New Analysis Developments for Reconstruction and Classification of CRES Signals in Project 8

Evan M. Zayas

S.B., Massachusetts Institute of Technology (2015)

Submitted to the Department of Physics in partial fulfillment of the requirements for the degree of Master of Science at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2019

Signature redacted

Author ....
Department of Physics
May 21, 2019

Signature redacted

Certified by ....
Joseph A. Formaggio
Professor of Physics
Thesis Supervisor

Signature redacted

Certified by ....
Peter H. Fisher
Thesis Reader

Signature redacted

Accepted by ....
Nergis Mavalvala
Associate Department Head

New Analysis Developments for Reconstruction and Classification of CRES Signals in Project 8

Evan M. Zayas

Submitted to the Department of Physics on May 21, 2019 in partial fulfillment of the requirements for the degree of Master of Science

Abstract

Determination of the absolute scale of the neutrino mass is among the most elusive and unique topics in experimental particle physics. Established direct measurement methods have reached the limit of their realistic potential, and still the values and ordering of the neutrino masses remain unknown. The Project 8 experiment pioneers a new technique called Cyclotron Radiation Emission Spectroscopy (CRES) to improve upon the obtainable neutrino mass sensitivity with ultra-precise frequency analysis. Motivated by a thorough understanding of Project 8 signal characteristics, I present new analyses to further develop the CRES technique. I detail an original algorithm to reconstruct a wider variety of CRES signal types, and a scheme to classify these types using machine learning with a demonstrated accuracy of 95%. I discuss the incorporation of these and other future developments with existing techniques to work toward a CRES analysis capable of probing the tritium endpoint with eV-scale precision or better. Lastly, I present preliminary studies of alternatives to the Fourier Transform for CRES signal processing and discuss the current limitations based on data as well as the future potential.

Thesis Supervisor: Joseph A. Formaggio
Title: Professor of Physics

Contents

List of Figures

List of Tables

Chapter 1

The Big Picture

For decades since the mid-20^{th} century, nuclear and particle physicists have developed and studied countless aspects of the Standard Model. It has been the collective goal of an entire generation of particle physicists across a wide range of topics, in both theory and experiment, to investigate the numerous predictions of the Standard Model and the questions it leaves unanswered. The neutrino sector provides some of the most prominent outstanding questions, including:

1. What are the precise values of the mixing parameters which govern neutrino oscillation? Is the CP-violating phase δ_{CP} nonzero?
2. Are the neutrino and the antineutrino distinct particles or the same particle with opposite chirality?
3. What is the absolute mass scale of the neutrino? What is the ordering of the known mass states?

Item (3) above is the grand-scale motivation for the work presented here. Historically, the Minimal Standard Model (MSM) [1] predicts all neutrino flavors to be massless. However, as the first neutrino detectors began to study neutrinos produced by nuclear fusion processes in the Sun, scientists quickly observed a discrepancy between the observed flux of solar neutrinos and that expected from the luminosity of the Sun. This discrepancy is known as the solar neutrino problem, and was first observed by Ray Davis and John Bahcall at the Homestake mine in the 1960s [2]. The solar neutrino problem provided the first evidence that the MSM description of the neutrino may be incomplete, and remained an open question for years after its initial discovery. Figure 1.1 shows several measurements from the Homestake experiment during the 1970s–1990s which clearly illustrate the consistent flux deficit.

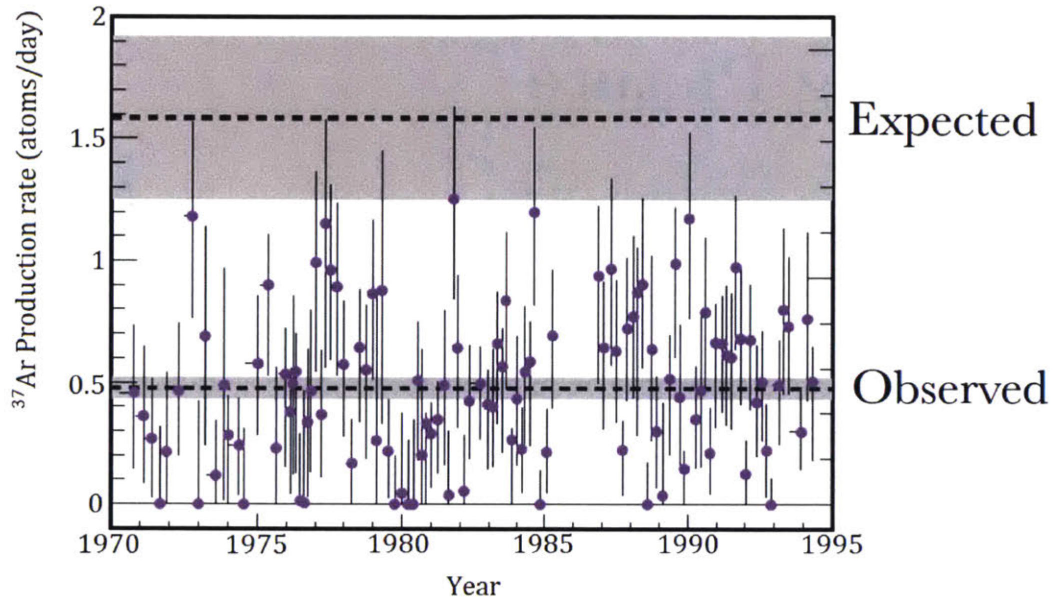


Figure 1.1: Results of the Homestake experiment over a period of 25 years, showing a consistent neutrino flux which is approximately 30% of that predicted from solar nuclear theory. Adapted from [3].

The introduction of nonzero neutrino mass provides a resolution to the problem. In the MSM, massless neutrinos are known to exist in three flavor states corresponding to the three lepton generations: electron, muon, and tau neutrinos. If the neutrino is instead massive, then the mass eigenstates are in general distinct from the flavor eigenstates and they (the mass states) must propagate as plane waves in time. Thus, propagation of a flavor state is not described by a simple plane wave, but instead with a time-varying amplitude which allows a neutrino initially of one flavor to be observed in a different flavor state at a later time. This is the well-known mechanism of neutrino oscillation. Since the Homestake detector was sensitive only to the electron neutrino flavor, this oscillation provides an explanation for the reduced flux: neutrinos are created in the Sun in a pure electron state, but arrive at the Earth in a mixed state with a reduced probability (~30%) to interact with the detector. While this theory of massive neutrinos and oscillations was well established even before the Homestake experiment, firm evidence for it eluded scientists until the turn of the century. In 1998 and 2001, the Super-Kamiokande and Sudbury Neutrino Observatory (SNO) experiments [4, 5] provided the first measurements of the neutrino oscillation parameters, and thus the first compelling evidence for a nonzero neutrino mass. The current global fits to these parameters are listed in Table 1.1. Since the oscillation depends on the square of the mass differences, these measurements do not provide information about the absolute scale of the neutrino masses m_1, m_2, m_3 or the sign of the splittings Δm_{12}, Δm_{13}, and Δm_{23}. Further questions about these mixing parameters, the nature of neutrinos, the origin of neutrino mass, and the scale of the individual masses remain unanswered and have since dominated discussion among 21^{st} century neutrino physicists.

Parameter	**Best-fit Value**	**3σ C.I.**
Δm^2_{12}	7.37×10^{-5} eV^2	$6.93-7.96 \times 10^{-5}$ eV^2
Δm^2_{13}, N.H.	2.56×10^{-3} eV^2	$2.45-2.69 \times 10^{-3}$ eV^2
Δm^2_{13}, I.H.	2.54×10^{-3} eV^2	$2.42-2.66 \times 10^{-3}$ eV^2
$\sin^2\theta_{12}$	0.297	0.250−0.354
$\sin^2\theta_{13}$	0.0215	0.0190−0.0240
$\sin^2\theta_{23}$	0.425	0.381−0.615
δ_{CP}/π, N.H.	1.38	2σ: 1.0−1.9
δ_{CP}/π, I.H.	1.31	2σ: 0.92−1.88

Table 1.1: Best-fit and 3σ confidence intervals for neutrino mixing parameters. The confidence interval of the CP-violating phase δ_{CP} is given to 2σ. N.H. (I.H.) corresponds to the normal (inverted) mass hierarchy. Adapted from [6], from results quoted in [7].

1.1 Direct Mass Measurement Approaches

A direct measurement effort capable of probing the absolute neutrino mass scale is one of the most sought-after prizes in neutrino physics. Currently, the most stringent bounds on the effective mass[1] of the electron antineutrino are obtained from the Mainz [8] and Troitsk [9] experiments:

$$m_{\bar{\nu}_e} < 2.05 \text{ eV} \tag{1.1}$$

where the above bound represents a 95% confidence limit from the Troitsk experiment as quoted by [6]. The most common methodology of direct measurement experiments involves electron calorimetry with tritium (^{3}H) β-decay:

$$^3\text{H} \;\rightarrow\; ^3\text{He} + e^- + \bar{\nu}_e \tag{1.2}$$

The neutron-proton mass difference is 1293.33 keV, and the binding energy difference between tritium and helium-3 is 763.76 keV; this leaves 529.57 keV to create the electron (511.00 keV) and the remaining $\sim$18.6 keV (called the Q-value of the decay) is the sum of the electron kinetic energy, neutrino kinetic energy, and the neutrino rest mass. In every current or past tritium β-decay experiment, only the electron is detected due to the weakly interacting nature of the neutrino. Direct mass experiments study the endpoint of the electron spectrum, where the maximum kinetic energy of the electron serves as a proxy for the Q-value; if all of the available kinetic energy goes to the electron, then the (extremely small) difference between this energy and the Q-value is precisely the effective mass of the electron neutrino. With this approach, a direct measurement of the neutrino mass can be achieved through ultra-high precision spectroscopy of the tritium β-decay endpoint.

The prevailing technique for such experiments is described in [10, 11] and I will briefly summarize it here. Electrons are emitted in a region of high magnetic field and guided through a MAC-E (Magnetic Adiabatic Collimation and Electrostatic) Filter which utilizes a much lower field magnitude in the center of the spectrometer and an electric retarding potential to select

---

[1] The effective mass of a flavor state is simply a weighted average of the mass eigenvalues: $m^2_{\nu_e} = \sum_j |U_{ej}|^2 \, m^2_{\nu_j}$ for the electron flavor where U_{ej} are the PMNS matrix elements.

electrons near the endpoint of the spectrum. Those electrons which surpass the retarding potential are then re-accelerated toward a calorimeter where the field is the same as at the location of the source. The energy of the electron when it reaches the calorimeter is very nearly the same as when it was emitted, and the uncertainty is dominated by the ratio of the magnetic fields in the spectrometer:

$$\frac{\Delta E}{E} = \frac{B_{\mathrm{min}}}{B_{\mathrm{max}}} \tag{1.3}$$

where B_{min} is the magnetic field magnitude at the center of the spectrometer and B_{max} is the magnitude at either end (the source and the calorimeter). From the conservation of magnetic flux, it is then immediately clear that this ratio is in turn limited by the physical size of the spectrometer, as the cross-sectional area at the center must scale inversely with the energy resolution. The Mainz and Troitsk experiments achieved a resolution of $\Delta E \approx 4.8$ eV corresponding to the neutrino mass bound quoted earlier (Equation 1.1). KATRIN [10] is a current experiment using the same technique with a spectrometer measuring 10 meters in diameter – the largest ultra-high-vacuum chamber in the world – an anticipated resolution of $\Delta E \approx 0.9$ eV and a corresponding neutrino mass bound of $m_{\bar{\nu}_e} < 0.2$ eV. The sensitivity at this point is limited primarily by rotational and vibrational states of the molecules. Thus, it is a reasonable assessment that KATRIN will utilize the MAC-E Filter technique with molecular tritium to its maximum potential.

1.2 A Novel Technique to Improve Precision

To improve upon the anticipated sensitivity of KATRIN, probe the inverted hierarchy region, and ultimately probe the normal hierarchy region to guarantee a direct mass measurement, a new methodology is needed. This is the motivation for the Project 8 experiment, which has pioneered such a technique and aims to achieve a sensitivity of 0.04 eV in its final phase of operations: a factor of 50 better than the present limit.

Project 8 measures the energy of an electron by subjecting it to a uniform magnetic field and detecting the resultant cyclotron radiation. This technique, called Cyclotron Radiation Emission Spectroscopy (CRES), has the potential to provide the necessary precision for an improved neutrino mass limit or measurement [12]. Project 8 is the first experiment to develop and utilize this technique, and the first to detect cyclotron radiation from individual electrons. The frequency of the cyclotron radiation is given by:

$$f = \frac{eB}{2\pi(m_e + K)} \tag{1.4}$$

where e is the electron charge magnitude, B is the magnetic field magnitude, m_e is the electron mass and K is its kinetic energy. Equation 1.4 provides an explicit relationship between the electron energy and radiation frequency, and the only appreciable source of uncertainty is the magnetic field magnitude. For the tritium endpoint energy and a field of 1 Tesla, the cyclotron frequency is $f \approx 27$ GHz; thus, Project 8 in effect turns a precision spectroscopy experiment into a radio frequency measurement. If the systematics are well-understood, such a frequency measurement has the ability to achieve very high precision provided the electron is detected for a sufficient amount of time. A CRES experiment such as Project 8 can ultimately be limited only by the magnetic field uniformity; in practice, there are numerous systematics pertaining to the electron dynamics, magnetic field configuration, and signal reconstruction which limit the sensitivity to a much greater extent. It is the overarching goal of Project 8 analysis teams to thoroughly understand these systematics, develop the CRES technique, and help the experiment progress toward a competitive neutrino mass limit and eventually a first-ever direct measurement.

In this work, I will discuss several recent efforts to better understand and advance the CRES technique. In Chapter 2, I present an overview of the Project 8 long-term plan, current apparatus, and introduce important signal characteristics. In Chapter 3, I discuss the analysis framework, signal reconstruction procedure, and related future strategies. Chapters 4–6 describe new advancements toward a sophisticated CRES analysis and the impacts on energy resolution.

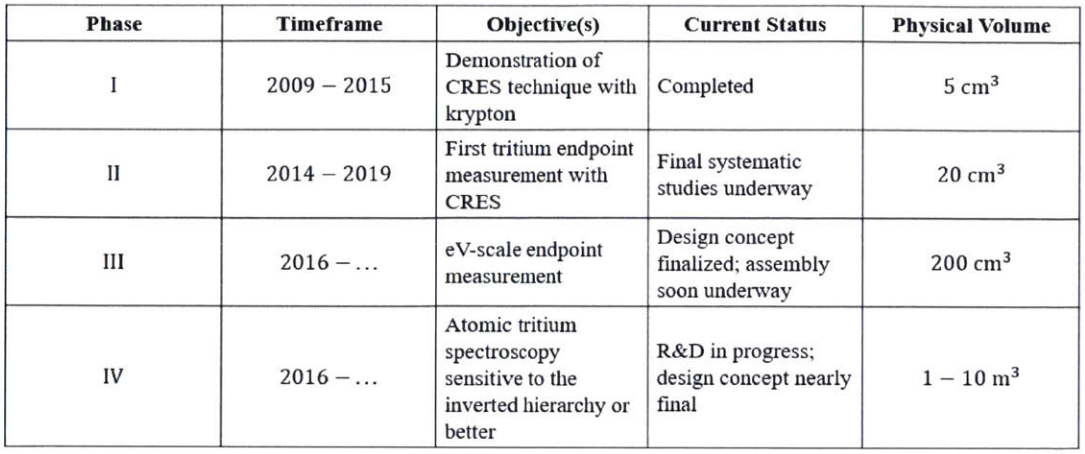
Chapter 2

Overview of the Project 8 Experiment

First conceived in 2008, the Project 8 experiment has steadily worked towards the goal of a neutrino mass measurement in a four-phased approach. Enumerated in Table 2.1, the phases can be briefly summarized as: (I) CRES demonstration, (II) first tritium spectroscopy, (III) eV-level sensitivity competitive with current limits, and (IV) atomic tritium spectrum with inverted hierarchy sensitivity or better. Phase I is fully completed, and Phase II is scheduled to conclude in late 2019. Both Phases III and IV will involve a considerable scaling up of the physical size of the apparatus.

As is evident from Table 2.1, there is significant overlap in the timeframes of many phases of the experiment. Project 8 owes much of its success so far to this parallel approach, as there are substantial research and development needs for every phase. Utilizing a wide range of collaborator expertise, we have so far made continuous progress on all phases of the project while accomplishing the objectives of Phases I and II.

The focus of this work is entirely limited to Phases I and II, the past and current generations of the experiment. The first phase, which reconstructed the first ever CRES signal in 2014 and concluded in 2015, used $^{83\mathrm{m}}$Kr gas as the electron source rather than a radioactive tritium source. These electrons are not produced via β-decay, but instead are conversion electrons produced in the electronic decays of a metastable krypton state. The krypton itself is



Phase	Timeframe	Objective(s)	Current Status	Physical Volume
I	2009 – 2015	Demonstration of CRES technique with krypton	Completed	5 cm^3
II	2014 – 2019	First tritium endpoint measurement with CRES	Final systematic studies underway	20 cm^3
III	2016 – …	eV-scale endpoint measurement	Design concept finalized; assembly soon underway	200 cm^3
IV	2016 – …	Atomic tritium spectroscopy sensitive to the inverted hierarchy or better	R&D in progress; design concept nearly final	1 – 10 m^3

Table 2.1: Description of primary objectives and current status of Project 8 phases. The timeframes quoted include concentrated R&D work as well as data campaigns. A (very approximate) fiducial volume is included with each phase to illustrate the extent to which Project 8 will expand the physical size of its detector.

produced from rubidium electron capture:

$$^{83}\mathrm{Rb} + e^- \;\rightarrow\; ^{83\mathrm{m}}\mathrm{Kr} + \nu_e \tag{2.1}$$

and pumped to a rectangular waveguide cell where it emits the conversion electrons. There are several transitions from this metastable krypton state with energies of the same order of magnitude as the tritium endpoint, listed in Table 2.2. Each transition is monoenergetic, up to a natural linewidth of order 1 eV which is substantially less than the energy resolution of Phase I. We will often refer to these emission lines simply by the groups listed in the first column; 17 keV, 30 keV, or 32 keV.

The waveguide is used for the cell to contain the source gas as well as to transport the emitted radiation toward the cryogenic receiver chain. The cyclotron radiation couples strongly to the fundamental (TE10) mode of the Phase I rectangular waveguide, and as a result a large fraction of the radiated power is detected. Because the source gas is transparent to cyclotron radiation, this measurement takes place with the electron still within the source volume and is nondestructive – an advantage over traditional methods that require

Group	**Transition Level**	**Line Energy (keV)**
17 keV	$1s_{1/2}$	17.83
30 keV	$2p_{1/2}$	30.42
30 keV	$2p_{3/2}$	30.48
32 keV	$3p_{1/2}$	31.93
32 keV	$3p_{3/2}$	31.94
32 keV	$4p_{3/2}$	32.14

Table 2.2: Emission lines of interest for the metastable Krypton-83 state, from [13].

extraction of an electron for measurement, limiting the viable gas density. The received signal is first mixed down with a fixed 24.2 GHz local oscillator, and then again with a variable mixing frequency within the digitizer. Together with a low-pass filter, this selects a 100 MHz frequency band of interest (the acquisition bandwidth) and the signal is digitized at 200 MS/s. For Phase II, a number of upgrades were made to this receiver chain as well as the waveguide cell and the digitizer which we will outline in Section 2.2.

2.1 Magnetic Bottle Trap

In discussing the CRES technique and the apparatus of Phases I–II, I place a particular emphasis on the magnetic trapping mechanism. An electron must persist within the waveguide cell for sufficient time to adequately detect the radiation power and measure the precise frequency. For an electron with kinetic energy 18.6 keV, the Lorentz factor is $\gamma = 1.036$ and the velocity is:

$$\beta = \sqrt{1 - \frac{1}{\gamma^2}} = 0.26 \tag{2.2}$$

Therefore, such electrons indeed have a relativistic velocity and will escape even a very large detector far too quickly for detection (recall from Table 2.1 the Project 8 krypton cells are at the centimeter scale). The electrons must be trapped if they are to be detected, and furthermore they must be trapped magnetically to ensure a proper energy measurement; any electric field in the trap would create unwanted kinetic energy smearing effects. Project 8 uses a magnetic bottle trap to contain the electrons, constructed with a series of pinch coils. Figure 2.1 illustrates the most basic Project 8 trap, in which one pinch coil with a current applied creates a small local depression in the background magnetic field. The coil is oriented along the same axis as the background field (left/right in Figure 2.1), which we define to be the $\hat{z}$ direction such that the field is well-described in only one dimension:

$$\vec{B}(\vec{r}) \approx B(z) \tag{2.3}$$

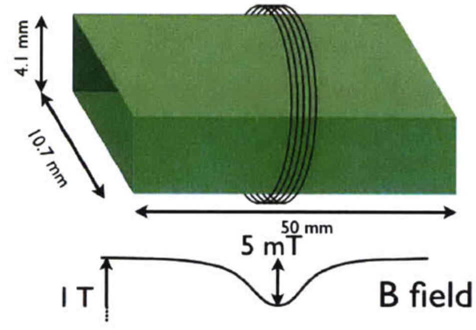


Figure 2.1: Illustration of a harmonic trap with a single pinch coil. Current in the coil creates a small local depression in the magnetic field, trapping electrons with sufficiently low axial momentum.

and we also commonly refer to $\hat{z}$ as the "axial" direction. As an electron which originates within the depression approaches a higher field near the edge of the trap, the Maxwell equations insist that the field lines converge to the center of the (x, y) plane to conserve magnetic flux. Then, the Lorentz force acquires a small component in the axial direction and pushes the electron back toward the center of the trap. This motion in z can be described quantitatively by the conservation of the adiabatic moment:

$$\mu = \frac{v_\perp^2}{2B} \tag{2.4}$$

where $v_\perp^2 + v_z^2 = v^2$, v_z is the axial velocity and $\delta\mu = 0$ in the adiabatic limit. The kinetic energy in the axial direction is $\frac{1}{2}mv_z^2$ which can be written with

Equation 2.4 as:

$$\mu B + \frac{1}{2}mv_z^2 = \frac{1}{2}mv^2 \tag{2.5}$$

and thus if the (total) kinetic energy $\frac{1}{2}mv^2$ is approximately constant over a short time, the quantity μB describes an effective potential for the axial motion. Since μ is also conserved, the field magnitude $B(z)$ alone can be interpreted as a 1D potential and the trap illustrated in Figure 2.1 describes a potential well. Figure 2.2 shows the other type of trap utilized in Project 8, where instead two coils with the current reversed create a pair of potential barriers. This type is called a "bathtub" trap, and the former type is referred to as "harmonic". Figure 2.3 shows more clearly the magnetic field profile $B(z)$ as calculated with the exact specifications of the Project 8 Phase II apparatus with the widest possible bathtub trap.

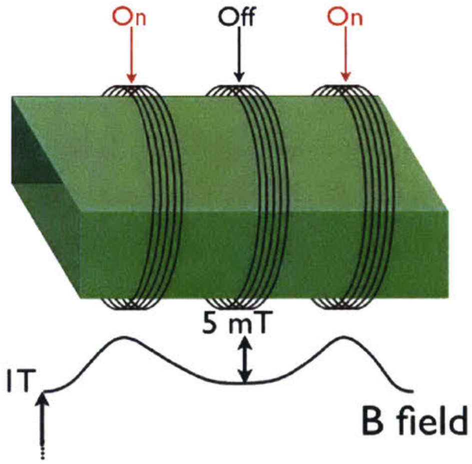


Figure 2.2: Illustration of a bathtub trap with a pair of pinch coils. Current is run through each coil with a polarity opposite that of the harmonic trap, to create a pair of high-field regions. Like the harmonic configuration, this traps electrons with sufficiently low axial momentum and has the added benefit of an elongated flat region in the center of the trap.

The magnetic bottle is also configurable by the use of many pinch coils. The Phase I apparatus consisted of three pinch coils, corresponding to 3 possible harmonic traps and 3 bathtub traps; for Phase II the apparatus was

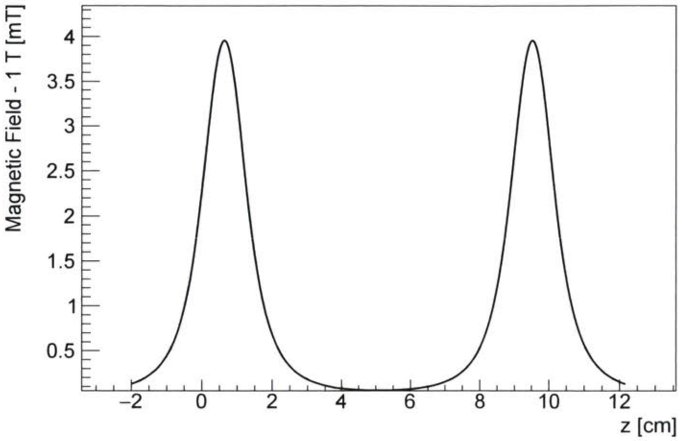


Figure 2.3: Magnetic field profile in the z direction, calculated from the Phase II cell specifications. The coils are separated in z by 8.9cm, the distance between the furthest-outside coils in Phase II. This creates a bathtub-type trap centered at $z = 0$. The background field used for this calculation is uniformly 1 Tesla.

upgraded to five coils to allow for longer bathtub traps as well as more configuration options: 5 harmonic and 10 bathtub options, plus more complicated configurations such as multiple simultaneous harmonic traps (poly-harmonic). These many trap configuration options have proven to be an extremely useful tool in studying systematics related to the trap position and geometry.

While a magnetic trap is clearly essential for any CRES experiment like Project 8, it also immediately complicates the relationship between frequency and kinetic energy given by Equation 1.4. The electron motion is no longer described completely by the cyclotron orbit, but also by axial motion within the trap. Again using the analogy of a 1D potential, a trapped electron will always explore a range of the magnetic field profile $B(z)$ which is bounded from below at the trap center and bounded from above by the "kinetic energy" in z of the electron. This leads directly to an effective smearing of the field value B in Equation 1.4, and in addition the axial motion has a significant impact on the frequency spectrum of emitted radiation. These effects are discussed in much greater detail in [14] and in Chapter 5, where we develop a new analysis to properly extract the cyclotron frequency with axial considerations.

2.2 Phase II Upgrades and Status

In addition to the two additional trap coils, the Phase II cell and receiver chain include several significant upgrades over Phase I:

- A reliable probe of the field strength using electron spin resonance (ESR). These ESR probes are installed in close proximity to the trapping coils to ensure high accuracy and reliability. ESR studies have proven very useful to learn about the shape of the background field, its stability over time, and to guide our studies of trap configurations.
- Increased cell volume by approximately a factor of 3. This is especially useful for preliminary tritium studies, as the region of interest near the endpoint has low activity, and increased cell volume allows for more source gas atoms (at equal pressure and temperature) and thus better statistics.
- The rectangular waveguide is replaced with a cylindrical waveguide. This further increases the effective trap volume and reduces the proportion of electrons which may come in contact with the cell walls.
- A circulator has been added to direct amplifier noise away from the waveguide and to a cold termination, substantially reducing the noise at the antenna.
- A new 3-channel digitizer and a separate triggering system to allow for both triggered and continuous data acquisition. The mixing frequency can be set independently for each channel.

These components and others are illustrated in Figures 2.4 and 2.5. Figure 2.4 is an annotated photo of the Phase II cell with the five trapping coils, ESR coils, and some other items labelled. Figure 2.5 shows a block diagram of the waveguide components and receiver chain, beginning with two cascaded low-noise preamplifiers. The amplifiers achieve an overall gain of 54 dB and an effective noise temperature of 145 K at a physical temperature of 50 K. The signal is then mixed down and split to the two data acquisition systems, the

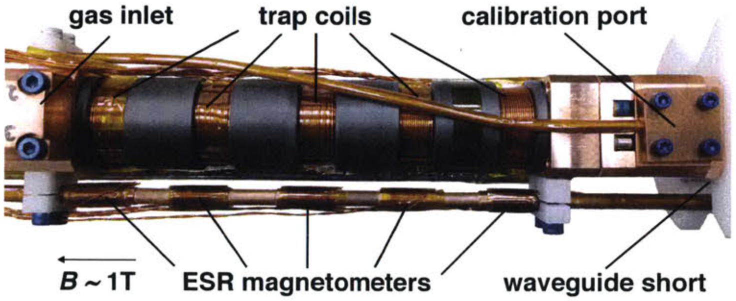


Figure 2.4: Photo of the Phase II krypton cell. The five trapping coils can be seen in the center, along with the waveguide reflector (short) on the right end of the apparatus and the gas inlet on the left. ESR coils are also labelled.

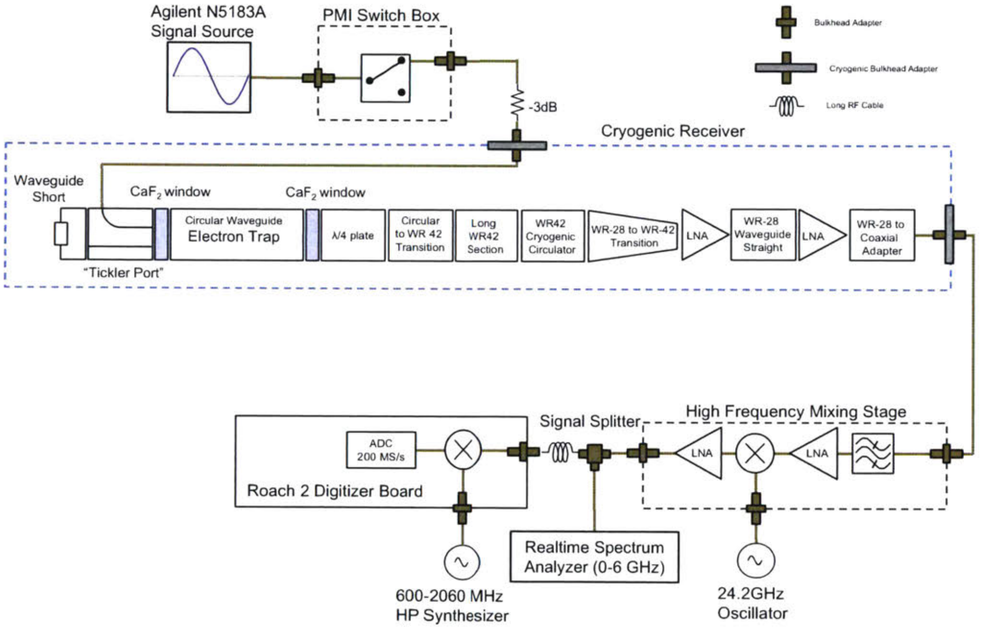


Figure 2.5: Detailed block diagram of the receiver chain for Phase II. The cell and waveguide components are shown in the center, followed by a high-frequency mixing stage and the two digitizers. A signal generator is also connected to the cell for injected-tone studies.

Realtime Spectrum Analyzer (RSA) of Phase I and the ROACH-2 of Phase II. Each digitizer performs a second-stage mixing, re-sampling, and writes the data to disk. The ROACH can be used alone to write continuous time-domain data, or in conjunction with an independent frequency-domain trigger.

For Phase II, the Project 8 group has constructed two versions of the cell: one to carry out all of the required systematic studies with krypton, and another to perform the tritium campaign. Both cells were designed with the exact same specifications to ensure maximum carryover from the results of the initial krypton studies, especially those which depend on the waveguide dimensions and positions of the trapping coils. The first cell is never tritiated, and we are able to make any necessary changes based the krypton results before introducing tritium to the system with the second cell.

The first cell also contained a reflector at one end of the waveguide, opposite the antenna (labelled 'waveguide short' in Figure 2.4); this reflects signal power back toward the antenna, which in principle boosts the SNR by a factor of 2. However, the reflected radiation introduces serious complications to the signal behavior due to an interference effect at the antenna between the incident and reflected radiation. A reflector was also present in the Phase I cell, but the noise floor was sufficiently high to mask the majority of the complex signal characteristics. In Phase II, the reduced noise and improved trap geometry illuminated these complications to an unprecedented extent and the reflector was removed in favor of a terminator for the tritium cell. Although it will not be utilized for the final Phase II results, the effects of the reflector in Phase II have enormously helped Project 8 to better understand CRES signals and model them in the detector, which is a necessary step when pioneering a new measurement technique.

Commissioning of the tritium cell for Phase II began in 2018, with first tritium data and results in Autumn of the same year. Figure 2.6 shows the results of a preliminary 7-day tritium run; this is the first tritium β-decay spectrum ever constructed with CRES. The endpoint (calculated, not fit) energy corresponds to 1369 MHz, which is qualitatively consistent with the spectrum.

No events were detected above[1] the endpoint, and the highest-energy event falls approximately 14 MHz, or 280 eV below the endpoint.[2] The full Phase II tritium campaign, which will accomplish its principal goals and conclude Phase II, is planned to begin in mid-2019 and will require 100 days of live time. The trap configuration will be similar to that of the preliminary run, but with some modifications to improve the detected event rate by up to a factor of 5 when combined with further analysis improvements.

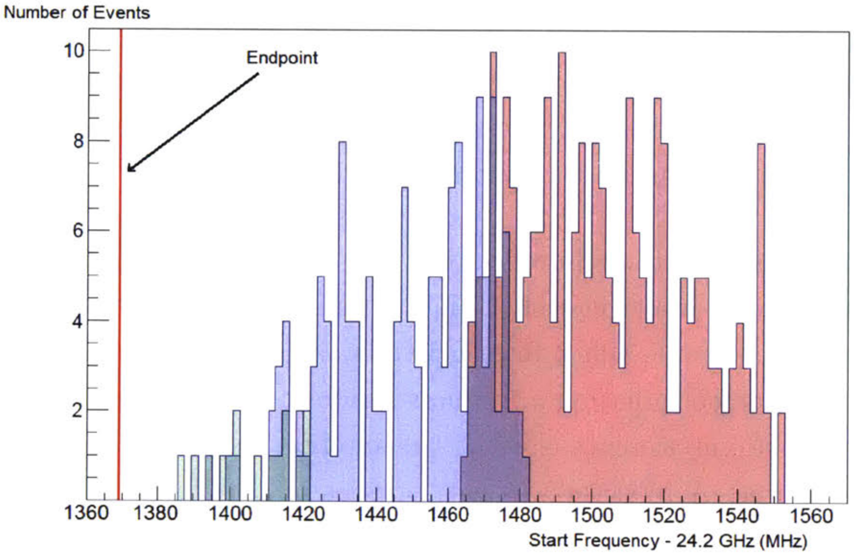


Figure 2.6: Frequency spectrum of a preliminary tritium data run in Phase II. The endpoint is known from magnetic field measurements to lie at approximately 1369 MHz. Colors correspond to three simultaneous acquisition channels which are each 85 MHz in width. Detection efficiency falls substantially in the high-frequency region, as a consequence of increased noise floor in the data acquisition; this behavior is expected, and qualitatively well-understood. At the edge of the highest-frequency channel (1550 MHz), the noise floor has overtaken virtually all of the signal power and no events are detected.

