

Radial Drift Modelling in the SuperNEMO Tracker

Claire Berschauer

August 18, 2023



MSc in Particle and Nuclear Physics
The University of Edinburgh
2023

Abstract

The SuperNEMO experiment searches for physics beyond the Standard Model in the form of a rare decay mode, neutrinoless double beta decay ($0\nu\beta\beta$). Its detector, the demonstrator module, includes an innovative tracker design that utilizes a grid of drift cells to find the position of travelling charged particles in three dimensions as they are emitted from the Selenium-82 source foils. Previous work on radial position determination resulted in a drift model designed for demonstrator module tracking cells. This dissertation utilizes that model on real calibration data from the detector, aiming to validate its results through analysis of drift time and radius distributions. It also compares drift model-calculated radii with radius data determined via convex optimization from randomly generated particle tracks.

Declaration

This dissertation was written entirely by myself.

Chapter 2 provides an introduction to the physics motivating the experiment I am working on and Chapter 3 details the background physics and previous work specific to my project. These sections review the work of others.

Chapters 4 through 7 and Appendices A, B, and C describe my work throughout this dissertation.

The programming I did during this project was done through CCLyon for data access and simulation setup, ROOT for data storage, and Python for data analysis. Falaise was used for simulation and visualization. The Python code was written entirely from scratch by myself, utilizing the numpy, pandas, matplotlib, uproot, time, cvxpy, random, and seaborn packages. Dataset access and simulation code was based on tutorials designed by SuperNEMO members for the January and July 2023 collaboration meetings.

Personal Statement

My involvement in the SuperNEMO collaboration began in January 2023, when I attended the winter collaboration meeting in Aussois, France. The meeting was a wonderful introduction to the group and provided a basis of knowledge on the collaboration's work.

I began working properly on my project in June. The first two weeks were spent reading through software documentation and background literature. I also started to get involved with planning for the upcoming collaboration meeting to be held in July. I began compiling resources for my project, started drawings for the collaboration meeting, and attended SuperNEMO and university meetings related to my project.

In weeks 3 and 4, I received access to CCLyon, the remote database that contains all of SuperNEMO's software and data. I also continued to contribute to collaboration meeting planning by testing out technology, visiting the meeting location, and making activity cards and stickers for the event.

The collaboration meeting was held in week 5 of my project. I spent the week taking notes on talks, making my own slides to be presented at the meeting, and attending discussions relating to the analysis group and communications planning. I also volunteered to be trained as a website contributor.

I encountered many technical issues in weeks 6 through 8. I needed to set up remote access to a Linux device to edit the website, find a workaround since I was unable to run Falaise simulations on my laptop, and I encountered errors when attempting to use ROOT on my device.

Analysis began in week 9 and continued through week 11. I developed Python code to enable my Jupyter notebooks to process ROOT files and save data as pandas data frames, then created a class to streamline calculations. From then on, I analyzed the data and compared it to a dataset I generated using cvxpy. During this time I also began working on writing up my dissertation, ultimately focusing on writing in the last week.

Acknowledgements

I would like to thank my supervisor, Cheryl Patrick, for all her help and guidance during this dissertation and over the course of my MSc. I couldn't have asked for a more supportive or enthusiastic mentor.

Thank you as well to the SuperNEMO collaboration at large. The collaboration meetings and weekly discussions were very informative and inspired my work over the summer.

I would also like to acknowledge the Particle and Nuclear Physics students – and the Physics MSc community in general – for their constant encouragement and ability to help me stay positive and enjoy life during the stressful times.

Finally, thanks to my non-physics friends and family who don't know the Standard Model by heart but are always willing to listen to me talk about graphs.

Contents

1	Introduction	2
2	Background Theory	4
2.1	The Standard Model and Beyond	4
2.2	Matter-Antimatter Asymmetry	4
2.3	Neutrino Oscillations	5
2.4	Nature of the Neutrino	7
2.5	Neutrinoless Double Beta Decay	8
2.6	SuperNEMO	10
3	Project Background	14
3.1	Drift Cell Specifics	14
3.2	Landells' Results	15
3.3	Križáč Track Fitting	17
3.3.1	Legendre Transformations	18
4	Design and Data Preparation	20
4.1	UDD .brio to ROOT and Falaise	20
4.2	ROOT to Pandas	21
4.3	Drift Time Calculations	21
5	Results and Analysis	23
5.1	Calculations and Data	23
5.2	Drift Time and Radius Visualization	25
5.3	Cell Region Analysis	25
5.4	Cell Type Analysis	27
5.5	Real vs. Generated Radius Data	28
5.6	Discussion and Future Work	29

6 Beyond Analysis	30
6.1 Artistic Director	30
6.2 SuperNEMO.org	30
6.3 SuperNEMO Starter Pack	31
7 Conclusion	32
A Code Repository	34
A.1 ROOT to Pandas	34
A.2 AnalyzeDrift	34
B Tech Support	36
C Artwork	37

List of Figures

1.1	<i>The Standard Model</i> — The first three columns contain fermions and correspond to the three generations of matter, with mass increasing from left to right. Fermions in the same row all have the same properties apart from mass [1]. The final two columns contain the force-carrying bosons.	3
2.1	<i>Decay Modes</i> — Drawn representation of the $2\nu\beta\beta$ and $0\nu\beta\beta$ decay modes.	9
2.2	<i>Decay Diagrams</i> — Feynman diagrams of two possible mechanisms by which $0\nu\beta\beta$ decay could occur. Diagram (a) shows the exchange of a Majorana neutrino where mass is given by the seesaw mechanism, while diagram (b) illustrates a right-handed current contribution. Image sourced from [2].	9
2.3	<i>Laboratoire Souterrain de Modane</i> — Depiction of LSM's location, sourced from [3].	11
2.4	<i>Demonstrator Module</i> — Layout of the detector's internal components, indicating the locations of the source foils, tracking chambers, and calorimeter walls.	11
2.5	<i>Drift Cell</i> — Diagram of a single tracker drift cell. The central anode wire is set at +1700 V, and is surrounded by 12 ground wires as shown in the top view on the right. The ends of the anode wire are capped by cathode rings.	12
3.1	<i>Signal Graph</i> — Signal output from the SuperNEMO tracker. R0 corresponds to the timestamp taken when the anode signal passes its initial threshold (-10mV). R5 and R6 refer to cathode timestamps triggered when the ion plasma reaches the two cathode caps. Graph taken from [4].	15
3.2	<i>E Field Map</i> — Electric field strength contour map showcasing 'Edge,' 'Corner,' and 'Center' quadrants of various drift cells. Each quadrant is 2.2x2.2 cm ² . Figure sourced from [4].	16
3.3	<i>Parameter Tables</i> — The top table gives the a, b, and cutoff time parameters for all quadrant types and pressures in the inner areas of the drift cells ($r < x$) where x is the cutoff radius determined by the chi-squared optimization. The bottom table gives the same parameters but for the outer areas ($r > x$). Tables taken from [4].	17

3.4	<i>Track Reconstructions</i> — Examples of (a) good and (b) bad reconstructions using the Legendre transform method. Figure courtesy of [5].	18
4.1	<i>Data Pipeline</i> — The data pipeline, with inputs from the detector and Falaise simulation. Image sourced from material presented in the January 2023 SuperNEMO collaboration meeting [6].	21
5.1	<i>Processed Dataframe Entries</i> — Each row of this data frame contains event, location, timestamp, drift time, cell type, and radius information for a single hit. This data frame can be accessed by inputting the pandas data frame of detector data into AnalyzeDrift, then calling drift_df.	23
5.2	<i>Drift Time Distribution</i> — Distribution of calibration data drift times. The maximum drift time in the dataset is $295\mu s$. Axes were set to the observed limits to better showcase the shape of the distribution.	24
5.3	<i>Drift Radius Distribution</i> — Distribution of radii calculated using Landells' drift model. The maximum radius is 4.28 cm, with the bulk of the data accumulating close to the center of the cell.	24
5.4	<i>Inner/Outer Region Drift Time Comparison</i> — Division of inner and outer cell hits within the full drift time distribution.	26
5.5	<i>Inner/Outer Region Radius Comparison</i> — Division of inner and outer cell hits within the full drift radius distribution.	26
5.6	<i>Cell Type Drift Time Comparison</i> — Comparison of drift time distributions for center cells, edge cells, corner cells, and the full dataset.	27
5.7	<i>Cell Type Radius Comparison</i> — Comparison of radius distributions for center cells, edge cells, corner cells, and the full dataset.	27
5.8	<i>Randomly Generated Radius Distribution</i> — Distribution of radii calculated by randomly generating tracks through a $4.4 \times 4.4 \text{ cm}^2$ area.	28
5.9	<i>Real vs. Generated Radius Comparison</i> — Distributions of real and randomly generated radii, including error bars based on normalization.	29
C.1	Rainbow SuperNEMO logo sticker.	37
C.2	Pollock Halls collaboration meeting sticker.	38
C.3	Whisky distillery sticker for Edinburgh whisky tour.	38
C.4	Whisky glass sticker for whisky tasting in Edinburgh.	38
C.5	"We are speed" SuperNEMO snail sticker.	38

Chapter 1

Introduction

The Standard Model characterizes the fundamental building blocks and forces of the universe but remains an incomplete description of natural processes. In its most basic form, the Standard Model contains two sets of elementary particles known as fermions and bosons that interact to create matter. These particles are shown in Figure 1.1. Fermions combine to create larger particles, like protons and neutrons, and bosons act as force-carriers to enable interactions between particles [7]. The Standard Model comprehensively describes the electromagnetic, weak, and strong interactions and the self-interactions between particles, and it has been very successful in predicting particles and their behaviors.

With the discovery of the Higgs boson in 2012, the Standard Model cemented itself as the most prominent model of sub-nuclear physics [1]. However, open questions remain on the nature of dark matter, the imbalance of matter and antimatter, the hierarchy problem, and more. Running experiments that look beyond the Standard Model may lead to validating hypotheses on the nature of these missing pieces with the aim of creating a more comprehensive model of the universe.

To this end, the SuperNEMO experiment, part of the Neutrino Ettore Majorana Observatory (NEMO) group, searches for evidence of neutrinoless double beta decay, a hypothesized decay mode that may be key to explaining matter-antimatter asymmetry and describing the nature of the Standard Model’s most mysterious fermions, neutrinos [8]. SuperNEMO performs this search by capturing energy and location data of double beta decays. Since neutrinos are neutral, they cannot be detected directly, so precise energy measurements are required to differentiate two-neutrino double beta decays from the evasive neutrinoless decay mode.

As of July 2023, the internal components of the SuperNEMO detector, the “demonstrator module,” and an anti-Radon tent are complete, with iron shielding installation underway [9]. The demonstrator module is positioned to make high precision particle measurements from decays of Selenium-82 with its innovative tracker design and to achieve lower background signals than its predecessor, NEMO-3.

