleaning**) is also used. All hits earlier than 6 μs before the trigger or later than 4 μs after the trigger are removed. Particle interactions are unlikely to produce light beyond these times and so TWCleaning provides a simple method of reducing noise hits at the edges of each frame.

4.5 DeepCore Filter

The main filter used for the identification of DeepCore events is known as the DeepCoreFilter (Abbasi et al., 2012). The filter, shown in Figure 4.3, accepts an I3RecoPulse series that has been cleaned using the SeededRT module. The hits are separated by location into DeepCore Fiducial and Veto region hits using pre-existing geometry defined by the yellow region in Figure 1.1.

Hits in the DeepCore region are averaged in time and outliers are removed. A **C**enter **o**f **G**ravity (**CoG**) is then calculated based on the positions of hit DOMs.

$$\vec{x}_{CoG} = \frac{1}{n_{hit}} \sum_{i}^{n_{hit}} \vec{x}_i \tag{4.2}$$

A "CoG time" estimate is also calculated. Each time is corrected for light travel assuming a isotropic emission from the CoG interaction point.

$$t_{CoG} = \frac{1}{n_{hit}} \sum_{i}^{n_{hit}} \left(t_i - \frac{\parallel \vec{x}_i - \vec{x}_{CoG} \parallel}{c_{ice}} \right) \tag{4.3}$$

where $c_{ice} = 0.221$ m/ns is the group velocity of light in ice. The time and position of the CoG is then used to calculate the apparent velocity between each hit in the Veto region and the CoG.

$$v_{Apparent} = \frac{\parallel \vec{x}_i - \vec{x}_{CoG} \parallel}{t_{CoG} - t_i} \tag{4.4}$$

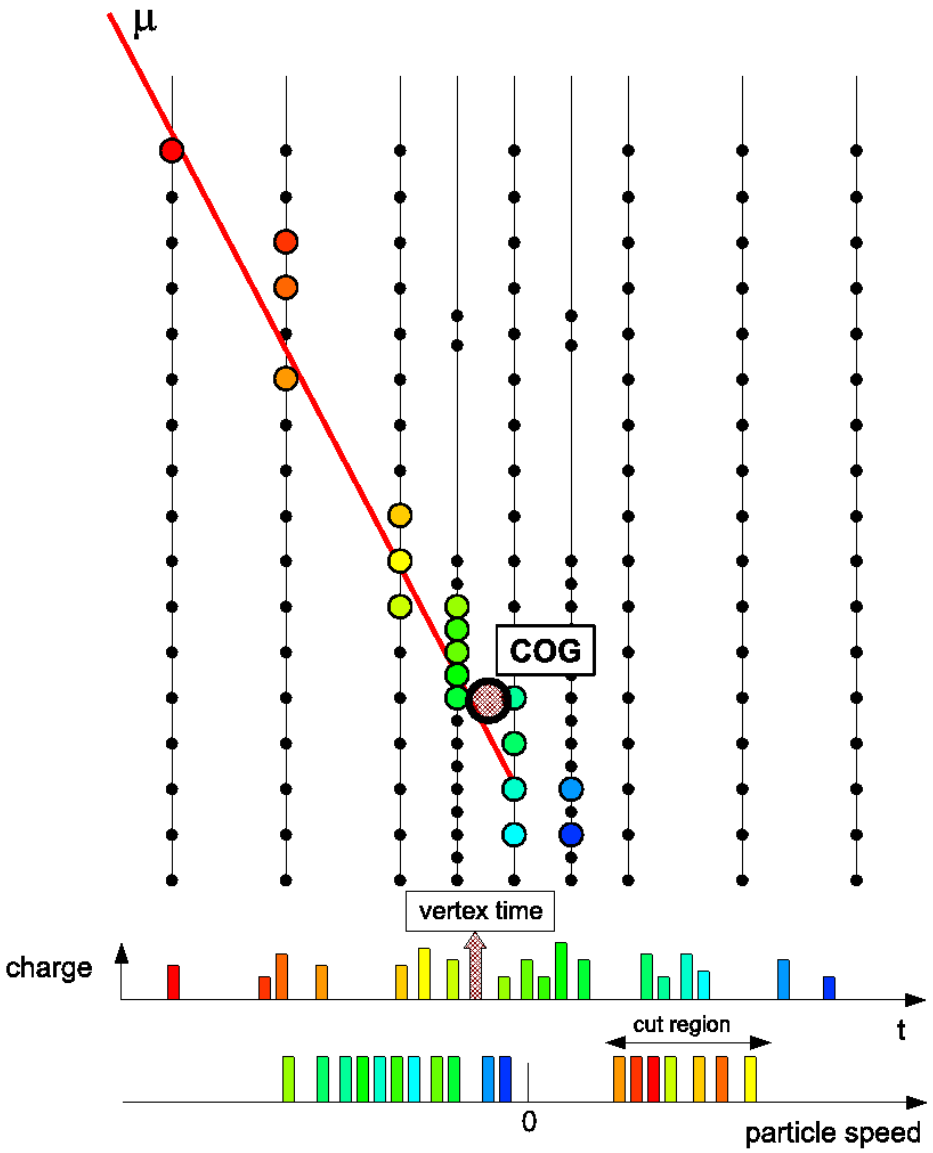


Fig. 4.3.: A schematic view of the DeepCoreFilter. The filter calculates a vertex using hits in the DeepCore fiducial volume before calculating apparent velocities to hits in the larger IceCube detector. If more than one hit in IceCube is causal with the vertex in DeepCore, the event is rejected.

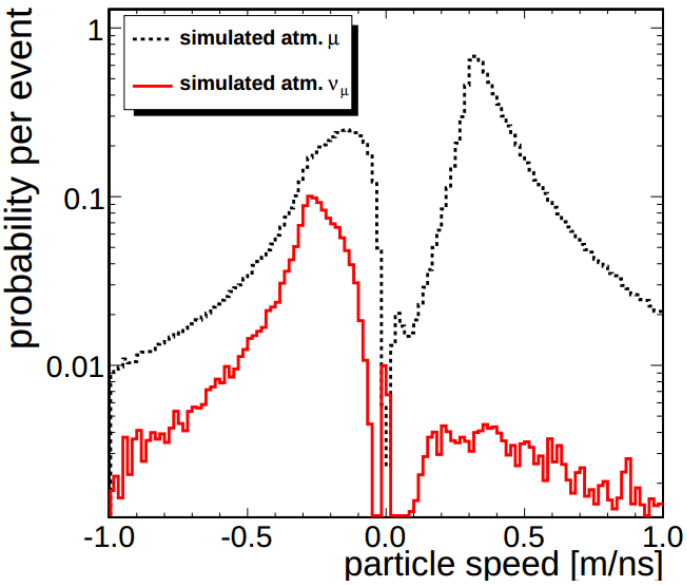


Fig. 4.4.: The velocities calculated during the DeepCoreFilter for muons (black) and neutrinos (red). Negative values indicate the interaction appears to begin in DeepCore while positive values indicate the interaction begins in IceCube.

A collection of these velocities for muon and muon neutrino events is shown in Figure 4.4. The number of hits in the Veto region with apparent velocities within a window of [0.25 m/ns, 0.4 m/ns] are counted. If more than one veto hit has a calculated apparent velocity within this window, then the event is rejected by the filter and is not processed further.

The DeepCoreFilter module runs exclusively on events passing the SMT3 trigger. Together, the DeepCoreFilter and SMT3 trigger reduce the atmospheric muon background by 100x relative to the IceCube SMT8 rate. The resulting background rate following the DeepCoreFilter is measured to be 13.5 Hz from data and 6.38 Hz from simulation (Daughhetee, 2012; Pepper, 2013).

The discrepancy in rate between data and simulation is most noticeable when looking at I3RecoPulses after SeededRT and TWCleaning. After the DeepCoreFilter is applied, the distributions of number of DOMs hit for each event ("NChannel") in simulation and data differ dramatically as shown in Figure 4.5. This shows that the excess events occurring in data appear with very few DOMs hit. This low NChannel excess is believed to be caused by spurious triggers caused exclusively by PMT noise. These triggers may be simulated separate from neutrino and muon interactions and are also shown in Figure 4.5 for reference. Note that the rate of noise triggers generated using the noise-generator module is inadequate to explain the discrepancy.

The discrepancy has historically been addressed by removing all events with fewer than 8 HLC hits in the detector. The selection is chosen due to constraints imposed by later likelihood recon-