---

[1] Above in energy, below in frequency. This inverse relationship follows directly from Equation 1.4; I will often use energy and frequency interchangeably in this way.

[2] This is another rough equivalence we will use often. From Equation 1.4, a frequency interval of $\Delta f = 1$ MHz corresponds to an energy interval of $\Delta E \approx 20$ eV in a 1 Tesla field.

2.3 Future Phases

Phases III and IV are the planned future generations of the Project 8 experiment which will aim to achieve a neutrino mass sensitivity at the eV and ultimately sub-eV scale. To accomplish this, each of the final two phases will require a significant upgrade to the apparatus capabilities, far more so than the upgrades discussed in this chapter between Phases I and II. For this reason, Project 8 has worked to design the future phases in parallel with the active commissioning and data campaigns of Phases I and II. The work presented in the remaining chapters pertains entirely to Phases I and II, but first I will briefly describe the planned designs and goals of Phases III and IV.

Phase III will increase the magnetic trap volume by at least an order of magnitude over Phase II. The improved statistics will give Project 8 the capability to measure the tritium endpoint to a sensitivity of $m_{\bar{\nu}_e} < 1{-}2$ eV with approximately 1 year of data, similar to the current bound achieved by the Mainz and Troitsk experiments. With the magnetic trap scaled up in size, the use of a waveguide is no longer viable as the mode coupling with a much larger waveguide to the cyclotron radiation would be extremely weak. Thus, Phase III will move the CRES technique into free space for the first time. The signal will no longer be obtained from 1 antenna at the end of a waveguide, but with an array of at least 30 antennas positioned radially outward from the cell (see Figure 2.7). With the use of beam-forming analysis techniques which are common in radio telescope array experiments, the antenna array can have targeted sensitivity to a small region within the trap (~1% of the full volume) and achieve sufficient SNR to detect the CRES signals. The beam-forming process will also most likely demand the ability to "follow" an electron in real-time as it moves through the trap by tuning the region of sensitivity accordingly. Development of this technology for the context of Project 8 is easily among the more substantial upcoming analysis challenges before the beginning of Phase III. The physical design concept is finalized and several other considerations including the use of a much larger magnet, data acquisition/management, etc. are well understood. Project 8 expects to begin building and commissioning the Phase III detector during the 100-day final data campaign of Phase II.

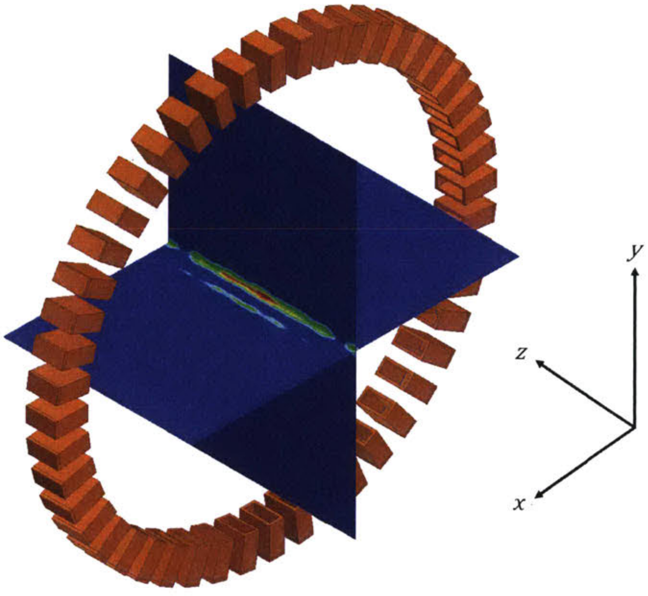


Figure 2.7: Illustration of the antenna array for Phase III with 48 elements. The sensitivity over a cross-section of the trap volume is shown in the interior, localized at the center. With digital beam forming processes, this region of sensitivity can be targeted to the location of an electron once it is detected.

Phase IV will again increase the trap volume, this time by several orders of magnitude, now to the cubic meter scale. It will utilize atomic tritium rather than molecular (diatomic) tritium to circumvent the sensitivity limit imposed by rotational and vibrational states of the molecules. This requires the design of a new magnetic trap which is capable of containing tritium atoms as well as electrons, since the affinity of hydrogen to bind with atoms on the cell wall makes recombination highly favorable. Such a trap will be the first of its kind, and with a sufficiently uniform magnetic field within the trap it will bring the sensitivity of Project 8 as low as 0.04 eV with between 3 and 5 years of data. This sensitivity by far exceeds that which is possible with molecular tritium, and has the potential to provide a direct mass measurement or rule out the inverted hierarchy. The design of this trap as well as other new components related to tritium dissociation and cooling are already well under way.

Chapter 3

Data Analysis and Signal Reconstruction

Project 8 has developed several software packages for analysis-related tasks in Phases I and II. The most extensive of these is called Katydid [15], and it is publicly available on GitHub.[1] Katydid is the principal C++ analysis package which processes raw data files from the digitizers and searches for CRES signals within them. It has been under continuous development for much of the project's history, primarily by Noah Oblath since 2012 and myself since 2015. In this chapter, I find it compelling to introduce in some considerable detail the framework in which Katydid operates and the current signal reconstruction process. This understanding provides a useful context for the new analysis presented in the remaining chapters – and in particular for the future prospects of it, which I will discuss broadly at the conclusion of this chapter.

3.1 Katydid Framework

The overarching task which Katydid performs is simple in concept: process raw time-series data produced by the digitizer, and output some collection of properties describing CRES events. To accomplish this, Katydid utilizes a modular framework that divides the analysis into individual steps similar to

---

[1] `https://github.com/project8/katydid/`

the Qt software development package.[2] The analyst then has the freedom to arrange these modules and the interactions between them to create a larger analysis. For example, one early step in the standard analysis is a Discrete Fourier Transform (DFT) which converts a segment of raw time-domain data to a frequency spectrum. In Katydid, each such analysis step is contained in an object called a processor. The processors are then linked together, most often by sending the output of one to the input of another, to form the full analysis chain. Processors are self-contained, meaning they exist outside of any particular analysis chain and can be used/tested individually.

The other main type of object in Katydid is data. Data classes hold the information which is passed between processors – for example, the DFT processor has the ability to pass a frequency spectrum to any other processor which can receive it. Like processors, the data classes are also self-contained and specific to allow for a great deal of freedom in the exchange of data objects between processors.

3.1.1 Signal/Slot mechanism and the Data Pointer

An analysis chain in Katydid is described by a collection of processors and connections between them to exchange data. As in the Qt framework, the names given in Katydid to these links in the chain are "signals" and "slots". The output of a processor is called a signal, and the input is a slot. Every processor has a specific set of slots which it can accept, and signals it can emit. In general (but not always), a slot requires one or more specific data objects to be present before it can proceed, and similarly a signal guarantees the presence of certain data object(s). An analysis chain can utilize any number of connections so long as for each one, the data object(s) required by the slot are also guaranteed by the signal. This arrangement gives the analyst a great deal of freedom to alter or construct a new analysis chain without the need to further develop the software.

The signal and slot objects themselves are void methods within a processor class, and the argument to each is a pointer to a linked list of data objects

---

[2] `https://www.qt.io`

called the data pointer. The data pointer allows every signal/slot connection to always pass the same type of object, and the linked list enables the transfer of multiple data objects between processors. A signal is emitted by sending it (the method) a data pointer, and a connection in turn calls the associated slot method with the same data pointer. The slot function then usually begins by checking the data pointer for all required data objects.

In some cases, a slot method simply appends a new or altered data object to the data pointer it received, and emits a specific signal; for example, the time series slot in the DFT processor always emits a signal containing the associated frequency spectrum. For these "linear" steps in the analysis chain, processors usually append the new data object to the data pointer, preserving a sort of processing history. Slot functions down the chain are then able to access data at each previous step from the single data pointer. Figure 3.1 at the end of this section illustrates a simple analysis chain which includes this behavior.

3.1.2 Configurable parameters

In addition to a collection of signals and slots, each processor also contains a set of member variables that control the behavior of the processor, and have public access and assignment methods. These variables can be set at runtime, and they are most often free parameters of the processor analysis which the analyst may vary for optimization studies. Other configurable parameters include input/output file names and boolean flags or options to select one out of a few similar processor functions. Not all member variables need be specified at runtime, as they are always given a default value in the constructor of the processor. However, the Katydid convention is to most often assign default values that are functional but not sensible. For example, many processors contain member variables for the minimum and maximum frequency range in which to operate, but they are initialized to 0 and 1 Hz by default (the full bandwidth for Project 8 data is 100 MHz). This convention makes obvious any member variable assignments which may be missing while still avoiding a crash.

3.1.3 Configuring the analysis chain

To execute a Katydid analysis, the user constructs the analysis chain with a configuration file. Every configuration file can be summarized in four sections, which are listed below and described with the aid of a simple example diagrammed in Figure 3.1:

1. Initialize all required processor instances. This first section initializes each processor needed for the analysis with a `type` to specify the processor class and a `name` to uniquely identify the instance.

2. List signal and slot connections. Each connection specifies a signal and a slot of the form `proc-instance-name:signal-or-slot-name`.

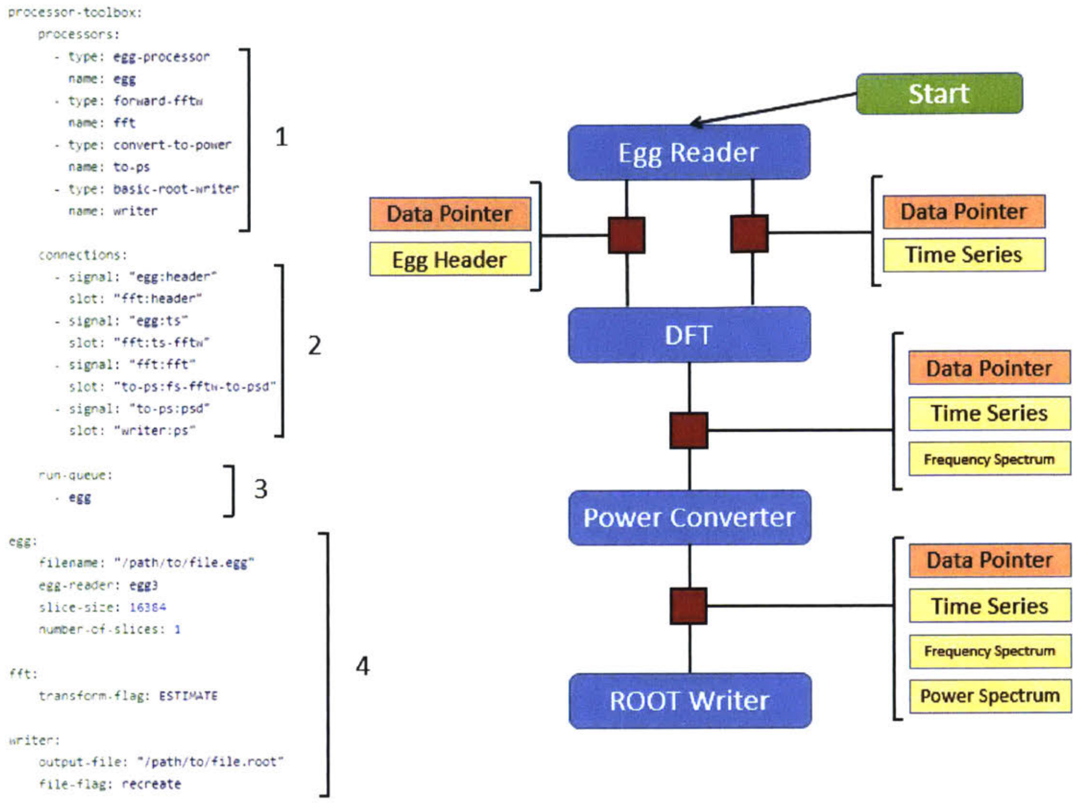
3. Specify the starting point for the analysis. This is almost always the "egg" processor, which reads in a raw data file.

4. Configure parameters for each processor as necessary. Each configurable variable is assigned with an entry of the form `variable-name:  value`.

The example shown in Figure 3.1 executes a standard Fourier spectrum analysis, with the DFT followed by the conversion to a real power spectrum. The power spectrum is then connected to an output processor which writes the contents to ROOT file and emits no signal, terminating the analysis chain.

3.2 Primary Track Reconstruction

In this section I will describe the current CRES signal reconstruction process which is executed by Katydid. The input files contain 5 seconds of I/Q voltage data sampled at 100 MHz by the ROACH2 digitizer. The digitizer has the ability to record data in both streaming (continuous) and triggered mode; triggered data files contain many $\mathcal{O}$(ms) acquisitions, and the last acquisition extends the total duration to or above 5 seconds.



```
processor-toolbox:
    processors:
      - type: egg-processor
        name: egg
      - type: forward-fftw
        name: fft
      - type: convert-to-power
        name: to-ps
      - type: basic-root-writer
        name: writer

    connections:
      - signal: "egg:header"
        slot: "fft:header"
      - signal: "egg:ts"
        slot: "fft:ts-fftw"
      - signal: "fft:fft"
        slot: "to-ps:fs-fftw-to-psd"
      - signal: "to-ps:psd"
        slot: "writer:ps"

    run-queue:
      - egg

egg:
    filename: "/path/to/file.egg"
    egg-reader: egg3
    slice-size: 16384
    number-of-slices: 1

fft:
    transform-flag: ESTIMATE

writer:
    output-file: "/path/to/file.root"
    file-flag: recreate
```

Figure 3.1: Katydid configuration file with a diagram of the associated analysis chain. This short example performs a Fourier Transform, computes the resultant power spectrum and writes it to a ROOT file. The steps enumerated in subsection 3.1.3 are indicated by the brackets numbered 1–4. Processors are represented by the four rectangles down the center, and the connections between them by the smaller red boxes. The data pointer at each connection is also shown, with its list of data objects. The primary three connections in this example (excluding the egg header) demonstrate the so-called linear processing steps mentioned earlier in this section.

3.2.1 Time slicing and Fourier Transform

The analysis begins with the egg processor which reads and segments the raw data into time slices. Egg files also contain a header section with information about the acquisition to ensure the data is self-descriptive. The egg reader first sends the header information to the DFT which initializes the appropriate FFT objects using the FFTW library [16]. Then, the time slice is passed to the DFT processor which transforms it to a frequency spectrum.

The complex frequency spectrum is converted to a real power spectral density (PSD), according to:

$$P(\omega) = \frac{1}{50\Omega}\frac{2\pi}{\Delta\omega}\left|\tilde{x}(\omega)\right|^2 \tag{3.1}$$

where $2\pi/\Delta\omega$ is the reciprocal frequency bin width $1/\Delta f$, or equivalently the time slice length Δt. When a full acquisition (many time slices) is processed, the power spectra from all slices together form a two-dimensional spectrogram of power over frequency and time. Figure 3.2 shows an example raw spectrogram for a 4.5 ms duration with a CRES event present.

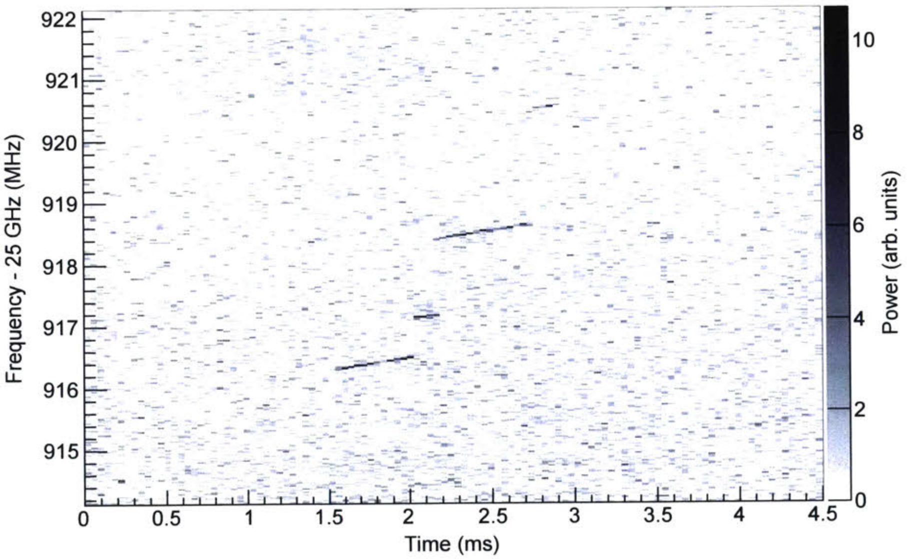


Figure 3.2: Example raw spectrogram with a CRES event. The signal power is sharply localized and its frequency increases with time due to radiative energy loss. The electron scatters with residual gas molecules at least two times, where it abruptly loses energy and the signal undergoes a discrete "jump" in frequency. The event shown here is the first β electron from tritium decay ever recorded with CRES.

3.2.2 Gain variation

As a consequence of the filters internal to the receiver chain and data acquisition system, the average frequency spectrum of only noise is not uniform. The noise power instead varies with frequency, and this dependence does not have an accurate functional form. To model and properly account for this variation, the raw power spectra are accumulated to determine the average $\langle P(\omega)\rangle$ and variance $\langle P(\omega)^2\rangle - \langle P(\omega)\rangle^2$ spectra. These two spectra are then passed to a Gain Variation processor, which constructs a spline for each spectrum using the ROOT TSpline library.[3] The spline contains 50 nodes, with each node equal to the average of all bins within the corresponding range of the spectrum.[4]

These two splines are used to evaluate the noise mean and standard deviation as a function of frequency. The entire data file is processed and the resulting spectra accumulated before this spline calculation is performed. Since the signal power is very sparse (especially in streaming data), we comfortably neglect the influence of signal power over the entire 5-second (5×10^8 samples) duration on the spline calculations.

3.2.3 Power discrimination

Once the data file has been fully processed and the gain variation splines stored, Katydid begins again at the beginning of the file. The power spectra are calculated exactly the same way, and this time sent to a discriminator processor. With the two gain variation splines, the discriminator has the ability to set SNR and sigma-based thresholds:

$$t(\omega) = k_{\text{SNR}} \times s_{\text{mean}}(\omega) \tag{3.2}$$

$$t(\omega) = s_{\text{mean}}(\omega) + k_\sigma \times \sqrt{s_{\text{var}}(\omega)} \tag{3.3}$$

where $t(\omega)$ is the threshold, $s_{\text{mean}}(\omega)$ and $s_{\text{var}}(\omega)$ are the mean and variance splines evaluated at ω, k_{SNR} is the SNR threshold and k_σ is the standard

---

[3] `https://root.cern.ch/doc/v608/classTSpline.html`

[4] The standard slice size for processing ROACH data is 4096, which corresponds to approximately 82 bins per spline node.

deviation threshold. The threshold mode and the threshold parameter k are both excellent examples of the runtime-configurable variables discussed in the previous section. Note that since the normalized noise PSD follows a χ^2-distribution with 2 degrees of freedom (an ordinary exponential distribution), the mean and standard deviation are equal and thus any SNR threshold k_{SNR} should be equivalent to a sigma threshold $k_\sigma = k_{\mathrm{SNR}} - 1$. In practice, the mean and standard deviation as evaluated by the splines are similar but differ at the percent level.

The collection of points which meet the discriminator threshold is called a sparse spectrogram. Figure 3.3 shows the sparse spectrogram of another CRES event with a threshold of SNR ≥ 6.

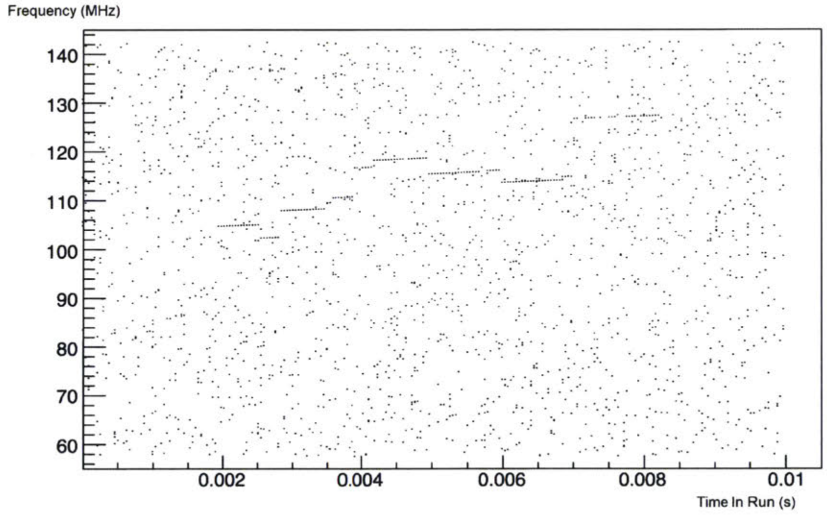


Figure 3.3: Example sparse spectrogram with $k_{\mathrm{SNR}} = 6$. The signal in between scattering interactions is now characterized by an excess of collinear points in the sparse spectrogram. This event also contains several scatters, and some appear to decrease the signal frequency which would suggest the electron gains energy during the scatter, an extraordinarily unlikely event. By the end of Chapter 5, we will better understand how these signals can decrease in frequency through other means.

3.2.4 Sequential Track Finder

As an electron persists in the trap, it loses energy to radiated power and increases its signal frequency. On the timescale it is trapped, the total fractional energy loss is small and thus the power is approximately constant. Consequently, a CRES signal in the sparse spectrogram is described by a pseudo-linear collection of points. The sequential track finder (STF) searches a sparse spectrogram for these linear signals, called tracks. It was developed by Christine Claessens based on an algorithm introduced by Dan Furse. The primary goal of this track reconstruction is to identify the start frequency of a signal, which is of course related to the energy of the electron immediately after the source atom decay. The STF processes slices of the sparse spectrogram from the discriminator in series, and sequentially constructs track candidates from an excess of approximately collinear points with high SNR. A modified version of this algorithm also serves as the real-time trigger for ROACH triggered acquisitions, where its ability to act on individual slices in time rather than a longer sparse spectrogram is essential. At each slice, the STF is given a set of new points where every point contains a time coordinate, frequency coordinate, and SNR. Because tracks can be wider in frequency than a single bin, the SNR is calculated from a local excess amplitude:

$$A_j = \frac{1}{b_j}\left(\left(\sum_{k=j-d}^{j+d} p_k\right) - 2d\, b_j\right) \tag{3.4}$$

where j and k refer to the frequency bin index, p_j is the power in bin j, d is a configurable size parameter and b_j is the background level.

The STF maintains a list of all active track candidates, and checks each new point for a match with each candidate. A point is appended to the candidate if its frequency coordinate matches the extrapolated frequency of the candidate at the same time:

$$|f_j - (f_c + q_c(t_j - t_c))| < \Delta f \tag{3.5}$$

where t_j and f_j are the coordinates of the point, t_c and f_c are the coordinates of the most recent point in the candidate, q_c is the slope of the candidate and Δf is a frequency tolerance parameter. If a point cannot be matched to any pre-existing candidate, it creates a new candidate with one point, itself. Until it receives another point, the slope of this new cluster is predefined with an initial slope parameter. When an existing candidate receives a new point, the slope is re-evaluated using the SNR from Equation 3.4 as a weight:

$$q_c = \frac{\sum_j A_j \times \frac{(f_j - \langle f \rangle)}{t_j - \langle t \rangle}}{\sum_j A_j}. \tag{3.6}$$

where j is the point index. In addition, new points in a sparse spectrogram slice are considered in descending order of SNR by the STF processor. Together with the SNR weights in the above equation for the slope, this ensures that the slope of a track candidate is not significantly deflected by occasional noise points.

A candidate is declared inactive when the time distance between the end of the candidate and the current time slice exceeds a time tolerance parameter Δt. At this stage, the candidate is trimmed of potential noise points at its periphery: any point at the start or the end (in time) of the track with SNR less than a trimming threshold is removed. This process repeats until at least one point at each end meets the threshold. The candidate is then declared a track if a minimum number of points remain, and passed to the next processing stage. The frequency and time tolerance parameters Δf and Δt, the trimming threshold and the minimum number of points are all configurable in Katydid.

3.2.5 Overlapping and Iterative Track Clustering

Since the STF constructs track candidates iteratively and with a strict termination condition (parametrized by the time tolerance Δt), a track can be declared inactive prematurely, and as a result the STF may reconstruct one track as many broken segments. This is obviously problematic for a proper extraction of the start frequency and the overall event rate, as a broken track will falsely yield several start frequencies and all but one (the first) is systematically shifted upwards. This creates high-frequency tails in the resultant spectrum. To minimize this effect, the STF contains a second stage in which two tracks labelled j and k are concatenated together if they overlap (describe the same track) or meet all of the following conditions:

1. The start time of track j and the end time of track k differ by less than the time tolerance: $|t_{j,\text{start}} - t_{k,\text{end}}| < \Delta t$
2. The start frequency of track j and the extrapolated frequency of track k at the start time of j differ by less than the frequency tolerance defined by Equation 3.5; this is the same condition for appending new points to a candidate. $|f_{j,\text{start}} - (f_{k,\text{end}} + q_k\,(t_{j,\text{start}} - t_{k,\text{end}}))| < \Delta f$
3. The slopes of the two tracks differ by less than a configurable slope tolerance: $\left|1 - \frac{q_j}{q_k}\right| < \Delta q$

Once no further pairs of tracks satisfy the above three criteria, they are passed to the final stage of reconstruction: the event builder.

3.2.6 Event Builder

Recall that a single electron signal may take the form of many reconstructed tracks. We have already observed this in the spectrograms shown in this chapter, where scattering interactions separate one event into several tracks distinct in time. It is also possible for an electron to radiate in multiple frequency bands simultaneously as a result of sideband power deposition, which we will study in great detail in Chapter 5. This creates a group of parallel tracks which coincide in time. The event building scheme groups tracks together in two sequential steps: the first addresses this sideband deposition by building multi-peak track (MPT) objects from individual tracks, and the second creates full events from the resulting MPTs to account for scattering interactions.

Multi-peak tracks are composed of tracks which match either start or end times to within a tolerance Δt_s (the "sideband tolerance"). Each track is iteratively checked against all current MPTs until the following condition is satisfied:

$$|\delta t_{\text{start}}| \leq \Delta t_s \text{ or } |\delta t_{\text{end}}| < \Delta t_s \tag{3.7}$$

where δt is the difference in timestamps between a track and a MPT; the MPT start time is simply the average start time of its constituent tracks, and the same for its end time. If the track does not match any MPT, a new one is created with only that track. Also note that although the purpose of the MPT builder is to collect parallel tracks, there is no explicit requirement or check on the track slopes.

After all tracks have been placed into MPTs, the MPTs are sent to the Multi-Peak Event Builder to construct the final events. Here, the MPTs are matched head-to-tail to group together those which are attributable to the same electron before and after a scatter:

$$|t_{\text{start, MPT}} - t_{\text{end, Event}}| < \Delta t_{\text{jump}} \tag{3.8}$$

where Δt_{jump} is the configurable tolerance for this step. Contrary to the first step, this time each MPT is checked against all MPTs within an event; it is appended to an event if matched with any of the constituent MPTs. In addition, the MPT is checked against all active events even after a match is found; if it is matched to more than one event, those events are then merged into one.

3.2.7 The full analysis chain

The multi-peak events produced by the event builder are the final output of the standard analysis for Project 8. Figure 3.4 illustrates how all of the components described in this section form the Katydid analysis chain. Processors which operate in the so-called linear fashion are omitted for brevity, and instead only the data pointer and last object are shown. I have also reduced the STF to a single processor for ease of illustration but in truth it consists of three. All other distinct data pointers and all processors which interact with at least two of them are shown, in addition to the egg reader and ROOT writer processors.

3.3 Analysis Outlook

In this chapter we have walked through the analysis framework, event reconstruction procedure, and introduced the basic properties of CRES signal

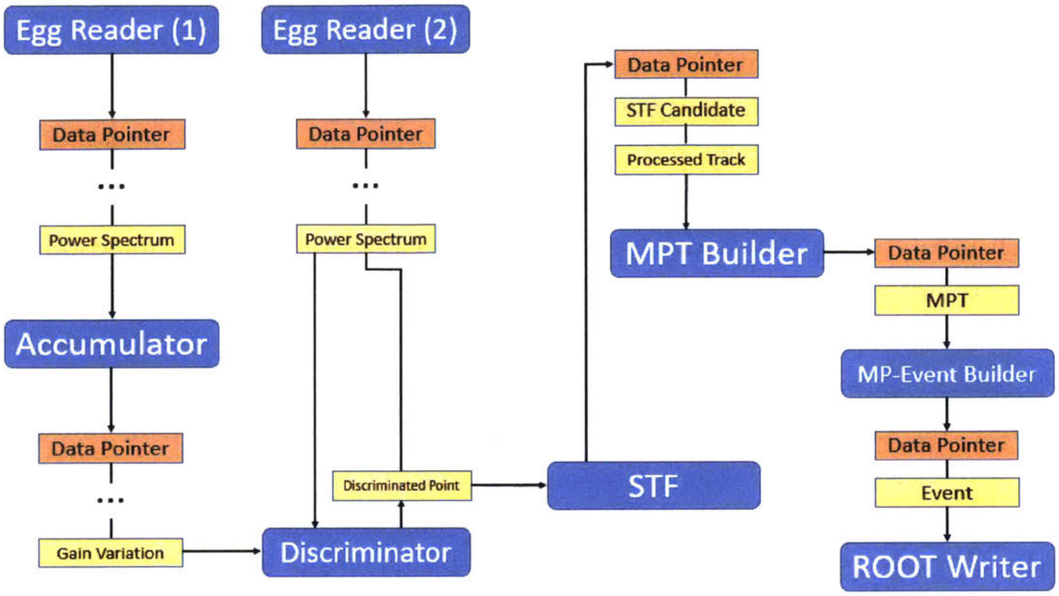


Figure 3.4: Analysis chain for the STF-based primary track and multi-peak event reconstruction. Processors which exchange with more than one data pointer are shown with blue rectangles; those which only involve one data pointer are excluded for brevity, and the three STF processors are represented as one. The first egg reader begins the analysis, terminating when the discriminator receives the gain variation data. Then, the second egg reader begins reading the data file again and the discriminator continues this chain by sending high-power points to the STF processors. From there, the track finding and event building algorithms proceed and the final events are sent to the ROOT writer which completes the analysis chain.

data. This event reconstruction has been very successful in the first phase of Project 8, and the same is expected of the upcoming final Phase II campaign. The science goals of these first two phases are largely restricted to demonstration of the CRES technique with both krypton and tritium, and adequate energy resolution. As we look forward to the remaining Project 8 phases, the central goals of a competitive energy resolution and enormously increased volume/statistics demand a much more sophisticated understanding of CRES signals and, in turn, a much more sophisticated analysis. In the remaining chapters I will present some of the first steps toward that goal. Chapter 4 describes an original track reconstruction algorithm which is implemented in a framework similar to but outside of Katydid, and may have greater versatility to handle low-SNR tracks and guided parameter optimization. In Chapter 5 we turn our attention from primary track finding to secondary track analysis, which involves processing data with prior knowledge of the track locations to study them more carefully. This is developed in Katydid, and the secondary track analysis takes place in a third pass of the data beginning with another egg reader instance. Finally, Chapter 6 presents some preliminary studies on altered Fourier Transforms which allow for extremely precise frequency extraction when the track parameters are already roughly known. Currently, Project 8's frequency resolution is mostly limited by imperfections in the event reconstruction and magnetic trap profile; but to achieve the target sensitivity of Phase IV and obtain a direct measurement of the neutrino mass with CRES, such extra steps to improve the ultimate limiting frequency precision will most likely be necessary. There is an almost limitless potential to expand upon the ideas developed here and in the following chapters, to begin the path toward an extensive Phase III analysis, and to ultimately help the CRES technique evolve into a leading method of ultra-precise spectroscopy.

Chapter 4

Likelihood Clustering

Here, we introduce an original track finding algorithm called the likelihood clustering. It is a new approach to primary track finding, which could serve as an alternative to or combined with the STF algorithm discussed in the previous chapter. Central to the algorithm is the likelihood metric, which estimates the probability that a track candidate is attributable to signal power, and not random noise fluctuations. The likelihood clustering builds linear tracks from a sparse spectrogram with the consistent goal of increasing this likelihood metric. It was conceived and developed entirely by the author and is fully implemented in C++. I will first present the motivation for the algorithm, walk through its steps in considerable detail, and then study its performance on recent Project 8 data.

4.1 Motivation and the Likelihood Metric

During the first several months of the Phase II krypton campaign, the Project 8 collaboration discovered many new complex characteristics of the observed CRES signals. The start of Phase II marked the first time in the project's history that the entire cell and surrounding components were replaced with a new, upgraded system. The upgrades between Phase I and Phase II (listed and discussed briefly in Chapter 2) were numerous, and boasted very promising improvements to the available trap geometries and SNR. However, as the first Phase II data was obtained it quickly became clear that the existing reconstruction, which was guided in development by Phase I data, had considerable difficulty working on the new Phase II data.

Figure 4.1 shows an example event from early in Phase II, with the STF reconstruction overlaid. This event highlights many aspects of the CRES signals which were not apparent, or not present at all in Phase I data:

- With the new Phase II data acquisition system, we gained the ability to take continuous (streaming) data and study long tracks. Note the timescale on Figure 4.1; the event lasts more than 150 ms until it leaves the acquisition bandwidth, and many of the individual tracks are 50 ms or more in length. This is far longer than the Phase I events, which were almost always limited to 10 ms by the RSA acquisition settings. Of course, the track length most directly depends on the gas pressure, which can be adjusted at will. However, Phase II still gave Project 8 the ability to acquire this type of data and study these very long events, which clearly illuminate some weaknesses in the signal reconstruction.