The two trackers in the demonstrator module are composed of over 1000 drift cells each. These drift cells measure drift times of passing charged particles in the horizontal and vertical directions. Through careful consideration of several variables that influence drift

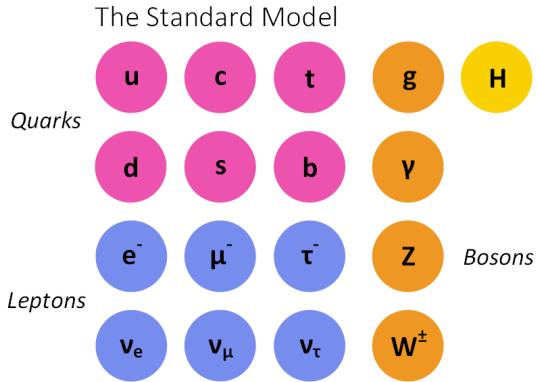


Figure 1.1: *The Standard Model* — The first three columns contain fermions and correspond to the three generations of matter, with mass increasing from left to right. Fermions in the same row all have the same properties apart from mass [1]. The final two columns contain the force-carrying bosons.

time, like electric field strength, pressure, and gas composition, drift times can be converted to three dimensional positions in post-data collection analysis.

This dissertation discusses how the tracker's radial position can be determined by transforming drift times derived from timestamp data to radii using a drift model and performs an analysis of how well the model makes this estimation. This work will contribute to the detector's precision and its ability to accurately map and identify decays, increasing the likelihood of signal detection. Previous research on this topic investigated how various factors impact drift times, establishing two drift model parameters that vary based on cell location within the tracker. Furthermore, this dissertation will also discuss plans for the tracking component of SuperNEMO's analysis, with a focus on integrating position calculation and track fitting.

Chapter 2

Background Theory

2.1 The Standard Model and Beyond

The Standard Model is a theory that aims to describe behavior at the sub-nuclear level and define the structure of the universe. It contains a set of elementary particles divided into two groups known as fermions and bosons that act as the building blocks of matter, with fermions combining to create larger particles, like protons and neutrons, and bosons acting as force-carriers to enable interactions between particles. Currently, this model accounts for three fundamental forces: the electromagnetic, weak, and strong forces, and predicts a single mass-giving boson known as the Higgs boson. Since the discovery of the Higgs in 2012 and with overall success in describing physics between low-energy and TeV scales, the Standard Model has become the most widely recognized model in particle physics [1][7].

However, despite its many successes in predicting particles and their behaviors, the Standard Model does not account for many unexplained phenomena like gravity, dark matter and energy, neutrino masses, and the disparity between the electroweak scale ($O(1 \text{ TeV})$) and the Planck scale ($O(10^{16} \text{ TeV})$) [10]. These missing pieces indicate that the Standard Model is not a comprehensive theory, but rather an effective low-energy approximation of a larger model. This conclusion necessitates new theories to extend the Standard Model, which has created a new section of particle physics that investigates phenomena Beyond the Standard Model (BSM) [1].

2.2 Matter-Antimatter Asymmetry

One key issue with the Standard Model is its inability to explain why there is an imbalance of matter and antimatter in the universe. This discrepancy can be directly observed by measuring the ratio of matter and antimatter in cosmic rays. The PAMELA and FERMI space observatories have found positrons and antiprotons in their probes of the solar system, but no heavier antinucleids [11]. Scientists theorize that the prevalence of matter derives from a small imbalance in the early universe which has grown over time.

Estimates of the initial ratio of baryons – composite particles of three or more quarks

– to photons can provide insight into the state of the early universe. These estimates have been created through two primary methods. The first method derives the ratio η mathematically through a series of Boltzmann equations relating to measured isotope abundances and has found $\eta = 5.80 \pm 0.27 * 10^{-10}$. The second method looks at temperature fluctuations in the Cosmic Microwave Background (CMB) and determines baryon content based on their power spectra [11]. This method has yielded a value of $\eta_{CMB} = 6.160^{+0.153}_{-0.156} * 10^{-10}$. Both methods, and observations of cosmic rays, agree that the early universe asymmetry is on the order of 10^{-10} [11]. Scientists have yet to determine why this imbalance exists, but the solution requires physics beyond the Standard Model.

2.3 Neutrino Oscillations

Another area of BSM investigation is the description of the neutral fermions, neutrinos. In 1965, a scientist named Ray Davis began an experiment deep underground in the Homestake mine located in South Dakota to count neutrinos emitted from the sun [12]. His experiment was based on the inverse beta decay $\nu_e + {}^{37}_{17}Cl \rightarrow {}^{37}_{18}Ar + e^-$, which occurred when neutrinos interacted with chlorine in the experiment's 615-ton tank of C_2Cl_4 and it aimed to detect high energy neutrinos [13][1]. To the surprise of the scientific community, Davis's experiment yielded a neutrino capture rate that was less than a third of the rate predicted by theory [12]. Specifically, the Homestake experiment measured a rate of 2.56 Solar Neutrino Units (SNU) to the standard solar model's 9.3 SNU [13]. Was this discrepancy the result of a problem with the Standard Solar Model that predicted neutrino emission, an issue with the design of the experiment, or an indication of new physics? This question became known as the ‘solar neutrino problem,’ and launched a new wave of neutrino detection experiments.

Even before the solar neutrino problem was posed, an Italian physicist, Bruno Pontecorvo, investigated a transition between muonium (u^-e^+) and antimuonium (u^+e^-) which would be forbidden in the Standard Model. Pontecorvo hypothesized that oscillations could occur between electrically neutral fermions, like neutrinos [14]. Pontecorvo's theories became the basis for the concept of neutrino oscillations, where neutrinos have nonzero mass and change flavor as they propagate through space.

Neutrinos in the original Standard Model were assumed to be massless, neutral particles. The solar neutrino problem, however, offered the opportunity to investigate a new description of neutrinos as neutrino oscillation could potentially resolve the observed flux discrepancy. Produced via the weak interaction, Standard Model neutrinos come in three ‘flavors’ associated with respective charged leptons: electron, muon, and tau. For instance, when an electron and a neutrino are produced in an interaction, that neutrino will always be an electron neutrino [1].

In a mathematical sense, neutrinos have three flavor eigenstates, and in interactions with charged leptons, the lepton and neutrino flavor will match. Neutrino oscillation proposes that neutrinos also have a set of three mass eigenstates, and that the flavor and mass eigenstates are not the same. When an electron is produced alongside a neutrino, that neutrino will be a well-defined flavor eigenstate but a linear superposition of the three

mass eigenstates [15].

$$|\nu_\alpha\rangle = C_1|\nu_1\rangle + C_2|\nu_2\rangle + C_3|\nu_3\rangle \quad (2.1)$$

As that neutrino propagates through space, if the eigenvalues of the mass eigenstates are different, the phases of the mass eigenstates will change independently of each other, so when the neutrino is detected at some distance from its origin, the proportions of the mass eigenstates may have changed. This means there is a nonzero probability that the flavor of the neutrino has changed [15]. For a two flavor oscillation between ‘a’ and ‘b’, the probability of flavor change is given by:

$$P(\nu_\alpha \rightarrow \nu_\beta) = \sin^2(2\theta)\sin^2(1.27\Delta m^2 \frac{L[\text{km}]}{E[\text{GeV}]}) \quad (2.2)$$

In this equation, θ is the mixing angle between the two flavors, where $\theta = 0^\circ$ indicates that there will be no mixing, and $\theta = 45^\circ$ results in maximal mixing. The Δm^2 term is the mass squared difference. This term is very significant, since it indicates that neutrinos must have nonzero mass, and that the mass eigenstates must have different masses for mixing to occur. The ratio $\frac{L}{E}$ is given by the distance from the source divided by the neutrino energy. For solar neutrino experiments, this is a fixed value, but other neutrino experiments can optimize this value in their detector design to maximize oscillation sensitivity [15].

The relationship between the three flavor eigenstates and three mass eigenstates is represented by a 3x3 mixing matrix known as the PMNS (Pontecorvo-Maka-Nakagawa-Sakata) matrix. In its expanded form, the two sets of eigenstates are related through three rotation matrices and three complex phases [15].

$$\text{Flavor-mass relation: } \begin{pmatrix} \nu_e \\ \nu_\mu \\ \nu_\tau \end{pmatrix} = \begin{pmatrix} U_{e1} & U_{e2} & U_{e3} \\ U_{\mu 1} & U_{\mu 2} & U_{\mu 3} \\ U_{\tau 1} & U_{\tau 2} & U_{\tau 3} \end{pmatrix} \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix} \quad (2.3)$$

$$\text{PMNS expanded matrix: } U = ABCD \quad (2.4)$$

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_{23} & \sin\theta_{23} \\ 0 & -\sin\theta_{23} & \cos\theta_{23} \end{pmatrix} \quad (2.5)$$

$$B = \begin{pmatrix} \cos\theta_{13} & 0 & \sin\theta_{13}e^{-i\delta_{CP}} \\ 0 & 1 & 0 \\ -\sin\theta_{13}e^{i\delta_{CP}} & 0 & \cos\theta_{13} \end{pmatrix} \quad (2.6)$$

$$C = \begin{pmatrix} \cos\theta_{12} & \sin\theta_{12} & 0 \\ -\sin\theta_{12} & \cos\theta_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.7)$$

$$D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{i\beta} & 0 \\ 0 & 0 & e^{i\gamma} \end{pmatrix} \quad (2.8)$$

The scientific community needed evidence to verify the hypothesis that the reduced flux observed in the Homestake experiment was due to neutrino oscillation. Several experiments were conducted to continue investigating the solar neutrino problem and its possible explanation.

One such experiment was Super-Kamiokande, based in Higashi-Mozumi, Gifu, Japan. Constructed within the Kamioka mine, Super-Kamiokande continues to be the largest water Čerenkov detector to date. The detector is comprised of a stainless-steel tank filled with 50,000 tons of water and has around 13,000 photomultiplier tubes surrounding the tank to detect scintillation [16]. In Super-K, neutrinos are detected indirectly when Čerenkov light is produced by charged particles emitted in neutrino interactions [17]. Unlike the Homestake experiment, which only counted neutrinos through a radiochemical technique by extracting and counting the Argon atoms produced in the inverse beta decay, Super-K also measured to high precision the spectra of recoiling electrons [1][17].

In parallel to Super-K, the Sudbury Neutrino Observatory (SNO) experiment in Canada was designed specifically to measure properties of solar neutrinos [18]. SNO's detector, like Super-K, was a water Čerenkov detector but utilized heavy water (D_2O) since the deuterium in heavy water enabled the experiment to detect all neutrino flavors. Researchers on the project predicted that to solve the solar neutrino problem, they must be able to make independent measurements of muon and tau neutrinos in addition to the typical electron neutrino detection [18].

Results from both Super-K and SNO confirmed the neutrino oscillation hypothesis: solar neutrino flux ratios matched those expected if oscillation occurred over long distances [1]. This evidence revolutionized the field of neutrino physics. The Standard Model needed to be revised to account for neutrino masses, these masses and mixing angles needed to be experimentally measured, and new avenues in particle physics emerged.