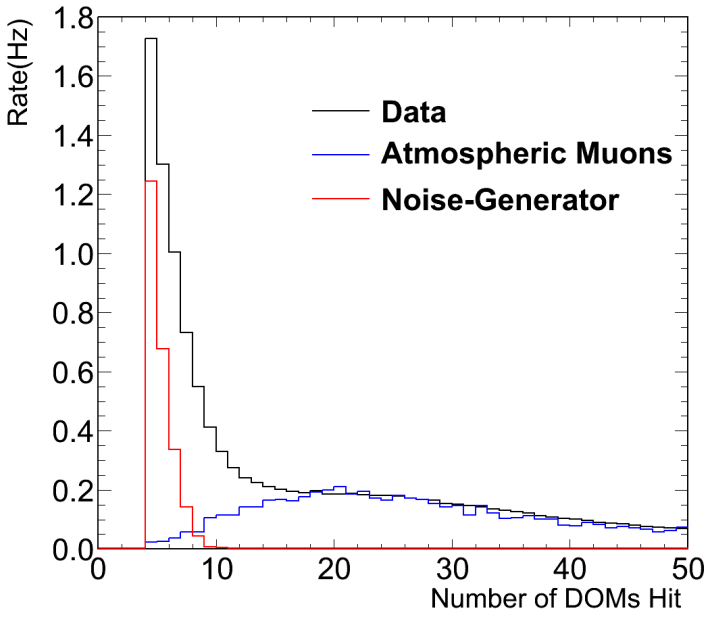
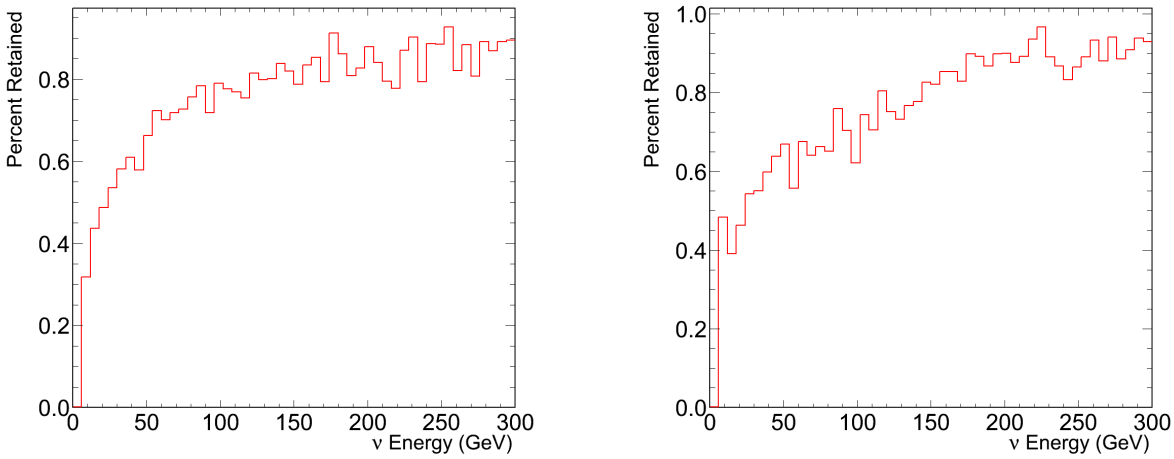


Fig. 4.5.: The number of DOMs hit for events passing the DeepCoreFilter. The hit series used has SeededRT and TWCleaning applied. The discrepancy for dim events is believed to be caused by noise events in the detector. For more information, see Section 6.1. The atmospheric muons include contributions from Noise-Generator produced detector noise.

struction techniques that traditionally have used well-cleaned hits. Removing low-brightness events in this way removes approximately half of all neutrinos while reducing the noise trigger rate to 214 mHz. After the cut, the electron neutrino trigger rate is 1.1 mHz while the muon neutrino rate is 5.6 mHz. The lowest energy neutrinos are preferentially removed. Figure 4.6 shows the percentage of neutrinos removed at each energy. The simulated noise trigger rate after applying this limit remains at two orders of magnitude above the neutrino trigger rate for DeepCore and remains a significant background for low energy analysis.



(a) ν_μ Passing Efficiency

(b) ν_e Passing Efficiency

Fig. 4.6.: Historically, the discrepancy between data and Monte Carlo simulation has been addressed by removing events containing fewer than 8 HLC hits. The efficiency as a function of energy is shown for for ν_μ (left) and ν_e (right). A significant fraction of low energy neutrino events are lost for both flavors of neutrinos.

Chapter 5: The Vuvuzela Noise Generator

5.1 Correlated Noise in IceCube DOMs

The lower energy threshold and fewer required hits for the triggering of DeepCore require an improved understanding of low energy detector systematics. The discrepancy between the data and simulation shown in Figure 4.5 requires detailed study. Identifying the characteristics of the disagreement provides some evidence of the source.

Assuming a Poisson process model, the mean time between hits is well defined and follows an exponential distribution. Measuring the time between hits from detector readout in Figure 5.2 demonstrates that the distribution of the time between hits on a DOM does not follow the expected distribution. This implies that the Poisson process model is an inadequate representation of the noise in IceCube and the noise may be correlated in time on a single DOM. Heereman (2013) showed that the correlated noise hits are not uniformly distributed, but instead occur in "bursts", consistent with other experiments (Meyer, 2010). These bursts, shown in Figure 5.1, occur with timescales of tens of μs, ruling out the possibility of undersimulated after-pulsing.

Work done during the initial planning of IceCube showed a similar behaviour. Helbing et al. (2003) demonstrated that the glass used in AMANDA's pressure spheres produced light when exposed to a radiation source. More recent work at the University of Wisconsin showed that removing the glass from IceCube pressure spheres significantly decreases the noise rate measured (Riedel, 2012). Figure 5.3 shows the noise rate could also be reduced by applying an artificially long deadtime, again indicating that the increased noise rate primarily occurs with very small gaps in time between hits.

The correlated noise observed in IceCube also appears to have a temperature dependence with

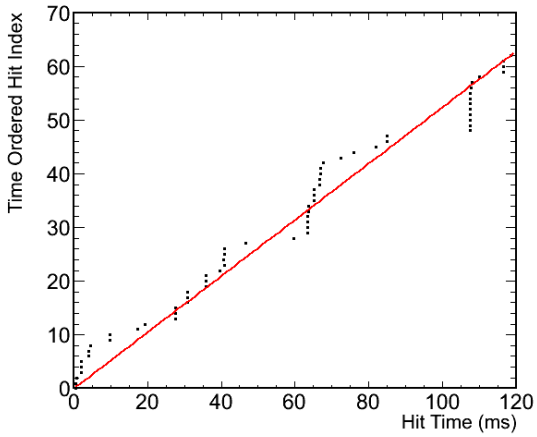


Fig. 5.1.: The hit times on DOM 37-12 for 120 ms. Each hit is ordered in time and assigned an index. The expectation from a Poisson process with the measured rate of 525 Hz is shown in red. An actual sample of hit times is shown in black. Note that the hits appear to occur in "bursts" with large gaps. It is from this behaviour that the "correlated" noise nomenclature arises.

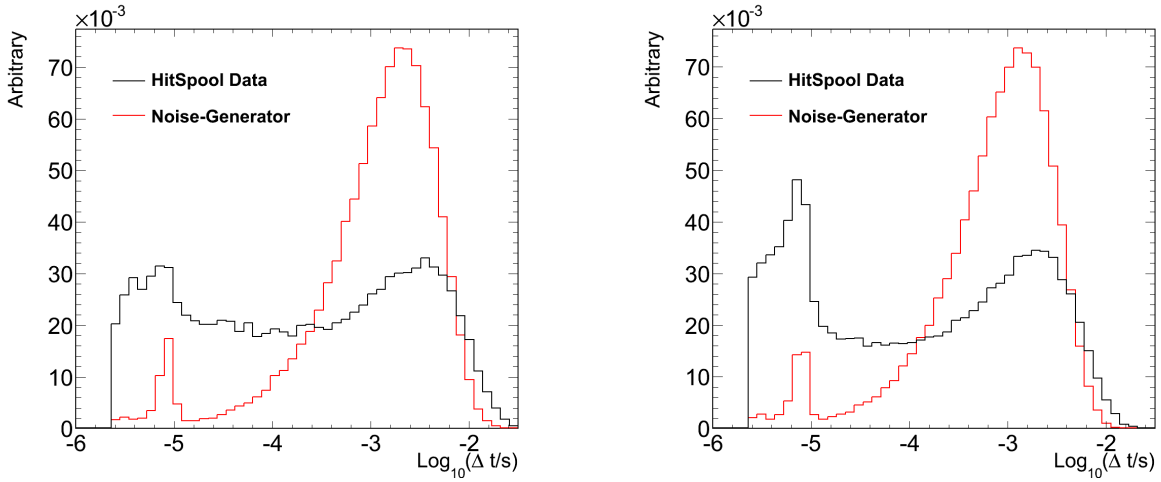
a minimum occurring around -10° C (Abassi & et al., 2010). This effect is well established in experiments using low temperature PMTs (Abassi & et al., 2010; Ankowski & et al., 2006; Briese et al., 2013; Hamamatsu Photonics, 2012; Helbing et al., 2003; Nikkel et al., 2007), although the cause is little understood (Meyer, 2010). Because the operating temperature of the IceCube detector lies in the range of -30° C to -10° C, both the correlated and uncorrelated noise are important.

Hamamatsu Photonics (2012) suggests that the borosilicate glass used in the manufacture of PMTs may be susceptible to radioactive decays and low efficiency scintillation. Applying the same model to the glass pressure spheres used for DOMs indicates that scintillation may be a reasonable explanation for the anomalous correlated noise.

5.2 HitSpool Data

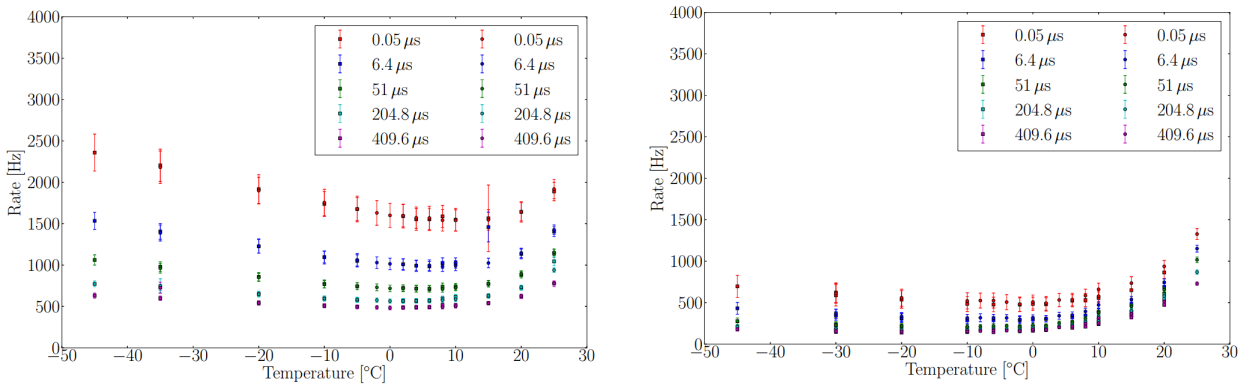
Standard IceCube data is triggered using the triggers described in Section 4.2.5. While useful for identifying particle interactions in the detector, these triggers are explicitly designed to reject much of the low-level noise. Triggered data is therefore unsuitable for noise studies.