- The tracks are faint (low SNR). Although some events do have the increased SNR as predicted, many events like the one pictured here have power which is comparable to or even less than those in Phase I. This, in conjunction with the long duration of the tracks, causes the reconstruction to often terminate track candidates prematurely. Consequently, the tracks appear "choppy" as one track is reconstructed in several pieces. The pieces themselves are usually quite disconnected, which then leads to many broken-up events as evidenced by the black dots which mark the event starts.

- At least one of the tracks has a completely new feature: a curve from approximately $t = 1.56 - 1.58$ seconds. This is especially difficult for the existing reconstruction techniques to handle, as the linear nature of the tracks appears no longer guaranteed. Combined with the other items listed above, it is perhaps unsurprising that the Phase I reconstruction misses this curved segment entirely. We now understand this type of curved track to arise from power deposition into the TM01 mode of the waveguide, which was not possible in the Phase I waveguide. Only radiation propagating in the primary TE11 mode is detected, and thus the additional coupling to TM01 manifests as an increase in the track slope. The curved nature of the track suggests this effect is frequency-dependent.

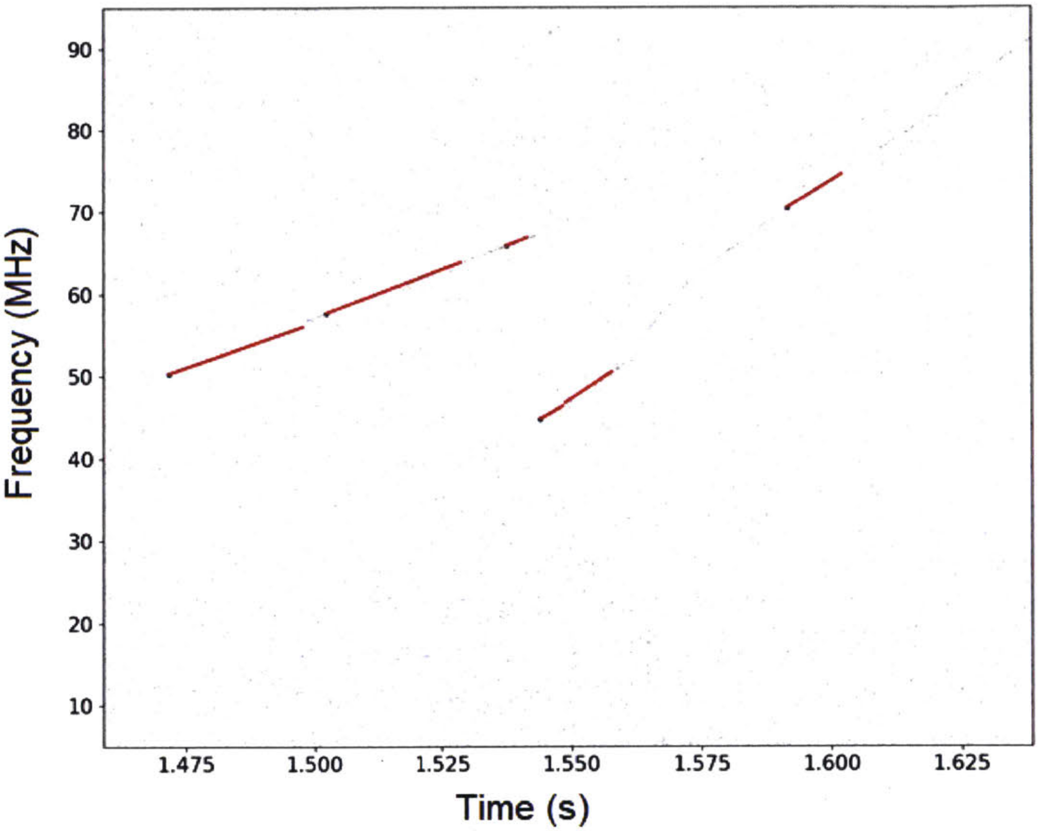


Figure 4.1: Example event from early Phase II with the Phase I reconstruction overlaid. The red lines indicate reconstructed tracks, and the black dots mark the event starts. The Phase I reconstruction is noticeably poor when applied to this data, as it breaks up one event into many (in an unpredictable fashion) and completely misses large portions of it as well.

As the Project 8 analysts worked to improve the existing reconstruction, in parallel we began to consider entirely new track finding ideas which could be better suited for Phase II data. Among these ideas were a phase-based approach studied by Alessandro Banducci [17], a tEMST tree algorithm implemented by Alec Lindman [18], and the likelihood clustering. The likelihood clustering in particular is motivated by the difficult problem of reconstructing long, faint, and (in some cases) even curved tracks as reliably as possible. It is evident from Figure 4.1 that there is a substantial disconnect between what the Phase I reconstruction "sees" and what a human observer sees by eye in the spectrogram. The likelihood clustering is an attempt to bridge this gap, and produce tracks which more closely align with human handscanning results.

The driving force behind the algorithm is the likelihood metric. Many of the tracks in Phase II-krypton data are even fainter than the ones in Figure 4.1, and are difficult to label accurately even by eye. In addition, streaming data contains vastly more data volume per event than triggered acquisitions, which is equivalent in the data processing context to a greatly reduced event rate; the data volume of the entire Phase I campaigns from beginning to end ($\sim$20 TB) is matched by streaming data in roughly 24 hours. This demands a Phase II reconstruction which is very strong for data with a sparse signal content.

The likelihood metric is a quantitative answer to the question: does a particular group of points in a sparse spectrogram contain a CRES signal? We can write down this metric quite easily using Poisson statistics, provided only that we define the nature of a CRES signal appropriately. Then, the clustering algorithm is developed around this likelihood metric with the singular objective to increase it at every possible opportunity.

4.1.1 Likelihood metric derivation

Consider a cluster of points as in Figure 4.2. We will assume the tracks we aim to reconstruct have a linear structure and a fixed width in the frequency domain; this is illustrated by the overlaid parallelogram shape. Note that with this assumption we abandon the possibility to reconstruct curved tracks, at least with the likelihood clustering alone. The parallelogram axis is calculated with a linear fit to the points, and the width is determined by the extrema of

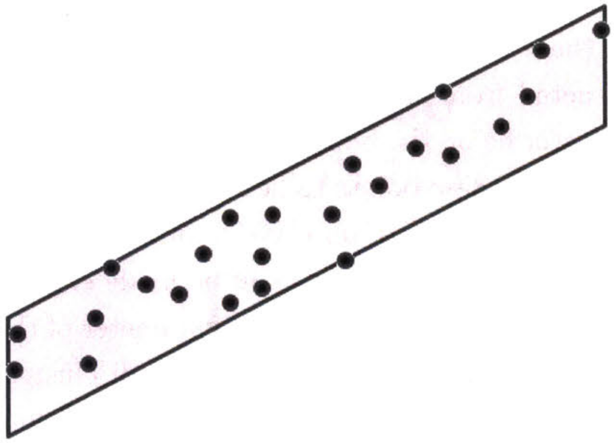


Figure 4.2: Cluster of points with the associated parallelogram overlaid.

the residuals. Precisely, if the points are labelled by (x_j, y_j) and the linear fit has intercept α and slope β, then the parallelogram is defined by:

$$\min_j [x_j] \leq x \leq \max_j [x_j] \tag{4.1}$$

$$\alpha + \beta x + \min_j [y_j - \beta x_j - \alpha] \leq y \leq \alpha + \beta x + \max_j [y_j - \beta x_j - \alpha] \tag{4.2}$$

With this construction, we have now defined a region in the frequency/time plane which this cluster of points occupies. The area of the region is:

$$A = \left(\max_j [x_j] - \min_j [x_j]\right) \times \left(\max_j [y_j - \beta x_j - \alpha] - \min_j [y_j - \beta x_j - \alpha]\right) \tag{4.3}$$

At this point, it is straightforward to calculate the likelihood metric with Poisson statistics. If the noise density (the average number of points per unit area with no signal) is ρ, then the expected number of points in the region k is simply:

$$k = \rho A \tag{4.4}$$

and the observed number of points is of course known, which we call n. The likelihood metric then follows from the Poisson distribution CDF:

$$\sum_{j=0}^{n} \frac{k^j e^{-k}}{j!} = Q(n+1, k) \tag{4.5}$$

where Q is the regularized Gamma function. Before we may write down the likelihood metric, there is one caveat to address. Since the parallelogram boundary is constructed from the points, this means in general four[1] of the points within the cluster lie on the boundary of the region, rather than strictly inside it. Do we interpret these points to lie within the boundary anyway, and thus include them in the number of observed points n, or do we exclude them and replace n with $n-4$? An argument can be made either way, and in the interest of choice I leave this as a configurable parameter of the analysis called the boundary correction factor. Especially for small clusters, the choice of this correction factor can have a substantial impact on the final reconstructed tracks, and thus it is valuable to keep as a free parameter for optimization.

The likelihood metric is defined by Equation 4.5 with this boundary correction factor c:

$$L(n,k) \equiv -\ln\big(1 - Q(n+c,k)\big) \tag{4.6}$$

where the sensible range of c is $-3 \leq c \leq 1$. We take the negative logarithm to avoid floating-point precision issues for extremely small numbers, and for ease of interpretation. For a distribution of only noise and no signal power, we assume the CDF is accurately described by Equation 4.5 and thus the likelihood metric defined by Equation 4.6 follows an exponential distribution with mean 1 (by construction due to the logarithm). Note that for this claim to hold exactly we must insist both $c = 1$ and the region with area k/ρ to be arbitrary; the former can be satisfied by choice but the latter cannot, as the cluster region is always determined by the point locations. With this caveat understood, we will not address it again and proceed with the natural interpretation of the likelihood metric.

4.1.2 Polynomial approximation of the likelihood metric

To accurately calculate $L(n,k)$ we can only use the regularized gamma function directly when the likelihood is reasonably small. For long signal

---

[1] Neglecting cases in which one point may define multiple edges, and thus fewer than four points define the boundary.

tracks, the likelihood can grow very large and proper evaluation of the gamma function is no longer an option as it becomes indistinguishable from 1. To circumvent this problem, I use an approximate form of the likelihood in the $n \gg k$ limit. The series expansion of $1 - Q(n,k)$ for $n \to \infty$ and finite k is, via Mathematica:

$$\begin{aligned} 1 - Q(n,k) = e^{n-k}\left(\frac{k}{n}\right)^n \frac{1}{\sqrt{2\pi}}\Bigg(& n^{-1/2} + \frac{12k-1}{12}n^{-3/2} \\ &+ \frac{288k^2 - 312k + 1}{288}n^{-5/2} + \frac{51840k^3 - 159840k^2 + 56340k + 139}{51840}n^{-7/2} \\ &+ \mathcal{O}(n^{-9/2})\Bigg) \end{aligned} \tag{4.7}$$

and the likelihood follows from the logarithm of the above equation:

$$\begin{gathered} L(n,k) = L_0(n,k) + \ln\left(\sum_{j=0}^{j_{\max}} \frac{p_j(k)}{c_j} n^{-j}\right) \\ \text{with } L_0(n,k) = n - k + n\ln(k/n) - \frac{1}{2}\ln(2\pi n) \end{gathered} \tag{4.8}$$

where $p_j(k)$ denotes the polynomial term of order j in Equation 4.7 and c_j is always the leading-order coefficient of $p_j(k)$. This approximation can be extended to arbitrary order $j_{\max}$ by calculating all required $p_j(k)$ terms. For the likelihood clustering, $j_{\max} = 3$ (as in Equation 4.7) shows good agreement with the exact function; this is demonstrated for example in Figures 4.3 and 4.4. For clarity, the first three $p_j(k)$ polynomials are listed in Table 4.1 as well.

j	$p_j(k)$	c_j
0	1	1
1	$12k - 1$	12
2	$288k^2 - 312k + 1$	288
3	$51840k^3 - 159840k^2 + 56340k + 139$	51840

Table 4.1: Polynomials for the likelihood approximation up to $j = 3$.

The polynomial approximation is valid for $n \gg k$, and recall that it is only needed as a substitute when $L(n,k) \gg 1$. For small k, these two conditions are

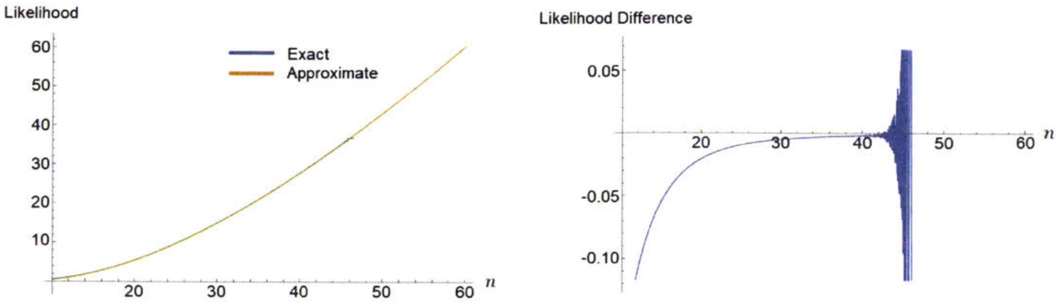


Figure 4.3: Comparison of exact and approximate likelihood functions vs. n with $k = 10$. The plot on the right shows the difference between the two functions. The agreement is excellent for $n \gg 10$, and the approximation allows for the calculation of very large (e.g. $L > 50$) likelihoods where the exact form does not.

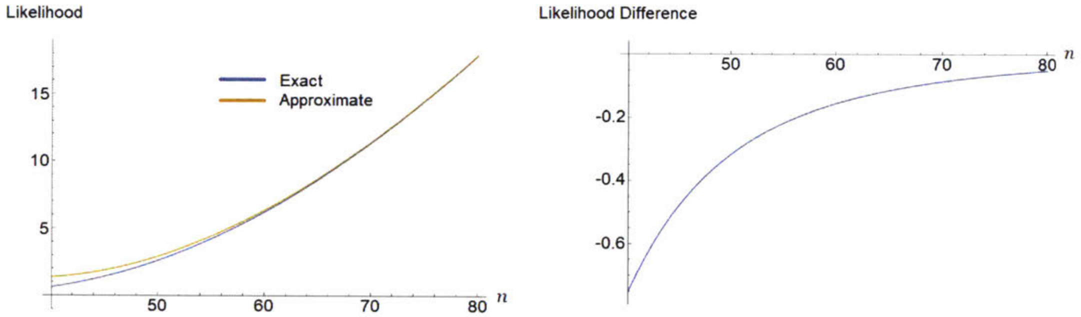


Figure 4.4: Comparison of exact and approximate likelihood functions vs. n with $k = 40$. The plot on the right shows the difference between the two functions, which is more substantial compared to the $k = 10$ case.

equivalent; high likelihood implies $n \gg k$. However, this equivalence begins to break down for large k, as can be seen in Figure 4.4. Here, the disagreement between the two functions remains noticeable even while the likelihood climbs much greater than 1 because the ratio n/k grows comparatively slowly when k is large. Therefore, we can be fully confident in the approximation except in the regime where $k \gg 1$ and $n > k$ but $n \not\gg k$.

For this application of the likelihood metric in practice, the effect of the

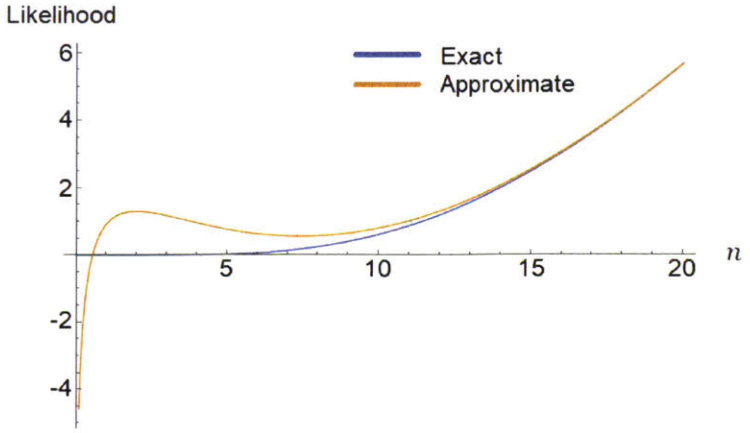


Figure 4.5: Comparison of exact and approximate likelihood functions vs. n with $k = 10$. The approximation clearly breaks down when $n < k$, as expected; the exact likelihood is used in this case.

discrepancy for large k is not a serious concern. We utilize the approximate form when $1 - Q(n, k) < 10^{-9}$ which corresponds to $L(n, k) > 9 \ln 10 \approx 20.7$; even for large but realistic values of k, the error when $L > 20$ is small and the qualitative behavior of the approximation is exactly the same as the true function. Still, I take one additional step to ensure a more accurate calculation: for a fixed value of n, I first calculate the difference between the exact and approximate likelihoods at the value of k corresponding to $1 - Q(n, k) = 10^{-9}$. This difference is then added back to the approximate likelihood with the true value of k, which roughly corrects for the error. To summarize the likelihood calculation, with a cluster of n points and k expected points we take the following steps:

- If $1 - Q(n, k) \geq 10^{-9}$, use the exact likelihood given by Equation 4.6.
- Otherwise, compute the error of the approximation at $L = 9 \ln 10$:

  $$\delta L = 9 \ln 10 - L(n, k') \text{ where } 1 - Q(n, k') = 10^{-9} \tag{4.9}$$

  and $L(n, k')$ is given by Equation 4.8 with $n \to n+c$. The final likelihood is then $L(n, k) + \delta L$.

4.1.3 Algorithm overview

The main components of the likelihood clustering algorithm can be divided into two stages: pruning and iterative extension. Each stage is an iterative process in which each iteration has the singular goal to increase the likelihood metric; the ideal result is then a track candidate which is a local maximum of the likelihood. The pruning stage searches a small section of a sparse spectrogram for a likelihood excess, indicating some structure above the noise. This initial cluster is then passed to the extension stage which has access to the full sparse spectrogram. The extension stage iteratively extends the cluster boundaries to further increase the likelihood and capture the full extent of the signal.

4.2 Clustering Algorithm: Pruning Stage

To begin the clustering, we divide a sparse spectrogram into square sections (blocks) on a grid; Figure 4.6 illustrates this scheme. The blocks are tessellated with a redundancy factor of 2, which is also illustrated in the figure, to avoid biasing the results based on the arbitrary locations of the block boundaries. The blocks are each processed through the pruning stage sequentially and independently.

For each block, the pruning stage first considers a cluster which contains every point in the block. In most cases, there is no signal power in the block and the cluster consists of a few[2] randomly distributed points. Since at least four points are required to guarantee a valid likelihood for $c = -3$, the block is immediately discarded if it has 3 or fewer.[3] For blocks with at least 4 points, the cluster begins the iterative pruning process according to the following steps:

1. Calculate the linear fit and residuals to determine the parallelogram

---

[2] The precise number of course depends on the block size, and is Poisson-distributed. In practice, we choose the average number of noise points per block to be $\mathcal{O}(1)$.

[3] This initial cut can be lowered if $c > -3$, but note that independent of c the cluster region is only well-defined with 3 or more points. I have chosen to neglect 3-point blocks nonetheless.

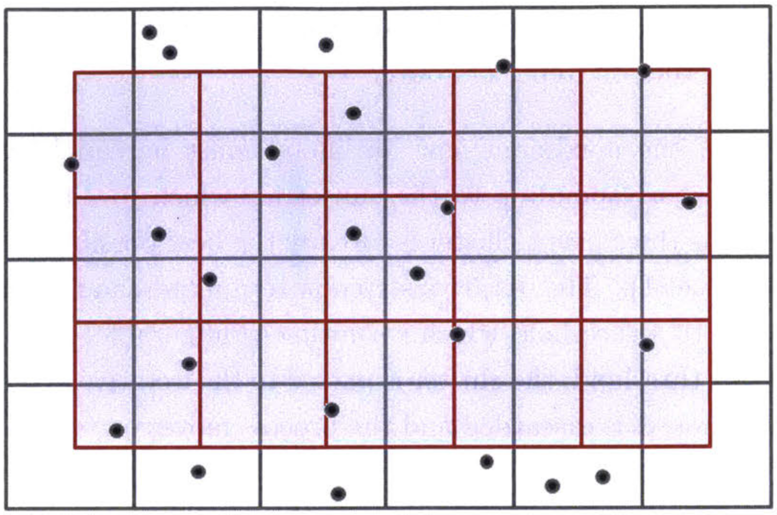


Figure 4.6: Illustration of the block division scheme for the pruning stage. The sparse spectrogram is segmented into square blocks with a specified size, and then again with the blocks shifted by half the block width. This tessellation helps to better capture any potential signals which are split across two or more of the initial blocks.

area A. From the known noise density, this also determines the expected number of points k with Equation 4.4.

2. Compute the likelihood and cache it for future comparisons. We refer to this as the initial likelihood.

3. Consider every possible subset of the cluster which contains at least $n - p$ points, where n is the number of points in the current cluster and p is a configurable parameter called the maximum pruning step. This effectively removes at most p points from the cluster. If $n - p < 4$, then only subsets with at least 4 points are considered.

4. For each subset, recalculate the likelihood according to steps (1) and (2) above. Determine the maximum likelihood among all subsets; this will be called the new likelihood.

5. If the new likelihood exceeds the initial likelihood, this indicates a set of points which it is advantageous to prune. These points are then removed

from the cluster, and the next iteration begins with the newly-pruned cluster. Return to item (1) above.

6. Otherwise, the maximum new likelihood must be equal to the initial likelihood, corresponding to the subset for which $p = 0$ (no pruning). This means the current cluster is already the best possible subset for the chosen value of p. The iterative step now terminates, and the likelihood is compared to a threshold which is runtime-configurable. If the likelihood meets this threshold, the cluster is passed to the iterative extension stage; if it does not, it is discarded and the process moves on to the next block.

The configurable parameters of note in this stage are the block size, the prune step p, and the initial threshold. The block size is specified by the average number of noise points $\langle k \rangle$, which lends itself to a more natural discussion with the other parameters. If the prune step is large, i.e. $p \gg \langle k \rangle$, then the process described above approaches an exhaustive search of every possible subset cluster. This effectively guarantees the ideal result, which is the overall maximum likelihood within the block. However, in practice, the runtime required for such a large p is unreasonable; the number of trial clusters searched if $n \leq p$ scales exponentially with n. This exponential scaling is manageable if n is small, which is usually the case if $\langle k \rangle$ is chosen to be small as well. Still, if the block contains signal power – which is of course the most important case – then $n \gg 1$ regardless of $\langle k \rangle$, and the exponential scaling prohibits large p. I have chosen $p = 6$ as a large but reasonable value which keeps the runtime nonprohibitive for $1 \leq \langle k \rangle \leq 10$.

The initial threshold has a straightforward mathematical interpretation: it is the minimum Poisson likelihood required to pass a candidate cluster forward to the extension stage. This threshold should be well above the expected value for blocks containing only noise, as the clustering should reject those blocks with near-perfect accuracy. The average likelihood for any arbitrary cluster is 1 by construction, but the pruning stage still searches for the best possible cluster and often considers many ($\sim 10^5 - 10^6$) individual clusters in the process. Thus, the likelihood of the final candidate at the conclusion of the pruning stage will in general be much larger than 1, and highly dependent on p and $\langle k \rangle$. For the results discussed at the end of this chapter, I use a threshold of $L \geq 25$, which corresponds to a probability of $\exp(-25) \approx 10^{-11}$.

4.3 Clustering Algorithm: Iterative Extension

Once a candidate cluster has passed the initial threshold, it is processed by the iterative extension stage. At this point, the full sparse spectrogram is available for consideration. The objective of the extension stage is to search near the boundary of the track for additional signal points which increase the likelihood. When the full track is captured, further extensions should decrease the likelihood as they add only noise. At each iteration, the boundary of the cluster is extended in one or many of the four possible directions (each edge and any combination thereof) up to a maximum amount. The maximum extent in the time domain is configured at runtime, which we call t_{max}. If the time length of the current cluster is Δt and the frequency width is Δf, then the maximum extent in frequency is $t_{\text{max}} \times \frac{\Delta f}{\Delta t}$. The iterative process at this stage is similar to the pruning stage:

1. Calculate the linear fit and residuals to determine k from Equation 4.4 as before.
2. Compute the initial likelihood.
3. Loop through the 15 possible directions of extension: any combination of the four edges (2^4) excluding the case where none are extended. For each of these directions, step each relevant boundary by 1 bin for time-domain edges, or 1 bin $\times$ $\frac{\Delta f \text{ [bins]}}{\Delta t \text{ [bins]}}$ for frequency-domain edges. The number of steps is $t_{\text{max}}/\Delta t$.
4. For each possible extension, search the surrounding area of the spectrogram and add any points which are now included within the boundary.
5. Recalculate the linear fit and likelihood. Note that this step re-shapes the boundary if any new points were added, so that the extension is no longer simply described by moving one or more edges.

6. Determine the best new likelihood from all possible extensions.

7. At this point, the new likelihood L_f is compared to the peak likelihood of the cluster L_p as well as a configurable tolerance δ. The tolerance allows the likelihood to decrease temporarily and thus avoid prematurely terminating tracks, which is the most obvious shortcoming of the reconstruction in Figure 4.1. New points are added with a status flag to indicate whether they increased the peak likelihood or utilized the tolerance. The likelihoods are compared according to the following cases:

    (a) $L_f > L_p$: the likelihood has improved. This extension is performed and the status flag is set to `NORMAL` for all points in the cluster to indicate the new peak likelihood. The algorithm then proceeds to the next iteration with this extended cluster.

    (b) $L_p - \delta \leq L_f \leq L_p$: the likelihood has decreased since its peak, but remains within the tolerance. Here, the next iteration proceeds anyway as in 7(a) but the new points are added with the status flag set to `TEMP`.

    (c) $L_f < L_p - \delta$ or L_f no longer meets the initial threshold from the pruning stage: the likelihood has fallen too severely and the cluster is terminated. No further extension is made and all points with the flag set to `TEMP` are removed, which restores the cluster to its state at the peak likelihood. The final likelihood is recalculated again and compared to a final threshold where the cluster is declared a track or discarded. When a cluster is declared a track, all points within it are removed from the sparse spectrogram and not considered in the pruning or extension stages of future clusters.

Here, we once again have three notable parameters to discuss: $t_{\max}$, the tolerance δ, and the final threshold. The maximum extension in time $t_{\max}$ establishes how far the extension stage is allowed to search for additional signal. For very weak tracks, which are not uncommon in the early Phase II data, there is substantial benefit to setting this parameter as large as 100 bins (4.1 ms) in order to capture sufficient signal power in each extension. By

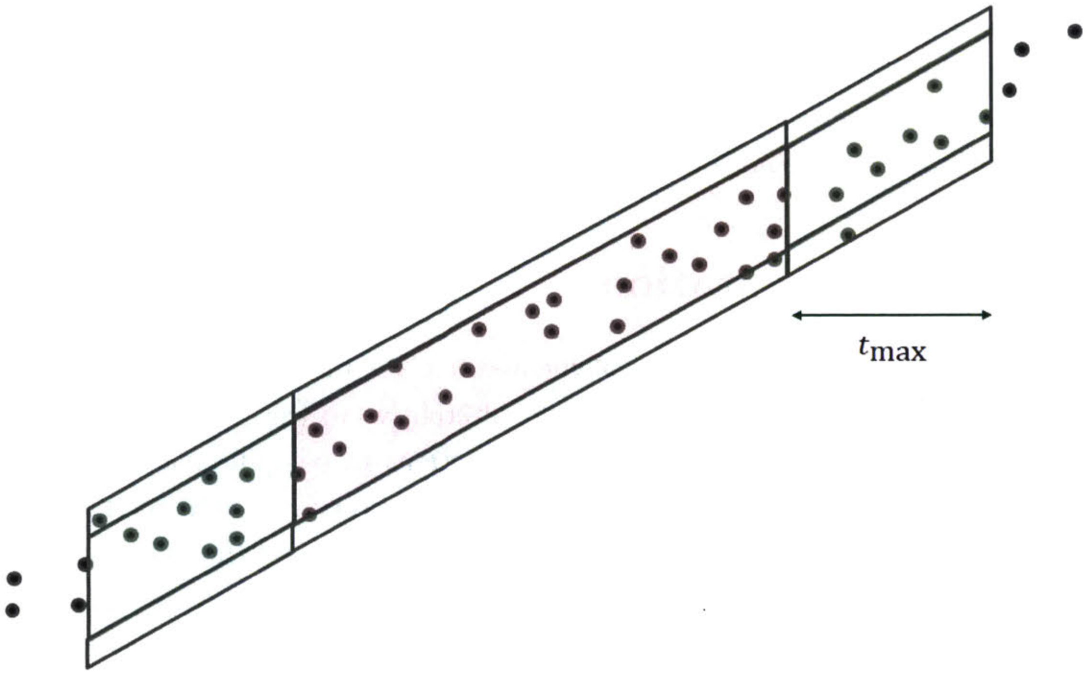


Figure 4.7: Diagram depicting the regions explored during the iterative extension stage. The red region indicates the current cluster, and the green regions indicate the areas accessible by an extension. The maximum extension in t is $t_{\max}$, specified at runtime.

contrast, strong tracks can be captured well with a maximum extension of $\mathcal{O}(1)$ bins which is more akin to the stricter conditions of the STF. Since $t_{\max}$ is only an upper bound and the algorithm includes protection against severe overextension (case 7(b) above), in principle there is no drawback to using a large value of $t_{\max}$ aside from runtime constraints. Indeed, values as large as 160 bins have shown minimal decline in the reconstruction of strong tracks. For the results presented at the end of this chapter, I have chosen $t_{\max} = 40$ bins which is generally sufficient for Phase II reconstruction.

The tolerance dictates the balance between premature termination and overextension. Since the cluster is restored to its peak likelihood before it is finalized, the risk of overextension is limited to the case where a track extends into noise but still reaches a new peak likelihood at least once. For this reason, I choose a fairly high tolerance of $\delta = 7.5$ in order to harshly protect against broken-up tracks. The final threshold parameter of course serves a purpose similar to the initial threshold which we discussed already in the pruning stage. For this, I have chosen $L \geq 40$ which yields zero events in 1 hour of streaming data with only noise (no source gas and no trap).

4.4 Implementation

The likelihood clustering is implemented to C++ fully outside the Katydid framework. It is meant as a lightweight alternative to the Katydid analysis, with minimal dependencies and easy instructions to compile and run. The code is hosted in a private Project 8 scripts repository on GitHub. The full Katydid reconstruction process includes many steps beyond the main clustering as discussed in Chapter 3 – for example, the Fourier Transform and Gain Variation. In the likelihood clustering implementation, these steps are generally mimicked as simply as is reasonable. Below I enumerate each such step and compare its function in the likelihood clustering to Katydid:

1. Time series slicing. The raw voltage samples are grouped into slices with a specified size. For both triggered and streaming standard ROACH data in Phase II, this step is equivalent to its Katydid counterpart.

2. Fast Fourier Transform. Like Katydid, the likelihood clustering accomplishes this step with the FFTW library. Katydid initializes a wide variety of FFT plans to handle different types of data and other preferences; the likelihood clustering instead utilizes just one FFT plan to work with complex-valued data. In addition, the power conversion used in the likelihood clustering is a simple norm-squared: $P(\omega) = |f(\omega)|^2$.

3. Gain variation. Katydid processes the entire acquisition file in a first pass solely to determine the mean and variance of the noise power at each frequency bin. This information is then used by the discriminator to construct a threshold spline for the second pass. In the likelihood clustering, the data file is processed only once, and there is no spline computation. Instead, the first 10,000 slices are utilized for this noise

power modeling and excluded from the track finding. Subsequent spectra are then discriminated with a sigma-based threshold as in Katydid, but via direct comparison to the threshold spectrum rather than a spline. The choice of 10^4 slices gives a fractional uncertainty of 10^{-2} or 1% on the threshold spectrum.

4. After the threshold spectrum is computed, the likelihood clustering must also determine the density of thresholded points in noise; recall that this quantity is crucial for computing any likelihood metric. The clustering first acquires 50,000 points above threshold to determine the noise density, corresponding to a fractional uncertainty of less than 0.5%. After these points are accumulated, the noise density is calculated and all future spectra are sent to the main trackfinding processors.

5. When track finding has concluded, the likelihood clustering attempts to combine every possible pair of tracks in order to further increase the likelihood. If the union of any two tracks, with the parallelogram axis and likelihood recalculated, has a likelihood which exceeds both of its constituent tracks then this concatenation is made. This step serves to partially correct for tracks which are still terminated prematurely, similar to the overlapping and iterative track clustering steps of the STF.

6. Lastly, the event building scheme is simplified to a bare-bones approach. For each track j with start time t_j and end time T_j, the minimum jump time:

$$\min_k \left[\, |t_j - T_k| \,\right] \tag{4.10}$$

   is computed over all other tracks k. If a track's minimum jump time is sufficiently small, then we conclude this track is not the first in an event, since its start time coincides approximately with another track's end time. This method does not capture any multi-peak track topologies, and does not actually group tracks into events; it only preserves those tracks which are likely to be the first within an event and discards the rest.

I have delegated an appendix of this chapter to a more detailed discussion of the structure of the likelihood clustering code. It uses a modular scheme similar to Katydid's signal/slot and processor framework, but with much less overhead complexity. A reader who is somewhat familiar with Katydid (or just comfortable with Chapter 3) might be inclined to further develop the likelihood clustering as well, and it is my sincere hope that such a task has a low barrier to entry compared to Katydid. To this end, I have also included in the appendix a discussion of some potential future developments to the algorithm, parameter optimization, and integration with the STF and Katydid.

4.5 Results on Phase II Data

The likelihood clustering has been run on a number of Phase II data sets for evaluation and for comparison with the standard Katydid reconstruction. Here, I present the results of that evaluation and discuss them briefly. The version of Katydid used for this comparison is v2.13, which includes several significant improvements to the optimization of parameters and the STF over the Phase I analysis. In addition, the data used for this comparison is much more recent than the initial Phase II runs and improvements in the trap configuration have substantially reduced the prevalence of faint tracks. As a consequence of these huge improvements in the standard reconstruction, the likelihood clustering should not be viewed through its initial motivation which was to be in consideration to replace the STF. Instead, the results primarily serve to demonstrate its performance on a large scale and at a similar quality level as the Katydid benchmark.

The data presented in this section is divided into two groups:

1. Noise-only data, nicknamed False Event Acquisition Rate (FEAR) data. This data was taken with all trap coils turned off, and the source gas pumped out of the cell. It consists of 24 hours of triggered data and 1 hour of streaming data. The triggered data produced only 3 data files, each of which is attributable to an anomalous sample in the initial time slice. While the cause of this anomaly has never been determined, we

can still safely neglect these files and thus the clustering is run only on the 1 hour of streaming data. These are runs 7849–7896 in the Project 8 catalog, or the FEAR-3 campaign.