2.4 Nature of the Neutrino

Neutrino oscillation experiments have confirmed that neutrinos are massive spin $\frac{1}{2}$ particles [1]. Since their masses are so small, they are considered to be relativistic, and like other Standard Model fermions, they obey the Dirac equation.

$$\text{Dirac equation of motion: } (i\gamma^\mu \delta_\mu - m)\psi = 0 \quad (2.9)$$

The solutions ψ of the Dirac equation are 4-component spinors. They can be represented in the Weyl basis by ξ , the left-handed chiral field, and η , the right-handed chiral field.

$$\text{Weyl basis: } \gamma^\mu = \begin{pmatrix} 0 & \sigma^\mu \\ \bar{\sigma}^\mu & 0 \end{pmatrix}, \gamma^5 = \begin{pmatrix} -I & 0 \\ 0 & I \end{pmatrix}, \psi = \begin{pmatrix} \xi \\ \eta \end{pmatrix} \quad (2.10)$$

In this basis, the left and right chiral states are eigenvectors of the γ^5 matrix with eigenvalues of -1 and 1, respectively. Chirality of Dirac spinors is not conserved; the left and right chiral fields are coupled and therefore experience mixing with time. When mass of the particle approaches 0, however, chirality and helicity become equivalent, allowing

ξ and η to decouple and evolve independently of each other since the helicity of free particles is conserved [19].

Physicist Ettore Majorana theorized a new type of particle: the Majorana particle. Like the Dirac particle, Majorana's particle is represented by a spinor ψ and moves according to the Dirac equation. The novelty of the Majorana particle has to do with how its chiral states ξ and η interact. For Majorana particles, one chiral state can be derived from the other. For Dirac particles, however, a chiral state cannot be known from its partner without more information. Because of this property, Dirac spinors have 4 degrees of freedom and Majorana spinors have only 2 [19]. Dirac and Majorana spinors are only subtly distinctive, but these differences have significant consequences.

$$\text{Majorana spinor: } \psi = \begin{pmatrix} \xi \\ -i\sigma_2\xi^* \end{pmatrix} \quad (2.11)$$

One such consequence is a property of the Majorana spinor's charge conjugated state. When a Majorana spinor undergoes charge conjugation, the result is the original spinor. When a particle is charge conjugated, it becomes its antiparticle, so this property implies, in contrast to Dirac particles which have distinct antiparticle spinors, that a Majorana particle's antiparticle is itself – the particle and antiparticle have the same spinors [19].

There are conditions that must be met for a particle to be a Majorana particle. The first is that the particle must be massive, since the distinction between Majorana and Dirac particles vanishes when mass is zero. Additionally, the particle must be spin $\frac{1}{2}$, and it must have a neutral charge. The only particles that meet these conditions in the Standard Model are neutrinos [19].

2.5 Neutrinoless Double Beta Decay

The most accessible test of neutrino nature and most sensitive probe of the Majorana neutrino mass is neutrinoless double beta decay ($0\nu\beta\beta$) [19]. In $0\nu\beta\beta$ decay a parent nucleus (Z, A) decays into a daughter nucleus ($Z + 2, A$) by emitting two electrons. This process is shown in Figure 2.1 in contrast to $2\nu\beta\beta$ decay, which emits two anti-neutrinos along with the electrons and conserves lepton number. $0\nu\beta\beta$ can occur only if neutrinos and anti-neutrinos are one and the same – if they are Majorana particles. As shown by Figure 2.2, this decay would take place via the exchange of a Majorana neutrino that connects two vector-axial vector (V-A) weak interactions [2].

For the $0\nu\beta\beta$ decay to occur, however, it is possible that the origin of the “light neutrino” mass that we observe is not the same as that of the masses of charged fermions. Accordingly, the mechanism by which neutrinos get their extremely small masses must be considered. The most prominent theory is the seesaw mechanism, where heavy right-handed neutrinos mix with left-handed ones, generating light Majorana neutrinos in the process [2].

In this case, the Majorana nature of the resulting neutrinos would be related to breaking lepton number and may contribute to the matter-antimatter asymmetry of the universe. The excess lepton numbers from the Majorana neutrino mass terms could translate into

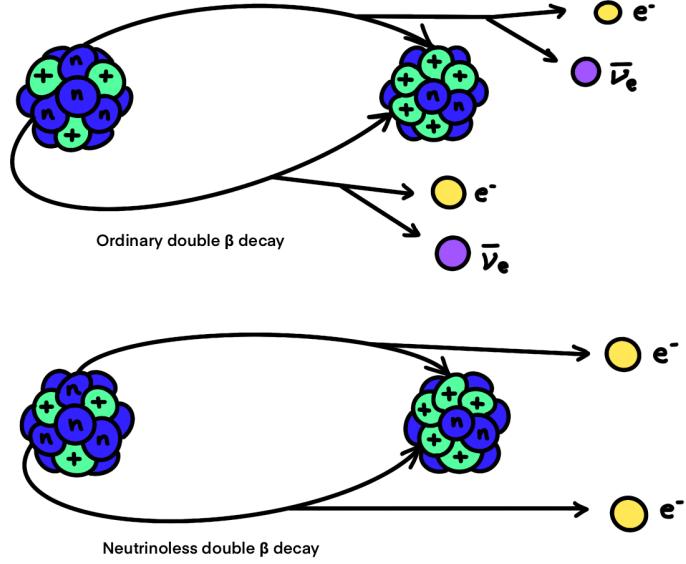


Figure 2.1: *Decay Modes* — Drawn representation of the $2\nu\beta\beta$ and $0\nu\beta\beta$ decay modes.

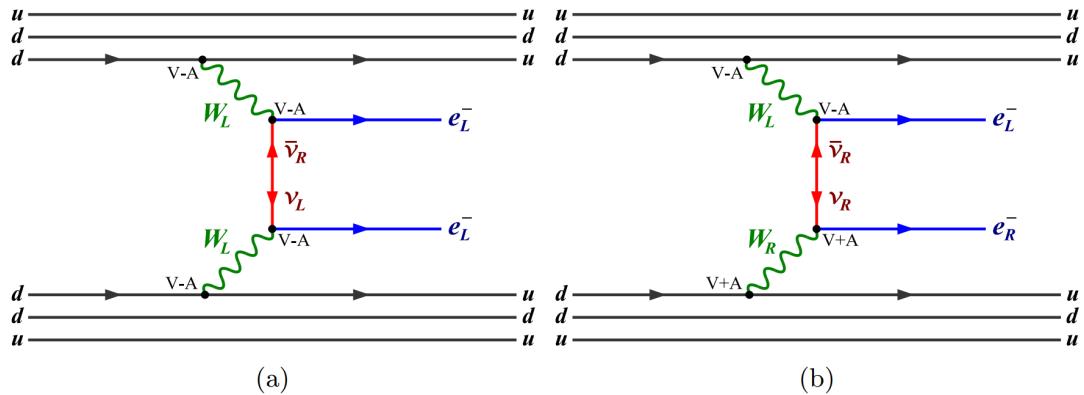


Figure 2.2: *Decay Diagrams* — Feynman diagrams of two possible mechanisms by which $0\nu\beta\beta$ decay could occur. Diagram (a) shows the exchange of a Majorana neutrino where mass is given by the seesaw mechanism, while diagram (b) illustrates a right-handed current contribution. Image sourced from [2].

number excesses of baryons through baryon number violation in high temperature electroweak processes [20].

Alternatively, $0\nu\beta\beta$ could occur via other mechanisms, including Left-Right Symmetry, R-Parity violating Supersymmetry, or Extra Dimension theories [2].

Observation of $0\nu\beta\beta$ decay would prove lepton number violation, provide direct evidence that neutrinos are Majorana particles, and support the theory that lepton violating processes contribute to the matter-antimatter asymmetry problem [2]. Whether $0\nu\beta\beta$ has already been observed is a matter of opinion, as researchers from the Heidelberg-Moscow collaboration claimed to have observed it with over 95% certainty in 2002 [21]. Regardless, the current $0\nu\beta\beta$ half-life limit has been set at $T_{1/2}^{0\nu} > 2.3 * 10^{26}$ years, with a 90% confidence level, by the Kamland-Zen experiment in 2023. Kamland-Zen, with its Xenon-loaded liquid scintillator, probes the $0\nu\beta\beta$ decay half-life in ^{136}Xe [22]. This half life is directly proportional to the square of the effective Majorana neutrino mass according to:

$$\text{Half-life: } [T_{1/2}^{0\nu}]^{-1} = \left(\frac{m_{\beta\beta}}{m_e} \right)^2 G_{01} |M_{m\nu}|^2 \quad (2.12)$$

$$\text{Effective mass: } \langle m_{\beta\beta} \rangle \equiv |\sum_i U_{ei}^2 m_{\nu i}| \quad (2.13)$$

In these equations, $m_{\beta\beta}$ is the effective Majorana neutrino mass, m_e is the mass of the electron, G_{01} refers to the phase space integral of the nuclear process, $M_{m\nu}$ is the mass mechanism nuclear matrix element, U_{ei} are elements of the PMNS matrix, and $m_{\nu i}$ are the light Majorana neutrino masses [2][22].

2.6 SuperNEMO

The SuperNEMO experiment is currently taking data for neutrinoless double beta decay detection using a detector called the demonstrator module. This detector is located within an underground section of the Laboratoire Souterrain de Modane (LSM) located in the Fréjus tunnel near Modane in the French Alps (Figure 2.3). With the protection of 1700 meters of rock above, LSM's location protects experiments like SuperNEMO from cosmic ray backgrounds that could interfere with signal detection [3].

SuperNEMO's demonstrator module takes advantage of this natural shielding. The detector's primary components, shown in Figure 2.4, are the central source foil, tracking chambers on either side of the foils, and a layer of calorimeters behind the trackers. As the laboratory is situated on the border between France and Italy, the half of the detector closest to the border is known as the 'Italian side', while the other half is the 'French side'. This naming scheme is commonly used to indicate which side of the detector is being mapped in data analysis and visualizations within the collaboration.

The demonstrator module's source foils contain Selenium-82, which is known to undergo ordinary double beta decay. These foils are 0.3 mm thick to allow as many electrons as possible to escape into the tracking chambers. Each tracker is composed of 113 columns of 9 drift cells – 1017 drift cells in total. The tracking chambers are filled predominantly with Helium gas, but also contain Argon and Ethanol. Each gas has a role in the tracker

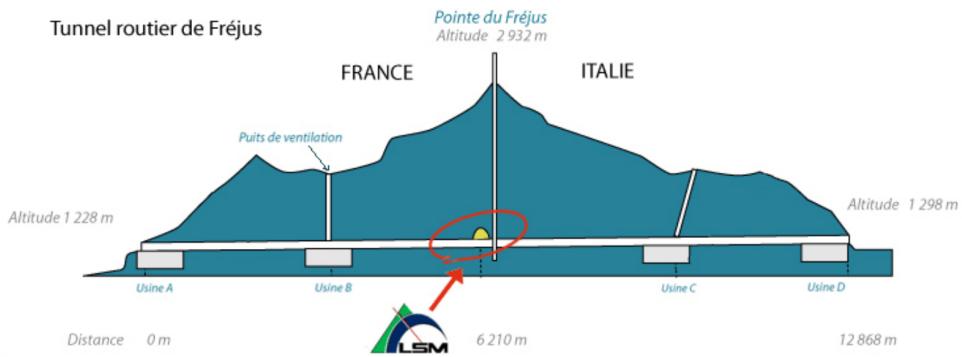


Figure 2.3: *Laboratoire Souterrain de Modane* — Depiction of LSM's location, sourced from [3].