A new method for data-taking was developed by Heereman (2013). Known as HitSpool, the new method is designed to record all hits within the detector in a continuous stream. This is different from typical triggered data in various ways. All hits are recorded as SLC hits, removing some of the burden from the processing system. The format is also highly compressed and must be parsed



(a) DOM 34-19 (b) DOM 85-30

Fig. 5.2.: The time between hits on DOMs 34-19 and 85-30. The black shows the distribution from data while the red shows the noise-generator simulated pure noise triggers. The peaks on the left of each distribution correspond to after-pulsing in the PMT while the peak on the right occurs at the characteristic time associated with the Poisson process.



(a) With Pressure Sphere (b) Without Pressure Sphere

Fig. 5.3.: Data taken at the University of Wisconsin. As the temperature of the DOM is decreased, the noise rate rises. Increasing the deadtime of the DOM reduces the noise rate. Removing the glass pressure sphere from the DOM also significantly decreases the noise rate.

outside of the IceTray framework. During detector operation, the HitSpool algorithm stores all hits registered by a DOM using a cyclical buffer. The buffer can store hits for up to 16 hours and can be written to disk at any time.

HitSpool removes the limitations imposed by standard IceCube triggered data, but can occupy a significant amount of limited bandwidth available for data transfer via satellite. Datasets taken with the HitSpool interface are therefore limited.

The system is external to the IceTray interface, using a unique compressed format to store

the HLC/SLC status, time, and charge information for all hits recorded by the DOM. While the HLC/SLC status is recorded, charge information is recorded in the SLC style for all hits. For more information on the difference between HLC and SLC hits, see Section 3.3. Using specialized scripts, the data may be read into IceTray to create continuous event readouts suitable for study. These hits may then be processed with IceCube modules for further analysis.

5.3 Vuvuzela's Correlated Noise Model

Simulation of the correlated noise component requires a new model for detector noise. The empirical model chosen consists of three components: the uncorrelated thermal noise, the uncorrelated radioactive noise, and the correlated scintillation noise.

The uncorrelated noise is treated as a Poisson process, with hits distributed uniform in time. The number of hits simulated for the thermal component is drawn from a Poisson distribution with a rate dependent on the temperature of the DOM and on the electronic properties of the given PMT.

The correlated component begins with another Poisson process component arising from proposed radioactive decays within the glass of the DOM and PMT. The number of hits from the radioactivity is drawn from a Poisson distribution with a rate independent of temperature and the hit times are again sampled from a uniform distribution.

Daughter particles from each radioactive decay then interact within the glass, producing light via scintillation. The light is produced in fast bursts followed by a long afterglow and enters the PMT, striking the photocathode to form a photoelectron signal. The number of hits in a burst is determined from a Poisson distribution with constant mean determined for each DOM. The time between subsequent hits is drawn from an empirically characterized log-normal distribution.

5.4 The Vuvuzela Noise Generator Module

The Vuvuzela module implements the model above. A direct replacement for the noise-generator module, Vuvuzela is designed to handle a frame of arbitrary length. Typical options used in configuring Vuvuzela are shown in Table 5.1. The module accepts a start and end time measured relative to existing hits caused by muon or neutrino interactions. These are used to determine the time rel-

Table 5.1.: Configurable options for Vuvuzela.

Parameter	**Usage**
StartWindow	Time relative to the first existing I3MCPE to simulate
EndWindow	Time relative to the last existing I3MCPE to simulate
DOMsToExclude	List of DOMs to avoid simulating
InputHitSeriesName	Name of existing I3MCPEs in the frame
OutputHitSeriesName	Name to save the produced I3MCPEs to the frame
SimulateNewDOMs	Enables simulation uncharacterised DOMs

ative to the earliest and last pre-existing hit within which to simulate noise hits. In general IceCube simulation, the StartWindow and EndWindow variables are set to -10 μs and +10 μs respectively, indicating that a buffer of 10 μs is simulated on each side of existing I3MCPEs.

Throughout simulation, the module records both the total number of hits produced as well as the total unweighted livetime of noise produced. All hits are internally stored relative to this unweighted livetime in order to simplify the production of especially long gaps in correlated hits.

The user may also identify any DOMs which should not be simulated. Vuvuzela accepts a list of DOMs to remove from simulation via the DOMsToExclude list. This is typically used to avoid simulating DOMs which may have problems or to simulate a subregion of the detector. Vuvuzela will attempt to build a list of DOMs to simulate using the DOMsToExclude list. This list is cached, eliminating the need to spend valuable processing time checking the DOMsToExclude list for each subsequent frame.

The list of DOMs to simulate is generated internal to the Vuvuzela module by cycling through all DOMs present in the GCD file. DOMs in the DOMsToExclude list are immediately dropped from simulation. If any DOM is discovered in the GCD file without the necessary parameters, a warning message notifies the user of the situation. The DOMs without proper parameters for the noise model are also ignored. This behaviour may be overridden for simulating unfit or new DOMs with the SimulateNewDOMs option. If enabled, the SimulateNewDOMs option will simulate new DOMs with configurable constant parameters. This option is unlikely to be used for IceCube and DeepCore simulation, but is useful for collaborators testing new geometries for upcoming detector extensions.

During the first frame visible to the module, Vuvuzela simulates 100 ms of correlated noise hits. The bursts occurring in this window are stored and used in subsequent frames, ensuring that the long afterglow of a correlated noise burst may begin well before the initial frames. This initial step must only be performed once.

Next, the Vuvuzela module identifies existing I3MCPEs in the frame via the configurable InputHitSeriesName parameter. The existing hits are checked to identify both the earliest and latest hit times. A time window for noise simulation is calculated using the StartWindow and EndWindow options described above.

For each DOM simulated, both the uncorrelated and correlated noise hits are produced. The uncorrelated thermal noise is simulated as a Poisson process with the number of hits, x, drawn from a Poisson distribution

$$f_{Uncorrelated}(x) = \frac{(\lambda \Delta t)^x}{x!} e^{(-\lambda \Delta t)} \tag{5.1}$$

where λ is the rate parameter associated with the process and Δt is the length of the time window being simulated. For the thermal noise, the rate typical of λ is on the order of 20 Hz and varies with depth. Each hit is then assigned a time drawn from a uniform distribution. The time is modified to store the produced I3MCPE in an internal buffer using the total unweighted livetime already simulated $T_{current}$.

$$t'_i = t_i + T_{current} \tag{5.2}$$

The correlated noise process begins with an uncorrelated Poisson process due to radioactive decays in the glass that is identical to the thermal noise component described in equations 5.1 and 5.2. The radioactivity occurs with a significantly higher rate constant $\approx$ 250 Hz.

Following the radioactive decay, the number of scintillation photoelectrons, y, is calculated following a second Poisson distribution independent of the time window being simulated.

$$f_{Correlated}(y) = \frac{\eta^y}{y!} e^{-\eta} \tag{5.3}$$

Here η is a constant expected value empirically fit for each DOM with a typical value of approximately 8. For each correlated hit, a standard normal distribution is sampled for a value z.

$$f_{Correlated}(z_i) = \frac{1}{\sqrt{2\pi}} e^{-z_i^2} \tag{5.4}$$

The z_i value is then converted to a time δt_i by translating to a log-normal distribution.

Table 5.2.: The input/output modes of Vuvuzela are based on the configurable options InputHitSeriesMapName and OutputHitSeriesMapName.

Mode	Input Name	Output Name	Includes Physics Hits	Replaces Input
1	Given	Given	Yes	No
2	Given	Missing	Yes	Yes
3	Missing	Given	No	N/A

$$\delta t_i = 10^{(\mu+\sigma z_i)} \tag{5.5}$$

The value of δt_i corresponds to the amount of time between hit i-1 and i. Because only HLC hits have full waveforms available for fitting, the distribution of correlated noise hits with $\delta t_i < 2$ μs is unknown. The simulated hits in this region are therefore removed pending further study.

The remaining gaps are sorted, yielding an initial burst of scintillation light with a long tail. The hit time may then be calculated relative to the initial radioactive decay time t_D.

$$t_i = t_D + \sum_{k=0}^{i} \delta t_k \tag{5.6}$$

Times are then stored following Equation 5.2. Scintillation hits are merged with the thermal and radioactive noise I3MCPEs. All noise hits are then merged with the pre-existing hit series. All hits within the correct time window are output to the frame with the configured name. Any correlated hits which do not occur within the current frame will be saved for later frames, providing an afterglow for later events.

Vuvuzela has three input/output "modes", which are described in Table 5.2. If no input hit series is provided, the module produces a noise-only hit series with the start and end times given by StartWindow and EndWindow options respectively. This hit series is then saved with the configured output name.

If input hits are provided but no output hit series name is configured, then the module merges the input hits with noise hits and replaces the input hit series in the frame.

The typical use specifies both the input and output hit series names. In this case, the input and noise hits are merged, then written to the frame using the configured output name. The input hit series is not deleted and can be available for other processing.

5.5 Fitting Procedure for Vuvuzela Parameters

Vuvuzela requires five parameters for simulation: the rates for the thermal and radioactive decay processes, the expected number of hits for the correlated component, and mean and variance for the log-normal distribution. The convolution of the three processes and other detector effects is nontrivial. In addition, detector effects, such as efficiency of the PMT and application of the discriminator, are present in data but occur only after noise generation in the simulation chain. In general, fitting the distributions taken from HitSpool data using standard methods yields values which are unsuitable for direct use in simulation due to these nonlinear detector effects.