2. Standard trap data. This data was taken with a reliable trap configuration, a 300mA harmonic trap with the center coil. It has been used for several other studies of track and event reconstruction during Phase II; it is run 7427 in the catalog.

All of the configurable parameter values are listed in Table 4.2, including some which were previously mentioned only in passing. The FEAR data was processed with both likelihood thresholds reduced to study the resulting false tracks, and in part to determine suitable threshold values for the processing of electron data.

4.5.1 False Event Acquisition Rate (FEAR)

With the likelihood thresholds reduced, a total of 5358 tracks were observed in the 1 hour of streaming FEAR data. The likelihood distribution of these tracks is shown in Figure 4.8; the largest observed likelihood is $L \approx 46$ and the bulk of the distribution is predictably exponential. The track with the highest likelihood is an anomaly which arises from the binning of the sparse spectrogram: since the point coordinates are discrete, it is sometimes possible to construct a cluster with at least 4 exactly collinear points. The width and thus the expected number of points k of the resulting cluster region is very nearly zero (up to floating point error) and thus the likelihood is artificially inflated. The clustering algorithm discards tracks which have a width of less than 1 Hz to protect against this problem, but this track has a width of 1.4 Hz.

Neglecting this track, the FEAR data produces no tracks with $L > 35$. This informs an appropriate final threshold for the electron data, to ensure the number of false tracks reconstructed is none or very few. I choose $L = 40$ to accomplish this.

Figure 4.9 shows the start frequency spectrum of the FEAR data. Aside from some minor structure near the edges of the bandwidth (< 5 and > 95

Processing Step	**Parameter**	**Value**
Slicer	slice size [samples]	4096
Accumulator	n initial slices	10,000
Discriminator	threshold k_σ	5.0
Block Maker	n initial points	50,000
Block Maker	stride fraction	0.5
Prune Stage	minimum points	4
Prune Stage	$\langle k \rangle$	1.0
Prune Stage	p	6
Prune Stage	initial threshold	13.0 (FEAR data) 25.0 (e^- data)
Iterative Extension	$t_{\max}$ [bins]	40
Iterative Extension	δ	7.5
Iterative Extension	final threshold	22.0 (FEAR data) 25.0, 40.0 (e^- data)
Event Builder	jump time tolerance	0.0
(Multiple)	c	-1.0

Table 4.2: Full list of configurable parameters used for the likelihood clustering results presented in this section.

MHz), the spectrum appears largely consistent with a uniform distribution as expected. The slight bias at the bandwidth edges is likely a consequence of the data acquisition and not unique to the likelihood clustering; the Katydid STF reconstruction restricts the sparse spectrogram to $[5, 95]$ MHz to mitigate exactly this same frequency bias.

4.5.2 Standard trap configuration data

The likelihood clustering is also run on a data set with typical electron data to evaluate its reconstruction quality. Overall, the results are largely on par with that of Katydid; Figures 4.10 and 4.11 show the frequency spectrum and track length distribution, respectively, for both analyses. There is very good agreement between the two results, which demonstrates the ability of the

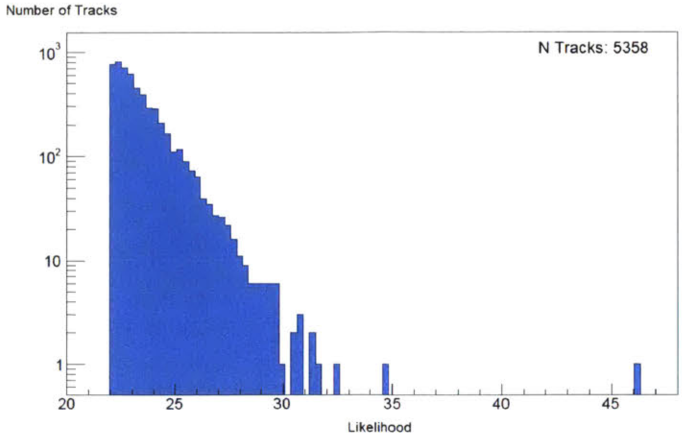


Figure 4.8: Likelihood distribution of tracks in the FEAR-3 data set. The initial and final thresholds were lowered to 13 and 22, respectively, to study the resultant track candidates. The distribution is clearly exponential as expected, and a threshold of at least 35 would remove all events from this data set; the event at $L \approx 46$ is artificially inflated due to the sparse spectrogram binning.

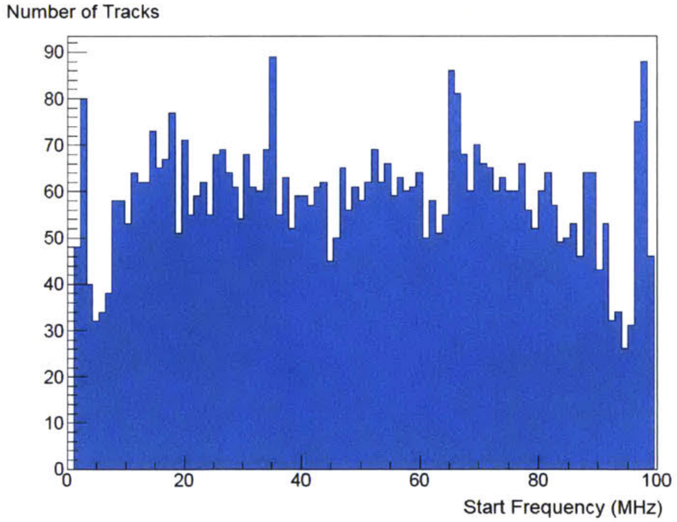


Figure 4.9: Frequency spectrum of tracks in the FEAR-3 data set. As expected, there appears to be no significant structure in the distribution except near the bandwidth edges ($<$ 5 and $>$ 95 MHz).

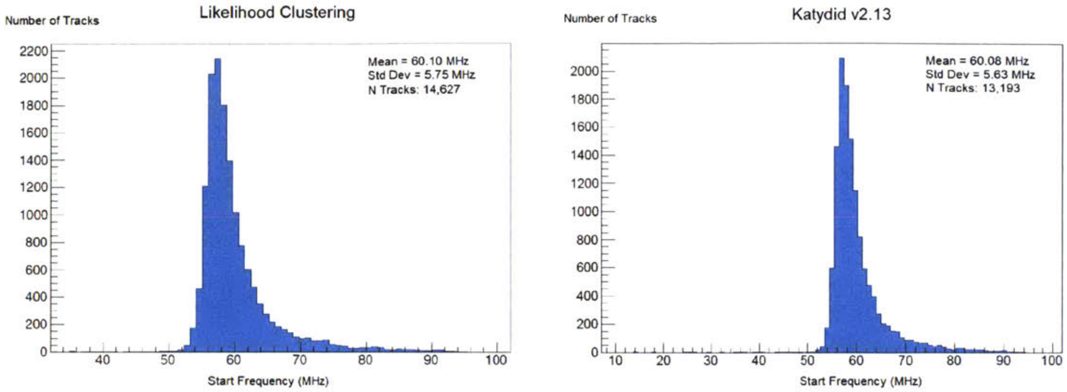


Figure 4.10: Start frequency distribution of tracks in run 7427 found by the likelihood clustering with $L \geq 40$ (left) and by Katydid (right). The mean start frequencies agree to within 20 kHz, the peak width (estimated by the standard deviation) is similar, and the number of tracks is similar as well. The high level of agreement between these spectra unequivocally demonstrates the likelihood clustering's ability to reconstruct typical electron data sensibly.

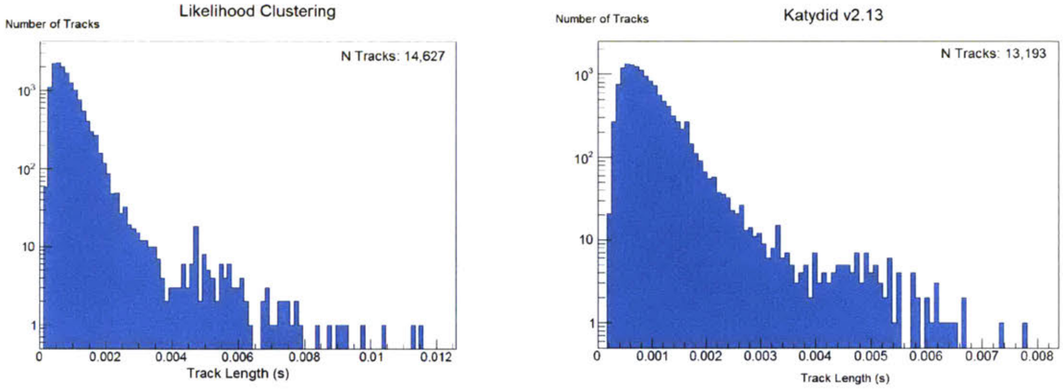


Figure 4.11: Track length distribution of tracks in run 7427 found by the likelihood clustering with $L \geq 40$ (left) and by Katydid (right). In both cases the distribution is not exponential for long tracks, but skewed upwards; this indicates systematic overextension of the tracks as a significant shortcoming of both methods.

likelihood clustering to reconstruct sensible tracks just the same as the STF. The overall event rate is also similar to within approximately 10%.

The final threshold used for the distributions in Figures 4.10 and 4.11 is $L \geq 40$. The event building stage is effectively skipped, as the jump time tolerance is set to zero (see Table 4.2) which passes all tracks through the event cut. This choice was made simply to study the results at the track level rather than the event level.

Perhaps the most significant shortcoming of the reconstruction is evident from the track length distribution, which shows an excess of long tracks. This excess is characteristic of systematic overextension of the tracks, and it is present in both the likelihood clustering and Katydid distributions. The reconstruction could therefore presumably benefit from reducing the lenience of the iterative extension stage, either by decreasing the tolerance δ, the maximum extension t_{max}, or both. In Katydid, the problem of overextension has been largely solved through further optimization of the reconstruction parameters since version 2.13.

Figure 4.12 shows the distribution of the track width (in frequency) for the likelihood clustering reconstruction. This is simply the "height" of the parallelogram calculated from the finalized track candidate, and it does not have an analogous quantity in the STF. Virtually every track has a width less than 200 kHz, and most are concentrated in the region between 25−100 kHz (1−4 bins). This further corroborates the success of the likelihood clustering, as the vast majority of tracks in this data set are expected to lie sharply within one or a few frequency bins at each time slice (this has been verified by visual inspection of the spectrograms).

4.5.3 Short track reconstruction

Figure 4.13 compares the track length distribution for short (< 2 ms) tracks. Since the length of CRES tracks is exponentially distributed, reliable reconstruction of short tracks often has a great impact on the event rate, optimal pressure of the source gas, and in turn even informs the required physical size of future detectors to reach the target energy resolution. Improv-

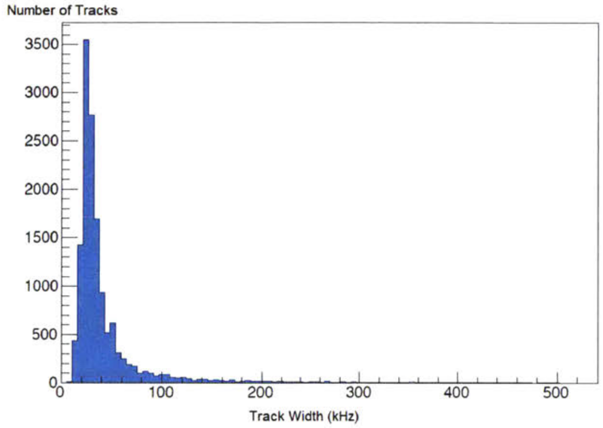


Figure 4.12: Distribution of the track width in run 7427 as found by the likelihood clustering; this quantity is not calculated in Katydid. The distribution is peaked sharply at approximately 25 kHz (1 bin), and the mean is less than 2 bins. This is clearly consistent with the expectation in Phase II, and in addition (a) provides a new way to study reconstruction quality that the STF does not, and (b) affirms that the start frequencies reconstructed by the likelihood clustering reliable to within $\mathcal{O}(1)$ bin.

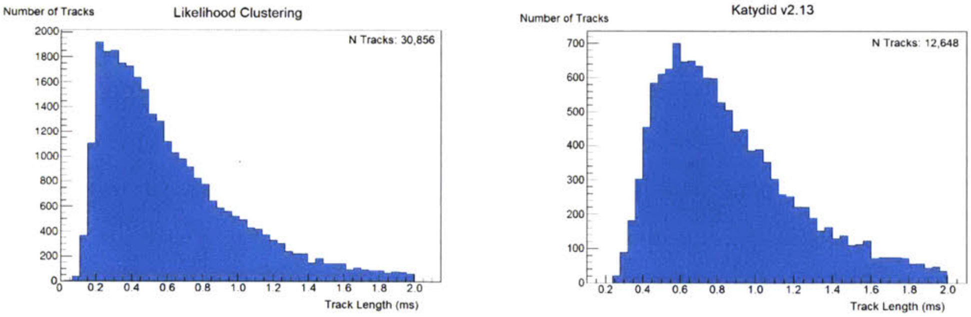


Figure 4.13: Track length distribution of short tracks in run 7427 found by the likelihood clustering. The likelihood threshold is reduced to 25, as insisting $L > 40$ removes most of the tracks at this length scale. As a consequence, there is most likely an excess of false tracks which is responsible for most or all of those below 0.2 ms where Katydid reconstructs none.

ing short track reconstruction in Katydid has been another primary focus of Project 8 analysts as we prepare to conclude Phase II. For this comparison, the likelihood threshold is reduced to $L \geq 25$ as the likelihood is very much correlated with track length, and imposing a safe threshold as before removes many of the shortest tracks. We observe considerably more tracks in the range 0.2−0.4 ms from the likelihood clustering compared to Katydid this way, but with the threshold reduced we also expect a fair number of false tracks to contaminate the distribution. The threshold is only marginally higher than was used for FEAR data (22.0), which yielded approximately 5300 tracks over the same 1-hour live time. For this reason, we conclude the likelihood clustering does not immediately offer an improved short track reconstruction over the STF but the possibility certainly exists. Furthermore, the use of a strict likelihood threshold directly and undoubtedly impacts the reconstruction efficiency as a function of track length; if such a cut is ever to be utilized for official analysis, either within the likelihood clustering or elsewhere, this track length dependence must be studied in much more detail.

4.5.4 Outlook

The results of this chapter have demonstrated the promising abilities of the likelihood clustering for track reconstruction. Based on the frequency spectrum and distributions of track dimensions, the reconstruction quality is largely comparable to that of a Katydid benchmark. There is an enormous potential to improve the likelihood clustering reconstruction solely by tuning the configurable parameters listed in Table 4.2; very little in the way of this optimization has been done already, and recall the initial motivation for the algorithm (long, faint tracks) is very different from the data used here to evaluate it. In addition, the likelihood metric as well as many other aspects of the clustering algorithm could be incorporated into the existing STF or another reconstruction. The discussion in this chapter and in the appendix is potentially a valuable starting point for future development of track and event reconstruction techniques as Project 8 moves toward Phases III−IV.

4.A Appendix

There remains a great deal of potential to further develop the likelihood clustering, and one of my chief goals when implementing the algorithm was to make it accessible for exactly that purpose. It is far less mature than other analysis techniques such as the STF and the track classifier (presented in the next chapter). Here, I describe the structure of the code and some ideas to explore in the future for any reader who is so inclined.

The code is located in the private `scripts` repository on the Project 8 GitHub account in the folder `zayas_analysis/likelihoodClustering/cppImp`. Documentation includes a list of external dependencies and compilation instructions.

4.A.1 Processor Operation

The clustering implementation has a similar structure to Katydid in its exchange of data objects between processors. This exchange is always mediated by a master processor called the "head": processors (a) send data to the head, and (b) access data from it. All necessary processors, data objects, methods to receive data, and the analysis chain exist in the head. The analysis chain is described by the `initiateProcessing()` method where the head iterates over the samples in a data file, calls the specified processor methods (akin to slot functions) and manipulates the data between them. Aside from a data object, these processor methods in general also send one of the following statuses to the head:

- OK: the processor has received and processed the data, and nothing else must be done with it. In this case the head will continue to the next sample from the data file.
- RETURNED: the processor has sent a data object back to the head. In this case, the head will proceed to the next processing step, usually with this data object as the input.
- ERROR: the processor has encountered a problem and cannot continue. The head will print an error message and abort the execution.

After the head calls a processor method, it checks the received status and takes the appropriate action. This behavior is accomplished concisely with the use of two macros, which are located in `logger.h`:

- PROCESS_SINGLE: given an input data object, a processor instance, and a method within that processor: call the method, check the status, and implement the behavior described above.
- PROCESS_MANY: given a vector of input data objects, process each one sequentially. If at least one object sends back a RETURNED status, then the overall status is RETURNED; the same applies for the ERROR status. Otherwise, every object returned OK and the status will be such.

The head operation is not strictly limited to the scope outlined here, and the `initiateProcessing()` method contains more than simply a series of method calls and status checks. Still, the use of this structure is advantageous because it allows the main processing method to be fairly concise, and outline well the analysis chain similar to a Katydid configuration file.

4.A.2 Future Work

The algorithm was initially developed to improve reconstruction of long, faint tracks while preserving the same ability for shorter and powerful tracks. The configurable parameters were studied and chosen with early Phase II tracks (as in Figure 4.1) as the primary concern, but as we prepare for the conclusion of Phase II we have all but entirely mitigated that concern with a better understanding of the detector. Very little has been done to re-optimize the likelihood clustering parameters in response; the quality of the results presented in this chapter on late Phase II data are in part a testament to the robustness of the algorithm. Still, its performance would undoubtedly benefit from an optimization campaign similar to the improvements made to Katydid since v2.13. These improvements were primarily directed toward (a) better short track reconstruction and (b) reduced overextension, both of which are at least slightly problematic in the current likelihood clustering results as well.

There are a number of outstanding questions and general inquiries related to these aspects of reconstruction and others:

- Is the cause of overextension related to the tolerance δ or another parameter? A large tolerance is demonstrably useful to reconstruct faint tracks, and in principle should not cause overextension when the cluster is reverted to its peak likelihood. It remains to be determined whether this is the case in practice. Other possible sources of the overextension include large $t_{\max}$ and block size $\langle k \rangle$. Both of these could likely be reduced by half or more without any sacrifice to short track reconstruction. Can a re-tuning of these parameters eliminate the problem of overextension with no drawbacks? Is there any tangible benefit to large δ in the final Phase II data?
- It is well understood that the likelihood of a track scales with its length in time. Is the likelihood metric an adequate tool to distinguish very short (< 5 bins) tracks from noise? Reconstruction of such short tracks yields a significant increase in the event rate, but this must be balanced against the risk of background events. By reducing the likelihood threshold from 40 to 25, we observed an event rate nearly triple that of Katydid but have good reason to believe at least some of this improvement is attributable to false events. How can the likelihood threshold and/or other quantities be optimally used to obtain the best possible event rate while maintaining a background-free expectation? How significantly does the boundary correction factor c influence short track reconstruction? Is there a benefit to considering 3-point clusters?
- Should some or all of the likelihood clustering be implemented in Katydid to work in tandem with the existing analysis there? The likelihood metric, for example, could be calculated for STF tracks and then utilized at a later stage. The core processors could be implemented in a very straightforward manner, as their functions in the existing code are directly transferable to the Katydid framework.

There are surely other questions to ask in addition to these which have not yet been realized. Phase II has importantly demonstrated to us that

CRES track reconstruction is never a solved problem, but instead a continuous challenge. There will forever be further potential to improve the existing scheme, new features to address, and new effects to understand. By exploring a wide range of projects such as the likelihood clustering and others, we can rise to this challenge and continue to strengthen our understanding of CRES reconstruction techniques as the project evolves.

Chapter 5

Track Topology Classification

We now turn our attention to the topic of secondary track analysis: reprocessing a spectrogram with knowledge of the tracks identified already by primary track finding. In Chapter 2, I briefly introduced the basics of the magnetic traps in Project 8 and the resulting electron dynamics. In this chapter, we will explore this topic in much greater detail and develop an extensive understanding of the numerous track characteristics observed in data. In particular, we will establish a set of three track types (called topologies) which each arise from different characteristics of the axial motion of a trapped electron. To extract the cyclotron frequency from these various topologies, we develop a series of machine learning (ML) models to classify unknown tracks with a topological label. This classification scheme can be utilized in conjunction with the phenomenological model developed in [14] to define the appropriate corrections between a track start frequency and the true cyclotron frequency; such corrections will be crucial to achieve eV-scale precision with CRES. The models are implemented as secondary track analysis steps in Katydid, which produce a classified frequency spectrum and have demonstrably high quantitative accuracy and robustness. After developing these models and studying the results, I outline the potential future applications of the classification scheme including energy corrections, ML-driven event building and the impact on tritium spectroscopy.

5.1 CRES Signal in a Waveguide

5.1.1 Pitch angle dynamics

Here, we summarize the analysis of electron dynamics from the phenomenological model developed by Ali Ashtari Esfahani, Gray Rybka and others in [14] to understand the resultant CRES signal characteristics. As mentioned in Section 2.1, an electron must have high transverse momentum and very small axial momentum to be contained within a Project 8 trap. We parameterize the axial momentum with the pitch angle, defined as the (instantaneous) angle between the momentum vector and the magnetic field:

$$\cos\theta(t) = \frac{\vec{p}(t) \cdot \vec{B}(t)}{p\ B(t)} \tag{5.1}$$

If we consider the magnetic field only in the axial direction, $\vec{B} = B_z\hat{z}$, the above equation reduces to simply:

$$p\ \cos\theta(t) = p_z(t) \tag{5.2}$$

Note that as in Section 2.1, we neglect the time dependence of the total momentum p over the relevant timescale since the fractional energy loss is small. An "ideal" electron exists only at the trap center with pitch angle $\theta(t) = 90°$ always, no axial momentum and emits a signal at true cyclotron frequency given by Equation 1.4. By contrast, a real electron explores a finite region of the trapping field as its z-momentum oscillates with time and thus so does the pitch angle via Equation 5.2. By definition, $\theta = 90°$ at the turning points of the axial oscillation since $p_z = 0$; at the center of the trap, the pitch angle and the magnetic field reach a minimum. Along with the trap geometry itself, it is this minimum pitch angle which governs the physical space explored by the electron. Going forward (but with the exception of the next subsection), we will refer to the pitch angle at the center of the trap simply by θ rather than the time-dependent instantaneous quantity in Equation 5.2.

5.1.2 Allowed pitch angle range

With Equation 5.2 and knowledge of the trap geometry, we can briefly study the range of pitch angles which allow the electron to be trapped. The minimum pitch angle is determined from the magnetic field at the trap boundary B_{max} and the maximum follows from the requirement that a trapped particle does not strike the waveguide wall. Both also depend explicitly on the birth position of the electron. From [19], the bounds for a rectangular waveguide as in Phase I are:

$$\theta_{\text{min}} = \cot^{-1}\left(\sqrt{\frac{B_{\text{max}}}{B(\vec{r})} - 1}\right) \tag{5.3}$$

$$\theta_{\text{max}} = \sin^{-1}\left(\frac{qB(\vec{r})}{\gamma m v}\left[\min\left(\frac{a}{2} - |x|, \frac{b}{2} - |y|\right)\right]\right) \tag{5.4}$$

where B_{max} is the maximum field at the trap boundary, x and y parametrize the guiding center of the cyclotron orbit, a and b are the cross-sectional waveguide dimensions, q and m are the electron charge and mass, γ is the Lorentz factor, v is the electron velocity, and $B(\vec{r})$ is the field magnitude at the electron initial position $\vec{r}$. With these bounds, the probability distribution function for the pitch angle is simply a truncated sinusoid:

$$P_{\theta,\vec{r}} = \frac{\sin\theta}{\cos(\theta_{\text{min}}(\vec{r})) - \cos(\theta_{\text{max}}(\vec{r}))} \text{ for } \theta_{\text{min}} \leq \theta \leq \theta_{\text{max}} \tag{5.5}$$

where the PDF of the momentum vector is assumed to be isotropic. For the Project 8 apparatus, both θ bounds are nearly 90° for every initial position $\vec{r}$ and the overall fraction of electrons which become trapped is 5.2% [19].

5.1.3 Power spectrum with pitch angle considerations

The phenomenological model uses piecewise functions to approximate Project 8 magnetic trap geometries and analytically describe the dynamics of trapped electrons. The resultant radiated power spectrum is calculated from the Poynting vector in the axial direction and can be decomposed as a sum of waveguide modes. For a single mode denoted by λ, we take advantage of the quasi-periodic motion of the electron in the trap to express the power averaged over the axial period [14]:

$$\begin{aligned}\tilde{P}_\lambda^\pm(\omega) = P_{0,\lambda} \sum_{n=-\infty}^{\infty} \left| a_n \left(\pm \frac{\Omega_0 + n\Omega_a}{v_{p,\lambda}} \right) \right|^2 \\ \times \left[\delta(\omega - (\Omega_0 + n\Omega_a)) + \delta(\omega + \Omega_0 + n\Omega_a) \right]\end{aligned} \tag{5.6}$$

where the amplitude $P_{0,\lambda}$ depends on the waveguide geometry, $v_{p,\lambda}$ is the mode phase velocity, and a_n are amplitudes dependent on the shape of the magnetic trap. The ± superscript refers to the $\pm z$ directions. This equation describes a power spectrum with a comb-like structure across many frequencies: the main-band (average cyclotron) frequency Ω_0 and a collection of sideband frequencies shifted by integer multiples of the axial oscillation frequency Ω_a. All of Ω_0, Ω_a and the band amplitudes a_n depend on the pitch angle and the trap geometry. Figure 5.1 shows a Phase II event which illustrates the behavior of Equation 5.6: the signal takes the form of multiple parallel tracks corresponding to the different values of the frequency band order n. We call this structure a multi-peak track (MPT). Note that (1) the relative power of each track within a MPT is not fixed but pitch-angle dependent, and (2) the total radiated power is encoded in the slope of each track, i.e. the change in frequency (energy) with respect to time.

5.1.4 Waveguide reflector effects

In the Phase I and Phase II-krypton apparatuses, the waveguide also contained a reflector opposite the antenna to boost the received signal power. This creates an interference effect between incident and reflected waves at the antenna position, further complicating the observed power spectrum. This interference effect is energy (wavelength)-dependent and can substantially enhance or suppress power in different frequency bands. The averaged power spectrum with this consideration is given by Equation (45) in [14], reproduced

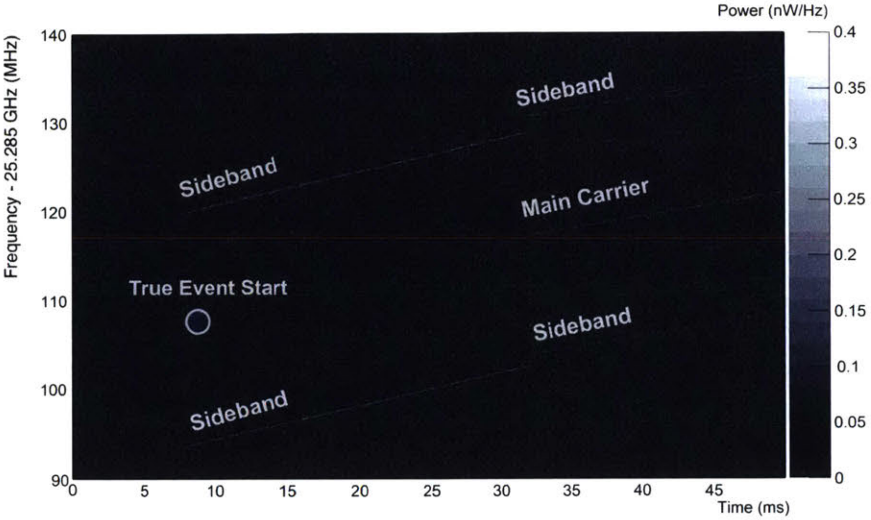


Figure 5.1: A multi-peaked electron event featuring five tracks in a bathtub trap configuration. The electron is born in the trap around 7 ms with a pitch angle that favors only 1st-order sidebands. After scattering with a residual gas molecule around 32 ms, the pitch angle changes to favor both the main carrier and sidebands.

here:

$$\begin{aligned} P_\lambda(\omega) = 4P_{0,\lambda} \sum_{n=-\infty}^{\infty} & \left| a_n \left(\frac{\Omega_0 + n\Omega_a}{v_{p.\lambda}} \right) \right|^2 \\ & \times \cos^2 \left[(z_t + l) \frac{\Omega_0 + n\Omega_a}{v_{p.\lambda}} \right] \\ & \times \left[\delta(\omega - (\Omega_0 + n\Omega_a)) + \delta(\omega + \Omega_0 + n\Omega_a) \right] \end{aligned} \tag{5.7}$$

where $z_t + l$ is the distance between the trap center and the reflector. Once again, the pitch angle dependence is clear through the detected frequencies Ω_0 and Ω_a which appear in both the amplitude coefficients a_n and the cosine term which describes the interference effect. Considerable discussion about the calculation of a_n for some trap models is given in [14].

Figure 5.3 illustrates concretely this pitch-angle-dependent power spectrum for a 32 keV electron in a Project 8 bathtub trap; both the total power and the individual band power for $n \leq 2$ are included. The individual band power

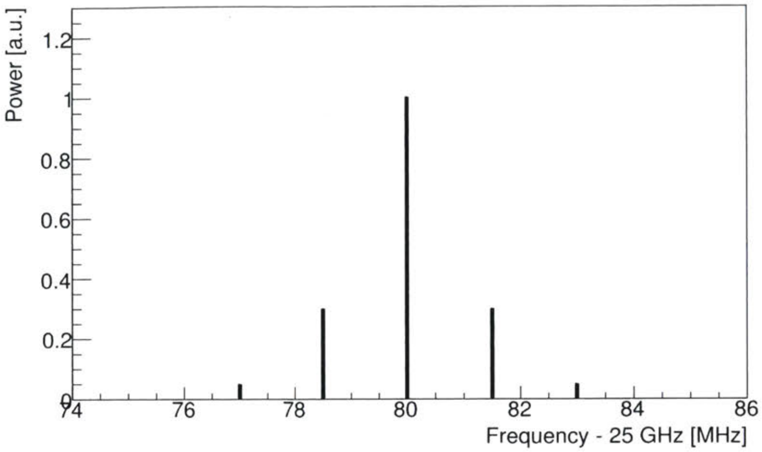


Figure 5.2: Illustration of the power spectrum comb structure described by Equations 5.6 and 5.7 where a main carrier at 80 MHz and several sidebands signals are observed. The amplitude of each peak is energy-dependent due to the waveguide short. Adopted from [14].

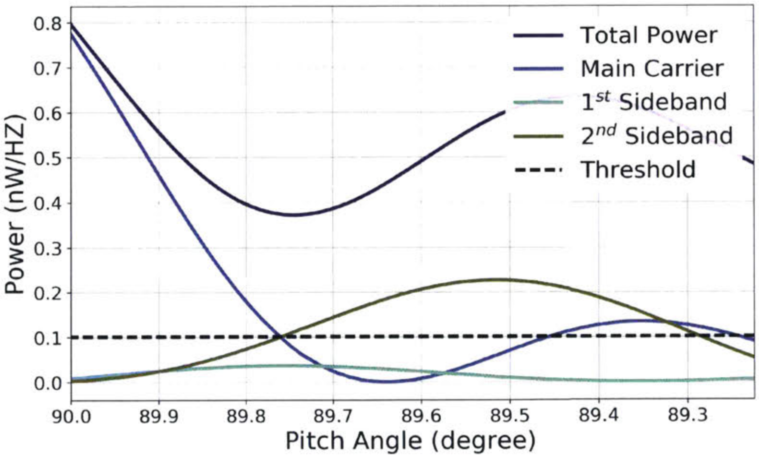


Figure 5.3: Power vs. pitch angle for a 32 keV electron signal in a bathtub trap with a shorted rectangular waveguide. An arbitrary detection threshold at 0.1 nW/Hz is included to aid in the illustration. Only the main carrier and 2^{nd}-order sidebands are above this threshold while the 1^{st}-order sideband is suppressed.

describes the power in a single track, and thus to analyze the effect of the pitch angle on track reconstruction we must consider this band power and not the total power. Since high-order bands never have sufficient power for reconstruction, we generally restrict our analysis to only the mainband ($n = 0$) and one detected sideband order ($n = 1$ or $n = 2$ depending on the reflector interference effect). The dashed line in Figure 5.3 depicts an example threshold for detection/reconstruction which is met only by the mainband and the $n = 2$ sideband for separate ranges of the pitch angle. Imposing such a threshold reveals a short list of detectable track types identified by (a) the band order, and (b) the allowed pitch angle range. Although the threshold in Figure 5.3 is for illustrative purposes only, it was chosen to match qualitatively these detectable regions with those observed in data:

1. Main carrier high pitch angle: closest to 90° for $n = 0$.
2. Main carrier low pitch angle: far from 90° for $n = 0$.
3. Sidebands: a single pitch angle range for $n = 2$.

Since the interference effect is energy dependent, we expect the specific structure of detectable pitch angles to vary with frequency. In the 32 keV case illustrated here, we expect multi-peak tracks with a single track (main carrier high pitch angle), with three tracks (main carrier low pitch angle and sidebands), and with two tracks (sidebands only); indeed, these are the types of multi-peak tracks observed in Phase I.[1] For 17.8 keV electrons in Phase I, the main carrier is almost completely suppressed and we detect only the $n = 1$ sidebands; this exemplifies the powerful influence of the reflector.

5.1.5 Radial gradient perturbation

So far, we have treated the axial motion as independent of the (x, y) plane on the basis that the magnetic field varies only with z, i.e. $\vec{\nabla} B$ is always

---

[1] Also recall the event in Figure 5.1 which changes pitch angle and MPT topology due to a scatter. This event showcases the (perhaps obvious) problem that in a MPT with no mainband, the true start frequency must be inferred.

parallel to $\vec{B}$.[2] However, to improve our description of the electron dynamics we must consider a small radial gradient of the form $\vec{\nabla} B \times \vec{B}$. This causes the guiding center of the cyclotron orbit to precess slowly (compared to the axial motion) in x and y, effectively perturbing the magnetic trap profile $B_z(z)$ with a slow time dependence. A model proposed by Gray Rybka [20] induces this precession with a small asymmetric tipping of the trap coils at an angle ψ from the axial direction. The result is a periodic drift in the axial frequency Ω_a:

$$\Omega_a'(t) \approx \Omega_a \left(1 + \frac{r}{l_1} \sin\psi \sin(\Omega_m t)\right) \tag{5.8}$$

where r is the radial position of the electron, l_1 is the characteristic trap length and Ω_m is the drift frequency.