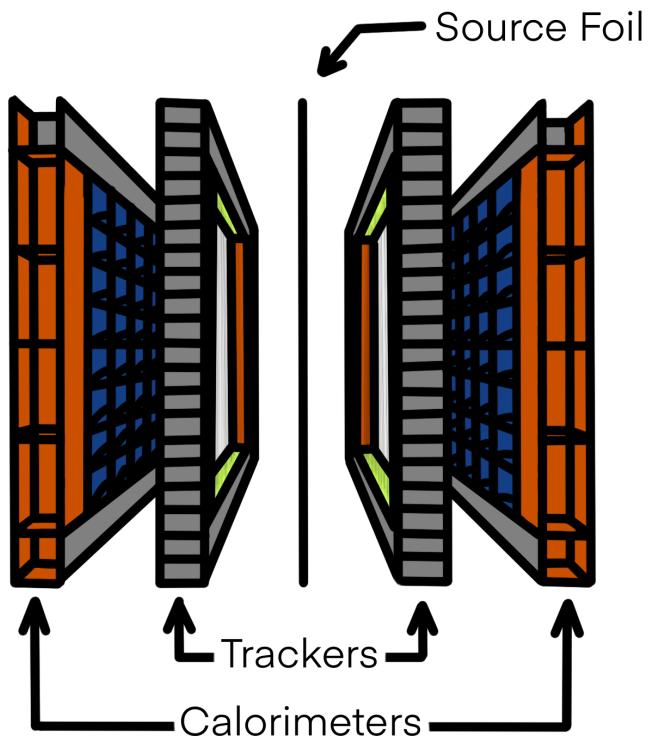


Figure 2.4: *Demonstrator Module* — Layout of the detector's internal components, indicating the locations of the source foils, tracking chambers, and calorimeter walls.

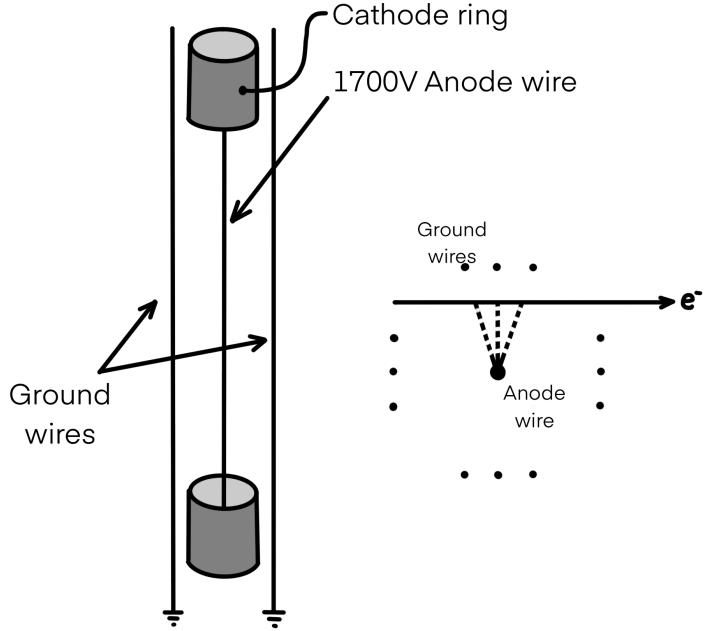


Figure 2.5: *Drift Cell* — Diagram of a single tracker drift cell. The central anode wire is set at +1700 V, and is surrounded by 12 ground wires as shown in the top view on the right. The ends of the anode wire are capped by cathode rings.

function. Helium (95%), as an unreactive, low atomic number gas causes less scattering and energy loss for passing particles than larger or more reactive substances. The Argon (1%) generates ionized electrons more easily due to its lower ionization point, and Ethanol (4%) acts as a quencher to absorb photons and prevent continuous discharge [4]. The gas temperatures and pressures, which can impact particle motion within the tracker, are monitored with a recently updated Graphics User Interface (GUI) that sends email alerts for any unexpected changes to temperature or flowrate [6]. Careful monitoring minimizes contamination in the tracker system.

A single tracker drift cell has two cathode ring caps, an anode wire set at +1700 volts, and 12 ground wires placed around the anode wire. Each drift cell is radially 4.4 cm in diameter and 4 m in height [2]. A diagram of a cell (not to scale) is shown in Figure 2.5. As charged particles pass through the drift cells, they ionize the surrounding gas. The cells then measure the drift times of the resulting particles radial to and along the anode wire. These times are translated into distances algorithmically, resulting in a three-dimensional map of particle position.

The last layer of the inner detector, the calorimeter walls, measure particles' energies. These walls are comprised of 520 optical modules each, with more above and below the tracking chambers to capture all particles exiting the detector. The optical modules are made of a polystyrene scintillator that produces flashes of light when struck by charged particles. Light emissions are captured by photomultiplier tubes (PMTs) at the back of the modules, and the PMTs output electrical signals based on the strength of the emitted light. By looking at these signals, researchers can measure the energy of the particles that caused them [8].

Beyond the design of its internal structure, the demonstrator module aims to achieve an ultra-low background by incorporating an anti-Radon tent and shielding walls [6]. The anti-Radon tent is fully installed, and construction on iron shielding to reduce background from gamma radiation is underway. Once the iron structure is installed, neutron shielding of polyethylene sheets and water-filled polyethylene blocks will be added for further background reduction [9].

Chapter 3

Project Background

3.1 Drift Cell Specifics

My dissertation project focuses on modelling drift times and converting them to distances for the radial component of a particle's position within the demonstrator module's tracker. As such, knowledge of the physics behind the drift cell mechanism is essential to understanding how to model particle behavior within the tracker. As mentioned in the previous section, a single drift cell contains a central anode wire set at a high voltage (1700V). This anode wire is surrounded by 12 ground wires, 3 to a side. The cell also contains a cathode cap at each end of the anode wire. When a charged particle passes through one of these cells, the gas in the chamber (95% Helium, 4% Ethanol and 1% Argon) ionizes. In NEMO-3, the anode wire in drift cells was set at +1800 volts and ionized gases yielded about six electrons per centimeter. Since the anode wires in SuperNEMO are at a slightly lower voltage, ionization likely yields an average of 5.7 electrons per centimeter. Once the surrounding gas is ionized, the electrons produced drift from some unknown distance r towards the positively charged anode wire under the electric force, $\mathbf{F}_E = q\mathbf{E}$.

The time it takes for the electrons to reach the central wire – the drift time – depends on a multitude of factors, including electric field strength, gas pressure, and gas composition. The gas properties are taken to be constant, but the electric field strength varies depending on where within the drift cell and within the tracker ionization occurs. The variation is greatest at the edges and corners of the tracker, due to edge effects. Detailed simulations of the electric field are needed to model particle location using the drift times given by the demonstrator module. Once the drift electrons reach the anode wire, the tracker, set to proportional or Geiger mode, records the hit as a timestamp. In proportional mode, anode signals with amplitudes proportional to electron energy are produced upon impact. This mode has small amplitude signals and short dead time where hits cannot be recorded, making it the preferred method for high frequency data recording. Geiger mode, on the other hand, produces a signal that is not proportional to electron charge. Rather, a single ionization event generates the signal on the order of 100mV, which can be read easily by electronics [4]. In Geiger mode, the anode wire is set to a high voltage in the range of 1600 – 1700 V. This mode experiences a longer dead time

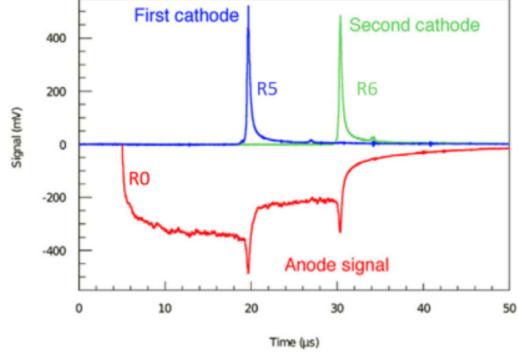


Figure 3.1: *Signal Graph* — Signal output from the SuperNEMO tracker. R0 corresponds to the timestamp taken when the anode signal passes its initial threshold (-10 mV). R5 and R6 refer to cathode timestamps triggered when the ion plasma reaches the two cathode caps. Graph taken from [4].

than proportional mode, around 10 ms, so it yields less event tagging efficiency.

The tracker outputs a series of timestamps corresponding to anode and cathode signals passing certain thresholds. These timestamps, R0, R5, and R6 can be seen in Figure 3.1. R0, the timestamp of most relevance to this dissertation, is the timestamp that is taken when the anode signal initially drops below -10 mV after an event has been triggered in the calorimeter. Drift time is taken to be the time between the trigger timestamp and the R0 timestamp.

3.2 Landells' Results

My predecessor, Betsy Landells, developed a drift model and uncertainty equation to calculate radii from drift times in the demonstrator module tracker. To accurately model electric field dependency, a detailed map of the tracker's electric field was needed. This was generated through simulation in COMSOL; a contour map showing electric field strength in a section of the tracker is shown in Figure 3.2. Landells needed to tackle the issue of varying electric field strength based on wire configuration and neighboring cells. She divided each cell into quadrants and categorized each section as ‘Corner,’ ‘Edge,’ or ‘Center’. Cells composed of all ‘Center’ quadrants (center cells) are symmetric across the x and y axes, while cells containing two ‘Edge’ quadrants (edge cells) are symmetric over only x or y, and those with ‘Edge’ and ‘Corner’ quadrants (corner cells) are diagonally symmetric.

The drift model was based on the distance-time relationship established during the NEMO-3 experiment.

$$t(r) = \frac{ar}{r^b} \quad (3.1)$$

This equation, when inverted, yields:

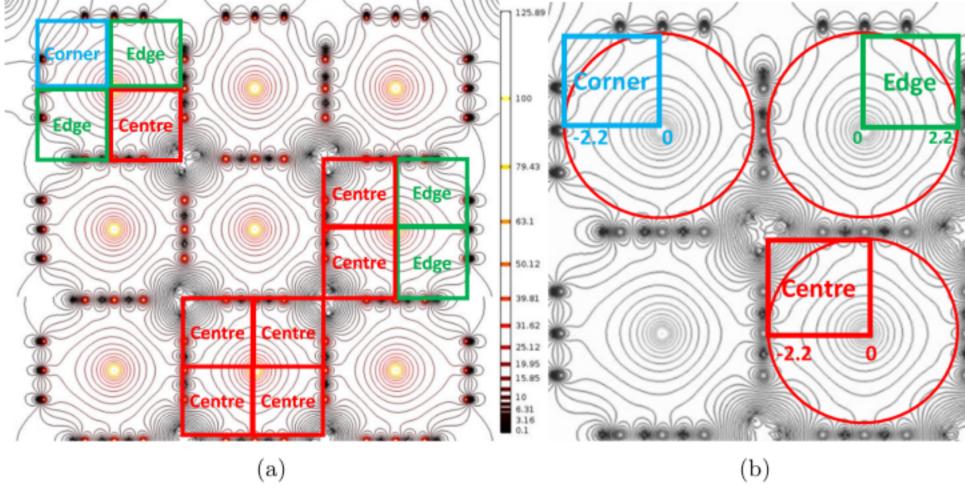


Figure 3.2: ***E*** Field Map — Electric field strength contour map showcasing 'Edge,' 'Corner,' and 'Center' quadrants of various drift cells. Each quadrant is $2.2 \times 2.2 \text{ cm}^2$. Figure sourced from [4].

$$r(t) = \left(\frac{t}{a}\right)^{\frac{1}{1-b}} \quad (3.2)$$

Where a and b are model-dependent parameters. To find a and b , Landells fit the equation to simulated data. This data, generated by SuperNEMO's SND drift software, simulated the drift times of 500 electrons in an inverse Gaussian distribution motivated by the driven Brownian motion of the particles. Landells performed separate fits on the inner and outer regions of the cells, using an experimentally determined cutoff radius r_{cut} .