This presents a unique challenge to identification of parameters for DOMs *in situ*. The standard method used in the laboratory prior to construction assumed an uncorrelated Poissonian noise model. Since DOM-to-DOM variations are unknown for a correlated noise model, each DOM needs to be fitted individually. The HitSpool data provides the sole means available for this kind of noise calibration study.

Initial values for DOMs, given in Table 5.3, were motivated by fitting a single DOM by hand. The DOM was given various values of parameters, simulated, then the resulting hits were used to form a distribution of time between hits in log-space similar to Figure 5.2. Due to the limitations of using a global constant after-pulsing probability, the after-pulsing region is not used in the fitting.

Table 5.3.: The five parameters fit by VuvuzelaFitter.

Parameter	**Units**	**Lower Bound**	**Seed**	σ
λ'_{Th}	Hz	N_μ	Eqn 5.7	2
λ_{Decay}	Hz	0	200, 260 (HQE)	10
η_{Scint}	Hits	0	6, 7 (HQE)	1
μ_{Scint}	Log_{10}(s)	None	-6	1
σ_{Scint}	Log_{10}(s)	0	2.7	1

A subset of the range extending from 10 μs up to 4 ms is used for fitting. The distribution was compared to data taken from early HitSpool tests, then new parameters were estimated, re-simulated, and re-evaluated. Once the parameters approximately reproduced the observed distribution, the initial estimation of parameters was used to seed all DOMs for an automated fitting procedure.

Muon hits form an irreducible background to the noise in HitSpool data. Fully simulating the muon background for 30 seconds of livetime is computationally infeasible, however. To properly

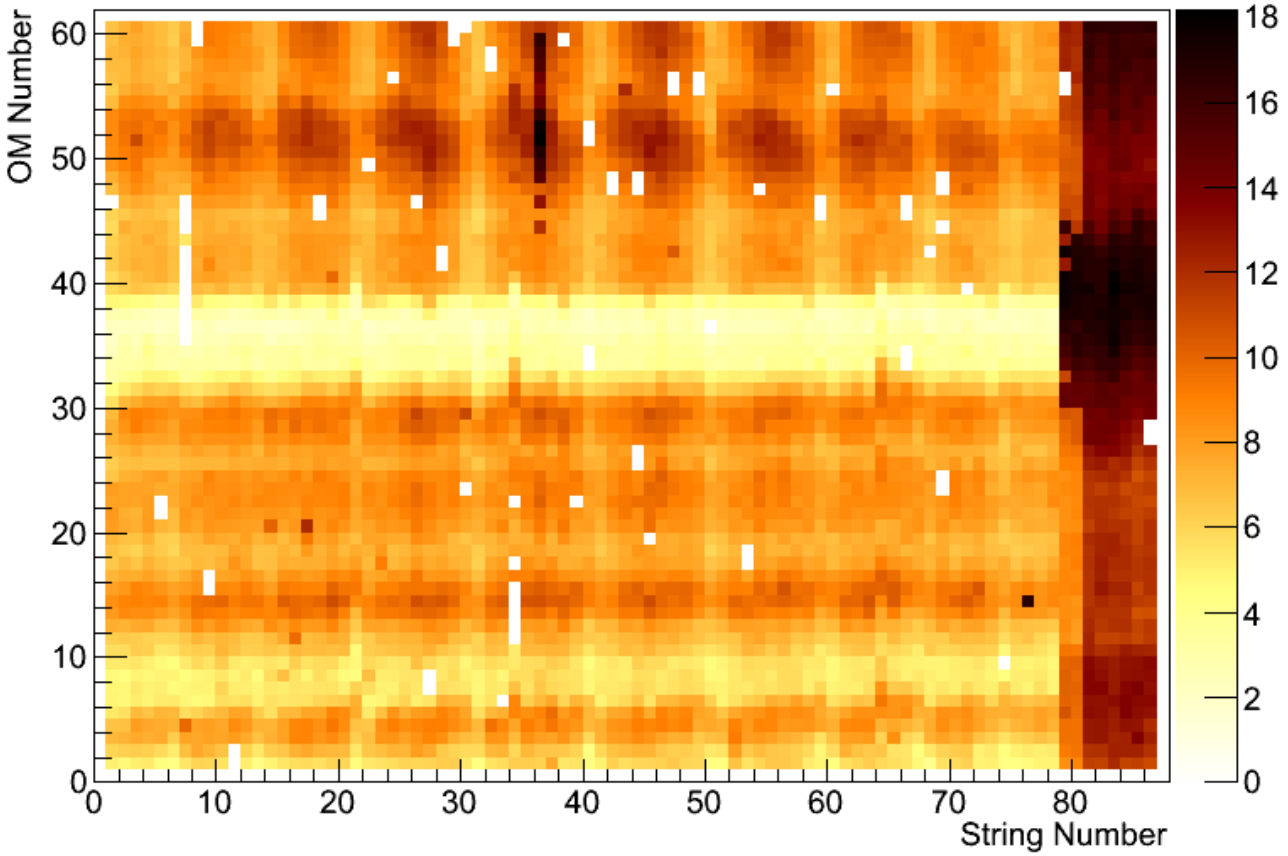


Fig. 5.4.: The rate of incident photoelectrons due to muons as simulated by CORSIKA and PPC. The color axis corresponds to the rate in Hz for each DOM. Deeper DOMs have higher OM numbers. Clearly visible are the HQE DOMs (red, black) and the dust layer (white). The deepest parts of the detector also have clearer ice as shown in Figure 4.1, facilitating detection of muons. Missing DOMs have known problems and are not simulated by PPC. These rates are used to correct for muon interactions when fitting parameters for Vuvuzela.

account for the hits, separate CORSIKA simulation is processed through propagation and the rate of I3MCPEs incident upon each DOM is calculated. The result is shown in Figure 5.4. On a given DOM, muon interactions are expected to be independent and rare compared to noise hits. The effect of muon interactions is therefore folded into the thermal noise rate λ_{Th}. The seed for the corrected parameter λ'_{Th} is then calculated using a simple temperature dependence to approximate the observed temperature dependence of the pre-existing DOM noise rates used in Noise-Generator.

$$\lambda'_{Th} = N_\mu + \frac{OMNumber}{60} \times 20Hz \tag{5.7}$$

The thermal noise rate for each DOM is simulated with a nonzero floor N_μ taken from Figure 5.4 during fitting. Once the fit converges, the final value of λ_{Th} is calculated using Equation 5.8.

$$\lambda_{Th} = \lambda'_{Th} - N_{\mu} \tag{5.8}$$

The fitting module, known as VuvuzelaFitter, uses a Metropolis-Hastings algorithm as defined in Cowan (2011) to optimize the noise simulation in the 5-parameter space. The module accepts initial values for each DOM on a single string which are stored both internally and in the frame. Vuvuzela uses the parameters written to the frame to simulate noise, which is then processed through DOMLauncher for detector and discriminator effects. The times of I3DOMLaunches produced are read into the VuvuzelaFitter module and stored for each DOM. The total livetime simulated is also stored by VuvuzelaFitter.

Limitations in the DOMLauncher project set an upper bound on the amount of time allowed for each frame of $\approx$ 100 ms. Because of this, multiple frames must be simulated and merged in order to provide enough livetime for fitting. VuvuzelaFitter accepts a configurable number of frames to process before evaluating the fit statistic. For the purposes of this study, 3000 frames have been merged following DOM simulation giving 30 seconds of livetime for each iteration of the fitting procedure.

Once VuvuzelaFitter has processed the required number of frames, the timing distribution shown in Figure 5.2 is produced using both the simulated hit times and Hitspool data. The test statistic is calculated using the weighted-unweighted χ^2 test for binned histograms available in the ROOT software environment (Gagunashvili, 2006). The returned probability, $P(\chi^2, N_{DoF})$, characterizes the goodness-of-fit for the simulated distribution assuming that the residuals in each bin are Gaussian. While this assumption may not be accurate for parameters far from the best-fit, $P(\chi^2, N_{DoF})$ is a valuable estimate for the similarity of the simulated and data distributions.

The calculated probability is used to determine the likelihood of accepting the current set of parameters over the previous most likely set of parameters.

$$\alpha = \frac{P(\chi^2, N_{DoF})}{P((\chi^2)', N'_{DoF})} \tag{5.9}$$

where the prime indicates the current best value. If α is greater than unity, the fit is always accepted. Otherwise, α represents the probability of accepting a step with a lower goodness-of-fit. The nonzero probability of backtracking allows the algorithm to step outward from a shallow local

optimum.

Regardless of whether the current parameters are accepted, new parameters are produced using a Gaussian distribution about each of the current best parameters. Each parameter is treated independently using a standard deviation according to Table 5.3. The radioactive decay rates, expected number of correlated hits from each decay, and variance of the log-normal distribution are each limited to positive values but are otherwise free to float. The mean of the log-normal distribution has no limits.

Each DOM is simulated until the probability returned by the weighted-unweighted χ^2 test is above a probability threshold and would therefore not be rejected given a standard test. Fits are subject to variation in the statistically limited 30 second samples which limits the ability to differentiate between similar parameters to high precision. Due to this, a lower limit for the χ^2 probability of 10% was chosen for this work.

Once this point is reached, the DOM is no longer simulated and the best fit values are saved. By removing DOMs with satisfactory fits from further simulation, the time taken for each iteration of remaining DOMs is reduced. This allows for DOMs far from the seed values to be fit within a few days.

Some of the early HitSpool data suffered from unrecoverable errors and therefore is missing some DOMs. To limit the number of DOMs affected, data from two separate tests of the HitSpooling interface is combined. This has the unfortunate effect that some DOMs have a significantly more limited set of data available to be fit than others, but allows nearly all DOMs to undergo at least some fitting.