The effect of this precession on the track signal now becomes clear: if the axial frequency varies sinusoidally, so will the frequency of the $n > 0$ (sideband) tracks. This oscillation has been observed in Phase I data, where it manifests as a track with an appreciable "width" in frequency; Figure 5.4 compares an example sideband track with this quality to a mainband without it. Since the observed period of precession is $\sim$100 μs, which is comparable to the DFT length (40.96 μs), this oscillation can be seen to some extent directly in the spectrogram. Figure 5.5 shows the same quality in a simulated sideband track with the coils tipped $\psi = 1°$ off-axis.

This qualitative property which distinguishes sideband tracks, illustrated in Figure 5.4(b) and modelled by the radial gradient contribution, represents one of the primary motivations for a machine learning approach to track classification.

5.1.6 Summary

At this point, it is useful to review the understanding we have gained about CRES signals in a waveguide with pitch angle considerations.

- The power spectrum at the antenna is calculated from the phenomenological model, revealing a comb-like structure centered at a mainband

---

[2] In fact, the assumption that $\vec{\nabla} B \times \vec{B} = 0$ contradicts the Maxwell equations and thus is clearly unphysical.

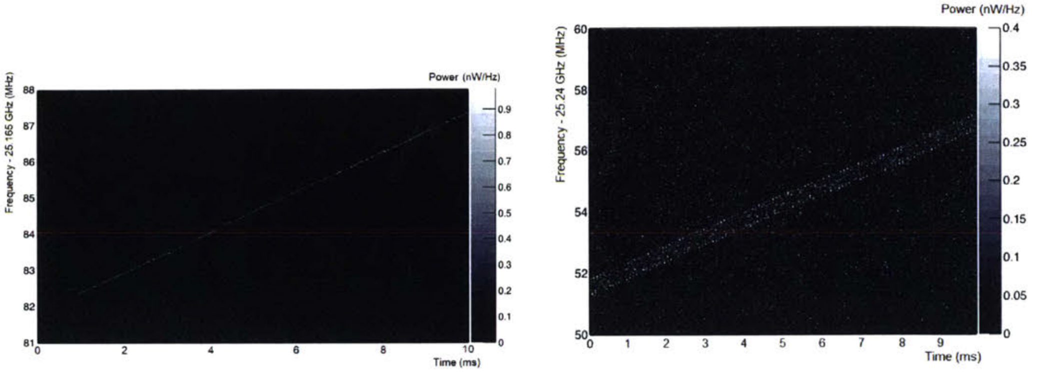


Figure 5.4: Spectrograms of a typical main carrier (left) and sideband track (right). While the main carrier track is sharply peaked in power, the sideband signal oscillates around the true frequency, depositing power in a range of frequency bins. The sideband power is largest at the extrema of the oscillation, since the particle spends the most time there.

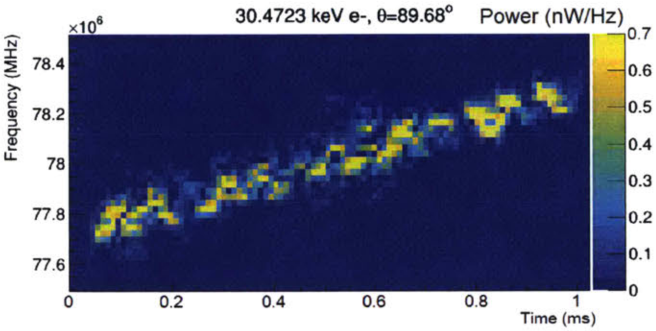


Figure 5.5: Simulated sideband track from a bathtub trap exhibiting oscillation width. Adapted from [21].

frequency Ω_0 with sideband power separated by the axial oscillation frequency Ω_a due to the Doppler effect. The mainband frequency is the average cyclotron frequency over the accessible trap region. Both the associated frequencies and the power in each band are pitch-angle-dependent.

- With a reflector opposite the antenna, the power spectrum gains an interference pattern described fully by Equation 5.7. This effect is manifestly energy dependent and has the ability to greatly enhance or suppress the various individual band powers.

- The pitch angle dependence of each band power creates distinct regions in the phase space of θ and the band order n with sufficient power for track reconstruction. The tracks in Phase I data may be categorized into three such regions, consistent with the phenomenological model prediction.
- Imperfections in the trap give rise to a considerable radial gradient in the magnetic field profile, which has been shown to induce a precession of the sideband track frequencies. This precession has been observed in data and simulation.

With all of these effects, the cyclotron frequency no longer corresponds to any actual starting frequency of a track. To accurately determine the electron energy, we require a more sophisticated approach which utilizes the phenomenological model, pitch angle information, and topological track labels. In the remainder of this chapter I will outline the scheme to accomplish this, and detail the current classifier models which make important early steps toward that end.

5.1.7 Energy corrections

The kinetic energy determination effectively amounts to a correction from the observed main carrier frequency Ω_0 to the cyclotron frequency Ω_c. Extracting the pitch angle information is critical to make this frequency correction, and with knowledge of Ω_0 it may be done in two ways:

1. *Axial Frequency*: for MPTs with a mainband and one or more sidebands, the axial frequency Ω_a is the frequency difference between the mainband and a sideband track divided by its order n. Ω_a may also be determined the same way from an event with multiple sidebands but no mainband. In Phase I, these cases comprise a minority of our data at about 10%.
2. *Track Slope*: for MPTs with only a mainband, the pitch angle may be extracted from the track slope which is proportional to the total radiated

power in Equation 5.7. Such cases comprise the majority of the data used for this work at about 90%.

The second method listed above has an ambiguity: in general, the track slope alone does not uniquely determine the pitch angle (this is evident from Figure 5.3). To resolve this issue, we must also differentiate between mainbands of the high and low pitch angle regions. Our task is then to assign every track an appropriate label from the list in subsection 5.1.4: mainband high pitch angle, mainband low pitch angle, or sideband. These three topologies encompass all of the Phase I and Phase II data as it is presently understood. Such a classification makes possible the extraction of a unique pitch angle from Ω_0 and the track slope and/or the axial frequency. The pitch angle information in turn yields the true cyclotron frequency and kinetic energy.

We approach this task with a machine learning model which uses a supervised learning method for classification. The overarching goal of the classification program is to use only those track features which are intrinsic to the signal itself and, through their inclusion, have the capability to improve the accuracy and robustness of the signal identification. As Project 8 moves to a tritium source, such a classification scheme will be vital to make an accurate measurement of the continuous spectrum and reach meaningful conclusions about the endpoint. In the remaining sections, I will develop the classification scheme and present the results on krypton Phase I data at all three energies of interest. The next steps of pitch angle calculation and the resultant energy correction are not yet implemented, but are of course a primary focus of future work to realize the full potential of track classification. We will also discuss the future impact of a more-developed classification process in the context of tritium endpoint sensitivity and full event reconstruction.

5.2 Feature Extraction

To develop a classification model, we must first determine a set of calculable track parameters which have some ability to discriminate between different topologies. If the distribution of a parameter among tracks of one topology is substantially different from that among tracks of a different topology, we say this parameter is a useful classification feature. An ideal feature would produce distributions which are completely separated, and thus perfectly determine the classification label with an ordinary cut. However, of course the distributions for any practical feature will overlap to some extent. The application of machine learning techniques for classification in essence extends this notion of an ordinary cut on one feature to a decision function on many features, each of which has partial discrimination power.

The track features which we use for classification result from two separate analysis techniques: primary track finding and the rotate-and-project algorithm. These methods give us a total of 14 parameters which we utilize in training the ML classification models. In this section, we describe the calculation of these parameters.

5.2.1 Primary track parameters

Primary track finding is the process of collecting high-power spectrogram bins into linear track signals, described in Section 3.2. We have not made any additional advances to the track finding procedure for this work. At its conclusion, several parameters are formally calculated to describe each track candidate; these include the slope, start frequency, time length, and many others. We utilize the following three quantities as inputs to the classification model:

- `TotalPower` (W/Hz): the sum of power spectral density values in all bins which comprise the track cluster.
- `TrackSlope` (Hz/s): the slope of the track as extracted from regression analysis and a Hough transform.
- `TimeLength` (s): the difference between the track end time and start time.

Recall that the slope of a track is directly proportional to the total power radiated by the electron, and the slope and individual track power together determine the pitch angle information as illustrated in Figure 5.3. Thus, the correlation between slope and track power has strong discrimination power between regions of high and low pitch angle. Figure 5.6 illustrates this correlation for 32 keV electrons and shows a clear separation of the two mainband populations which agree with the phenomenological model prediction. This provides a clear motivation for the use of these two features as inputs to the classification model.

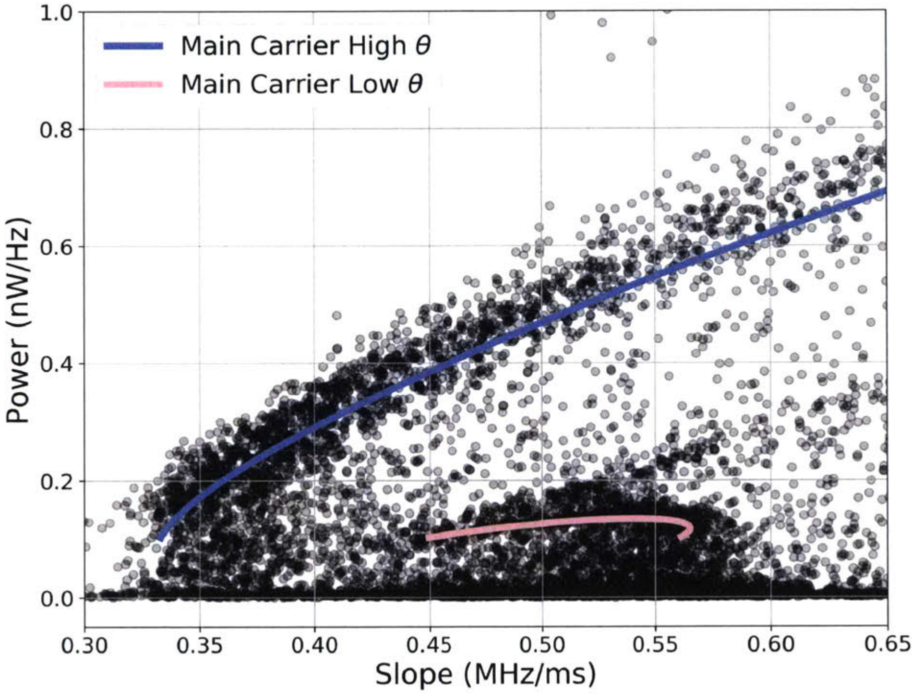


Figure 5.6: Power and slope correlations in Phase I 32 keV bathtub trap tracks (black scatter) with a start-time cut of 0.5 ms around the pre-trigger time. Fits using the phenomenological model for high (blue) and low (pink) pitch angle carriers demonstrate the separation of mainband populations with disjoint pitch angles.

The utility of the track length is primarily based in its correlation with the track power as well. Mainband tracks in general have a strong profile with power concentrated across one or a few frequency bins at each point along the track. By contrast, sideband tracks often have power distributed across many points due to the axial frequency precession and contain less power overall in the case of the 30 and 32 keV peaks. Consequently, the track power can have considerable dependence on the number of points which comprise the track, and the track length helps to bolster the discrimination ability between all three types in conjunction with the slope and power.

5.2.2 Rotate-and-Project distribution

Aside from these track parameters, the axial frequency smearing creates a clear qualitative difference between the appearance of sideband and mainband tracks in a spectrogram. To extract a set of parameters which quantify this difference, we first simplify the problem. After primary track finding, sparse spectrograms with a known track are reprocessed with a "Rotate-and-Project" operation, where they are effectively projected along the axis perpendicular to the track. This reduces the useful information in the two-dimensional spectrogram to one dimension (the projected spectrum).[3] The precise procedure is as follows:

1. The known track is characterized by a slope q and intercept f_0.
2. The full spectrogram is reduced to a sparse spectrogram of only points with SNR > 4.0 and which lie within the time bounds of the track. These points will be described by their time and frequency coordinates (t_j, f_j).
3. The projected spectrum is calculated as a function of the intercept, which we call β and which sweeps the range $f_0 \pm \Delta f$ in discrete steps of $\delta\beta$ (both Δf and $\delta\beta$ are runtime-configurable parameters):

$$s_k = \sum_j \exp -\frac{(f_j - qt_j - \beta_k)^2}{2\sigma^2} \tag{5.9}$$

---

[3] It should be noted that with this projection, the oscillatory nature of the sideband track (in time) is partially lost. Some effort has been made to better utilize this oscillation via Fourier analysis, but the results have been mostly inconsistent. The rotate-and-project procedure still captures the extent, or "width" of the track in frequency.

where $\beta_k = f_0 - \Delta f + k\,\delta\beta$ and σ is another runtime-configurable variable which describes the resolution of the spectrum.

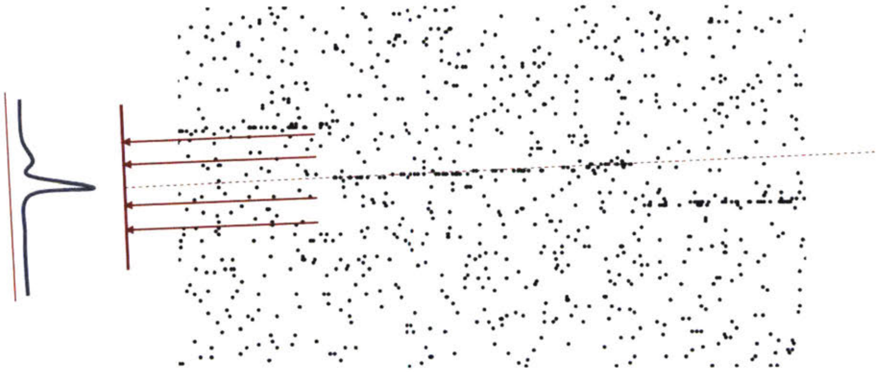


Figure 5.7: Illustration of the rotate-and-project operation on a sparse spectrogram.

At a minimum, σ should reflect the inherent uncertainty in each point location which is roughly the bin size; this way, the spectrum is not largely affected by how precisely the choices of Δf, $\delta\beta$, or the reconstructed value f_0 coincide with the discrete binning of the spectrogram. σ can also be much larger than the bin size, and the projected spectrum gains sensitivity to structures which span a similarly larger bandwidth; however to retain good sensitivity to the sharp mainband tracks, we keep σ similar to the bin width. There is no advantage to matching it exactly with the bin size, so for convenience we choose $\sigma = 50$ kHz which is approximately 2 bins. For the step size, we choose $\delta\beta = 25$ kHz, or half the resolution σ and approximately 1 bin. The only requirements on the sweep range $2\Delta f$ are that it should be many times larger than the frequency bin size, and at a minimum large enough to capture the full amplitude of the sideband oscillation ($\sim$1 MHz). We choose $\Delta f = 4$ MHz.

Figure 5.8 shows typical projected spectra corresponding to a mainband and a sideband track. The qualitative differences between the two remain clear: sideband spectra typically feature a wide doubly-peaked structure, contrasted with the sharp and high-amplitude profile of the mainband spectrum. The sideband spectrum amplitude is largest near the edges of the signal region because the axial motion is slowest at its turning points, thus depositing more power per bin; this effect can also be seen in the spectrogram (recall Figure 5.4). This comparison confirms the promising discrimination power of features extracted from the projected spectrum. Next, we use the ROOT library TSpectrum[4] to characterize peaks in the projected spectrum. This library fits a linear background $b_k = ak + b$ and labels a point k as a peak if it meets all of the following criteria:

1. Value is at least twice that of the background level: $s_k \geq 2\ b_k$.
2. Peak amplitude meets or exceeds a minimum fraction r of the highest peak: $s_k - b_k \geq r\ \max_j\{s_j - b_j\}$.
3. Value is a local maximum within m steps: $s_k > s_j\ \forall\ j \mid 0 < |j-k| \leq m$. The frequency range corresponding to m steps is $m\ \delta\beta$.

We choose $r = 0.4$ and $m = 5$. The values of these and the other configurable parameters from Equation 5.9 are listed in Table 5.1. Once the peak locations are determined, the full spectrum is fit to a sum of n Gaussian functions where n is the number of peaks found. Only a handful of tracks in our studies produced a spectrum with 3 or more peaks, and none with more than 6.

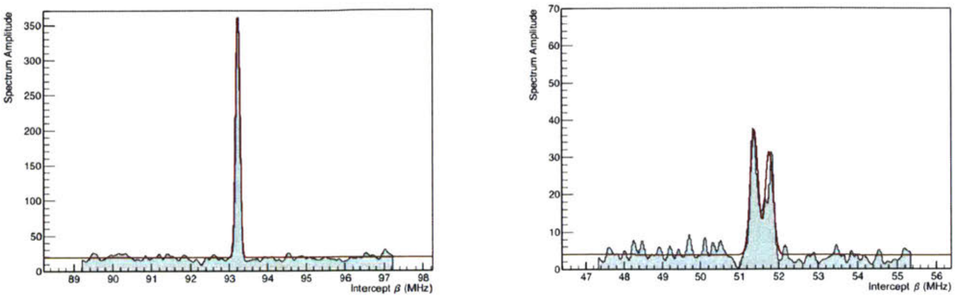
Parameter	**Value**
Δf	4 MHz
$\delta\beta$	25 kHz
σ	50 kHz
r	0.4
m	5

Table 5.1: Standard parameter values for rotate-and-project analysis.

From the spectrum s_k and the results of the Gaussian fit, we extract a total of 11 additional parameters for use with the classifier:

---

[4] `https://root.cern.ch/doc/v608/classTSpectrum.html`



(a) Mainband track projected spectrum (b) Sideband track projected spectrum

Figure 5.8: Spectra from rotate-and-project analysis for a typical (a) mainband track and (b) sideband track. The mainband track is sharply peaked, whereas the sideband track is spread over a wider frequency range and doubly-peaked. Each spectrum also shows the associated Gaussian fit for comparison. Note the y-axis in (b) is scaled down in comparison to (a). The sideband used for (b) is the same track as illustrated in Figure 5.4.

- `Average`, `RMS`, `Skewness`, `Kurtosis`: First four statistical moments of the spectrum s_k. `Average` and `RMS` are in units of MHz, and `Average` is shifted by f_0 so that 0 corresponds to the center of the spectrum.

- `MeanCentral`, `SigmaCentral`, `NormCentral`, `MaximumCentral`: Extracted fit parameters of the Gaussian with mean closest to f_0 (the most central peak). `MeanCentral` is shifted by f_0 as described above, and `MaximumCentral` $= b_0 + (2\pi)^{-1}$ `NormCentral` / `SigmaCentral` where b_0 is the background level at the peak location. Both `MeanCentral` and `SigmaCentral` are in units of MHz.

- `NPeaks`: Number of peaks found by TSpectrum for the Gaussian fit.

- `RMSAwayFromCentral`: RMS $(\langle s^2 \rangle - \langle s \rangle^2)^{1/2}$ of points greater than 3 times `SigmaCentral` away from the most central peak.

- `CentralPowerFraction`: average value $\langle s \rangle$ of bins within 3 times `SigmaCentral` of the most central peak divided by the overall average of the spectrum.

In the event that the Gaussian fit fails to converge, all of the parameters which depend on it are obviously unreliable and some can be undefined. To circumvent this, we simply remove any tracks from the analysis which have an unsuccessful fit, or which have any parameters undefined as a result of an improper fit. These represent between 5−10% of all tracks in various data sets we have used. Combining these with the track parameters discussed in the previous subsection, we have a total of 14 parameters to use for machine-learning-based classification. The slope, power, and track length have the ability to distinguish between all three track topologies based on the phenomenological model; the projected spectrum provides an additional 11 parameters which distinguish sidebands from the two mainband topologies. The discrimination power of some parameters is illustrated concretely in Figures 5.9 and 5.10: the former shows distributions of 30 keV tracks over the space of one parameter, and the latter over 3D slices of the feature space. Both show a clear difference (partial separation) between the labelled track types which indicates good predictive power that a ML algorithm can extract. Of course, it is the goal of the ML classification scheme to utilize the full 14 dimensions of the feature space and produce the best possible separation between the track topologies. In the next sections, we discuss the implementation and results of this new analysis.

5.3 Learning Sets: Data Selection and Statistics

The data utilized for development of the ML model is from a Phase I campaign in 2015 which spans all three energy regions of interest for the krypton emission lines: 17 keV, 30 keV, and 32 keV. It is partitioned into three important subsets of interest: training, cross-validation and test sets. The training set is used to fit the ML model during its optimization campaign, the cross-validation set is used to optimize the classifier's hyperparameters[5], and finally the test set is used to assess the accuracy and robustness of the final classifier.

---

[5] Hyperparameters are model-intrinsic parameters which govern the learning process itself and thus must be fixed before learning with the training set begins.

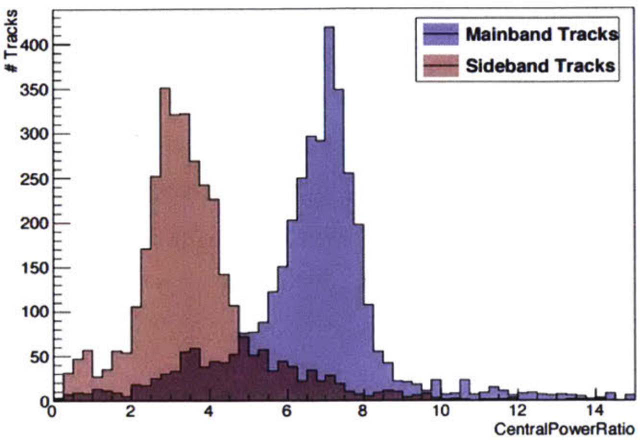


Figure 5.9: Training point distribution for the central power ratio r where $r/(1+r)$ is the central power fraction. The distribution is clearly not the same between mainband and sideband tracks, which indicates the classification power of this parameter.

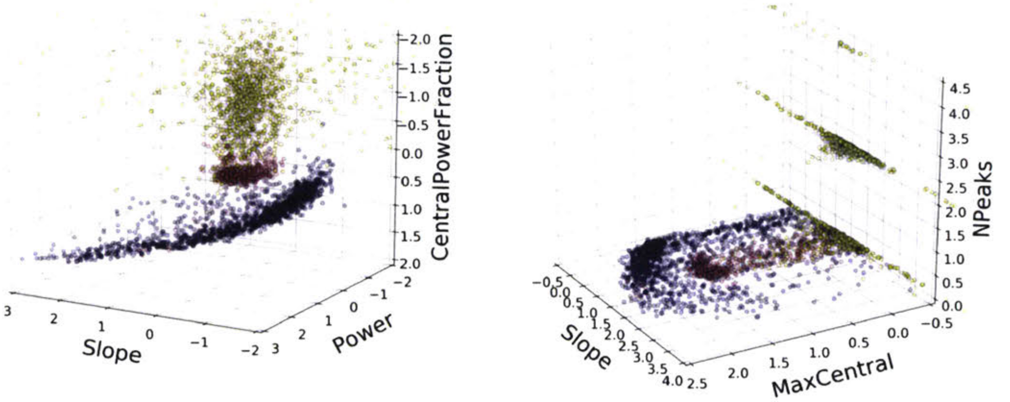


Figure 5.10: Example 3D slices of the classification feature space. Here the class separations may be seen by eye. The blue scatter are mainband high pitch angle, the pink mainband low pitch angle and the yellow sideband tracks. All features are scaled (unitless) as described in subsection 5.4.2 for use with the SVM training model.

We require the true classification label of every track to be known for these learning sets, but they are entirely comprised of real CRES event data and no simulated data.[6] Consequently, we cannot determine the true labels with perfect accuracy or else the ML model would serve no real purpose. However, the phenomenological model and simulations make predictions about the signal behavior which confidently agree with our observations in data and thus inform the selection process.

For data selection, we first make a cut on the track start time. The Phase I DAQ system acquires triggered acquisitions of a 10-millisecond duration with a pre-trigger time of 1 millisecond, such that a strong track which promptly triggers the acquisition has a start time near 1 ms. By contrast, a faint track can persist for some time before triggering and the eventual reconstructed start time is well before the pre-trigger time. Aside from the obvious problem that such a track which initially evades the trigger may also evade good reconstruction, many are reconstructed perfectly well but with a start time at or very near $t = 0$; this of course indicates the true starting point lies before the acquisition window, and the reconstructed start frequency is automatically unreliable. We keep only those tracks with a start time between $0.75-1.25$ milliseconds, which we are confident implies the acquisition was promptly triggered and the track reconstructed well.

The remaining tracks are given topological labels based on the predictions of the phenomenological model. For tracks near the appropriate krypton peak (presumed main carrier), the correlation between the slope and start frequency is used to assign a label of either high or low θ. Similarly, tracks which are far from the krypton peaks (sidebands) are labelled using the phenomenological prediction for the relative track power with respect to frequency. In both cases, only those tracks which agree sufficiently well with the phenomenological prediction are included and outliers are discarded.

Lastly, as briefly mentioned in the previous section, we remove tracks for which the rotate-and-project analysis fails to properly converge on a multi-peak fit. We also discard severe outliers in each of the 14 parameters by

---

[6] Producing simulated data in a manner and quality compatible with this type of machine learning application is one of the more significant current analysis efforts.

keeping only those with within 5σ of the respective population; we assume the rest come from poorly reconstructed tracks not suitable for training. This yields a total of 7347 tracks for the ML model. This learning set is split in a $2:1$ ratio for training (training and cross-validation) and testing, respectively. The split is done via a random shuffle using a stratified method which keeps the relative size of labelled populations in mind. This avoids biasing the training set with a user-defined splitting strategy which may increase or decrease the representation of any one class during training.

We develop a total of three ML models with this learning set. The first utilizes a Binary Decision Tree (BDT) and only the rotate-and-project parameters to perform two-label classification: mainband and sideband. This classifier was developed before the phenomenological model. The second classification model incorporates all 14 features and a Support Vector Machine (SVM) algorithm, and trains on only 30 and 32 keV data. The third model utilizes the same features and SVM, and trains simultaneously on all three krypton lines: 17, 30, and 32 keV.

5.4 Model Description

5.4.1 Binary Decision Tree overview

The first model we will discuss uses a Binary Decision Tree to discriminate between two labels: signal (mainband) and background (sideband). This model was implemented when our understanding of the track parameter correlations – especially slope and power – was very limited, and the only method of sideband discrimination was a harsh power cut. The BDT uses only those parameters derived from the rotate-and-project analysis. The learning process iteratively constructs a decision tree with all available parameters, as follows:

1. For each parameter, (a) sort all training points by this parameter and (b) determine a threshold value which best separates signal from background.

2. Select the parameter which achieves the overall best discrimination accuracy.

3. Create two daughter nodes in the decision tree with this one-parameter threshold.

4. Recursively repeat (1)–(3) for each daughter node until a termination condition is reached.

The result is a decision tree which consists of an ordered sequence of one-parameter cuts, and returns a binary classification label. Note that a path through the tree may use the same parameter at many decision points, so the depth of the tree is not limited by the number of parameters. If a node achieves perfect separation (all points have the same classification label), the termination condition is automatic. Otherwise, a node may be terminated if the separation improvement from its parent is negligible or if it exceeds a maximum depth in the tree (the former case is overwhelmingly more common in this application). When a node is terminated, it is assigned the appropriate binary label.

5.4.2 Support Vector Machine overview

For a detailed introduction to the SVM algorithm we refer to [22, 23]. In short, the SVM is a model which minimizes a unique loss function to obtain a fit parameter vector $\vec{w}$ for each classification label. The loss function is calculated over m training points where each point is represented by a 14-dimensional vector of the extracted features, and the fit vector exists in the same 14-dimensional feature space. The SVM determines a hyperplane in this feature space which maximally separates training points of the correct label from those of any other label (called a One-Vs-Rest approach); $\vec{w}$ is the vector normal to this hyperplane. Classification is then performed by projecting data points onto each fit vector to obtain a set of probabilities, and the label corresponding to the highest probability is assigned.

To aid in minimization of the loss function, learning points are rescaled such that each feature has a mean of 0 and a standard deviation of 1 (now dimensionless). This aids the ability of the SVM to compute reliable distance measures from data points to the decision hyperplane with the scale of each dimension treated equally.

5.4.3 Cross-validation and hyperparameter optimization

The BDT has no hyperparameters, so all of its training set is utilized for the loss minimization stage. The SVM model features two hyperparameters which are, by definition, external configurations that cannot be estimated from data. The two hyperparameters are called C and γ; qualitatively, C encodes the leniency of the SVM in trading misclassification for model stability and γ dictates the influence of training points defining the decision boundary to the rest. C has considerable impact on overfitting of the model, since a large value gives little weight (penalty) to high order terms in the SVM loss metric allowing them to dominate the minimization strategy. On the other hand, large γ can introduce a significant bias since the sphere of influence of the support vectors themselves becomes sharp around the boundary and less so at other points while a small γ usually leads to low variance as the sphere of influence of the support vectors is large.

To optimize these hyperparameters, we explore the 2D parameter space via a randomized grid search technique with exponential distributions[7] for each parameter. The parameters are randomly drawn for a total of 100 instances, and the instance with the largest accuracy score:

$$\text{accuracy}\,(y, \hat{y}) = \frac{1}{m}\sum_{j=0}^{m-1} 1\left(\hat{y}^{(j)} = y^{(j)}\right) \tag{5.10}$$

is selected. Here, $\hat{y}^{(j)}$ is the predicted label for point $\{x^{(j)}, y^{(j)}\}$. To avoid overfitting, we perform this optimization using a K-folds cross-validation strategy. For every (C, γ) tuple, the learning set is split into K smaller sets (folds). Each fold acts as the validation set exactly once, and the model is trained on the remaining $K-1$ folds. We thus obtain K individual accuracy scores, and the average is defined to be the overall accuracy of this (C, γ) instance. This

---

[7] The chosen distribution scale for C is 100 and for γ is 0.1.

optimization yields the following SVM hyperparameters:

$$C = 108.01 \tag{5.11}$$

$$\gamma = 2.9470 \times 10^{-3} \tag{5.12}$$

These values are indicative of a "smooth" model ($\gamma \ll 1$) which captures little of the data complexity in the feature space, but is balanced by the large value of $C \gg 1$ which allows for highly nonlinear terms in the loss metric minimization, recovering some complexity in the decision plane.

5.4.4 Receiver Operating Characteristic

The Receiver Operating Characteristic (ROC) curve is an important metric in evaluating each ML model. The ROC measures the robustness of the model across all classification thresholds by comparing the proportions of true and false positives for each classification label. If the threshold for a certain label is extremely low, the rate of both true and false positives should be high (nearly every track is assigned this label), and thus the proportion of each is nearly 100%. Similarly, with a very high threshold the proportion of each is close to 0%. The ROC curve, a 2D graph of true vs. false positives, then always includes the points $(0, 0)$ and $(1, 1)$. It is of course the behavior between these limits which is important:

- A random classifier will always have the same proportion of true and false positives, regardless of the classification threshold. It has no bias whatsoever in its assignment of uniformly random labels. The ROC curve of such a classifier is a straight line passing through the limit points $(0, 0)$ and $(1, 1)$.

- A perfect classifier will create a threshold which fully separates each track topology from the rest, corresponding to a perfect rate of true positives and no false positives $(0, 1)$. Any more permissive threshold will allow false positives, but retain the perfect rate of true positives. This ROC curve then consists of two segments, from $(0, 0)$ to $(0, 1)$ and from $(0, 1)$ to $(1, 1)$.

- A practical classifier will always have a ROC curve in between those of the two items above. A robust classifier will maintain a high true positive rate even with a reasonably low false positive rate.

Figure 5.11 illustrates three ROC curves corresponding to the two limits and one practical case listed above. To quantitatively measure the robustness of the model, we consider the area under the ROC curve (AUROC). In addition, we construct comprehensive ROC curves by two averaging methods:

- Micro-averaging: count true and false positives in all classes together as a single problem, and then compute the resulting ROC curve.
- Macro-averaging: construct the ROC for each classification type separately, and then simply average the results.

With the accuracy score, optimized hyperparameters, and ROC curves, we are now prepared to evaluate the performance of each classification scheme. In the next section, we compare these metrics and the associated new frequency spectra.

5.5 Supervised Classification: Results

Here we report the results of each track classifier described in Section 5.4. All three models distinguish different track topologies with high accuracy and robustness, leading to a great improvement in the frequency spectrum across all energy ranges.

5.5.1 BDT model

The BDT classifier was trained simultaneously on 30 and 32 keV data, and the trained decision function is implemented in Katydid with the TMVA library.[8] Tracks are processed with the rotate-and-project algorithm and then input to the decision function which appends a classification label. The classified 32 keV spectrum is shown in Figure 5.12 and the ROC in Figure 5.13. The

---

[8] `https://root.cern.ch/tmva`

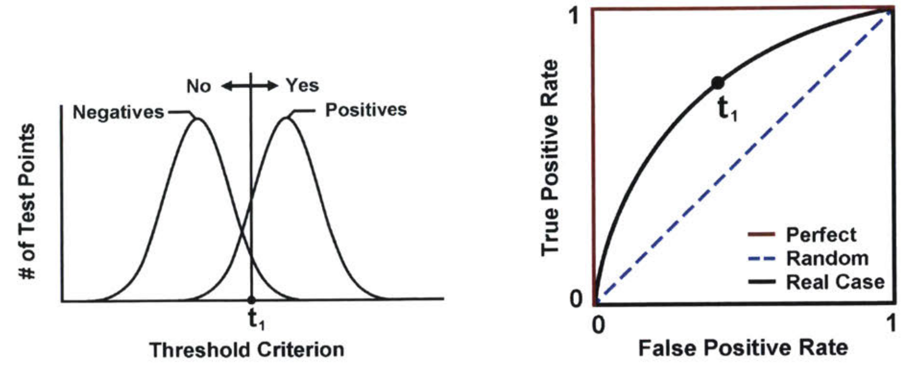


Figure 5.11: (Left) Positive and negative overlapping test populations subject to a sliding minimum threshold t_1 for classification. (Right) Illustration of resulting ROC curves for three cases: a perfect classifier (red) with area 1.0, a random classifier (blue) with area 0.5, and a real-case of intermediate strength (black).

classification accuracy on the training set is 95.0% for sidebands and 81.8% for mainband tracks. Although this model does not utilize the track slope, power, or length, these intrinsic track parameters of course have an indirect influence on the calculation of the rotate-and-project quantities. The comparatively reduced accuracy of mainband classification may be in part attributable to the difference in those track parameters for the high and low pitch angle regions. The total accuracy is 90.0% and the AUROC metric is similar at 0.908.