To find r_{cut} , Landells utilized a reduced chi-squared analysis for optimization. A reduced chi-squared weights chi-squared:

$$\chi^2 = \sum_i \frac{(O_i - P_i)^2}{\sigma_i^2} \quad (3.3)$$

Which sums over the squared difference between observed and predicted datapoints divided by the variance σ^2 .

$$\chi^2_{reduced} = \frac{\chi^2}{\nu} \quad (3.4)$$

The reduced chi squared is weighted by the number of datapoints, or degrees of freedom. This method avoids bias due to overfitting.

Landells' model fitting and chi-squared optimization gave parameters a and b and the cut-off radius for all three quadrant types and for three different pressures. These parameters are recorded in the tables in Figure 3.3.

Landells also developed an equation for uncertainty, based on the equation below, which assumes a , b , and t to be variables of a function f , where a and b are independent of t :

Best fit parameters $r < x; t < t_x$									
Cell type: x (cm) pressure (mb)	Corner			Edge			Centre		
	2.15	2.2	1.9	850	880	910	850	880	910
t_x (μ s)	3.34	3.45	3.59	3.73	4.15	4.12	2.95	2.97	3.06
a_{lo} ($10^{-7}\mu$ s/cm)	8.79	9.13	9.36	9.32	9.62	9.88	9.61	9.88	10.2
b_{lo}	-0.89	-0.88	-0.90	-0.91	-0.95	-0.92	-0.93	-0.93	-0.93
a ($10^{-7}\mu$ s/cm)	7.66	7.92	8.16	8.05	8.35	8.56	8.28	8.53	8.77
b	-0.87	-0.87	-0.87	-0.90	-0.92	-0.90	-0.90	-0.90	-0.90
a_{hi} ($10^{-7}\mu$ s/cm)	6.77	6.95	7.20	7.05	7.37	7.53	7.25	7.48	7.69
b_{hi}	-0.85	-0.86	-0.85	-0.88	-0.88	-0.87	-0.88	-0.87	-0.87

Best fit parameters $r \geq x; t \geq t_x$									
Cell type: x (cm) pressure (mb)	Corner			Edge			Centre		
	2.15	2.2	1.9	850	880	910	850	880	910
t_x (μ s)	3.34	3.45	3.59	3.73	4.15	4.12	2.95	2.97	3.06
a_{lo} ($10^{-7}\mu$ s/cm)	5.16	4.91	5.17	3.29	3.29	3.89	4.18	4.33	4.86
b_{lo}	-1.61	-1.69	-1.68	-2.30	-2.34	-2.20	-2.19	-2.17	-2.10
a ($10^{-7}\mu$ s/cm)	5.18	4.94	5.25	3.34	3.39	4.03	3.86	4.19	4.55
b	-1.40	-1.48	-1.45	-2.04	-2.07	-1.91	-1.99	-1.93	-1.90
a_{hi} ($10^{-7}\mu$ s/cm)	5.17	4.93	5.28	3.42	3.48	4.14	3.80	4.16	4.50
b_{hi}	-1.23	-1.31	-1.27	-1.82	-1.84	-1.69	-1.79	-1.72	-1.68

Figure 3.3: *Parameter Tables* — The top table gives the a, b, and cutoff time parameters for all quadrant types and pressures in the inner areas of the drift cells ($r < x$) where x is the cutoff radius determined by the chi-squared optimization. The bottom table gives the same parameters but for the outer areas ($r > x$). Tables taken from [4].

$$\sigma_r = \sqrt{\left(\frac{\delta f}{\delta t}\right)^2 \sigma_t^2 + \left(\frac{\delta f}{\delta a}\right)^2 \sigma_a^2 + \left(\frac{\delta f}{\delta b}\right)^2 \sigma_b^2 + 2\left(\frac{\delta f}{\delta a}\right)\left(\frac{\delta f}{\delta b}\right)\sigma_{ab}} \quad (3.5)$$

Substituting in $r(t)$ as f:

$$\sigma_r = r \sqrt{\left(\frac{1}{(1-b)t}\right)^2 \sigma_t^2 + \left(\frac{1}{(b-1)a}\right)^2 \sigma_a^2 + \left(\frac{\ln(\frac{t}{a})}{(1-b)^2}\right)^2 \sigma_b^2 + 2\left(\frac{\ln(\frac{t}{a})}{(b-1)^3 a}\right)\sigma_{ab}}. \quad (3.6)$$

3.3 Křížák Track Fitting

Determining the radius at which the decay particle passes through each drift cell is only part of the story when it comes to reconstructing its path. Accurate reconstruction of particle trajectories is essential to achieve the best background reduction. For his Bachelor project, SuperNEMO member Tomáš Křížák developed the first version of an algorithm using Legendre transforms to reconstruct linear tracks in the demonstrator module [5]. His project focused on the more challenging (x,y) reconstruction since drift model outputs give radial distance but not exact position.

Each radius calculated by the drift model gives a circle of potential points the particle may have passed through. Reconstructing a track is a matter of finding a trajectory that ideally is tangent to all the circles defined by tracker cell radii. Figure 3.4 shows examples of good and bad track reconstructions.

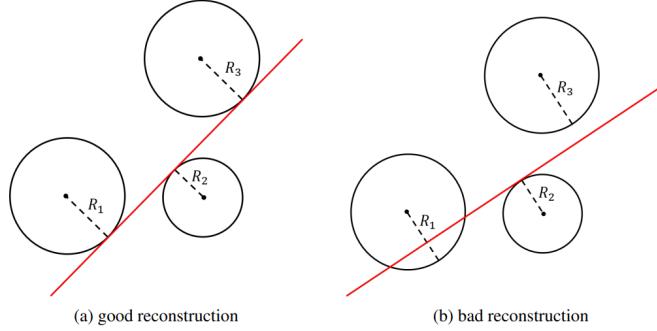


Figure 3.4: *Track Reconstructions* — Examples of (a) good and (b) bad reconstructions using the Legendre transform method. Figure courtesy of [5].

The Legendre transform reconstruction method utilized by Křížák has a basis in thermodynamics and theoretical physics applications. It also has a geometric interpretation that matches the radial data generated by the SuperNEMO tracker. Advantageously, this method is robust to noise and missing data points, but it has a long computation time due to the involvement of multiple iterative searches [5].

Křížák’s algorithm looks for the intersection of Legendre images of the circles described by the drift cell radii then constructs the tangent line that most closely matches the hits.

3.3.1 Legendre Transformations

Let $y = f(x)$ be a continuous, differentiable, convex function that defines a set of points $(x, f(x))$. Each point has a line $y = px - g$ that runs tangent to the function, where g and p are arbitrary parameters. The set of parameters that generate these tangent lines can be represented by a function $g(p)$.

Performing a Legendre transform on the function $y = f(x)$ yields a “Legendre Image” $g(p)$, the set of tangent line parameters. This transform is shown in the equation below:

$$g(p) = x(p)p - f(x(p)) \mid \text{Where: } x(p) = (f')^{-1}(p) \quad (3.7)$$

Applied to a circle centered at (x_0, y_0) of radius r , the function $y(x)$ is given by the combination of a convex and concave semicircles:

$$f_{\pm}(x) = y_0 \pm \sqrt{r^2 - (x - x_0)^2} \quad (3.8)$$

Applying the Legendre transform to f_- , $p(x)$ is obtained by differentiating f_- with respect to x , then $p(x)$ is inverted, giving $x(p)$. This can then be plugged into Equation 3.7, resulting in:

$$g_-(p) = x_0p - y_0 + r\sqrt{1 + p^2} \quad (3.9)$$

Similarly:

$$g_+(p) = x_0p - y_0 - r\sqrt{1 + p^2} \quad (3.10)$$

These images describe the tangent lines to the circle with slope p and intercept $-g$. Re-parameterizing these lines in terms of R , the distance from the line to the origin at the tangent point, and θ , the angle between the line and the vertical axis, describes all the tangents of the circle uniquely with polar coordinates.

$$g = \frac{R}{\sin\theta}, \quad p = -\frac{\cos\theta}{\sin\theta} \quad (3.11)$$

$$R_{\pm} = -x_0\cos\theta - y_0\sin\theta \mp r \quad (3.12)$$

In SuperNEMO, R is limited by the size of the drift cells.

Chapter 4

Design and Data Preparation

With a preliminary drift model in place for radial position in the demonstrator module, my project aimed to utilize the model on real datasets. The initial goal was to validate the model and see how well it was able to reconstruct radii from drift times given by the timestamps in the UDD (Unified Digital Data) format outputted by SuperNEMO’s software, Falaise. To this end, I developed methods to convert UDD ROOT files to pandas data frames, unpacked the timestamp-format data, and performed analysis on calibration data using python programs. As one of the first members of SuperNEMO to use python in analysis, I developed conversion infrastructure to access data in python-based Jupyter notebooks and created a class to facilitate faster analysis.

4.1 UDD .brio to ROOT and Falaise

Falaise is the software developed and used by the SuperNEMO collaboration. It includes modules to visualize data in the trackers and calorimeters of the demonstrator module, create simulated data of background or signal events, and perform rudimentary track reconstructions [23]. The Unified Digital Data (UDD) format is derived from the detector’s raw data and is a transitional Falaise-based format in the data pipeline that bridges the raw data of the detector and the Calibrated Data (CD) – the ultimate output. Figure 4.1 shows how SuperNEMO’s data is transformed between formats and is connected to the simulated data generated by Falaise [6].

This project utilized the ROOT version of the UDD format. ROOT files are constructed much like trees; there is a main tree that is divided into branches, which are further divided out into sub-branches. In the case of the calibration data, the main tree contains around 150 branches with information about the run, the nature of the data, and outputs of the calorimeter and tracker hardware. Of interest to my analysis were the **digicalo** and **digitracker** sets of branches. These groups contain data on the locations and times of events in the trackers and calorimeters needed to test the drift model.