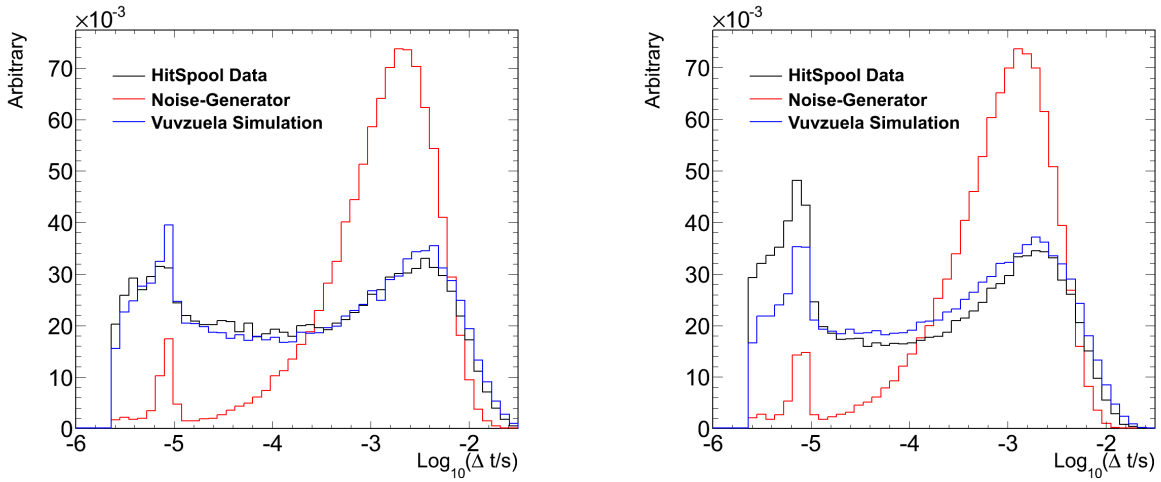
5.6 Results of the Vuvuzela Module

Fitting of Vuvuzela's parameters takes approximately 4 days for the entire detector using a high performance computing cluster. Each of the 86 strings is fit in parallel. Parameter is output to a file which may then be uploaded to the collaboration's GCD files to be used for simulation. Examples of the final distributions of IceCube and DeepCore DOMs are shown in Figure 5.5. The values of the various parameters are shown in Figure 5.6.

Vuvuzela does not significantly affect the number of hit DOMs during events triggered by muons.

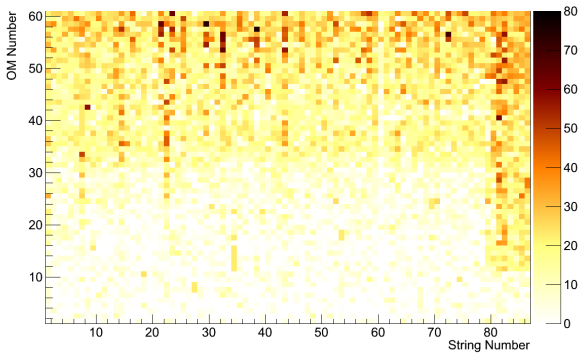
This is expected due to the short time window used for trigger readouts. Instead, Vuvuzela allows noise hits to be significantly more likely to be upgraded to HLC by effectively increasing the time available for local coincidence according to the length of the scintillation bursting effects. This effect is particularly notable when looking at triggers simulated without muon and neutrino interactions in the ice.

Vuvuzela simulation files have more noise triggers than noise-generator files. As shown in Figure 5.7, Vuvuzela therefore provides a better explanatory model than the uncorrelated noise simulated by noise-generator.

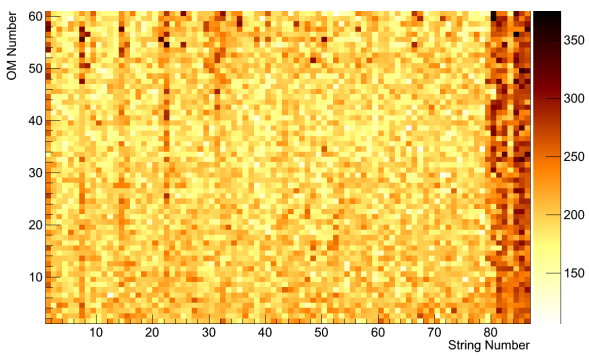


(a) DOM 34-19 (b) DOM 85-30

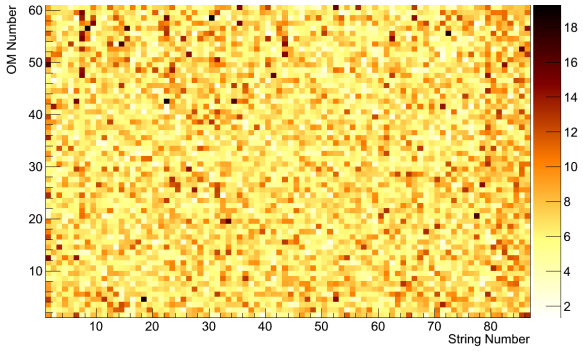
Fig. 5.5.: The time between hits on a single DOM. The black and red curves remain as in Figure 5.2. The new Vuvuzela model is shown in blue. Both of the DOMs show significantly more accurate simulated distributions using Vuvuzela.



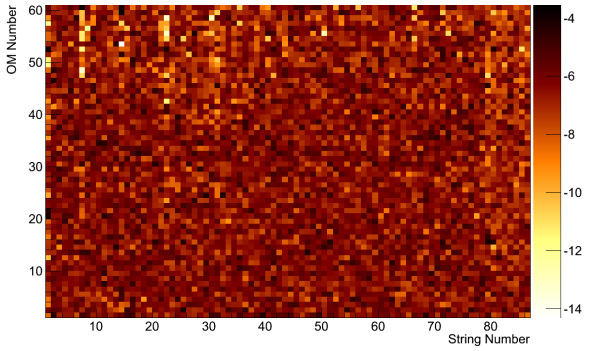
(a) Thermal Rate (Hz)



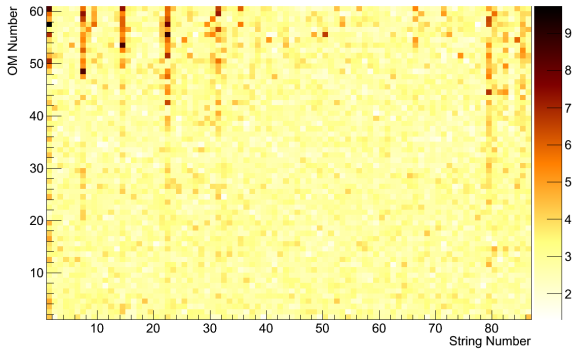
(b) Decay Rate (Hz)



(c) Expected Scintillation Hits



(d) Log-normal Mean (Log_{10}(s))



(e) Log-normal Sigma (Log_{10}(s))

Fig. 5.6.: The parameters of each DOM after convergence. DOMs located deeper in the ice have warmer temperatures and higher OM numbers. The HQE DOMs are visible on the right inn (a),(b), and (d). The effects of the temperature weighted values are clearly visible in (a). Newly deployed strings are also visible in data taken in 2011.

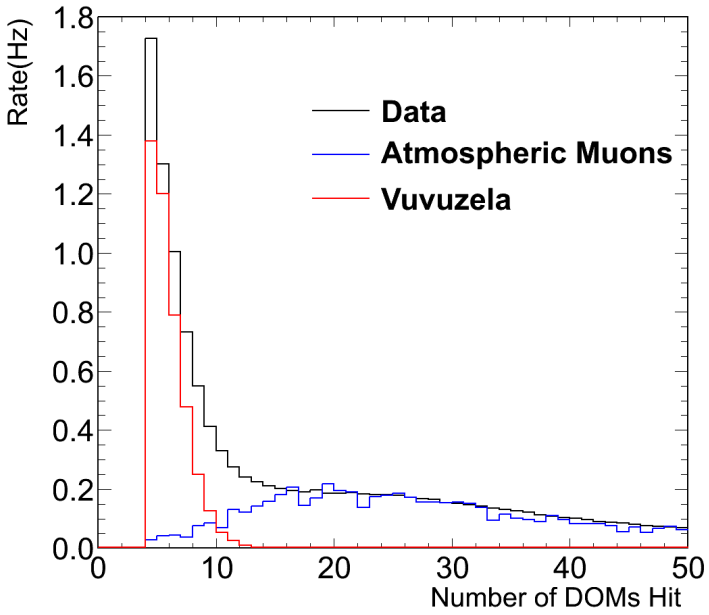


Fig. 5.7.: The number of DOMs hit for events passing the DeepCoreFilter. The hit series used matches that of Figure 4.5. Vuvuzela noise triggers are now included, showing a better agreement than triggers produced by noise-generator.

Chapter 6: The NoiseEngine Filter for Pure Noise Triggers

6.1 Noise Triggers using Vuvuzela

Because DeepCore uses the hit time and coincidence of hits for triggering, noise triggers should occur with some non-zero probability. The triggers caused exclusively by noise ("noise triggers") are predictable using Poisson statistics if the noise itself is a Poisson process, but are nontrivial for the new noise model. Noise triggers are expected to occur rarely, with very few DOMs reaching the strict requirements imposed by HLC cleaning. This effect is negligible for data streams dominated by large, bright events, but can be a significant background to low energy interactions with few DOMs hit. DeepCore should therefore be significantly affected by noise triggers. This is observed in data passing the DeepCoreFilter, with the number of HLC hits within the trigger window falling rapidly as shown in Figure 4.5. Typical noise triggers occur with fewer than 7 DOMs reaching HLC status.

In the past, this discrepancy between the simulation and data as well as limitations of fitting procedures forced IceCube analyses to remove all events with fewer than 8 DOMs recording HLC hits. In the case of DeepCore, this is a significant loss of signal, with only 60% of sampled atmospheric muon neutrinos and 70% of sampled atmospheric electron neutrinos kept at 100 GeV as shown in Figure 4.6. Replacing this cut is therefore a priority.