This model provides an excellent first look at the classification ability of a simple ML approach. With the additional track parameters and the phenomenological model predictions, we can further build upon these results and separate all three track topologies. In the remainder of this section we present the results of the SVM models.

5.5.2 Narrowband model

With a tritium spectrum, the region of interest will be a window spanning approximately 4 keV (~200 MHz) about the endpoint value $Q \approx 18.6$ keV.

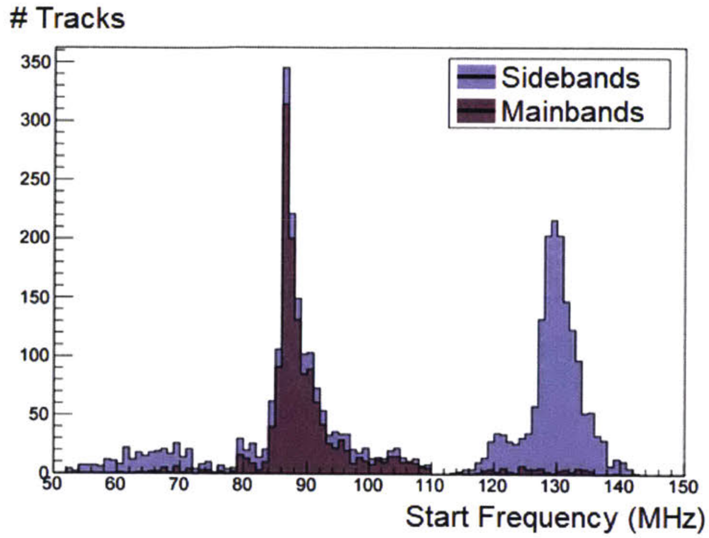


Figure 5.12: Classified 32 keV spectrum with the BDT model. Nearly all of the sideband tracks and most of the mainband tracks are accurately identified, and the mainband-only spectrum is almost completely free of the sideband peaks.

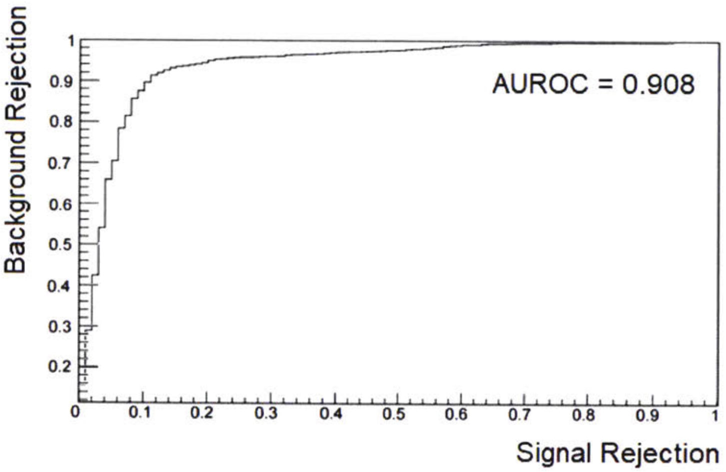


Figure 5.13: Receiver Operating Characteristic curve of the trained BDT model which demonstrates its capacity for high background rejection (i.e. sideband identification) with simultaneously good signal retention.

Here we study the results of a SVM classifier trained on the 30 keV peak and applied to both 30 and 32 keV peaks simultaneously. This 2 keV energy separation serves as a test of the classifier's reliability across an energy range similar to the tritium window; we call this configuration the narrowband classifier. The SVM is implemented in Katydid with the DLIB library [24] and independently in python [25].

The test set accuracy (on the 30 keV data partitioned for evaluation) of this model is 91.1%. The ROC curves for each label as well as the micro and macro-averaged overall curves are shown in Figure 5.14. Every AUROC is well above 0.9, which indicates a consistent ability to accurately discern each individual topology. The micro-average AUROC is 0.97 and the macro-average is 0.96; the largest AUROC is that of sidebands individually with 0.984. Applied to the 32 keV data, the narrowband classifier obtains an accuracy of 91.61% which surpasses the test set score.

As can be seen in Figure 5.15, the frequency spectra at both emission lines boast a clean separation between all three track topologies. The sideband peaks are identified with especially high accuracy, and the two allowed pitch angle regions within the mainband peaks are clear as well. These results in particular are very promising for future pitch angle reconstruction and cyclotron frequency corrections. The success of the narrowband model at 32 keV assuredly suggests a similar technique could be applied to the tritium endpoint region with training on the 17 keV krypton peak.

5.5.3 17 keV peak: sidebands and energy dependence

The 17 keV peak presents a unique problem for the classifier; as discussed in Section 5.1, the CRES signal power distribution across the sideband spectrum is energy-dependent as a consequence of the waveguide reflector. We have come to understand that the 17 keV Phase I data consists entirely[9] of pairs of 1^{st}-order sideband tracks, due to a large suppression of the mainband peak from the reflector interference effect. With this knowledge, I have studied more

---

[9] Either entirely or almost entirely; a small fraction (~1%) of observed tracks at 17 keV are hypothesized to be genuine mainband signals from shake-off electrons [26].

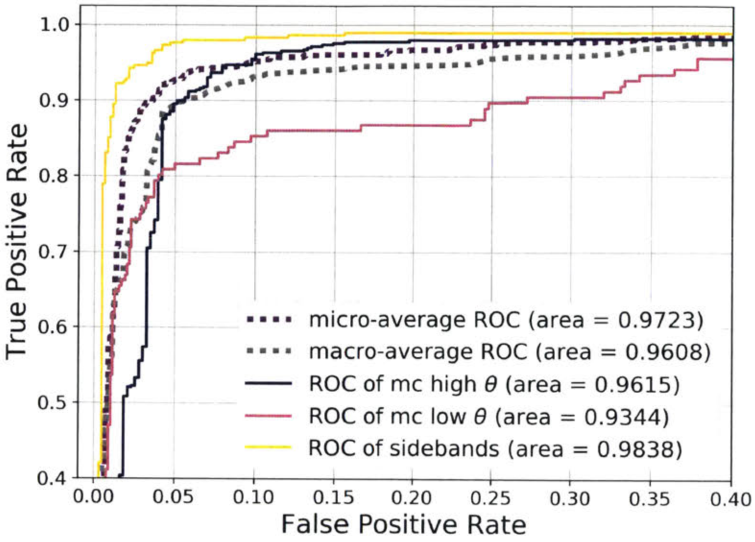
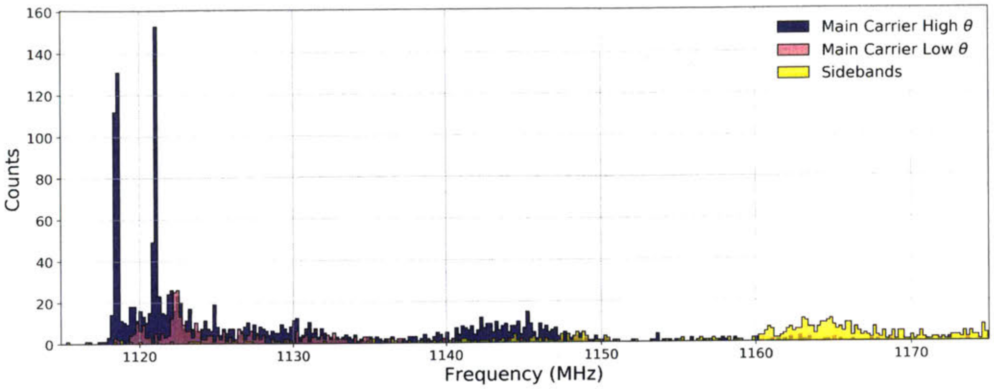


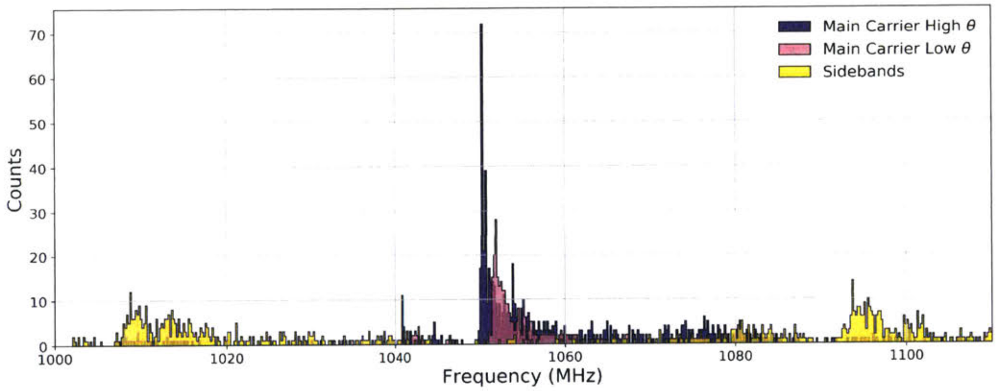
Figure 5.14: ROC curves of individual classes and averages for the Narrowband model.

closely the region between the 17 keV sideband peaks and found an excess of power corresponding to the "hidden" mainband peak. To do this, I select MPTs with exactly 2 tracks separated by at least 10 MHz; these correspond to pairs of $n = 2$ sidebands and the expected location of the mainband follows immediately. I then process the spectrogram in the vicinity of this expected mainband track with the rotate-and-project algorithm and the power excess is revealed. Figure 5.16 shows this projected spectrum analysis and the peak pseudo-SNR (SNR in the projected spectrum) in this region is 1.26. While this quantity is not directly comparable to a true SNR, recall the discriminator threshold in primary track finding is much greater: SNR ≥ 6.0. The power in this region directly confirms the sideband nature of the 17 keV line, and further confirms the expectation that none or very few hidden mainband tracks are sufficiently powerful to show up in the frequency spectrum. Consequently, we cannot train the classifier separately on the 17 keV peak since the training data necessarily contains only one track topology.

Instead, we evaluate the narrowband classifier (trained on 30 keV) when applied to 17 keV tracks. The results are shown in Figure 5.18(a). While the



(a) 30 keV classified frequency spectrum



(b) 32 keV classified frequency spectrum

Figure 5.15: 30 and 32 keV frequency spectra classified with the narrowband SVM model.

majority of tracks are still classified properly, the overall performance is subpar in comparison to the higher-energy peaks. 75.5% of all tracks are identified as sidebands, and we are confident that very nearly all of the remaining 24.5% are incorrect. However, this result is not surprising given the energy dependence of track power and slope correlations which we discussed in Section 5.1. These two parameters are among the most decisive in the narrowband model, and the accuracy suffers greatly when their behavior in the training data differs from that in the evaluation data. It should be noted that this concern of

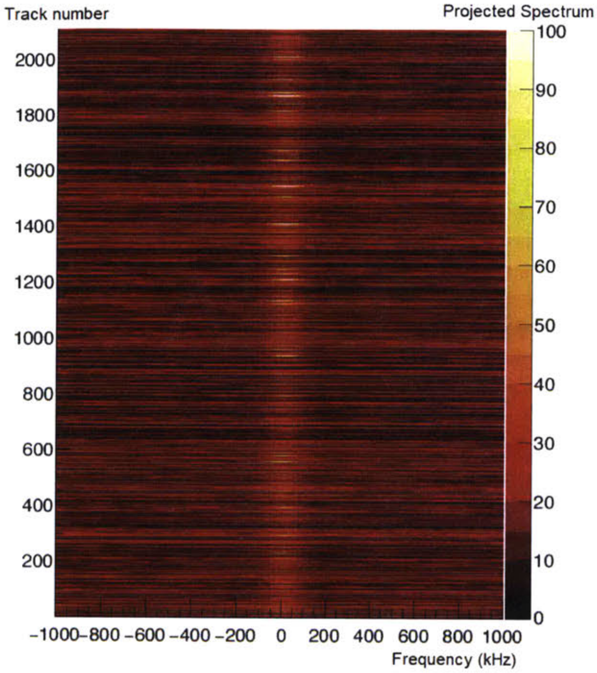


Figure 5.16: Projected spectrum near the hidden mainband location of the 17 keV peak for MPTs with exactly 2 tracks separated by more than 10 MHz. The frequency is shifted by the average of the two sidebands, such that the mainband peak is expected to lie at zero. Each y-axis bin corresponds to a different MPT, hence the axis is labelled "track number". The hidden mainband peak is revealed here as a consistent excess in the spectrum amplitude.

energy dependence is still valid at 32 keV since the narrowband is trained on 30 keV. However, from the comparable accuracy scores we may conclude it is insignificant for this small energy difference, much to our advantage.

5.5.4 Wideband model

To improve the classifier performance at 17 keV, we cannot train the model on 17 keV tracks alone, so we must train it on all three energy ranges simultaneously; this model is called the wideband classifier. The wideband model has the capability to learn the energy-dependent slope and power correlations by considering a much larger energy range than the narrowband model. Figure

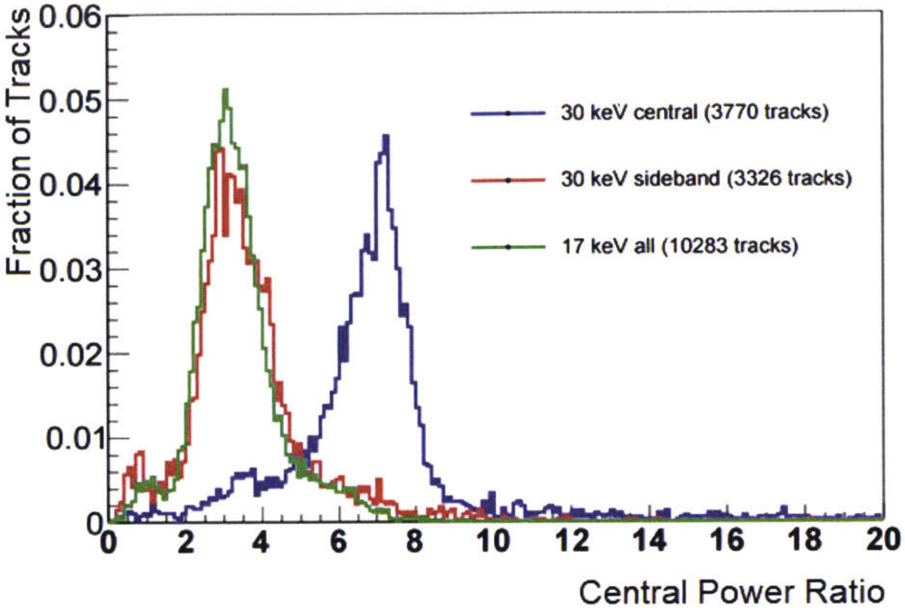
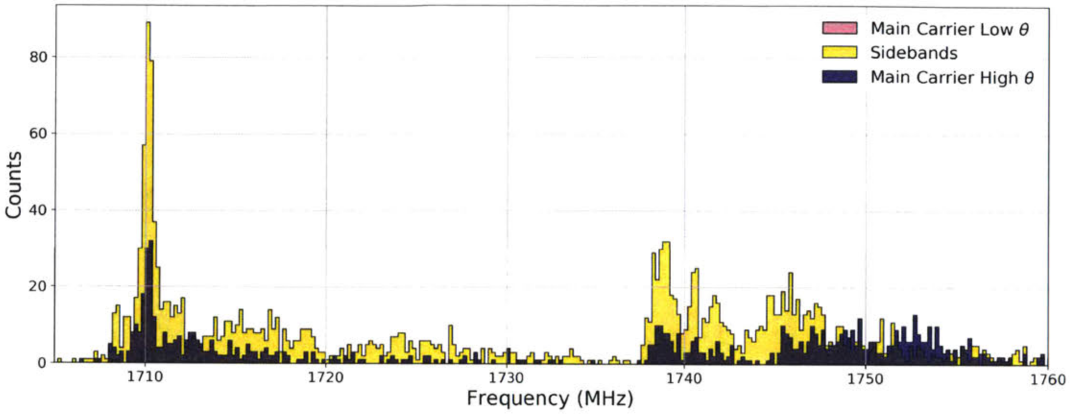
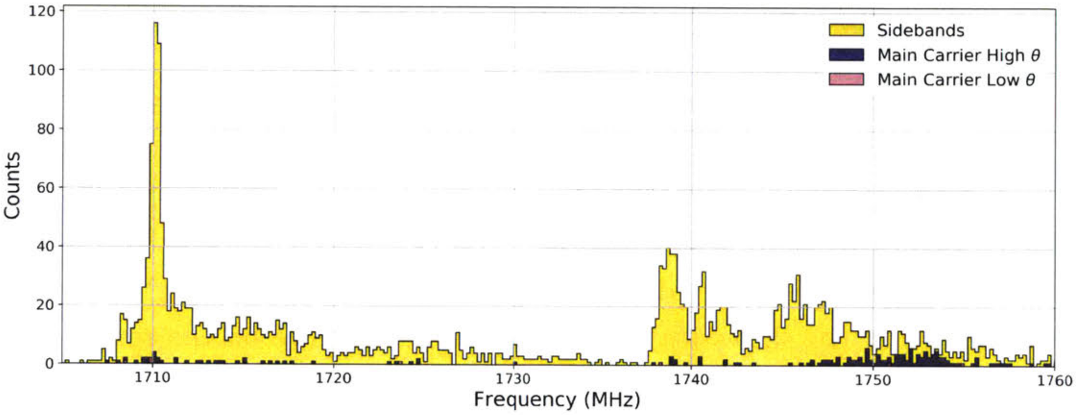


Figure 5.17: Comparison of the central power ratio distribution for 17 keV, 30 keV mainband and 30 keV sideband tracks. The agreement between 17 keV and the higher-energy sidebands (a) further bolsters the 17 keV sideband hypothesis and (b) shows little or no energy dependence on the rotate-and-project analysis, contrary to the track power and slope. The central power ratio r is related to the central power fraction f by $f = r/(1+r)$.

5.18(b) shows the classified 17 keV spectrum from the wideband model. It is immediately clear that this model has a far better performance, with an accuracy of 96.05%. The largest population of main carrier tracks is broadly centered about ~1750 MHz, which is at least 30 MHz above the (suppressed) main carrier peak. From informal inspection of the slope-power correlation among these tracks, as well as the individual spectrograms, we are inclined to believe these tracks are genuine mainband signals and most likely a consequence of satellite shake-up/shake-off conversion electrons [26]. If this hypothesis is correct, then the true accuracy of the wideband classifier at 17 keV is even higher than reported here. The 30 and 32 keV lines also show improved accuracy scores compared to the narrowband model: 92.27% (+1.19%) and



(a) 17 keV classified frequency spectrum with narrowband model



(b) 17 keV classified frequency spectrum with wideband model

Figure 5.18: 17 keV frequency spectra classified with the SVM models. The accuracy of the narrowband model is subpar due to energy-dependent slope and power correlations; the wideband model shows a decisive improvement.

95.64% (+4.03%), respectively.

5.5.5 Optimal set of classification features

We can improve the performance of the narrowband SVM model by examining how useful each parameter is for accurate classification. An exhaustive search of the unique combinations of features from the 14-dimensional feature space results in $2^{14} - 1 = 16,383$ iterations of the training algorithm. We train a SVM and evaluate the accuracy and the AUROC for each feature subset. The subset with the best overall performance may have improved accuracy compared to the full feature set, and will require less intense computation resources to train on a large data campaign.

To evaluate each subset SVM with a single metric, we use the sum in quadrature of the accuracy and the AUROC:

$$\Delta_{\text{opt}} = \sqrt{x^2 + r^2} \tag{5.13}$$

where x is the accuracy of the model and r is the mean of the micro- and macro-averaged AUROCs. The maximum possible value of Δ_{opt} is $\sqrt{2} \approx 1.414$.

Perhaps unsurprisingly, the value of Δ_{opt} is in general large for subsets with many features; a complex model has a greater capability for increased performance. However, some single-feature models are able to achieve high values of Δ_{opt} as well. `fAverage` exemplifies this, which alone yields a model with 90.10% accuracy and $\Delta_{\text{opt}} = 1.314$. A maximum of $\Delta_{\text{opt}} = 1.359$ is achieved with a model utilizing 6 parameters: `fTotalPower`, `fTrackSlope`, `fAverage`, `fRMS`, `fKurtosis`, and `fMaximumCentral`. This amounts to an accuracy of 94.89%, an increase of 4.81% compared to the original result earlier in this section, and an AUROC of 0.9725.

5.5.6 Model comparison

Next, we will discuss the comparative advantages and drawbacks of each of the SVM-based classifiers: the initial narrowband model, and the wideband model. It is important to keep the context of future tritium data and analyses in mind for this discussion, although all of the data presented here is from a krypton source. I choose not to consider the BDT model for use with tritium, as it does not distinguish the high and low pitch angle regions of main carrier tracks. The overall accuracy scores and AUROCs for each classifier are summarized in Table 5.2.

With the introduction of the wideband model, we improved the classification accuracy at 17 keV by training on all three peaks simultaneously; this

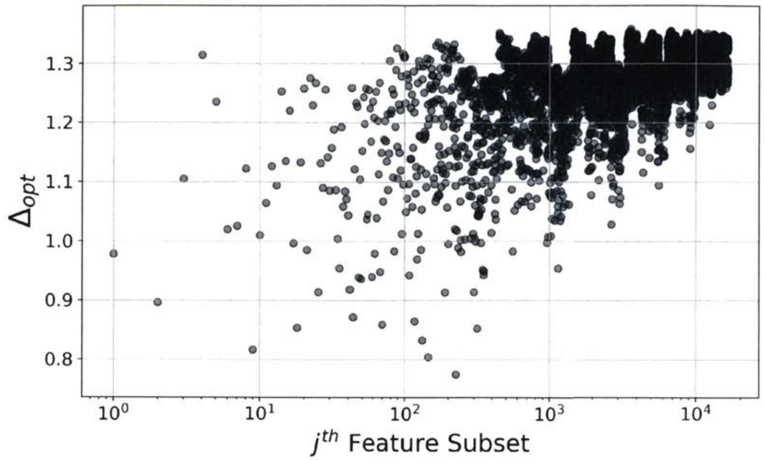


Figure 5.19: Value of Δ_{opt} for each classification feature subset. The x-axis shows the subset index, in ascending order of complexity (number of features). We utilize the log scale to illustrate the ability of some models with very few features to achieve reasonably high accuracy and robustness.

Model	**Accuracy** 17 keV	 30, 32 keV	**AUROC**
BDT		90.0%	0.908
Narrowband	75.5%	91.4%	0.967
Wideband	96.1%	94.0%	0.985
Optimized NB		94.9%	0.973

Table 5.2: Summary of classification model accuracy scores and AUROC metrics. I exclude 17 keV from the calculation of the narrowband AUROC; thus the wideband AUROC reflects all three energy regions and the rest describe only the high-energy peaks.

model also boasted percent-level improvements to the accuracy at 30 and 32 keV. We understand well that the narrowband classifier performed poorly at 17 keV due to energy-dependent correlations between parameters, and the lack of appropriate training at 17 keV. In a tritium analysis, the data acquisition will be contained to not more than ~4 keV around the endpoint; this includes the 17 keV krypton peak which can be used for magnetic field calibration. Therefore, if we construct a narrowband model in this context – trained on 17 keV krypton data and applied to T_2 data from approximately 15–19 keV – it is reasonable to expect a performance similar to the current narrowband model at 30 and 32 keV. The wideband model then only offers improvement at the percent level, and at the expense of utilizing a much larger energy range when we know the parameters of the classification scheme to be energy-sensitive.

In the previous subsection, we saw that the choice of an optimal subset of the classification features improved the narrowband model accuracy and AUROC metrics at the percent level, bringing them to a point comparable to the wideband model. In the next section, we will also examine the effect of less-than-perfect classification on the resultant tritium spectrum. With the intended application of tritium spectroscopy in mind, we are convinced that the marginal improvement at 30 and 32 keV from the wideband model is insignificant. The effect of such an improvement on the reconstructed spectrum is expected to be minimal, and the optimal feature set anyway provides a greater improvement. The narrowband model instead has the benefit of direct tritium applicability; it has already been evaluated over a ~4 keV energy range (~29–33 keV), and performs essentially as well as any model we have studied. Therefore, we conclude the narrowband classifier model to be most suitable for future tritium analysis, trained on 17 keV krypton events and not incorporating 30 and 32 keV.

5.6 Future Work and Outlook

5.6.1 Improved event reconstruction

Recall the event building procedure detailed in subsection 3.2.6. This is the last stage of event reconstruction; at its conclusion, Katydid writes the final MPT events which are used for spectrum analysis. However, it is important to note that this event builder makes no statement about the identity of the main carrier within a MPT structure. The start frequency of an event is simply that of the first (earliest start time) track within the first MPT. The classification scheme thus creates the potential for a more intelligent event building procedure which takes advantage of the labelled track topologies.

One obvious potential improvement utilizing the classification labels is to add consistency checks in the event building process. Tracks which are grouped together in a MPT structure should logically contain no more than one main carrier, and the start frequency of the MPT should be determined by this main carrier track if it is present. MPTs with two or more sidebands should be checked to ensure the frequency spacing between them is consistent with a unique axial frequency. If a main carrier track decreases in frequency after a scatter (recall Figure 3.3), this indicates a sharp increase in the pitch angle and therefore a decreased axial frequency, both of which should be corroborated by the sideband locations if they are present.

This kind of more sophisticated event logic might be incorporated in a number of ways to build a smart event builder. Those which fail one or more checks could be discarded, flagged for further analysis, or the checks could be included in the event building conditions given by Equations 3.7 and 3.8. Figure 5.20 shows several typical event topologies which could benefit from a smart event builder. These are, of course, in addition to single-track MPTs which would benefit from pitch angle corrections independent of the event builder.

- Top left: an event with a scatter which changes the pitch angle from the high to low region, according to the classifier. A faint sideband is also visible above the mainband after the scatter, but it was not reconstructed.

- Top right: A MPT with both a mainband and sideband track, yielding two independent methods to derive the pitch angle: from the slope and from the extracted axial frequency.

- Bottom left: A MPT with two lone sidebands. In this case, we may reconstruct the hidden mainband start frequency from the axial separation and determine the pitch angle correction.

- Bottom right: A MPT which the classifier has identified to contain two mainbands. This indicates an error, either in the classification or the

MPT construction. To address this, we might consider the relative probabilities that either track is in fact a sideband (i.e. work event building information into the classifier), or simply discard the event if it cannot be made sensible.

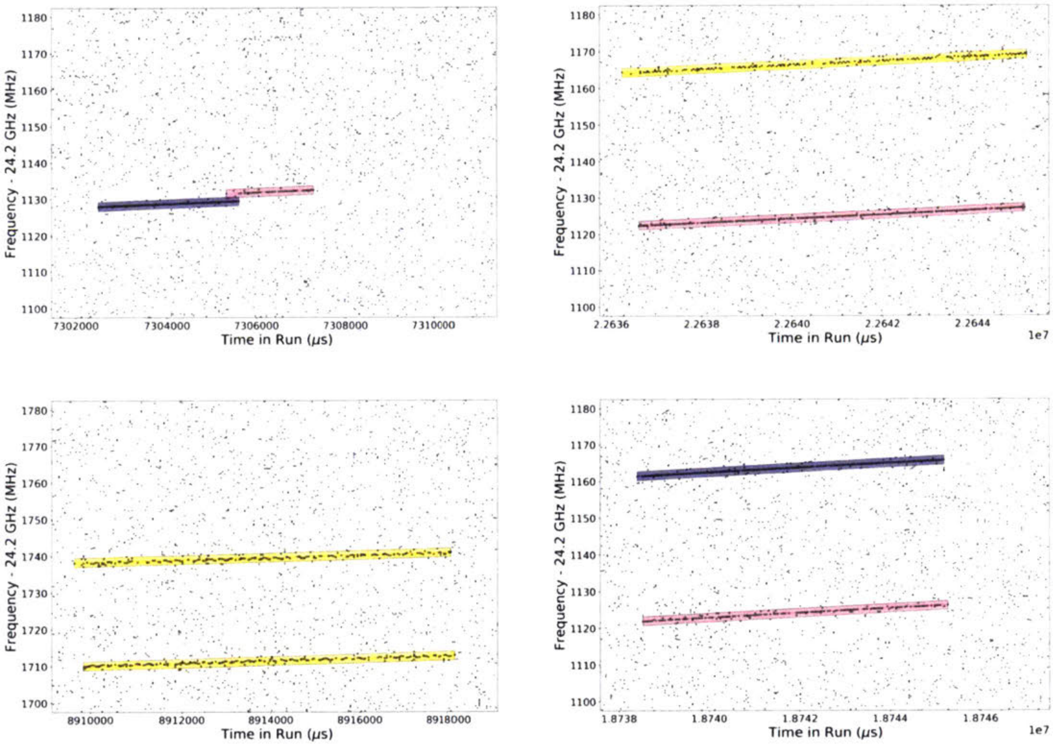


Figure 5.20: Classified tracks of candidate MPT events exhibiting multiple topological combinations.

An improved event builder which makes use of the classifier information in some of the ways presented here will be crucial for proper reconstruction of a continuous tritium spectrum using CRES. Complex event topologies and sideband proliferation from lower energies will demand a sophisticated and comprehensive understanding of the underlying nature of tracks and we expect this model will prove to be a decisive step toward that success. There is an enormous potential to incorporate the numerous aspects of CRES signals discussed in this chapter into the Project 8 analysis chain. The classifier alone has already been implemented and utilized in Katydid for the results presented here. An advanced event building scheme can further the success of the classifier, perhaps even utilizing machine learning itself, and build MPTs with the necessary information for pitch angle corrections as discussed in Section 5.1. These corrections can then be implemented after the event builder to produce the desired end result: an event which confidently contains every track attributed to signal power from one electron, and the calculated cyclotron frequency at the start of the event. The advanced event builder and the implementation of pitch angle corrections are undoubtedly the most promising open-ended areas to expand upon this work from the analysis perspective. Figure 5.21 illustrates a hypothetical analysis chain which includes these elements as extensions to the current analysis in Katydid.

5.6.2 Effects of misclassification on tritium spectroscopy

While the practical implications of an intelligent event builder are vague, we can study the effect of a classifier at the track level on the tritium spectrum more concretely through simulation. Using the Morpho[10] interface to perform Hamiltonian Monte Carlo simulations with the Stan package [27], we model the electron kinematic variables and compute the detected track frequencies according to the discussions in Section 5.1. The kinetic energy is drawn from the tritium spectrum PDF with an endpoint of exactly $Q = 18600$ eV and zero neutrino mass. The power in each of the mainband and the pair of $n = 2$ sideband tracks is then calculated for a circular waveguide with the same bathtub trap configuration as for the data utilized in this chapter.[11] An electron is trapped if it satisfies the pitch angle bounds presented in subsection 5.1.2, and for those trapped a uniform detection threshold is enforced on each track. The detected tracks are collected into a mainband spectrum, denoted $p_0(E)$, and a sideband spectrum denoted $p_1(E)$. Here, E represents the inferred kinetic energy from the detected track frequency; this allows us to study the effects of both misclassification and the lack of pitch angle corrections. It is important

---

[10] Morpho is the Project 8 spectrum analysis package developed primarily by Mathieu Guigue. It is available publicly on GitHub: `https://github.com/morphoorg/morpho/`

[11] Although the data is from Phase I which contained a rectangular waveguide.

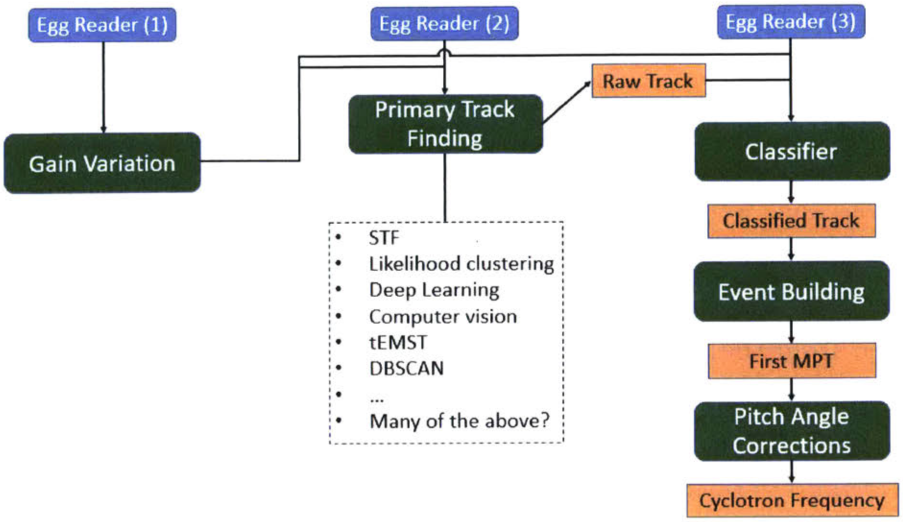


Figure 5.21: Diagram of an extended analysis chain in Katydid which utilizes a classifier, an improved event builder, and the pitch angle corrections discussed earlier in this chapter. For brevity the only individual processors shown are the egg readers, and each of the larger-scale processing steps is represented by the green rectangles. The gain variation and primary track finding blocks describe the current analysis chain, minus the event builder. I have also included a (non-exhaustive) list of efforts in primary track finding to emphasize that potential future work is not limited to the classifier and related elements. Furthest to the right are the future analysis steps which fall under the broad category of secondary track analysis: the classifier, an advanced event builder, and the frequency correction (only the classifier is currently implemented). The output of this extended analysis chain is the true cyclotron frequency, which with the magnetic field determines the kinetic energy at the event start.

to note that this simulation serves only as a toy model and does not reflect the complete design of the Project 8 detector in Phase I (from which the data presented here in earlier sections was taken), or Phase II which has since reconstructed the first ever tritium spectrum with CRES. However, the toy model is relevant as a means to highlight and evaluate some of the challenges that sideband presence will bring when reconstructing any CRES spectrum.