I specifically used calibration data from the ROOT file on Run 840. This dataset measures 9590 events, so each feature branch contains 9590 sub-branch arrays. For instance, **digitracker.anodetimestampR0**, which gives the first anode timestamp of each tracker

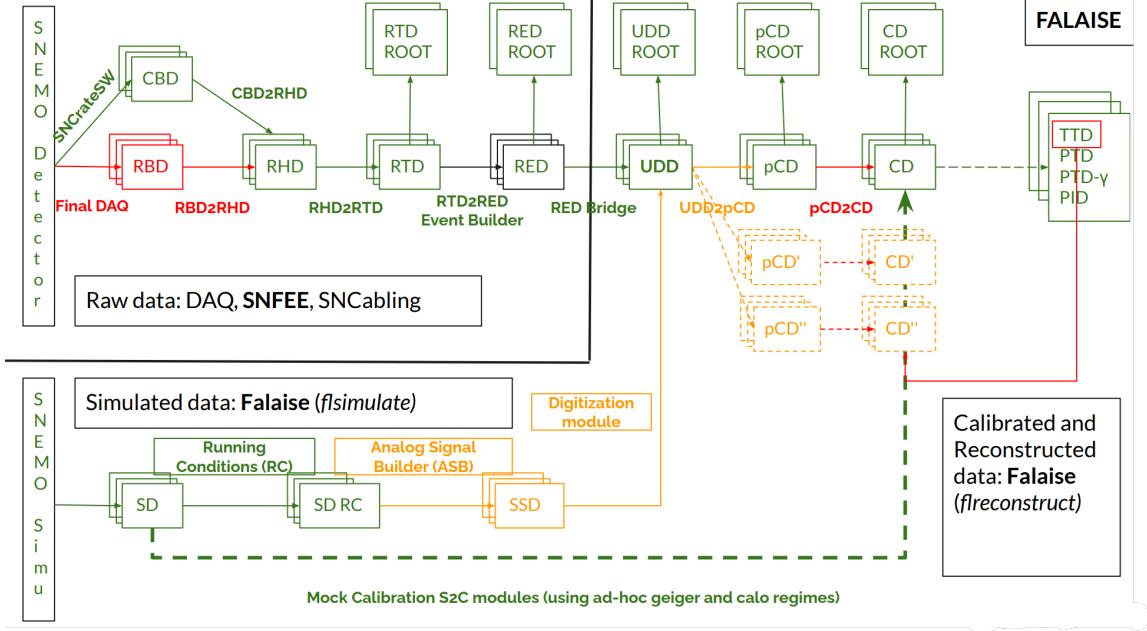


Figure 4.1: *Data Pipeline* — The data pipeline, with inputs from the detector and Falaise simulation. Image sourced from material presented in the January 2023 SuperNEMO collaboration meeting [6].

cell that was hit in any given event, contains array-like sub-branches that differ in size for each event. The length of an array depends on how many drift cells register a hit during the corresponding event.

4.2 ROOT to Pandas

Due to the Windows nature of my computing system, I was unable to retrieve data directly from the ROOT files. Instead, I converted the data of interest to a pandas data frame with a numpy-based python library called uproot [24]. Uproot allowed me to load and access the data in my Jupyter Notebooks, but to fully convert the data to pandas format, I designed a program to access the desired branches and systematically set up a data frame with event number, side, row, column, and anode and calorimeter timestamp data for each tracker hit. This code can be found in Appendix A.

4.3 Drift Time Calculations

The next step towards analysis was converting the timestamp datapoints to drift times. In the demonstrator module, the tracker and calorimeter run on different clocks. This necessitates independent conversions for the two timestamps from ticks to microseconds along with adjustment factors due to cable lengths and other offsets. In the UDD format, the tracker timestamps are stored in **digitracker.anodetimestampR0** and the calorimeter timestamps in **digicalo.timestamp**. While there is an anode timestamp for

each hit, I only took the first entry in the calorimeter timestamp data for each event, as that indicates the “trigger time” of the event. Similarly, the offsets required data from **digicalo.rising_cell**; this data is also event-based rather than hit-based. The conversion equation from timestamps to drift time is of the form:

$$t_{drift} = TDC - 400[\text{ns}] + PT + PST \quad (4.1)$$

In this equation, TDC is the timestamp of the calorimeter hit (or trigger timestamp) in units of 6.25 ns, PT is the post-trigger setting configured to 250 ns. This component is a fixed parameter to tune the position of the signal pulse in the recording window. PST is the pulse start time, calculated by:

$$PST = \frac{RC * T_{sampling}}{256} \quad (4.2)$$

The rising cell data, RC , stored in **digicalo.rising_cell** is the start time measured by firmware using the constant fraction discriminator method. It is equivalent to the time when the pulse height reaches 25% of its maximum amplitude. $T_{sampling}$, the sampling period, is 0.39062500 ns. One important note on calculating drift time is that the tracker timestamps are recorded in units of 12.5 ns per tick, while the calorimeter timestamps are in 6.25 ns per tick, so a factor of 2 is needed when comparing the two. Both timestamps represent the absolute time since the start of the run.

Chapter 5

Results and Analysis

5.1 Calculations and Data

With the drift times calculated using Equation 4.1, the next step was to implement the drift model described in Section 3.2. Based on the layer and column labels of each hit, each data frame row was tagged as ‘center,’ ‘edge,’ or ‘corner’ and then its drift time was put through a threshold to determine whether the particle passed through the inner or outer section of the cell. This threshold t_x is cell type-dependent and values are recorded in the parameter data tables in Figure 3.3. With cell type and cell location determined, a column of cell types, e.g. “center_in,” was added to the data frame. Each cell type label corresponds to a specific set of parameters a and b to be used in radius calculation.

With those parameters in place, it was a simple matter of plugging drift times into Equation 3.2 and adding a new column “Drift_radius” to the data frame. The first five rows of the final version of the Drift Data Frame are shown in Figure 5.1. All these data setup steps are automated in the AnalyzeDrift class (refer to Appendix A) I developed.

	Event	ID	Side	Layer	Column	R0	Calo_time	Rising_cell_time	Drift_time	Cell_type	Drift_radius
0	0	0	1	0	7	847289447.0	1.694578e+09	82928.0	4.017214	edge_in	0.683123
1	0	1	1	8	0	847289127.0	1.694578e+09	82928.0	0.017214	corner_in	0.037672
2	0	2	1	5	3	847289197.0	1.694578e+09	82928.0	0.892214	center_in	0.304760
3	0	3	1	6	2	847289122.0	1.694578e+09	82928.0	-0.045286	center_in	NaN
4	0	4	1	1	7	847289281.0	1.694578e+09	82928.0	1.942214	center_in	0.458947

Figure 5.1: *Processed Dataframe Entries* — Each row of this data frame contains event, location, timestamp, drift time, cell type, and radius information for a single hit. This data frame can be accessed by inputting the pandas data frame of detector data into AnalyzeDrift, then calling drift_df.

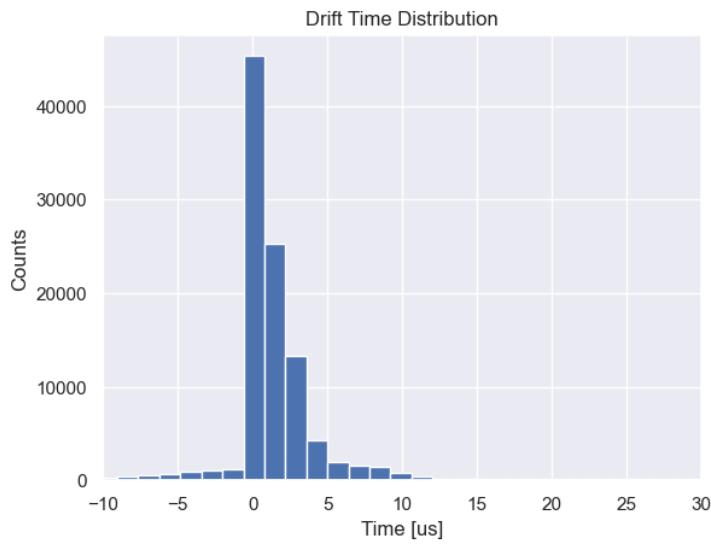


Figure 5.2: *Drift Time Distribution* — Distribution of calibration data drift times. The maximum drift time in the dataset is $295\mu\text{s}$. Axes were set to the observed limits to better showcase the shape of the distribution.

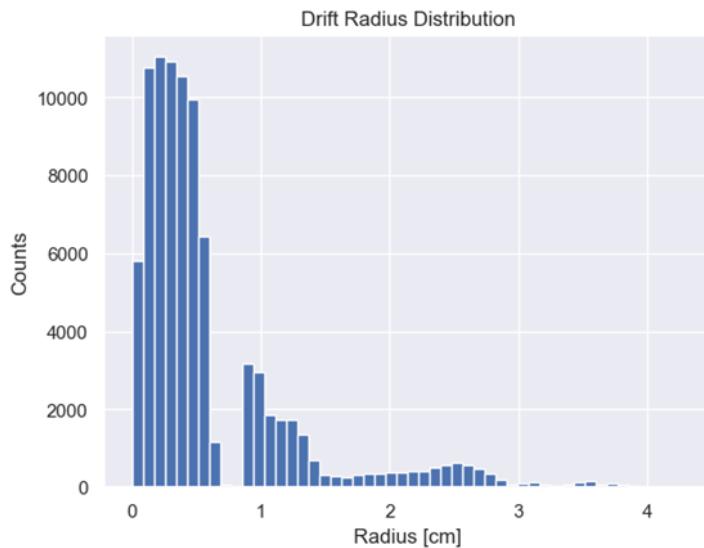


Figure 5.3: *Drift Radius Distribution* — Distribution of radii calculated using Landells' drift model. The maximum radius is 4.28 cm, with the bulk of the data accumulating close to the center of the cell.

5.2 Drift Time and Radius Visualization

Figure 5.2 and Figure 5.3 are histogram plots that show the drift time and radius distributions of the dataset.

The drift time distribution also has a small minority of data points around 10^{-17} microseconds due to missing data that causes division by 0. Those points have been removed from the histogram in Figure 5.2. The remaining data points range primarily from -10 to $12\mu s$, with several large outliers up to $300\mu s$. On the range $[0, 12]$, the distribution of data points appears to decay somewhat exponentially.

The radius distribution also drops off steeply. The nature of the drift model does not allow any negative radii, so the negative drift times are automatically cut from impacting the distribution. This distribution extends to 4.28 cm at its tail, far beyond the limits of a single drift cell. As drift cells are 2.2 cm in radius, the expected maximum value for drift radius is $2.2\sqrt{2}$, or 3.26 cm. The larger values could be explained if a particle passed through a dead cell, causing the electron shower to drift a greater distance to the nearest powered anode. In this scenario, the maximum radius to an adjacent cell would be 4.4 cm, which covers the outlier radii. This situation also accounts for the outlier drift times observed.

The shape of the drift radius distribution is quite interesting. The histogram is largest at small radii, around 0.5 cm, but has secondary and tertiary local maxima at about 0.9 and 2.5 cm. The observed drop off before 3 cm is expected, since the farthest reaches of a single drift cell should be 3.26 cm, but the gap in counts at 0.8 cm is yet unexplained. It could be a physics phenomenon, but more likely, as will be discussed in the next section, it could be a consequence of the drift model.

5.3 Cell Region Analysis

In her drift model, Landells fit different parameters to the “inner” and “outer” portions of drift cells. She chose cell type-dependent radii around 2 cm as the cut-off points, which translated to various threshold drift times. Figure 5.4 plots the division of inner and outer cell hits within the dataset. This graph clearly shows an overlap in drift times between the categories, resulting from the varying threshold times depending on cell type. For instance, the threshold for a center cell is $2.97\mu s$, while the threshold for an edge cell is $4.15\mu s$.