Noise triggers are expected to occur with relatively few hits and no preferred direction. Using these two characteristics, an algorithm to identify noise triggers has been developed in order to limit their effect on analyses.

6.2 The NoiseEngine Module

The NoiseEngine module is based upon an earlier module called TrackEngine (Wernhoff, 2010) used to identify muon tracks in the detector. The original TrackEngine module used the same algorithm with parameters optimized for tracks. A rewrite of the TrackEngine code forms the basis for the NoiseEngine module.

The module accepts I3RecoPulses from IceTray. If there are more than 30 hits recorded in the given hit series, then the event is immediately passed and no further processing occurs. This limits the processing time for bright events that are unlikely to be caused by noise alone.

The time of each hit in the given hit series is recorded and time ordered by the module. Using a fixed length τ specified by the user, NoiseEngine loops over the sorted hit times. For each hit, a time window is defined, extending from the hit time t_i until $t_i + \tau$. The optimal time window is chosen by maximizing the number of hits falling within the time window. Hits occurring outside of the time window are removed from the module's processing.

Once a time window is chosen, the module maps the azimuth and zenith angles from each hit in the window to all later hits and records them by using a binned **H**ierarchical **E**qual **A**rea iso**L**atitude **Pix**elization (**HEALPix**) sphere. HEALPix is chosen for convenience, as each bin in the HEALPix mapping of the sphere has equal solid angle, limiting issues where binning is biased. In addition, HEALPix binning provides constant-latitude rings, again limiting the effects of bin edges biasing against particular directions. The HEALPix map has a configurable "order" parameter N_{side} which determines the number of bins N_{bins} and, thus, the resolution.

$$N_{bins} = 12 \times 4^{N_{side}} \tag{6.1}$$

Because the lowest energy neutrino events visible to DeepCore are expected to have few hits, N_{side} is set to 0, yielding 12 bins in the HEALPix map for NoiseEngine. This allows NoiseEngine to concentrate hits into few bins. Low energy neutrinos forming an isotropic cascade are therefore expected to pass more easily with few bins.

The averaged charge of each pair of pulses may be used by the algorithm when mapping to the binned sphere. If no charge is available or the user disables the function, then each pair of hits is assigned a weight of 1.

The binned HEALPix sphere is then processed to look for any bins recording a total weight above a threshold. If the event being processed has any bin with above this threshold, the event passes. Otherwise the event is deemed too diffuse and likely to be caused by noise and is rejected by the filter.

6.3 Optimization of NoiseEngine Parameters

Cleaning noise triggers requires enough information to be able to differentiate between noise and particle interactions. Strict HLC hit cleaning is therefore to be avoided to prevent information from being lost. Because NoiseEngine connects each hit within the chosen time window to one another, the number of operations is expected to grow as $O(n^2)$ and a large hit series should also be avoided.

To simplify processing, a pre-existing hit series is selected based on the DeepCore standard processing for 2013. The I3RecoPulses cleaned via the SeededRT algorithm described in Section 4.4 and used during processing of the DeepCoreFilter described in Section 4.5 is selected. This hit series, labelled TWSRTOfflinePulsesDC after its processing chain involving time window cleaning, SeededRT cleaning, and reduction to the DeepCore fiducial volume only, is cleaned to remove most noise hits and geometrically limited. Because most DeepCore neutrino events are expected to be contained in the DeepCore geometry, little effect is expected from limiting to the fiducial volume.

The NoiseEngine module requires a time window τ, an apparent velocity window V_{app}, N_{side} of a HEALPix map, and a threshold to be used for filtering. The velocity window will be identified first, then all remaining parameters are tested using a simple grid search technique.

Table 6.1.: The parameters required for the NoiseEngine module. The velocity window is identified first, then remaining parameters are optimized using a discrete grid search algorithm.

Parameter	**Range**	**Effect**
V_{app} Window	N/A	Limits NoiseEngine to causal hits.
τ	[500, 1500] ns	Time window used to select hits
Threshold	[1, 8]	Number of pairs mapped to one HEALPix bin to pass
N_{side}	0	Order used to determine the number of HEALPix bins

The velocity window is tested by examining all potential I3RecoPulse pairs in TWSRTOfflinePulsesDC. These velocities may then be plotted to observe trends in the neutrino, muon, and noise triggers. A velocity window is chosen to minimize the number of pairs occuring in noise triggers while retaining pairs occurring in neutrino events.

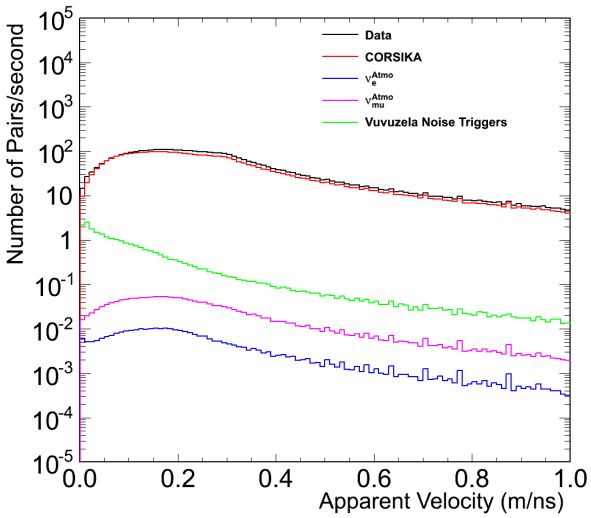


Fig. 6.1.: The apparent velocity as calculated by NoiseEngine. The velocity is calculated from each hit to every later hit, producing only positive values. Note the non-zero peak values for both muon and neutrino interactions.

Once a velocity window is identified, the remaining parameters are discretized to form a grid using the ranges shown in Table 6.1. The time windows tested ranged from 500 ns to 1500 ns in steps of 250 ns. An N_{side} value of 0 is expected to provide the largest rejection of noise triggers while retaining neutrino and muon events. Finally, the threshold for acceptance is varied from 1 pair to 8 pairs mapped to a single HEALPix bin.

6.4 Results of Removing Noise Triggers with NoiseEngine

The apparent velocities between hits is shown in Figure 6.1. To maximize the probability of accepting neutrino events, a velocity window of [0.05, 0.4] m/ns has been chosen. This ensures that the peaks of the neutrino distributions are accepted while the peak for the noise distribution is rejected.

Figure 6.2 shows the results of the grid search. For neutrinos, a majority of events are retained for any combination time window and threshold whereas a majority of noise triggers are rejected. The optimal value is identified from Figure 6.3, which shows the neutrino and noise histograms of Figure 6.2 subtracted. Choosing the values labelled as "minimum" cleaning in Table 6.2 rejects 94% of noise triggers while retaining 80% of low brightness ν_e events and 87% of low brightness ν_μ events in the 10-300 GeV range. The efficiency of the NoiseEngine filter using these settings is shown for neutrino events in Figure 6.4. These events have energies near the lower threshold for

detection in DeepCore (Abbasi et al., 2012).

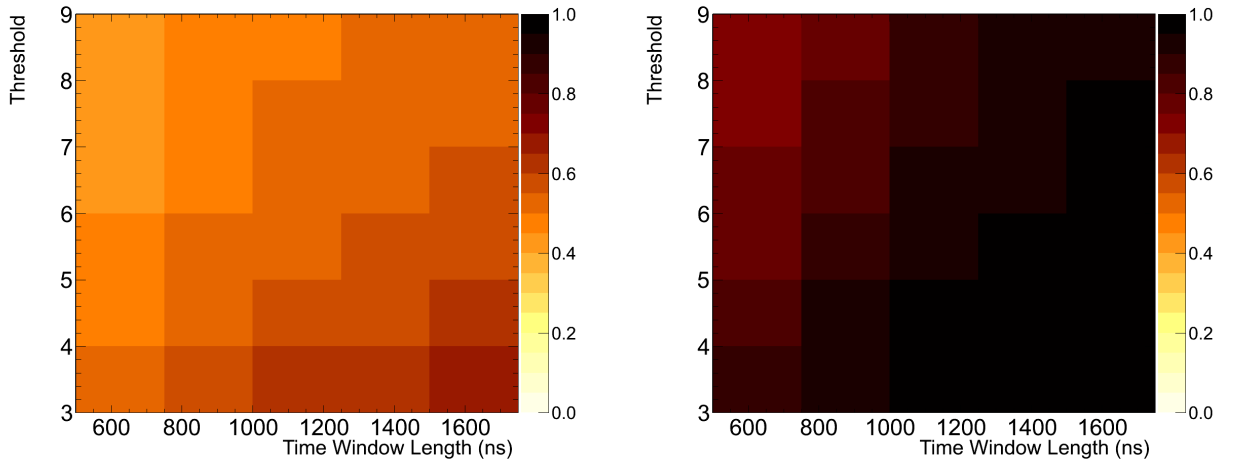
Table 6.2.: The parameters selected for the NoiseEngine module.

Parameter	**Minimal Cleaning**	**Stringent Cleaning**
V_{app} Window	[0.05, 0.40] m/ns	[0.05, 0.40] m/ns
τ	1000 ns	500 ns
Threshold	4 Pairs	7 Pairs
N_{side}	0	0

If necessary, more stringent cleaning is also possible. Using the stringent NoiseEngine parameters listed in Table 6.2 results in rejection of 99.9% of noise triggers while retaining 55% of low brightness ν_e events and 60% of low brightness ν_μ. Stringent rejection parameters reduce noise trigger rate to be approximately comparable to the neutrino rate, reducing a significant background. The distributions of number of DOMs hit after both minimal and stringent filtering is shown in Figure 6.5. Rates for both signal and background are given in Table 6.3. The efficiency of the NoiseEngine filter using stringent settings is shown for neutrino events in Figure 6.4.