If the probability for a sideband to be falsely classified as a mainband track is α, then the spectrum of classified mainband tracks is:

$$p(E) = p_0(E) + \alpha\, p_1(E) \tag{5.14}$$

where $p_0(E)$ represents the main carrier spectrum and $p_1(E)$ the sideband spectrum. Note that we neglect misclassification in the other direction (mainband as sideband), purely for simplicity. With $\alpha = 0$, we obtain the main-carrier-only spectrum from a "perfect" classifier. Since the energy of each event is calculated from the measured (main carrier) frequency and not the cyclotron frequency, we expect to observe a shift to lower energy (higher frequency) in the endpoint. Indeed, Figure 5.22 shows the simulated Kurie plot for perfect classification $\alpha = 0$ (in black) and the fitted Q-value:

$$Q(\alpha = 0) = 18.461 \text{ keV} \tag{5.15}$$

which deviates from the true modelled value by 139 eV. This demonstrates concretely the need for frequency corrections, as even with perfect classification the pitch angle effects are expected to shift mainband tracks at the 100 eV scale. In the same figure, we show the spectrum with $\alpha = 0.5$ (random classification); now, the region above the endpoint is contaminated by sidebands. As a result, the measured endpoint is:

$$Q(\alpha = 0.5) = 18.972 \text{ keV} \tag{5.16}$$

which exceeds the true value by 372 eV, or 19 MHz: the scale of the observed axial frequency. In addition, the quality of the Kurie fit is noticeably poor.

Lastly, we perform the Kurie fit for many values of α to see the dependence of Q in a more continuous form; Figure 5.23 illustrates these results. Each Q-value represents the mean of 50 unique spectrum simulations for a fixed value of α; the band in light red spans the standard error of the mean on either side, and is dominated by the statistical uncertainty of each simulation. Though the simulations are themselves independent, we use the same set of simulations for all values of α. Thus, the fluctuations of Q over this range don't reflect the standard error directly. For this reason, we should avoid drawing inappropriate parallels between this analysis and future Project 8 phases; for example, the general endpoint precision depicted in Figure 5.23 does not say anything

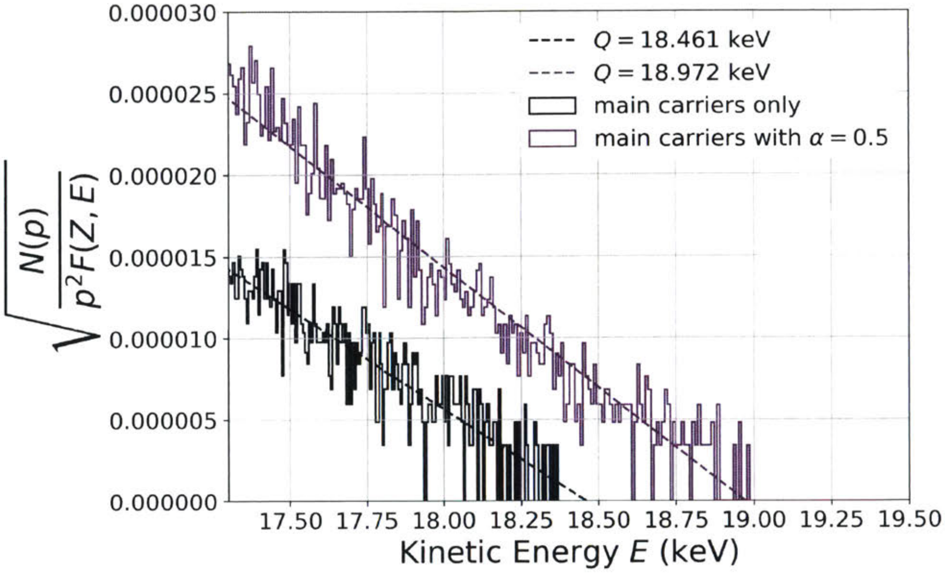


Figure 5.22: Simulated tritium spectra with main carriers only (black) and sideband-contaminated with misclassification rate $\alpha = 0.5$ (purple).

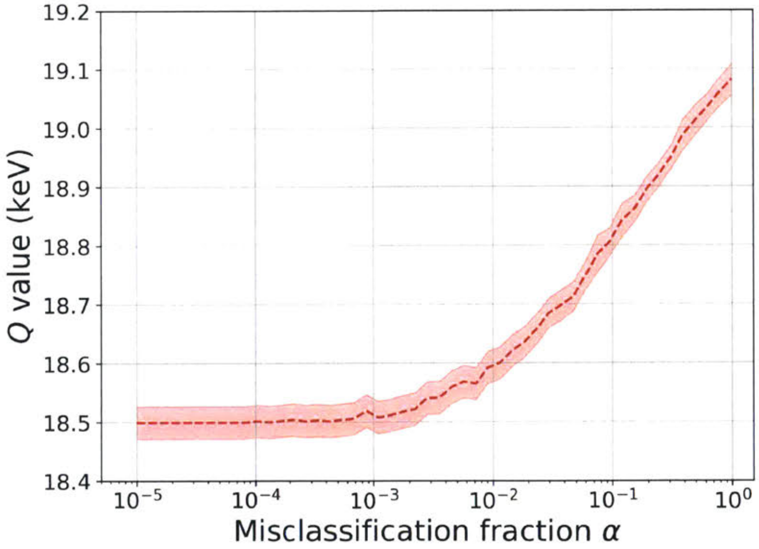


Figure 5.23: Extracted Q-value from Kurie fit to sideband-contaminated tritium spectrum simulations for different values of the misclassification fraction α.

about the real precision of any Project 8 phase. However, it is evident that even for values of $\alpha \approx 10^{-2}$ which are small compared to our actual models, the endpoint shift can be significant. The actual scale at which α begins to shift the endpoint is not necessarily transferable to Project 8, but we can safely conclude from these results that we should have the highest reasonable standard for accuracy with classification. With our thorough understanding of the configurable trap geometry and related systematics, we can optimize future phases of the experiment to help obtain a highly accurate and robust classification model.

Clearly, both energy corrections and track classification will have a substantial influence on a CRES tritium spectrum. We have shown through simulation that these effects shift the measured endpoint on the scale of $\mathcal{O}(100$ eV$)$ or $\mathcal{O}(10$ MHz$)$, which is far from a competitive precision. For a CRES experiment like Project 8 to take the next step and achieve eV-level sensitivity, proper track classification and pitch angle considerations must be handled with near-perfect accuracy.

Chapter 6

Chirp and Fractional Fourier Transform Studies

In this final chapter, I will present preliminary studies on the use of two alternatives to the Fourier Transform for CRES analysis: the chirp transform and the fractional Fourier Transform. The chirp transform multiplies an input signal by a quadratic phase:

$$x(t) \rightarrow e^{-i\beta t^2} x(t) \tag{6.1}$$

followed by the ordinary Fourier Transform. This in effect transforms a linear CRES signal to a Fourier peak, exactly as a Fourier Transform does to a monotone signal. The use of a chirp or related transform is key to unlocking the true potential of an ultra-precise frequency measurement in a CRES experiment like Project 8 [28, 29]. Here, I discuss early work on the implementation of a chirp transform and new analysis of a fractional Discrete Fourier Transform (FrDFT) algorithm.

6.1 Motivation

First we consider the effect of the ordinary DFT on a linear chirp signal over a time interval Δt. This time interval corresponds to the slice length in a Katydid analysis. If the signal at the start of the time slice has frequency f

and constant slope q, then it will sweep in frequency from f to $f + q\Delta t$ over the duration of the time slice. The frequency bin size and the time slice length obey the familiar relation:

$$\Delta f \Delta t = 1 \tag{6.2}$$

where Δf is the frequency bin size in the transformed spectrum. If we imagine a future CRES experiment with perfect track and event reconstruction, pitch angle corrections, etc. the precision of the extracted cyclotron frequency is limited ultimately by the width of a frequency bin in the spectrogram. Naturally, we can seek to improve this precision by increasing Δt. However, this comes at a cost: for sufficiently large Δt, the chirp signal will traverse many frequency bins in each time slice and thus the Fourier spectrum will have signal power smeared across a frequency range, rather than localized in one or a few bins. The condition to avoid this problem follows easily from Equation 6.2 and the statement that $q\Delta t$ must not exceed Δf:

$$\Delta t \leq \frac{1}{\sqrt{q}} \tag{6.3}$$

or equivalently:

$$\Delta f \geq \sqrt{q} \tag{6.4}$$

If the slope q is fixed, then clearly the attainable frequency precision is strictly bounded from below. The chirp transform provides a means to circumvent this problem in theory, as it alters the time/frequency space such that the chirp signal is instead monotonic, i.e. $q = 0$. Then in the ideal case, the frequency precision is no longer limited by q but only by the length of the signal via Equation 6.2.[1] It is worth noting explicitly that the chirp transform requires knowledge *a priori* of the track slope q, which is adequately obtained from primary track finding (with the ordinary FFT and slice size). This places the chirp transform in the category of secondary track analysis. It is also worth noting that the current standard slice size ($\Delta t = 0.04096$ ms) is chosen to approximately saturate the bound in Equation 6.3 for typical Project 8 tracks, where the observed slope varies from roughly 0.2–0.8 MHz/ms.

---

[1] The slice size will still inevitably be bounded by the non-linear nature of the track, which becomes more significant for large Δt as the first-order approximation $\delta E/E \ll 1$ begins to fail.

6.2 Chirp Transform in Katydid

The chirp transform is implemented in Katydid as a processor which performs the quadratic phase shift on a real or complex time series. Each sample is extracted in the form:

$$x_j = r_j e^{i\phi_j} \tag{6.5}$$

where for a real time series, the phase ϕ_j is always zero. The phase shift is given by:

$$\phi_j \rightarrow \phi_j - q(t_s + j/f_{\text{acq}})^2 \tag{6.6}$$

where q is the configurable chirp rate, t_s is the timestamp of the slice within the acquisition, and f_{acq} is the acquisition rate. The use of t_s allows this transform to flatten a chirp signal across many time slices, rather than in each slice independently. The phase-shifted time series signal can then be sent to the existing DFT processor to produce a de-chirped frequency spectrum.[2]

The chirp rate parameter q can be given to the chirp transform in the form of a track (where it calculates q from the track slope), or specified directly in the configuration file. In the broader context of a track finding analysis, this means the chirp components take place during an additional pass through the data after primary track finding, much like the classifier. Figure 6.1 diagrams an analysis flowchart which includes the classifier as well as chirp components, for a total of four passes through the data. This chart is a comprehensive summary of the interaction between existing analyses described in Chapter 3, the new analyses introduced in Chapters 4, 5, and this one, and many components of the future work discussed there. Its further development and implementation in full should be a primary directed effort for future analysts in Project 8 between now and the completion of Phase III.[3]

Next, I present a series of spectrograms containing both real and simulated CRES tracks to study the performance of the chirp transform. In particular, I

---

[2] The DFT processor itself is of course unaware of any distinction between this and an ordinary (unaltered) time series.

[3] In addition to the beam forming and grid search techniques, which I have neglected in this diagram but will be needed to see anything in Phase III in the first place. Development of those algorithms will be arguably even more important.

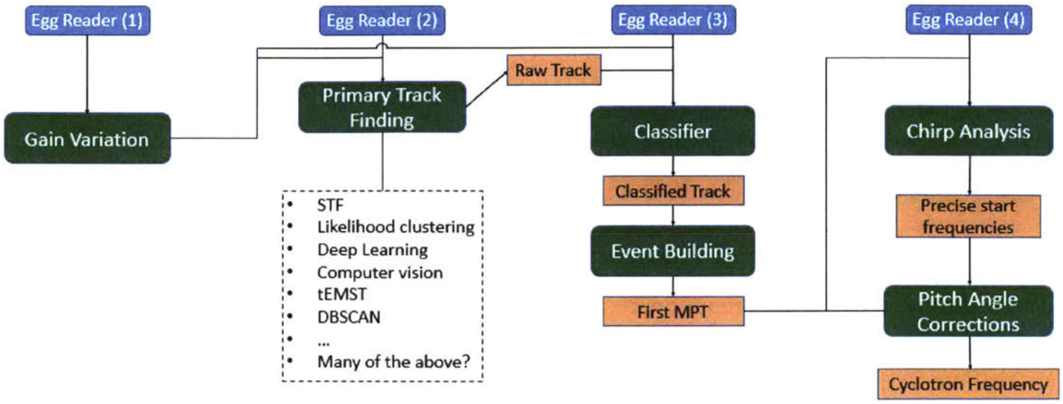


Figure 6.1: Illustration of a full analysis chain which incorporates the existing primary track finding, the classifier, an advanced event builder, chirp analysis and pitch angle corrections. The end result is a cyclotron frequency calculated to very good precision from a comprehensive understanding of the CRES signal characteristics discussed in this work.

vary the slice size from the original (4096), increased by a factor of 8 (32,768), and again (262,144) to study the precision of the de-chirped start frequency. These slice sizes and the associated bin sizes Δt and Δf are enumerated in Table 6.1.

Slice size	**Time bin size (Δt)**	**Frequency bin size (Δf)**
4096	40.96 μs	24.41 kHz
32,768	327.68 μs	3.0518 kHz
262,144	2.62144 ms	381.469 Hz

Table 6.1: Slice parameters used for chirp studies shown in this section.

6.2.1 Results on simulated data

The simulated CRES signal is generated from a component of the Locust simulation [30] known colloquially as the "Fake Track Generator" (FTG) [31]. Locust is a software package also available publicly on GitHub[4] which simulates

---

[4] https://github.com/project8/locust_mc

the antenna and receiver chain response to an incident EM field. The field itself can be generated within Locust or input from a separate particle-tracking simulation, e.g. Kassiopeia [32]. The output of Locust is a data file which looks (to Katydid) exactly like a real data file from the digitizer. For this study, we use the FTG to produce a CRES-like signal above Gaussian noise. The chosen slope is similar to that of observed tracks, and the chosen length is very long to allow for easy spectrogram examination with large time bins.

Figure 6.2 shows the first 30 ms of the spectrogram with the standard slice size. The track begins at approximately 13 ms and chirps upward at 0.447 MHz/ms. The corresponding slope which saturates the bound in Equation 6.4 for the standard slice size is 0.596 MHz/ms; thus the track SNR does not initially suffer from the chirp.

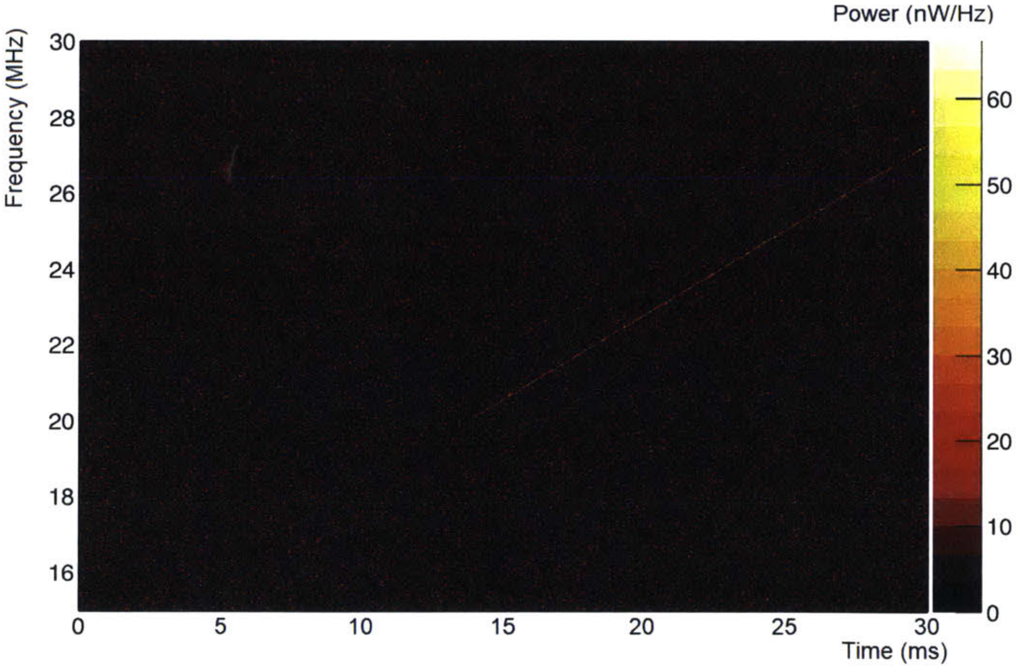


Figure 6.2: Raw spectrogram of a simulated CRES signal with the standard slice size.

We first clearly motivate the use of a chirp transform by increasing the slice size but retaining the ordinary DFT. A large slice size in principle allows for a more precise start frequency determination, but as expected the SNR per bin quickly drops (Figure 6.3). The frequency bin size has been reduced from 24.4 kHz to only 3.05 kHz, but the track now smears across roughly 6 frequency bins per time bin which greatly hinders the potential for good reconstruction.

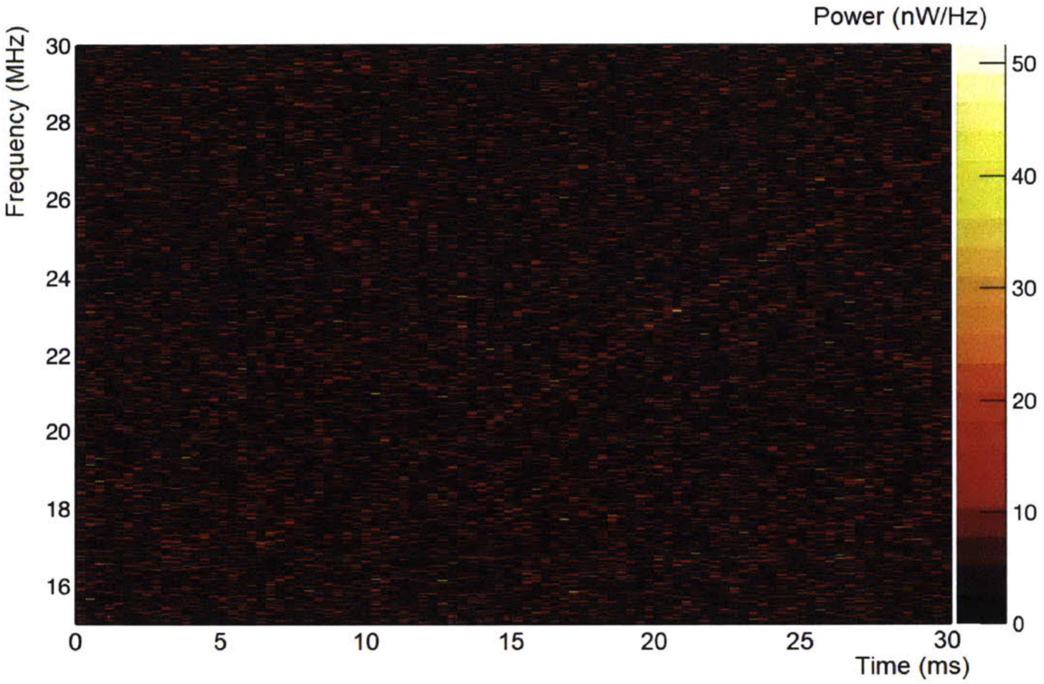


Figure 6.3: Raw spectrogram of a simulated CRES signal with the slice size increased by a factor of 8 to 32,768.

With the slice size still increased, we now de-chirp the signal by $q = 0.441$ MHz/ms. This effectively decreases the track slope by the same amount, to 0.006 MHz/ms which satisfies Equation 6.4 for this new slice size. The de-chirped spectrogram is shown in Figure 6.4; it is immediately clear that the chirp transform works as intended. The simulated signal is de-chirped perfectly and as a result the track power per bin has increased by the same factor as the slice size. Reconstruction of this track in Katydid is very easy and the de-chirped start frequency can be reliably determined to within 1 bin.

Figure 6.5 shows the same signal with the slice size further increased and

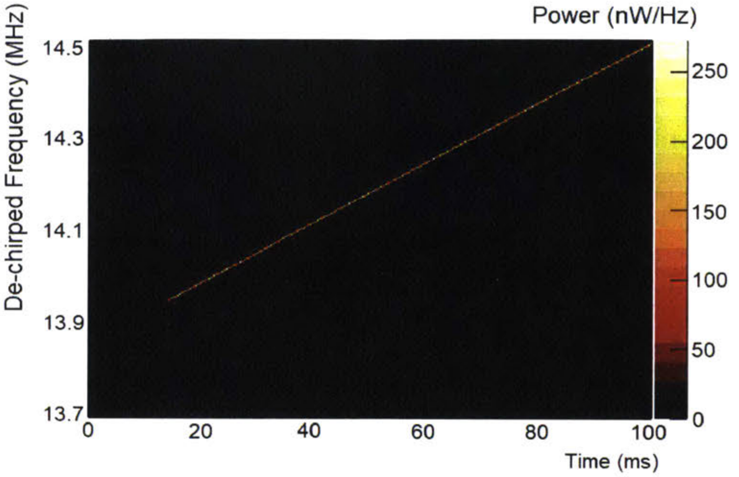


Figure 6.4: Raw spectrogram of a simulated CRES signal with slice size 32,768 after de-chirping by 0.441 MHz/ms. The resultant track slope is greatly reduced and the signal power is once again contained within one or a few frequency bins per time bin. This increases both the SNR and the accessible start frequency precision by the same factor as the slice size increase.

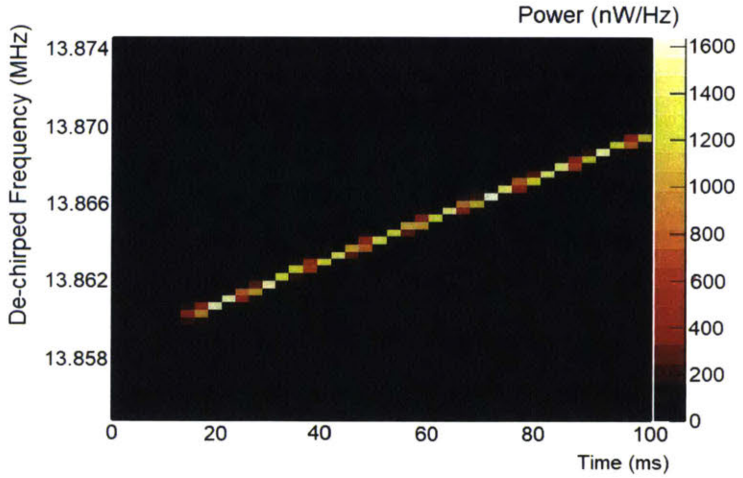


Figure 6.5: Spectrogram with the slice size further increased to 262,144 and de-chirped by 0.447 MHz/ms. The precision and track SNR both continue to increase.

the slope almost fully de-chirped. The slice length is now 2.6 milliseconds – longer than many real tracks – and the power per bin has once again increased dramatically. With a simulated signal, it is clear that this technique has essentially no limiting factor and the de-chirped frequency can be determined with extraordinary precision and confidence.

6.2.2 Results on electron data

Next, we apply the same process to a real CRES event and study the effect on the spectrogram. In the interest of space, I will only present the de-chirped spectrograms for a single mainband and sideband track. These are shown in Figure 6.6, and the slice size for each is 32,768. Each has been fully de-chirped, i.e. by the same amount as the reconstructed track slope.

The de-chirped mainband boasts good SNR per bin, but not in comparison to the simulated signal with the same slice size. At each time slice, the signal is smeared across multiple bins spanning approximately 10–20 kHz; this is not a consequence of the chirp, but instead reveals some fluctuations at this scale which are fast compared to the time bin size (0.328 ms). It is not yet clear whether these fluctuations are fast compared to the standard time bin size as well. Still, it appears the de-chirped frequency can be determined with marginally better precision than the standard frequency bin size (24.4 kHz). In addition to the smearing due to fast fluctuations, we observe a new structure: the de-chirped track moves through a frequency range of roughly 80 kHz on a timescale of several milliseconds, in a slow but somewhat chaotic manner. We can only hypothesize that its source is similar to the radial gradient perturbation discussed in subsection 5.1.5, where the guiding center of the cyclotron orbit precesses in (x, y). However, this precession is slower by a factor of 100, affects the mainband, and is not confidently periodic due to the somewhat unpredictable behavior between 44–48 ms.

The sideband spectrogram shows a similar ~100 Hz precession in addition to the larger familiar "width" due to the radial gradient. At this timescale, it is unsurprising that the radial gradient oscillation is completely smeared in each time bin. Notably, the signal power per bin due to this smearing

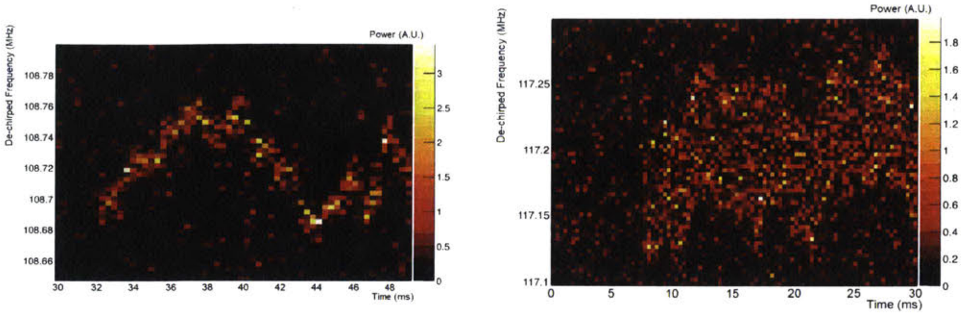


Figure 6.6: De-chirped spectrogram of a mainband CRES signal (left) and a sideband (right). The mainband spectrum reveals new structure in the track frequency fluctuations at the 10 millisecond timescale. The sideband spectrum shows a similar chaotic precession and preserves its width from the radial precession of the cyclotron orbit.

does not worsen with the increased slice size; therefore the chirp transform does not have a direct negative impact on the potential for track reconstruction/analysis. In practice, the track reconstruction process would certainly require some adaptation to better handle these types of heavily altered tracks.

6.3 Discussion

The simulated track results demonstrate the best possible outcome of the chirp transform. An idealized linear CRES signal can be de-chirped to localize the signal power in one bin, in principle with an arbitrarily long time slice. The logical extreme of this approach would be to contain the entire length of a track in one slice, and with the chirp transform create a 1D spectrum which contains the track in $\mathcal{O}(1)$ bins.[5] The precision of the de-chirped frequency is then limited only by the ordinary uncertainty principle (Equation 6.2) where Δt is the full signal length. With real CRES data, we observed two new properties of the mainband signal:

---

[5] The rotate-and-project operation utilized for much of the work in Chapter 5 can be viewed as a means of approximating exactly this idea.

1. A fast fluctuation in the track frequency of order 10 kHz, which smears the track power. It is unclear whether this fluctuation is very fast (perhaps a consequence of axial motion), or only marginally faster than the time bin scale.

2. A slow precession of the track frequency by approximately 80 kHz. This most likely indicates the electron motion in the (x, y) directions is influenced by small imperfections in the magnetic field which are not yet understood.

Both of these phenomena prohibit the ability of a chirp transform to achieve sub-kHz precision as with the simulated data; however, they do not decrease the signal power per bin. The total signal power per slice increases with the slice size, but when smeared over many bins the number of bins increases at the same rate. Thus, the signal power per bin remains constant with respect to the slice size and the chirp transform provides a potential avenue for comparable (albeit very different) track reconstruction of de-chirped signals.

6.3.1 Linear chirp limit

Although the de-chirped frequency precision of the simulated signal appears virtually limitless, recall that a true CRES signal will gradually decrease its slope over a sufficiently large timescale as the radiated power falls with energy. Thus, the linear nature of the track – which persists for all time in the simulation – in truth breaks down for long tracks, and even a perfectly de-chirped real signal will not forever be localized to one frequency bin. We can estimate the timescale at which this happens by again using Equation 6.3, where q this time represents the change in the track slope from $t = 0$ to $t = \Delta t$ (i.e. the slope of the de-chirped track). We will also utilize the cyclotron radiation power:

$$-\frac{dK_e}{dt} = \frac{\sigma_t B^2 c}{\mu_0}\beta^2 \tag{6.7}$$

where K_e is the kinetic energy, σ_t is the Thomson cross section, $B = 1$ Tesla, c is the speed of light, μ_0 is the permeability of free space and β is the velocity divided by c. We maintain the assumption that the fraction of the total energy

$K_e + m_e$ lost over the timescale of interest is negligible. The time derivative of the cyclotron frequency follows from Equations 1.4 and 6.7:

$$\frac{df}{dt} = -\frac{eB}{2\pi E}\frac{dE}{dt} \tag{6.8}$$

$$= \frac{eB^3\sigma_t c}{2\pi E^2 \mu_0}\beta^2 \equiv S\beta^2 \tag{6.9}$$

thus with E constant, the track slope is proportional to β^2 with the constant of proportionality S. With $\beta = 0.26$ (as in Equation 2.2), $B = 1$ Tesla and $E = 530$ keV, the proportionality constant evaluates to $S \approx 5$ MHz/ms (note that multiplication by β^2 recovers a typical Project 8 slope). With the same parameters we similarly find $-d\beta/dt = 0.055$, which we approximate as a constant as well: $\beta(t) = \beta - q_\beta t$ and $q_\beta = 0.055$. Then the change in the track slope over a time Δt is:

$$\Delta q = S\left(\beta^2 - \beta(\Delta t)^2\right) \tag{6.10}$$

$$= S\left(2q_\beta\beta\Delta t - q_\beta^2(\Delta t)^2\right) \tag{6.11}$$

$$\Rightarrow \frac{1}{S(\Delta t)^2} = 2q_\beta\beta\Delta t - q_\beta^2(\Delta t)^2 \tag{6.12}$$

where the last equality comes from saturating the bound in Equation 6.3 with $\Delta q = (\Delta t)^{-2}$. S, q_β, and β are known so we may solve the above equation for Δt:

$$\Delta t \approx 2 \text{ ms} \tag{6.13}$$

With this estimation, we conclude that a Project 8 CRES signal de-chirped perfectly by its initial slope will still decrease in frequency by $1/\Delta t = 500$ Hz after time $\Delta t = 2$ milliseconds. Thus, the ability of the linear chirp transform to achieve the best possible precision is limited to $\Delta t < 2$ ms, and $\Delta f > 500$ Hz. This is quite similar to the precision demonstrated for the simulated track in Figure 6.5, with the largest slice size.

6.3.2 Start frequency vs. de-chirped frequency

In the context of frequency precision or determination, throughout this discussion I have carefully referred to the domain of the chirp transform output as the de-chirped frequency. It is important to distinguish this from the ordinary frequency, because it is the start frequency of a signal we want to extract but the chirp transform provides exceptional precision of the de-chirped frequency. The two are of course simply related:

$$f_s = f_0 + qt_s \tag{6.14}$$

where f_s is the start frequency, f_0 is the de-chirped frequency, q is the chirp rate and t_s is the start time. If the start time is known, then we may match it with the start time of the chirp transform and $f_s = f_0$. However, the more important statement is about the uncertainty:

$$(\Delta f_s)^2 = (\Delta f_0)^2 + q^2 (\Delta t_s)^2 \tag{6.15}$$

where I have assumed there is no uncertainty on the slope q. The chirp transform allows Δf_0 to become very small, but the resultant uncertainty on the true start frequency still has a contribution from the start time. Neglecting Δf_0, we may simply rewrite the above equation as:

$$\Delta f_s = q\Delta t_s \tag{6.16}$$

While the de-chirped frequency precision improves with large Δt according to the uncertainty principle, the actual start frequency precision (somewhat paradoxically) worsens as Δf_s grows with Δt_s. We can understand this with Equation 6.15 where Δf_0 and Δt_s are related by the uncertainty principle $\Delta f_0 \Delta t_s = 1$; multiplying each side by $(\Delta t_s)^2$ we obtain:

$$\Delta f_s \Delta t_s = \sqrt{1 + \left[q (\Delta t_s)^2\right]^2} \tag{6.17}$$

Therefore, if both the start time and frequency are determined from a chirp transform alone then the product of their uncertainties is always greater than 1, i.e. the Heisenberg bound is not saturated.[6] However, this setback is not as restrictive as it may at first appear. The start time can still be determined independently, for example with ordinary track finding. In fact, the uncertainty principle says nothing about the obtainable precision of a signal

---

[6] The exception of course is when $q = 0$, but this describes an ordinary DFT.

start time; thus it is possible in theory to completely remove Δt_s from the RHS of Equation 6.15 and the remaining uncertainty is simply $\Delta f_s = \Delta f_0$. The start frequency precision is then the same as that of the de-chirped frequency, and limited only by the full track length.

6.3.3 Summary

From this analysis, we have demonstrated the ability of the chirp transform to extract the de-chirped frequency of a simulated CRES signal with sub-kHz precision. We have also calculated the limiting precision of the linear chirp approximation to be at a similar scale (500 Hz). To fully realize this level of precision in data, a number of factors must be addressed:

- The track frequency is smeared at the 10 kHz scale over short times. This prohibits the use of a chirp transform to achieve a precision below this scale, no matter the slice size used.
- The de-chirped track frequency also precesses irregularly but very slowly, on the timescale of 10 milliseconds. It is unclear to what extent this behavior might limit a future chirp analysis, until it is better understood.
- An independent method must be developed to determine the start time to a similarly extraordinary precision, such that the resultant uncertainty on the start frequency is negligible. Alessandro Banducci [17] and Nick Buzinsky have already begun to explore this topic. With Project 8 chirp rates, the required start time precision to reach sub-kHz frequency precision is $\Delta t \leq 1$ μs. Currently, the start time precision is limited by the spectrogram bin size $\Delta t = 40.96$ μs.

6.4 Fractional DFT Discussion

In this section, I explore the possibility of a Fractional DFT (FrDFT) as another possible alternative to the Fourier Transform and the chirp transform. It serves primarily as an aside, since I have not found a promising use for this transform in CRES data. But the analysis is interesting and somewhat novel, so it is a worthy digression.

The concept of a track as a linear object in the space of time and frequency is now quite familiar. The Fourier Transform takes data from the time domain to the frequency domain: a "rotation" by the angle $\pi/2$ in this space, if the time and frequency axes are taken to be orthogonal. A monotone signal exists as a constant frequency, and its Fourier Transform is of course a delta function; with discrete data, this is equivalent to the statement that the signal power is localized to one bin. The frequency of a chirp signal is not fixed, thus in Fourier space it is not localized to one bin. The chirp transform fixes this problem because the quadratic phase shift converts a chirp signal to a monotone. However, we found in the previous discussion that this comparison is slightly dishonest because while the signal power in the de-chirped frequency domain is localized, it is still not the case in the true frequency domain. Furthermore, the chirp transform does not saturate the uncertainty bound between time and frequency, a disadvantage compared to the ordinary DFT. The idea of the FrDFT is to convert a linear signal in the time/frequency plane to a constant frequency by a pseudo-rotation. We already understand the ordinary DFT as a rotation by $\pi/2$ in the same way, so we can describe the FrDFT as more general rotation by an angle α. Then α may be chosen with knowledge of the track slope to ideally localize the signal power and avoid the precision disadvantage of the chirp transform.