In contrast, the plot of radii for inner and outer cell hits in Figure 5.5 shows no visible overlap. The division between inner and outer hits occurs exactly at the mysterious 0.8 cm dip, indicating a correlation between the sharp decrease and a transition between the inner and outer cell regions. Interestingly, the time threshold categorized the range $[0.8, 4.5]$ cm as ‘outer’ despite the radius threshold for all cell types being around 2 cm. The drift model conversions from radius to time and vice versa do not match up for the threshold radii and drift times.

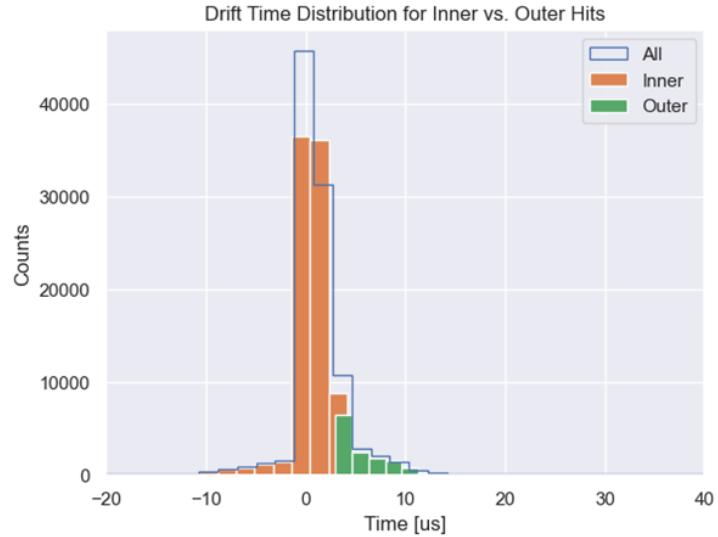


Figure 5.4: *Inner/Outer Region Drift Time Comparison* — Division of inner and outer cell hits within the full drift time distribution.

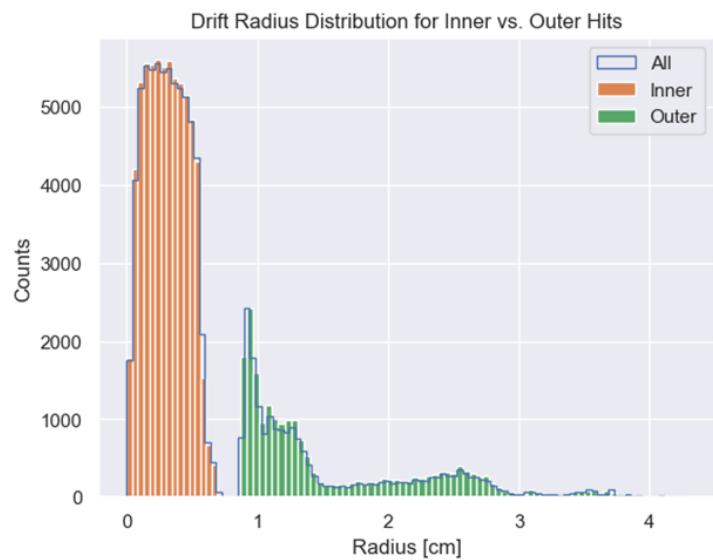


Figure 5.5: *Inner/Outer Region Radius Comparison* — Division of inner and outer cell hits within the full drift radius distribution.

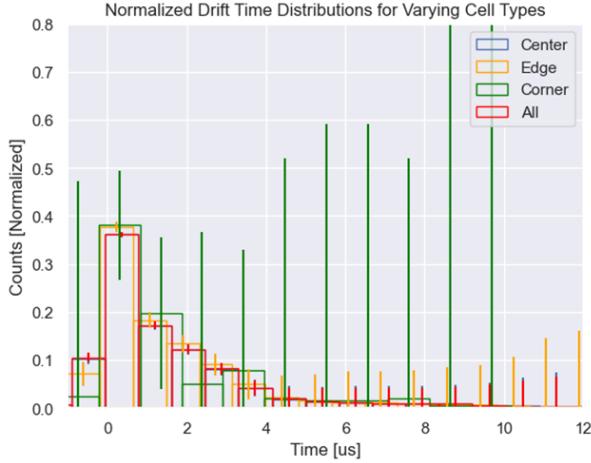


Figure 5.6: *Cell Type Drift Time Comparison* — Comparison of drift time distributions for center cells, edge cells, corner cells, and the full dataset.

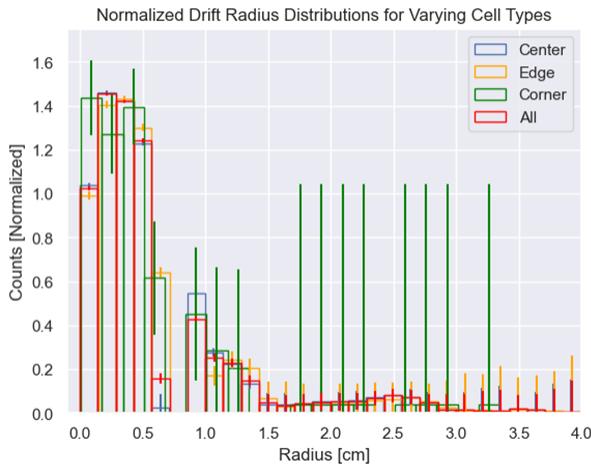


Figure 5.7: *Cell Type Radius Comparison* — Comparison of radius distributions for center cells, edge cells, corner cells, and the full dataset.

5.4 Cell Type Analysis

Beyond looking at the full time and radius distributions alone, I considered how cell type impacted distribution shape. Figure 5.6 compares the drift time distributions of center, edge, and corner cell types to the distribution for all cell types combined. These histograms are normalized, with error bars determined by $\frac{\sqrt{counts}}{counts}$ for each bin. Corner cells noticeably have the largest error bars, likely due to the scarcity of data in corner cells. Figure 5.7 shows the same comparison, but for radii. In both figures, all the cell types are consistent, with the largest deviations occurring for corner cells. More data would need to be considered to discuss whether these deviations are due to actual differences between cell types or a simple lack of data. Regarding these comparisons, my implementation of the corner, edge, and center parameters was across the full cell, rather than in specific quadrants as it is impossible to determine which quadrant of a cell the

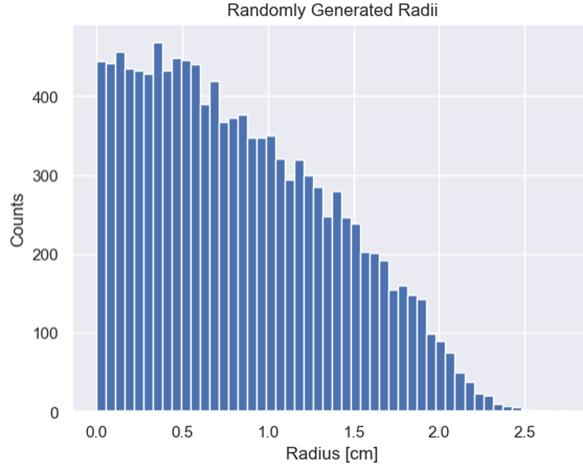


Figure 5.8: *Randomly Generated Radius Distribution* — Distribution of radii calculated by randomly generating tracks through a 4.4x4.4 cm² area.

particle passed through without preliminary track fitting. More precise representations of cell type distributions would take quadrant into account.

5.5 Real vs. Generated Radius Data

With such a strange radius distribution, I wanted to test how the distribution would look if the radii were calculated from randomly generated particle tracks.

To construct the dataset, I randomly generated two sets of (x, y) data points, (x_1, y_1) and (x_2, y_2) , 12,000 times where both x and y were within the range $[-2.2, 2.2]$ so they would always be within the boundaries of a single tracker cell. With two data points, it is possible to construct a line using the point-slope equation:

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) \quad (5.1)$$

Therefore, two randomly generated data points define a unique particle trajectory that passes through an arbitrary drift cell. To find the drift radius from the line, I needed to find the minimum distance from a point on the line to the center of the cell — essentially, I had to minimize r , where $r = \sqrt{x^2 + y^2}$ for all (x, y) pairs described by Equation 5.1.

This minimization was done using CVXPY, an open-source python library for convex optimization. CVXPY takes input parameters, variables, and constraints, then converts the specified problem to standard (“DCP”) form which is solvable by convex optimization solvers, calls the solver, and delivers the optimal results [25].

For each particle path, I defined four variables x_1, y_1, x_2 , and y_2 and two parameters $m = \frac{y_2 - y_1}{x_2 - x_1}$ and $b = y_1 - \frac{y_2 - y_1}{x_2 - x_1}x_1$. I defined $R^2 = x^2 + y^2$ and set the constraint that x and y must lie on the line $y = mx + b$. I then programmed the function to minimize R^2 subject to the constraint and return the (x, y) pair at which R^2 is closest to the origin.

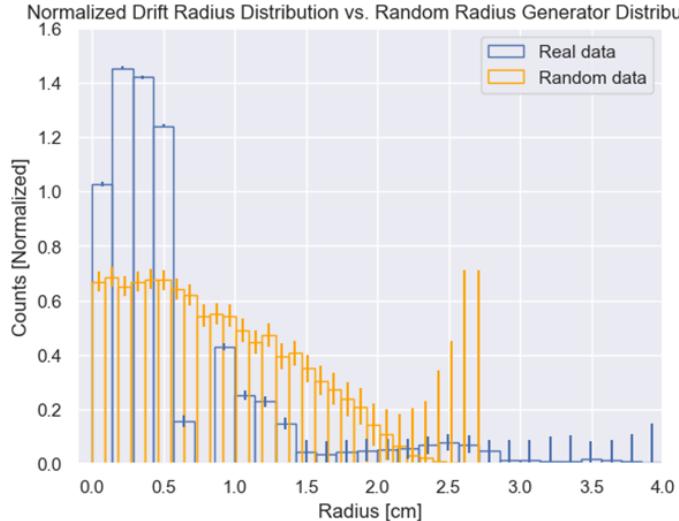


Figure 5.9: *Real vs. Generated Radius Comparison* — Distributions of real and randomly generated radii, including error bars based on normalization.

Figure 5.8 shows the distribution of the radii from these randomly generated tracks, and Figure 5.9 plots the random data against the radii calculated from the calibration dataset. The generated data looks like half of a Gaussian distribution, with a mean at 0 cm and tail extending to 2.75 cm. Apart from an overall decreasing trend, the real and generated radii behave very differently, with the dip at 0.8 cm completely absent from the generated data.

5.6 Discussion and Future Work

Work on the radial drift model is only beginning. I looked at the impact of using Landells' drift model on real calibration data, creating new tools for python analysis of SuperNEMO data in the process. I considered the effects of cell type and location within the cell and compared the resulting radial distribution to a randomly generated version.

More analysis needs to be done with this data to reach conclusions on the meaning of the unexpected dip at 0.8 cm in the radial distribution and whether it is a result of a particular drift model. This could be done by calculating new radii from the same dataset using a different drift model.

Additionally, the cause of outlier radii could be determined by looking at the locations of dead cells in the tracker in relation to the cells that produce outlier radii. My class `AnalyzeDrift` has a dictionary of all known dead cell locations, and the next step in drift model implementation would be to locate neighboring cells that would be impacted by them and tag the neighbors as edge cells.