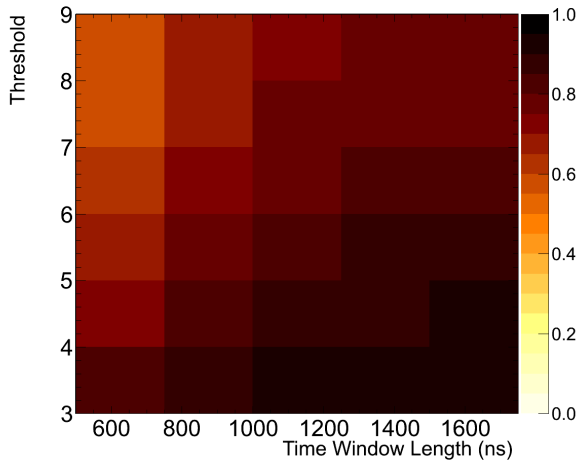
Table 6.3.: The passing rates for the NoiseEngine module using both minimal and stringent parameters. Stringent parameters reduce the noise trigger rate to be comparable to the neutrino rate.

	Without NoiseEngine	**Minimal**	**Stringent**
Data	11.850 Hz	6.422 Hz	4.108 Hz
CORSIKA	5.440 Hz	5.211 Hz	3.427 Hz
Vuvuzela Triggers	5.253 Hz	0.347 Hz	9.722 mHz
ν_μ	8.101 mHz	7.057 mHz	4.506 mHz
ν_e	1.875 mHz	1.570 mHz	0.981 mHz

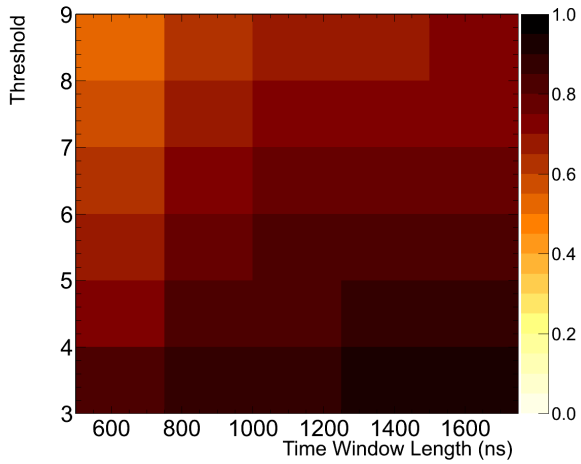


(a) Data

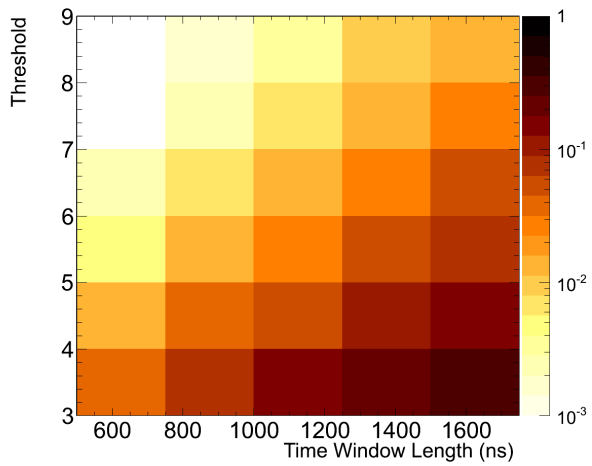
(b) CORSIKA



(c) ν_{μ}^{Atmo}

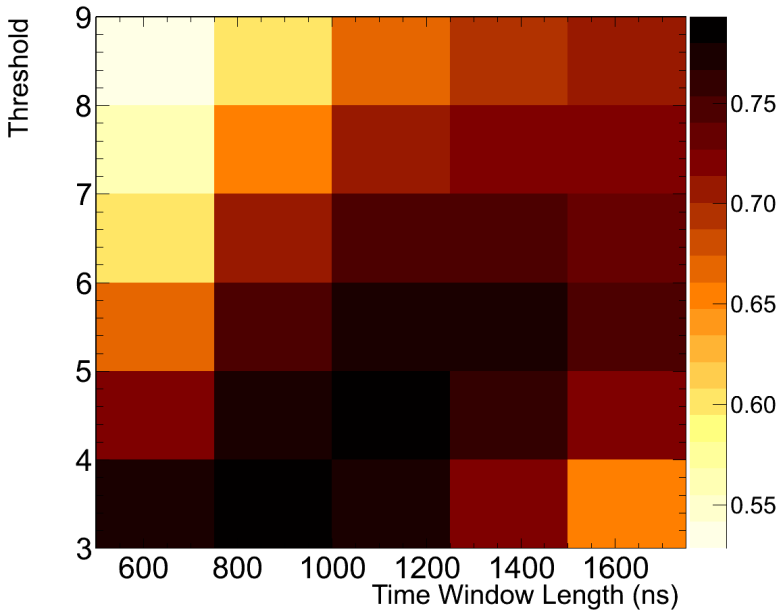


(d) ν_{e}^{Atmo}

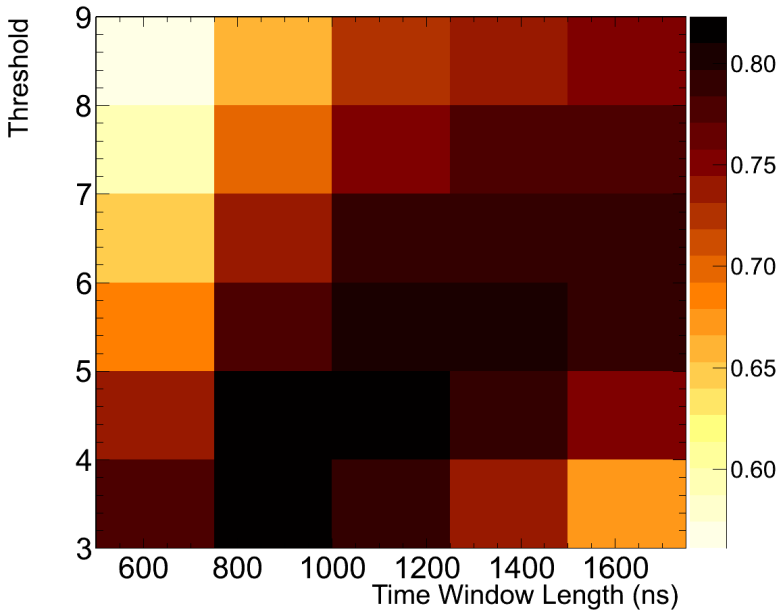


(e) Noise Triggers

Fig. 6.2.: The results of a grid search of the NoiseEngine Time Window τ and Threshold parameters. Each bin corresponds to values along its left and bottom edges. The color axis corresponds to the percentage of events passing the NoiseEngine module. Note that the color axis on (e) is logarithmic instead of linear.

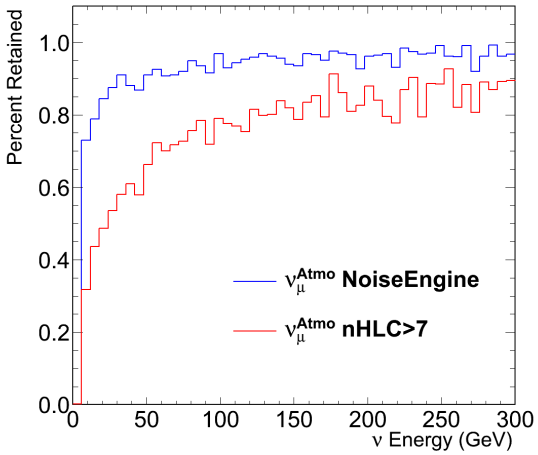


(a) ν_e - Noise Triggers

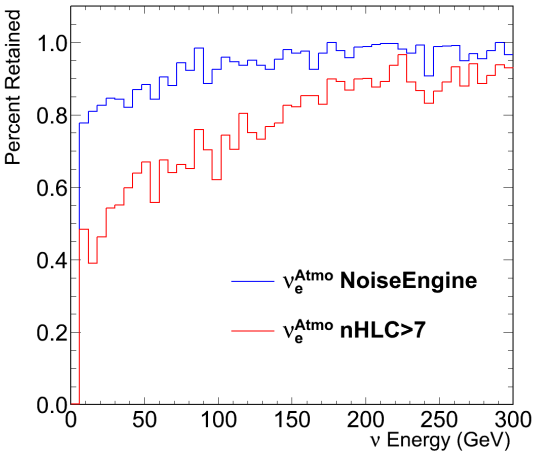


(b) ν_μ - Noise Triggers

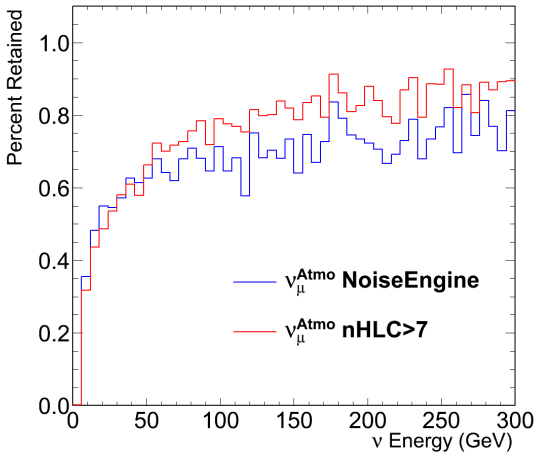
Fig. 6.3.: The relative passing percentages of neutrinos to noise, defined to be the percentage of noise triggers passing subtracted from the percentage of neutrinos passing. Darker regions correspond to better separation between the neutrino and noise triggers.