Drift model implementation could also be improved by incorporating track fitting as an early step. Knowing the quadrant of the cells through which the particle trajectory passes would allow for a more precise application of the drift cell parameters.

Chapter 6

Beyond Analysis

My time as a member of SuperNEMO was not limited to analysis work alone. Over the course of the summer, I contributed to the communications group, updated the SuperNEMO website, acted as Artistic Director for the July collaboration meeting, and began putting together documentation for a new member starter pack.

6.1 Artistic Director

In advance of the summer collaboration meeting, I was given the title of Artistic Director. In this position, I designed a collection of stickers for SuperNEMO members which are shown in Appendix C and created activity sheets on areas of Edinburgh like Arthur's Seat and the Royal Mile for collaboration members unfamiliar with the city. All artwork on the stickers and activity sheets barring the SuperNEMO logo was original work.

6.2 SuperNEMO.org

After the collaboration meeting in July, I was trained on how to edit the SuperNEMO website, supernemo.org. The website is built using Jekyll software through GitLab and requires a Mac or Linux operating system to visualize changes before they are enacted. As a Windows user, I had to find a workaround, which is detailed in the tech support section in Appendix B.

As webmaster, I predominantly wrote articles for the news page. For instance, I wrote a news article on the summer collaboration meeting detailing presentation content and information on the guest speaker. I also updated the publication list as a new manuscript on NEMO-3 was recently completed.

6.3 SuperNEMO Starter Pack

My struggle with technical difficulties stemming from my Windows operating system and my work on Python analysis put me in a position to be able to help future SuperNEMO members avoid pitfalls and perform analysis with greater ease. As such, in the coming weeks I will be contributing to a SuperNEMO ‘Starter Pack’ proposed by Miroslav Macko and Cheryl Patrick with documentation on getting started with SuperNEMO software, using that software on a Windows computer, and performing analysis with the programs I wrote.

Chapter 7

Conclusion

The Standard Model, while extensive in its abilities to describe the natural world, is an incomplete representation of reality. New experiments aim to investigate physics Beyond the Standard Model to extend our understanding of the universe.

SuperNEMO specifically searches for the elusive neutrinoless double beta decay ($0\nu\beta\beta$) to investigate the nature of the neutrino, consider the issue of matter-antimatter asymmetry, and achieve new levels of neutrino detector precision.

The demonstrator module's tracker, with its grid of drift cells, measures particle position in three dimensions using anode and cathode signal measurements that are taken when triggered by a calorimeter hit. The complexity of the electric field within the tracker and the physics of the drift cells necessitate a detailed drift model to convert drift times to radii.

In her previous work on radial drift modelling, Betsy Landells created a drift model with varying parameters based on cell type and location of the particle within the cell. In my dissertation, I implemented this model and analyzed the results using real calibration data from Run 840 in Python. During this analysis, I developed a program to convert data from the UDD ROOT format to a pandas data frame that contained all relevant data for a single tracker cell hit in each row. I also designed a python class to automate drift time and drift radius calculations to easily implement the drift model.

The analysis itself looked at distributions of time and radius, investigating the effects of cell type and particle location. I also compared the radius distribution of the full dataset to a simulated version, noting significant differences between the two.

Many open questions on the drift model and many opportunities for improvement remain. Some immediate changes to make include tagging dead cells using the dictionary in the AnalyzeDrift class and categorizing cells that neighbor dead cells as ‘edge’ cells. Investigation into the causes of outlier drift times, which may be related to dead cells, and potential physics or model effects on drift radius is also needed to better understand the radius distribution shape, particularly the multiple local maxima observed in the real data distribution. Another action item for future work would be to compare radius calculations across multiple drift models and datasets to better understand which model works best for the SuperNEMO data.

Furthermore, more precise model implementation could be achieved by combining drift modelling with track fitting. Tomáš Křižák has developed a Legendre transform-based method to fit linear tracks in the SuperNEMO detector. Incorporating advance track fitting to identify particle location on a quadrant level rather than a cell level would enable a better implementation method of Landells' drift model, likely yielding more accurate results.

Appendix A

Code Repository

While working on this dissertation, I retrieved data, made simulations, and performed analysis with C++, ROOT, and Python. All of my original code is stored in the repository here: https://github.com/claireberschauer/MSc_Dissertation.

My programs for ROOT to pandas data conversion and the `AnalyzeDrift` class can be found in the `AnalyzeDrift.ipynb` notebook, though I plan to convert the class to a module. Consult the README in the github repository for updated information on the files.

A.1 ROOT to Pandas

My function to convert ROOT file data to pandas data frame format is called `process_data`. It takes in at minimum a UDD ROOT file and has several keyword arguments for more detailed features like the number of events to convert or whether an index column should be specified. The function utilizes a single for loop to create lists of anode timestamps, calorimeter timestamps, cell location information, event numbers, and more. After the loop is completed, the program creates a data frame where each column contains one of those lists and saves it as a .csv file so you don't have to re-run the function.

This function was a labor of love, as it took me about a week to get it to its current form. The initial version had nested for loops and took over 3 hours to run 1000 events. Now `process_data` can complete a full dataset of over 9000 events in that same time frame. It still has a long run time but is a huge improvement over its first form.

A.2 AnalyzeDrift

The `AnalyzeDrift` class is intended to simplify the transition from raw data to analysis. When called, the class requires an input data frame, preferably one outputted from `process_data` since it uses certain column names, and performs several actions. The `__init__` function of the class creates a dictionary of Betsy's model's parameters, a list of dead cells, and lists of center, edge, and corner cells. It then loads the input data frame, calcu-

lates and concatenates a list of drift times with the function `calc_drift_time()`, identifies and concatenates cell type labels with `define_io(t_drift, region)` and `find_region()`, and finally calculates and makes a column of drift radii for each individual tracker hit. This results in a data frame, the `drift_df`, containing all of that information.

I hope to improve the functionality of the class by adding new plotting functions and defining dead cell neighbors in the future.

Appendix B

Tech Support

I encountered many, many technical difficulties this summer. Most had to do with the incompatibility of my laptop's Windows processing system with SuperNEMO software, since I am apparently the only Windows user in the collaboration. To prevent future fellow Windows users from battling technology like I have, I've prepared a document of all the workarounds I used to run software, access data, and edit websites.

I call it "SuperNEMO on a Windows PC" and the most recent version is uploaded to my github here: https://github.com/claireberschauer/MSc_Dissertation. I hope this saves at least one person some pain.

Appendix C

Artwork

I was Artistic Director for the July 2023 SuperNEMO collaboration meeting in Edinburgh. For this event, I designed several stickers for attendees. These designs are shown in this section. I also made several activity sheets for locations around the city and included original artwork. These activity sheets can be accessed in the repository at: https://github.com/claireberschauer/MSc_Dissertation



Figure C.1: Rainbow SuperNEMO logo sticker.

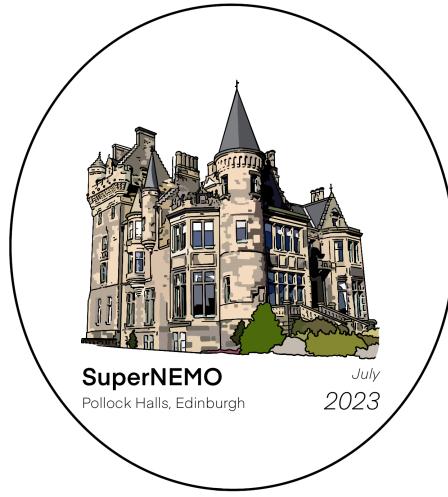


Figure C.2: Pollock Halls collaboration meeting sticker.



Figure C.3: Whisky distillery sticker for Edinburgh whisky tour.



Figure C.4: Whisky glass sticker for whisky tasting in Edinburgh.



Figure C.5: "We are speed" SuperNEMO snail sticker.

Bibliography

- [1] M. Thomson. *Modern Particle Physics*. Cambridge University Press, 2013.
- [2] R. Arnold *et al.* Probing new physics models of neutrinoless double beta decay with SuperNEMO. *The European Physical Journal C*, 70(4):927–943, Nov. 2010.
- [3] LSM. Laboratoire Souterrain de Modane. <http://www.lsm.in2p3.fr/laboratoire/presentation/lab01.htm>.
- [4] Betsy Landells. Development of the electron drift model for the SuperNEMO experiment, 2022.
- [5] Tomáš Křižák. Development of an algorithm for linear particle track reconstruction in SuperNEMO detector, 2023.
- [6] SuperNEMO Collaboration. SuperNEMO Winter Collaboration Meeting, Jan. 2023.
- [7] CERN. The Standard Model. <https://home.cern/science/physics/standard-model> [Accessed: 4 August 2023], 2022.
- [8] SuperNEMO Collaboration. The SuperNEMO Experiment. <https://supernemo.org/about.html>, 2017.
- [9] SuperNEMO Collaboration. SuperNEMO Summer Collaboration Meeting, Jul. 2023.
- [10] Tommaso Dorigo. Hadron collider searches for diboson resonances. *Progress in Particle and Nuclear Physics*, 100:211–261, 2018.
- [11] Laurent Canetti, Marco Drewes, and Mikhail Shaposhnikov. Matter and antimatter in the universe. *New Journal of Physics*, 14, 2012.
- [12] Sanford Underground Research Facility. Davis Experiment. <https://sanfordlab.org/experiment/davis-experiment>, 2020.
- [13] Bruce Cleveland, Timothy Daily, and Raymond Davis. Measurement of the Solar Electron Neutrino Flux with the Homestake Chlorine Detector. *The Astrophysical Journal*, 496:505–526, 1998.
- [14] Vadim Bednyakov. Pontecorvo and neutrino physics. *CERN Courier*, 53:78, 2013.
- [15] Steve Boyd. Neutrino Oscillations, 2016.

- [16] Super Kamiokande. Overview. <https://www-sk.icrr.u-tokyo.ac.jp/en/sk/about/outline/>.
- [17] S. Fukuda and Y. Fukuda *et al.* Determination of solar neutrino oscillation parameters using 1496 days of Super-Kamiokande-I data. *Physics Letters B*, 539(3-4):179–187, Jul. 2002.
- [18] SNO. The SNO Detector. <https://sno.phy.queensu.ca/sno2.html>, 2002.
- [19] B. J. P. Jones. The Physics of Neutrinoless Double Beta Decay: A Primer, 2022.
- [20] M. Fukugita and T. Yanagida. Baryogenesis without grand unification. *Physics Letters B*, 174(1):45–47, 1986.
- [21] H. V. Klapdor-Kleingrothaus, A. Dietz, H. L. Harney, and I. V. Krivosheina. Evidence for Neutrinoless Double Beta Decay. *Modern Physics Letters A*, 16, 2001.
- [22] S. Abe and S. Asami *et al.* Search for the majorana nature of neutrinos in the inverted mass ordering region with KamLAND-zen. *Physical Review Letters*, 130(5), jan 2023.
- [23] SuperNEMO Collaboration. Falaise 4.0.1: SuperNEMO Software Toolkit. <https://supernemo.org/Falaise/index.html>.
- [24] Jim Pivarski and Henry Schreiner *et al.* Uproot, Sep. 2017.
- [25] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.