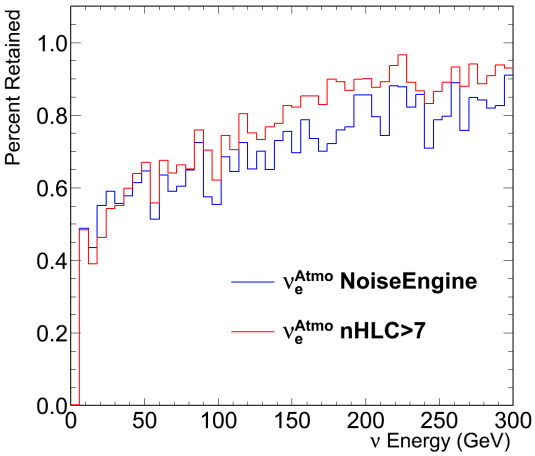
(a) ν_μ Minimal Passing Efficiency



(b) ν_e Minimal Passing Efficiency

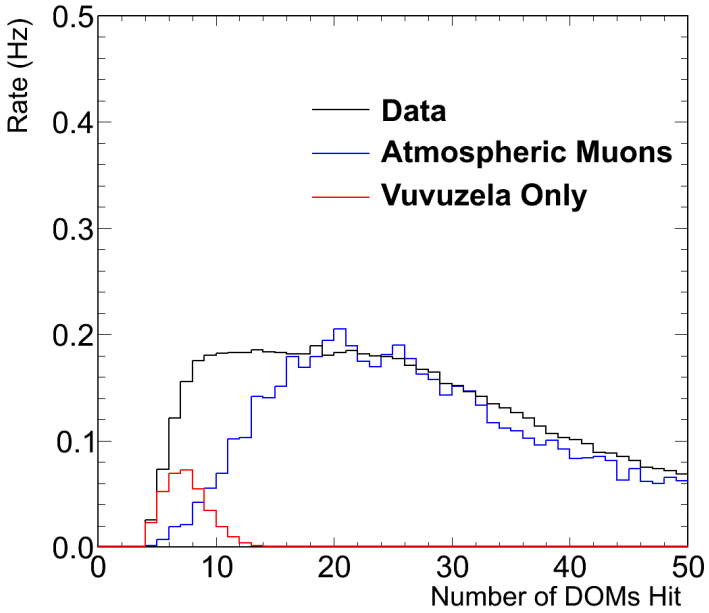


(c) ν_μ Stringent Passing Efficiency

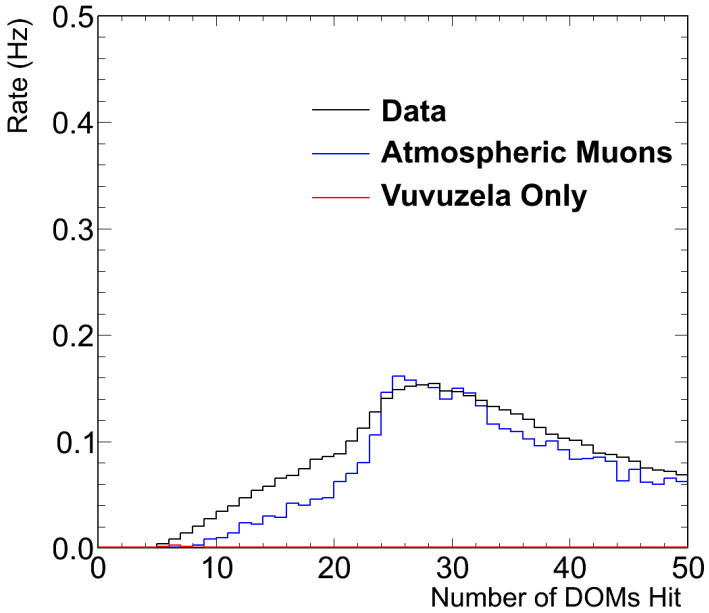


(d) ν_e Stringent Passing Efficiency

Fig. 6.4.: The efficiency of the NoiseEngine cleaning is shown as a function of neutrino energy. The effect of removing events with fewer than 8 HLC hits is shown in blue while the effect of NoiseEngine is shown in red. Figures (a) and (b) show the efficiency of the NoiseEngine minimal cleaning. Figures (c) and (d) show the efficiency of the NoiseEngine stringent cleaning. Muon neutrino events are shown on the left while electron neutrino events are shown on the right.
Top: More low energy neutrinos pass the minimal cleaning than pass the historical cut on the number of HLC hits in the detector. Both show comparable removal of noise triggers. Bottom: Removing 99.9% of noise triggers reduces the noise trigger rate to approximately comparable to the low energy neutrino trigger rate. The neutrino efficiencies for this stringent cleaning are comparable to the historical cut on the number of HLC hits in the detector.



(a) Minimal



(b) Stringent

Fig. 6.5.: The number of DOMs recording hits is shown after applying the NoiseEngine minimal cleaning (top) and stringent cleaning (bottom). The distribution for data is shown in black and the CORSIKA simulation is shown in blue. Compare to Figure 5.7. The peak at low NChannel is greatly reduced for minimal cleaning and eliminated for stringent parameter sets. Remaining discrepancies are due to limited statistics and undersimulated high energy atmospheric muons in CORSIKA simulation.

Chapter 7: Conclusions

This thesis has shown that uncorrelated noise is insufficient to explain the timing profiles and triggers in the IceCube neutrino detector. These effects are particularly important for dim, low energy neutrino interactions, where they provide a significant background. The new noise model incorporating correlated noise is a more accurate description of the detector noise in the IceCube detector. Low quality fits increase the noise trigger rate from approximately 3 Hz to 6 Hz, reducing the overall discrepancy between data and simulation. Fit quality is computationally limited, although further processing is possible. This work is currently ongoing.

These noise triggers occur as very dim events and at rates significantly above the atmospheric neutrino flux that is to be studied using the DeepCore sub-detector. Noise triggers must therefore be identified and removed. The NoiseEngine module is capable of reducing the number of noise triggers in data more efficiently than previous methods. By using the directionality of events, noise triggers may be reduced using the minimal cleaning, which rejects more than 90% of noise triggers while keeping more than 80% of both electron and muon neutrinos. Stringent cleaning has also been investigated, which reduces noise triggers to a level comparable to the neutrino signal.

The understanding and rejection of noise triggers reduces a significant background for low energy neutrino analysis with DeepCore. The NoiseEngine filter gives DeepCore better sensitivity to effects occurring in the 10-50 GeV range, where oscillation effects should be prominent. These events, which would have been far below the noise trigger background using previous methods, may hold clues to better understanding neutrino oscillation.

The Vuvuzela noise model and NoiseEngine algorithm can also be applied to potential future upgrades. Identification of noise components may allow better testing and characterization prior to deployment. The planned low-energy extension to DeepCore, the Precision IceCube Next Generation Upgrade (PINGU) detector, in particular can benefit from both Vuvuzela and NoiseEngine, granting the ability to observe neutrinos with energies as low as 1 GeV.

Bibliography

Aartsen, M., & et al. 2013, submitted to Nucl. Instr. and Meth.

Aartsen, M. G., Abbasi, R., Abdou, Y., et al. 2013a, ArXiv e-prints, arXiv:1301.5361

—. 2013b, ApJ, 765, 55

Abassi, R., & et al. 2010, Nucl. Instr. and Meth., A618, 139

Abbasi, R., Abdou, Y., Abu-Zayyad, T., et al. 2012, Astroparticle Physics, 35, 615

Achterberg, A., Ackermann, M., Adams, J., et al. 2006, Astroparticle Physics, 26, 155

Ackermann, M., Ahrens, J., Bai, X., et al. 2006, Journal of Geophysical Research: Atmospheres, 111, doi:10.1029/2005JD006687

Andreopoulos, C., Bell, A., Bhattacharya, D., et al. 2010, Nuclear Instruments and Methods in Physics Research A, 614, 87

Ankowski, A., & et al. 2006, Nucl. Instr. and Meth., A556, 146

Beatty, J., Matthews, J., Gaisser, T., & Stanev, T. 2011, in Phys. Rev. **D**86

Bichsel, H., Groom, D., & Klein, S. 2011, in Phys. Rev. **D**86

Briese, T., Bugel, L., Conrad, J. M., et al. 2013, ArXiv e-prints, arXiv:1304.0821

Chirkin, D. 2013, Nuclear Instruments and Methods in Physics Research A, 725, 141

Cowan, G. 2011, in Phys. Rev. **D**86

Daughhetee, J. 2012, DeepCore 2012 Filter Proposal, Tech. rep., internal Report

Gagunashvili, N. D. 2006, ArXiv Physics e-prints, arXiv:physics/0605123

Gazizov, A., & Kowalski, M. 2005, Computer Physics Communications, 172, 203

Hamamatsu Photonics. 2012, Photomultiplier Tubes and Assemblies for Scintillation Counting & High Energy physics

Heereman, D. 2013, in 33rd International Cosmic Ray Conference

Helbing, K., Goldschmidt, A., Köpke, L., et al. 2003, http://butler.physik.uni-mainz.de/~amauser/download/afterscint.pdf

Hörandel, J. R. 2005, Nuovo Cimento B Serie, 120, 825

Kampert, K. ., & KASCADE-Collaboration. 2002, ArXiv Astrophysics e-prints, arXiv:astro-ph/0204205

Meyer, H. O. 2010, EPL (Europhysics Letters), 89, 58001

Nikkel, J. A., Lippincott, W. H., & McKinsey, D. N. 2007, Journal of Instrumentation, 2, 11004

Pepper, J. 2013, DeepCore 2013 Filter Proposal, Tech. rep., internal Report

Richardson, O. 2003, Thermionic Emission from Hot Bodies (Wexford College Press), 196

Riedel, B. 2012, IceCube Collaboration Meeting

Wernhoff, C. 2010, Master's thesis, Stockholm University