6.4.1 Procedure

The actual implementation of a FrDFT has been the subject of some considerable study over the last 25 years. In particular, a number of procedures for calculating a transformation matrix $F(\alpha)$ have been proposed [33, 34]. The transformation matrix is defined by:

$$\tilde{x}_k = \sum_{j=0}^{N-1} F_{kj}(\alpha)\; x_j \tag{6.18}$$

where x_j is the input vector of dimension N and $\tilde{x}_k$ is the transformed vector of the same dimension. An interesting method to calculate an expression for F_{jk}

arises from a fractional FFT algorithm published by Garcia *et. al* [35]. The algorithm consists only of two kinds of operations: (a) an ordinary DFT, and (b) multiplication by a quadratic phase. This is especially intriguing because it is clearly related to chirp transforms, and its implementation in Katydid is very easy since the necessary components already exist. In addition, the quadratic phase multiplication is an $\mathcal{O}(N)$ operation, and the ordinary DFT is $\mathcal{O}(N \log N)$ as is well established by Cooley and Tukey [36]. Thus, the FrDFT as described by this method is also achievable in $\mathcal{O}(N \log N)$ time – just as fast as the FFT – while the direct computation of $F(\alpha)$ is necessarily $\mathcal{O}(N^2)$ or worse.

The algorithm consists of five steps:

1. Multiplication by the quadratic phase $\exp\left(-\frac{i\pi}{N} q_1 j_1^2\right)$
2. The FFT
3. Multiplication by the quadratic phase $\exp\left(-\frac{i\pi}{N} q_2 j_2^2\right)$
4. The inverse FFT
5. Multiplication again by the quadratic phase $\exp\left(-\frac{i\pi}{N} q_1 j_3^2\right)$

where $0 \leq j_1, j_2, j_3 < N$ denote the distinct sample indices at each transform step, and the chirp rates q_1, q_2 are related to the rotation angle α by:

$$q_1 = \tan(\alpha/2) \tag{6.19}$$

$$q_2 = \sin(\alpha) \tag{6.20}$$

The purpose of the remainder of this section is to study this algorithm from the viewpoint of a transformation in a 2D vector space in order to build on the work of Garcia *et. al.* and gain some understanding in the general study of fractional Fourier Transforms.

6.4.2 A rotation in the time/frequency plane

To concretely define the time/frequency plane, we write:

$$x(t) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{6.21}$$

$$\tilde{x}(\omega) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{6.22}$$

where $x(t)$ denotes a time-domain function and $\tilde{x}(\omega)$ a frequency-domain function. The ordinary DFT is represented by the matrix which takes $x(t) \rightarrow \tilde{x}(\omega)$, which we will call B:

$$B = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \tag{6.23}$$

and we can easily confirm that B is indeed a rotation matrix with the angle $\pi/2$. The continuous chirp transform is associated to a Fourier Transform with the replacement $\omega \rightarrow \omega + qt$:

$$\begin{aligned} f_{\text{chirp}}(\omega, q) &= \int_{-\infty}^{\infty} f(t) e^{-iqt^2} e^{-i\omega t} \, dt \\ &= \int_{-\infty}^{\infty} f(t) e^{-i(\omega+qt)t} \, dt \\ f_{\text{chirp}}(\omega, q) &\Leftrightarrow \tilde{f}(\omega + qt) \end{aligned} \tag{6.24}$$

where $\tilde{f}$ denotes the Fourier Transform of f. The notation of the last line is obviously dubious, since $\tilde{f}$ is by definition only a function of ω and not t; but the relation expressed in Equation 6.24 is perfectly sensible in the 2D vector space of (t, ω). The quadratic phase shift is the matrix which performs the substitution $\omega \rightarrow \omega + qt$ and preserves $t \rightarrow t$:

$$\begin{pmatrix} t' \\ \omega' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ q & 1 \end{pmatrix} \begin{pmatrix} t \\ \omega \end{pmatrix} \tag{6.25}$$

Now, we have determined the matrices $A(q)$ and B which describe the quadratic phase shift and the Fourier Transform, respectively:

$$A(q) = \begin{pmatrix} 1 & 0 \\ q & 1 \end{pmatrix} \tag{6.26}$$

$$B = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \tag{6.27}$$

The fractional FFT from the Garcia algorithm is:

$$F(\alpha) = A(q_1)B^{-1}A(q_2)BA(q_1) \tag{6.28}$$

and the claim is that with q_1 and q_2 related to α by Equations 6.19 and 6.20, the matrix $F(\alpha)$ is a rotation by precisely the same angle α. I will next explicitly show this claim to be true.

$$\begin{aligned} F(\alpha) &= \begin{pmatrix} 1 & 0 \\ q_1 & 1 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ q_2 & 1 \end{pmatrix}\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ q_1 & 1 \end{pmatrix} & (6.29) \\ &= \begin{pmatrix} 1 - q_1 q_2 & -q_2 \\ q_1(2 - q_1 q_2) & 1 - q_1 q_2 \end{pmatrix} & (6.30) \end{aligned}$$

With Equation 6.20 we can immediately replace $F_{12}(\alpha) = -q_2$ with $-\sin(\alpha)$. The diagonal elements both simplify to:

$$\begin{aligned} 1 - q_1 q_2 &= 1 - \tan(\alpha/2)\sin(\alpha) \\ &= 1 - \frac{\sin(\alpha/2)}{\cos(\alpha/2)} 2\sin(\alpha/2)\cos(\alpha/2) \\ &= 1 - 2\sin^2(\alpha/2) \\ \Rightarrow 1 - q_1 q_2 &= \cos(\alpha) & (6.31) \end{aligned}$$

and all that remains is to massage the last element F_{21}:

$$\begin{aligned} q_1(2 - q_1 q_2) &= q_2(2q_1/q_2 - q_1^2) \\ &= \sin(\alpha)\left(\frac{2\tan(\alpha/2)}{2\sin(\alpha/2)\cos(\alpha/2)} - \tan^2(\alpha/2)\right) \\ &= \sin(\alpha)\left(\sec^2(\alpha/2) - \tan^2(\alpha/2)\right) \\ \Rightarrow q_1(2 - q_1 q_2) &= \sin(\alpha) & (6.32) \end{aligned}$$

Indeed, we have shown that this representation of $F(\alpha)$ as a product of linear shifts and $\pi/2$ rotations is the 2D rotation matrix:

$$F(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \tag{6.33}$$

This result is still very bizarre to me and demands a deeper understanding of the connections between chirp transforms and rotations.

6.4.3 N-dimensional data space

We have studied the continuous fractional Fourier Transform in the space of time and frequency, and extracted from it the interpretation as a 2D rotation matrix. Next, we return to the N-dimensional data space of the discrete transform and once again find an explicit expression for $F(\alpha)$.

The DFT is defined as:

$$\tilde{x}_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \exp\left(-\frac{2\pi i}{N} jk\right) x_j \tag{6.34}$$

and we can easily read off the matrix element B_{jk}:

$$B_{kj} = B_{jk} = \frac{1}{\sqrt{N}} \exp\left(-\frac{2\pi i}{N} jk\right) \tag{6.35}$$

We can do the same for the quadratic phase expression to find A_{jk}:

$$\tilde{x}_k = \exp\left(-\frac{i\pi}{N} qk^2\right) x_k = \sum_{j=0}^{N-1} \exp\left(-\frac{i\pi}{N} qjk\right) x_j \delta_{jk} \tag{6.36}$$

$$A_{kj}(q) = A_{jk}(q) = \exp\left(-\frac{i\pi}{N} qjk\right) \delta_{jk} \tag{6.37}$$

where δ_{jk} is the Kronecker delta. For brevity I will introduce the quantity $\zeta = e^{-i\pi/N}$, the $2N^{\text{th}}$ root of unity; the simplified expressions for A_{jk} and B_{jk} are:

$$A_{jk}(q) = \delta_{jk} \zeta^{qjk} \tag{6.38}$$

$$B_{jk} = \frac{1}{\sqrt{N}} \zeta^{2jk} \tag{6.39}$$

The matrix element $F_{jk}(\alpha)$ follows from the matrix product:

$$\begin{aligned}
F_{jk} &= \sum_{l,m,n,p=0}^{N-1} A_{jl}(q_1) B^{-1}_{lm} A_{mn}(q_2) B_{np} A_{pk}(q_1) && (6.40) \\
&= \sum_{l,m,n,p=0}^{N-1} \zeta^{q_1 jl} B^{-1}_{lm}\ \zeta^{q_2 mn} B_{np}\ \zeta^{q_1 pk}\ \delta_{jl}\delta_{mn}\delta_{pk} \\
&= \sum_{m=0}^{N-1} \zeta^{q_1 j^2} B^{-1}_{jm}\ \zeta^{q_2 m^2} B_{mk}\ \zeta^{q_1 k^2} \\
&= \zeta^{q_1(j^2+k^2)} \sum_{m=0}^{N-1} B^{-1}_{jm}\ \zeta^{q_2 m^2} B_{mk} \\
&= \frac{1}{N}\zeta^{q_1(j^2+k^2)} \sum_{m=0}^{N-1} \zeta^{-2jm}\ \zeta^{q_2 m^2}\ \zeta^{2mk} \\
&= \frac{1}{N}\zeta^{q_1(j^2+k^2)} \sum_{m=0}^{N-1} \zeta^{q_2 m^2 + 2m(k-j)} && (6.41)
\end{aligned}$$

Substituting in ζ we have the explicit expression:

$$F_{jk} = \frac{1}{N} \exp - \frac{i\pi}{N} q_1 (j^2 + k^2) \sum_{m=0}^{N-1} \exp - \frac{i\pi}{N}\Big(q_2 m^2 + 2m(k-j)\Big) \tag{6.42}$$

This is perhaps a simpler closed-form expression for a FrDFT matrix than presently exists in the literature.

In this section, we have confirmed that the algorithm described by Garcia *et. al.* is consistent with the understanding of a fractional Fourier Transform as a rotation between time and frequency. We have also used the algorithm to calculate an explicit expression for the associated transformation matrix F_{jk}. In the appendix to this chapter, I continue with some additional analysis of this expression for F_{jk} to check for consistency in the two limiting cases of the rotation angle:

(a) For $\alpha = 0$, the transformation matrix F_{jk} reduces to the identity: $F_{jk}(0) = \delta_{jk}$.

(b) For $\alpha = \pi/2$, the fractional transform corresponds to an ordinary DFT: $F_{jk}(\pi/2) = B_{jk}$ up to an overall phase.

6.4.4 Katydid implementation

The FrDFT is implemented in Katydid within the chirp transform processor. The processor contains a separate slot for each transform type, and like the chirp transform the rotation angle for the fractional transform is determined either by user specification, or from an input track. The transform utilizes the algorithm described in this section internally to perform the rotation at the same $\mathcal{O}(N \log N)$ runtime as the ordinary FFT. However, the results on CRES data are somewhat underwhelming (though not unexpected). The fractional transform does not localize the signal power from a CRES event and sharpen the frequency resolution as the chirp transform does. Figures 6.7 and 6.8 show a spectrogram of the simulated event after a FrDFT with $\alpha = 0.002$ and $\alpha = 0.01$. It is clear that the signal is effectively destroyed by this operation as the magnitude of α becomes appreciable at all. While this result is obviously disappointing, it is unsurprising given the expression for F_{jk} (Equation 6.42). If the fractional transform would localize the signal power, then it should be true that $F_{kj} \cdot x_j$ returns a delta function when x_j is a linear chirp signal; clearly this is not the case. Indeed, as we have already discussed it *is* the case for the chirp transform alone, as demonstrated by the excellent results in Figure 6.4.

While the Fractional FFT is an intriguing idea, a suitable application to CRES signal analysis is not immediately apparent. It also remains to be determined whether the transformation does in fact preserve the uncertainty bound $\Delta f \Delta t = 1$, as the chirp transform does not. While we have learned a substantial amount about the curious "mixed" domain of time and frequency accessible through this transform, it appears this is not a useful avenue to extract CRES signal properties.

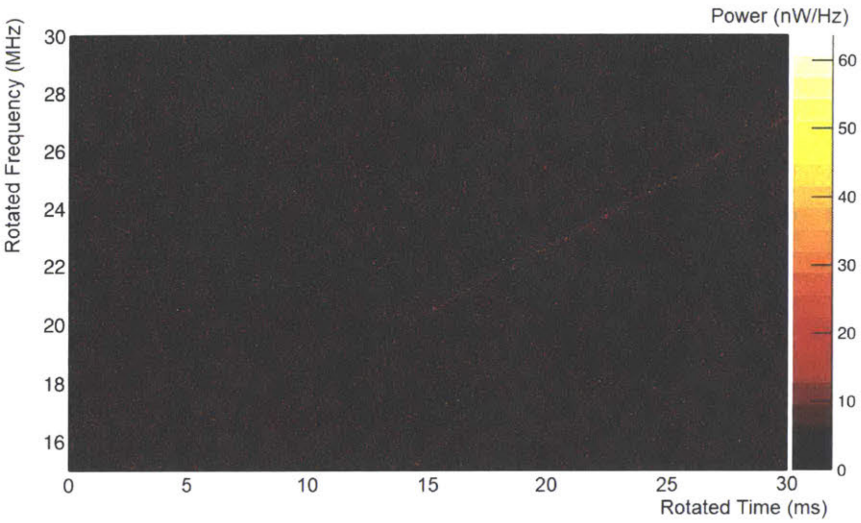


Figure 6.7: Spectrogram of a simulated CRES event rotated with the FrDFT by $\alpha = 0.002$. The transform has marginally deteriorated the spectrogram.

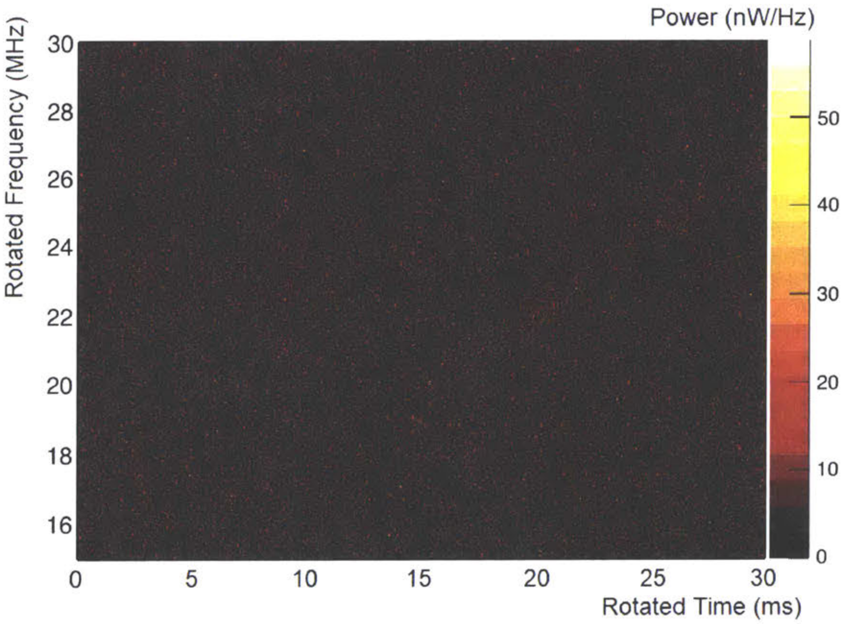


Figure 6.8: Spectrogram of a simulated CRES event rotated with the FrDFT by $\alpha = 0.01$. The transform has substantially deteriorated the spectrogram.

6.A Appendix

We would like to show that the expression for $F(\alpha)$ given by Equation 6.42 behaves as we expect in the limiting values of the rotation angle α; specifically, that $F(0) = I_N$ and $F(\pi/2) = B$. It is quite straightforward to show that with $\alpha = 0$ we recover the identity; both q_1 and q_2 are zero in this case, and thus $A_{jk} = \delta_{jk}$ we may simply replace it with the identity. Then the expression for F is:

$$F = A(0)B^{-1}A(0)BA(0) = B^{-1}B = I_N \tag{6.43}$$

Alternatively we may look at F_{jk} directly, which simplifies to the following when $q_1 = q_2 = 0$:

$$F_{jk} = \frac{1}{N}\sum_{m=0}^{N-1} \exp - \frac{2\pi i}{N} m(k-j) = \frac{1}{N}\sum_{m=0}^{N-1} \zeta^{2m(k-j)} \tag{6.44}$$

Which is precisely equal to δ_{jk} via the geometric series:

$$\begin{aligned} F_{jk} &= \frac{1}{N}\sum_{m=0}^{N-1} \left(\zeta^{2(k-j)}\right)^m \\ &= \frac{1}{N}\left(\frac{1}{1-\zeta^{2(k-j)}} - \frac{\zeta^{2(k-j)N}}{1-\zeta^{2(k-j)}}\right) \\ &= \frac{1}{N}\frac{1}{1-\zeta^{2(k-j)}}\left(1-\zeta^{2N(k-j)}\right) \\ \Rightarrow F_{jk} &= \frac{1}{N}\frac{0}{1-\zeta^{2(k-j)}} \end{aligned} \tag{6.45}$$

where the last equality follows from the fact that $\zeta^{2Ns} = 1$ for any integer s. The quantity $k-j$ is bounded both above and below by $-N+1 \leq k-j \leq N-1$; therefore the denominator only vanishes in the trivial case $j = k$, where it is irrelevant since clearly $F_{jj} = 1$. When $j \neq k$, the denominator is finite and $F_{jk} = 0$, thus $F_{jk} = \delta_{jk}$ for $\alpha = 0$.

The second statement we would like to show is that for a rotation angle of $\pi/2$, we recover the ordinary DFT: $F(\pi/2) = B$. With $\alpha = \pi/2$ we have $q_1 = q_2 = 1$, and F_{jk} simplifies to:

$$F_{jk} = \frac{1}{N}\zeta^{j^2+k^2}\sum_{m=0}^{N-1} \zeta^{m^2+2m(k-j)} \tag{6.46}$$

Combining both factors of ζ, we have in the exponent:

$$j^2 + k^2 + m^2 + 2m(k-j) = (m+k-j)^2 + 2jk \tag{6.47}$$

The last term is precisely what we need to isolate B_{jk}:

$$\begin{aligned} F_{jk} &= \frac{1}{N}\zeta^{2jk}\sum_{m=0}^{N-1}\zeta^{(m+k-j)^2} \\ \Rightarrow F_{jk} &= B_{jk}\frac{1}{\sqrt{N}}\sum_{m=0}^{N-1}\zeta^{(m+k-j)^2} \end{aligned} \tag{6.48}$$

Thus, up to a normalization of $\sqrt{N}$ and an overall phase, we must only show that the sum in the above equation reduces to 1, i.e. it is independent of j and k. To show this, I will prove and then make use of the following trick. Consider a sum S of the form:

$$S = \sum_{s=k}^{k+N-1}\zeta^{s^2} \tag{6.49}$$

and define $g \equiv \lfloor k/N \rfloor + 1$. We separate the sum into two parts: the first contains the terms for which $s < gN$, and the second contains the remaining terms where $s \geq gN$. It is clear from the definition of g that the former is never empty and the latter is empty only when k is divisible by N. We may assume for the moment that k is not divisible by N, and return to this case shortly. The separated sum is:

$$S = \sum_{s=k}^{gN-1}\zeta^{s^2} + \sum_{s=gN}^{k+N-1}\zeta^{s^2} \tag{6.50}$$

Next, we shift the second part with $t = s - N$:

$$S = \sum_{s=k}^{gN-1}\zeta^{s^2} + \sum_{t=(g-1)N}^{k-1}\zeta^{(t+N)^2} \tag{6.51}$$

$$= \sum_{s=k}^{gN-1}\zeta^{s^2} + \sum_{t=(g-1)N}^{k-1}\zeta^{t^2}\zeta^{2Nt}\zeta^{N^2} \tag{6.52}$$

and recall that $\zeta^{2Ns} = 1$ for any integer s, or equivalently that $\zeta^{Ns} = 1$ for any even integer s. Then, provided only that N is even the sum simplifies to:

$$S = \sum_{s=k}^{gN-1} \zeta^{s^2} + \sum_{t=(g-1)N}^{k-1} \zeta^{t^2} \tag{6.53}$$

and the two parts can be recombined into one:

$$S = \sum_{s=(g-1)N}^{gN-1} \zeta^{s^2} \tag{6.54}$$

Note that if k is instead divisible by N, the above equation is identical to Equation 6.49. Finally, we again shift the summation variable this time with $s \to s - gN + N$. Using the same property of ζ, we simplify the result and compare it with the initial form of S (Equation 6.49):

$$S = \sum_{s=k}^{k+N-1} \zeta^{s^2} = \sum_{s=0}^{N-1} \zeta^{s^2} \text{ for any integer } k \tag{6.55}$$

Therefore, the sum is independent of the range of s so long as it contains exactly N terms. This relationship is ultimately a simple consequence of the fact that ζ^{s^2} is periodic in s with period N, and thus any sequence of N terms will sum to the same value. We can also rewrite the above equation with k in the exponent:

$$\sum_{s=0}^{N-1} \zeta^{s^2} = \sum_{s=0}^{N-1} \zeta^{(s+k)^2} \text{ for any integer } k \tag{6.56}$$

Let us now apply this trick to the expression for F_{jk}. From Equation 6.48 we have $s = m + k - j$ which we may replace via Equation 6.56 with simply $s = m$ to obtain:

$$F_{jk} = B_{jk} \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} \zeta^{m^2} \tag{6.57}$$

and as desired, the j, k dependence is now completely isolated to B_{jk}:

$$F_{jk} = \omega \; B_{jk} \text{ with } \omega = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} \zeta^{m^2} \tag{6.58}$$

All that remains is to show ω has unit norm. This is easily accomplished with the same trick; first note that ω^* has the same form as ω with the sign of the exponent flipped since $(\zeta^s)^* = \zeta^{-s}$. The norm squared of ω is:

$$|\omega|^2 = \frac{1}{N}\sum_{j=0}^{N-1}\zeta^{j^2}\sum_{k=0}^{N-1}\zeta^{-k^2} \tag{6.59}$$

We displace the exponent in the sum over k by $-j$:

$$\begin{aligned}
|\omega|^2 &= \frac{1}{N}\sum_{j=0}^{N-1}\zeta^{j^2}\sum_{k=0}^{N-1}\zeta^{-(k+j)^2}\\
&= \frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}\zeta^{j^2-k^2-j^2-2jk}\\
&= \frac{1}{N}\sum_{k=0}^{N-1}\zeta^{-k^2}\sum_{j=0}^{N-1}\zeta^{2j(0-k)}\\
&= \frac{1}{N}\sum_{k=0}^{N-1}\zeta^{-k^2}N\delta_{0k}\\
&= \frac{1}{N}N\zeta^0\\
\Rightarrow |\omega|^2 &= 1
\end{aligned} \tag{6.60}$$

Thus, up to an overall phase $\omega = e^{i\alpha}$ we have recovered the ordinary DFT matrix and obtained the desired result:

$$F_{jk}(\pi/2) = B_{jk} \tag{6.61}$$

Chapter 7

Conclusions and Outlook

In this work, we have discussed in detail several new advancements in CRES signal processing across a wide range of analysis topics. Phases I and II of Project 8 have revealed valuable information about the structure of CRES tracks which has served as the motivation for much of the analysis developed here. Early Phase II data demonstrated the powerful effects of waveguide mode coupling and the reflector, which are both sensitive to the energy region of interest and the magnetic trap geometry. The likelihood clustering algorithm presented in Chapter 4 is one of many avenues explored to reformulate primary track finding for Phase II, in addition to optimization of the Phase I reconstruction. On recent Phase II data and with minimal optimization, this clustering boasts results comparable in quality to the Sequential Track Finder. In Chapter 5, we outlined the effects of pitch angle considerations based on the phenomenological model to explain several phenomena present in both Phase I and Phase II. With machine learning techniques, we have developed several models to classify Phase I tracks according to the associated pitch angle range and frequency band order with very good accuracy. This classification scheme lays the necessary groundwork for a more sophisticated analysis to fully account for pitch angle effects. In Chapter 6, we discussed exploratory studies of the chirp transform and fractional Fourier Transform as a more effective means to extract localized signal power from CRES tracks.

Through all of these studies, we have learned that a full description of an electron in a CRES experiment requires much more than only the cyclotron orbit. Axial motion in the trap and the resultant doppler shifts create sideband power deposition, as demonstrated in data and numerous simulations. Imperfections in the magnetic trapping field and the background field lead to a slow precession of the cyclotron orbit guiding center, and even slower variations of the main carrier frequency as revealed with the chirp transform. Table 7.1 enumerates the many frequency scales which characterize these effects, including the (rather surprising) rate of photon emission as calculated from the Larmor power and $E/\hbar = 2\pi \cdot 26$ GHz. Also included in the table are the associated timescales and energy sensitivity scales of these effects, as well as anticipated Project 8 precision benchmarks for comparison. In summary, a full understanding of CRES dynamics spans a very wide range of frequencies from the cyclotron orbit down at least to the 100 Hz scale.

Frequency	**Time**	**Energy**	**Process**
26 GHz	38 ps	530 keV	Cyclotron orbit
100–200 MHz	5–10 ns	2–4 keV	Data acquisition
50 MHz	20 ns		Photon emission
20–40 MHz	25–50 ns	400–800 eV	Axial motion
750 kHz	1.3 μs	15 eV	Phase II endpoint precision
200–500 kHz	2–5 μs	4–10 eV	Axial motion variation
100 kHz	10 μs	2 eV	Current neutrino mass limit
24 kHz	41 μs	0.5 eV	Spectrogram resolution
10–20 kHz	50–100 μs	0.2–0.4 eV	Demonstrated chirp precision
10 kHz	100 μs	0.2 eV	KATRIN sensitivity
5–10 kHz	100–200 μs	0.1–0.2 eV	Radial gradient precession
2 kHz	0.5 ms	40 meV	Phase IV sensitivity
500 Hz	2 ms	10 meV	Linear chirp model limit
100 Hz	10 ms	2 meV	Mainband track precession

Table 7.1: Magnitude scales of several phenomena related to CRES measurement, Project 8, and direct measurement of the neutrino mass. The energy scale associated to the photon emission rate is omitted to avoid confusion with the photon energy.

Project 8 is now preparing to conclude Phase II with a 100-day tritium data campaign; this is expected to yield approximately 15,000 CRES events and an endpoint precision of ~15 eV. The next phase has the ambition to improve this precision to the eV scale, where it will be competitive with current limits on the neutrino mass from direct measurement. A comprehensive understanding of CRES signal characteristics will be essential to achieve this goal, and with the work presented here we have begun to make use of that understanding. It remains to extend this work toward a grand Phase III analysis which (a) confidently reconstructs primary tracks with accurate start and end locations, (b) incorporates the classification scheme to perform pitch angle corrections, and (c) utilizes as much information as possible to group tracks into events which describe all of the signal power attributable to one electron. To this end, there are still several open-ended areas and questions which must be explored further:

- Optimization of the likelihood clustering, and incorporation of some or all of it with existing primary track finding techniques. The likelihood clustering was designed to be more versatile and accessible than current methods, and to some extent this is already demonstrated by its good performance on Phase II data. The configurable parameters of this analysis are well-documented and were discussed at some length in Chapter 4. Can the likelihood metric or a similar approach benefit the existing STF procedure? How can the reconstruction quality improve, especially of short tracks, with better-optimized parameters?
- The classification scheme must be extended to improve performance on Phase II data, and potentially again for future Phase III data. Does the projected spectrum and do the associated classification features have good discrimination power for Phase II tracks? Do the slope and power correlations?
- Pitch angle corrections must be implemented in some form, preferably within Katydid. We have demonstrated that even with perfect classification, the difference between the detected mainband frequency and the true cyclotron frequency can exceed even the precision goals of Phase II.

Properly utilizing this information to correct for pitch angle effects is a step which cannot be ignored.

- Intelligent event building which makes use of the phenomenological model and the classifier information. When pitch angle corrections and a more developed classifier are in place, an overhaul of the event builder to incorporate these changes will be a valuable next step. This procedure could also use a machine-learning approach.
- Can the chirp transform be used effectively as a part of secondary track analysis to improve the start frequency precision? What about other methods?
- We have shown that in the case of an idealized CRES signal, the limiting factor of the chirp transform precision is the start time. What potential exists in independent methods to determine the start time to an improved level of precision? What are the other limiting factors?

With these items (and others outside the scope of this work) further investigated and questions answered, Project 8 can develop the necessary analysis to move from a demonstration of new technology to a competitive scientific result. The CRES measurement provides a wealth of information about trapped electrons which we have only begun to thoroughly understand. By building upon the work presented here, future analysts in Project 8 can push the technique to a new standard, achieve the late-stage goals of the project, and work toward a truly historic scientific accomplishment.

Bibliography

[1] H. Davoudiasl et al. *The New Minimal Standard Model.* arXiv:hep-ph/0405097v2. (2005)

[2] R. Davis et al. *Search for Neutrinos from the Sun.* Phys. Rev. Lett. **20**, 1205. (1968)

[3] C. Ignarra. *Recent MiniBooNE Results and the Status of Sterile Neutrinos.* Presented at the Harvard LPPC Seminar. (2013)

[4] Y. Fukuda et al (The Super-Kamiokande Collaboration). *Measurements of the Solar Neutrino Flux from Super-Kamiokande's First 300 Days.* Phys. Rev. Lett.**81**, 1158-1162. (1998)

[5] Q. R. Ahmad et al (SNO Collaboration). *Measurement of the rate of* $\nu_e + d \rightarrow p + p + e^-$ *Interactions Produced by* ^8B *Solar Neutrinos at the Sudbury Neutrino Observatory.* Phys. Rev. Lett. **87**, 071301. (2001)

[6] M. Tanabashi et al. (Particle Data Group). *Neutrino Masses, Mixing, and Oscillations.* Phys. Rev. D **98**, 030001. (2018)

[7] F. Capozzi et al. *Global constraints on absolute neutrino masses and their ordering.* Phys. Rev. D **95**, 096014. (2017)

[8] C. Kraus et al. *Final Results from phase II of the Mainz Neutrino Mass Search in Tritium* β *Decay.* Eur. J. Phys. C **40**, 447. (2005)

[9] V. N. Aseev et al. *Upper limit on the electron antineutrino mass from the Troitsk experiment.* Phys. Rev. D **84**, 112003. (2011)

[10] C. Weinheimer, KATRIN Collaboration. *KATRIN, a next generation tritium β decay experiment in search for the absolute neutrino mass scale.* Progress in Particle and Nuclear Physics, **48**(1). (2002)

[11] N. Steinbrink et al. *Neutrino mass sensitivity by MAC-E-Filter based time-of-flight spectroscopy with the example of KATRIN.* New J. Phys. **15**, 113020. (2013)

[12] B. Monreal and J. A. Formaggio. *Relativistic Cyclotron Radiation Detection of Tritium Decay Electrons as a New Technique for Measuring the Neutrino Mass.* Phys. Rev. D **80**, 051301. (2009)

[13] A. Picard et al. *Precision measurement of the conversion electron spectrum of 83m Kr with a solenoid retarding spectrometer.* Zeitschrift f´ur Physik A Hadrons and Nuclei **342**(1), 7178. (1992)

[14] A. A. Esfahani et al (Project 8 Collaboration). *A Study of Electron Radiated Power in Cyclotron Radiation Emission Spectroscopy Experiments.* Phys. Rev. C **99**, 055501. (2019)

[15] N. S. Oblath et al. *project8/katydid: Docker container build.* DOI: 10.5281/zenodo.2621293

[16] M. Frigo and S. G. Johnson. *The Design and Implementation of FFTW3.* Proceedings of the IEEE **93**(2), 216-231. (2005)

[17] M. Guigue. *Status of the phase analysis method for tracks start frequency extraction.* Internal Project 8 note on Basecamp. (2018)

[18] A. Lindman. *Description of the tEMST Algorithm.* Internal Project 8 note on Basecamp. (2018)

[19] L. Saldaña and M. Guigue. *Pitch Angle Distributions of Electrons in Project 8.* Internal Project 8 report. (2017)

[20] G. Rybka. *Sideband width and radial position.* Internal Project 8 note on Basecamp. (2016)

[21] P. Slocum. *Simulation Update: Comparing Magnetron Trajectories in Kassiopeia with Power Measurements in Locust.* Internal Project 8 note on Basecamp. (2017)

[22] C. Cortes and V. Vapnik. *Support-Vector Networks.* Machine Learning **20**(3), 273. (1995)

[23] C.-C. Chang and C.-J. Lin. *LIBSVM: A Library for Support Vector Machines.* ACM Transactions on Intelligent Systems and Technology **2**, 27:1. (2011)

[24] D. E. King. *Dlib-ml: A Machine Learning Toolkit.* Journal of Machine Learning Research **10**, 1755. (2009)

[25] F. Pedregosa et al. *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research **12**, 2825. (2011)

[26] R. G. H. Robertson. *Examination of the calorimetric spectrum to determine the neutrino mass in low-energy electron capture decay.* arXiv:1411.2906v2. (2015)

[27] B. Carpenter et al. *Stan: A Probabilistic Programming Language.* Journal of Statistical Software **76**(1). (2017)

[28] M. Guigue. *Reconstruction algorithm optimization.* Internal Project 8 note on Basecamp. (2016)

[29] M. Guigue and E. Zayas. *Study and optimization of the reconstruction algorithm.* Internal Project 8 report. (2019)

[30] A. Ashtari Esfahani et al (Project 8 Collaboration). *Locust: Software Interface Between Calculated Electromagnetic Signals and Simulated RF Detection.* In preparation.

[31] A. Ashtari Esfahani et al (Project 8 Collaboration). *Reconstruction of Electron Events from Cyclotron Radiation Emission Spectroscopy for the Project 8 Experiment.* In preparation.

[32] D. Furse et al. *Kassiopeia: A Modern, Extensible C++ Particle Tracking Package.* arXiv:1612.00262. (2016)

[33] C. Candan et al. *The Discrete Fractional Fourier Transform.* IEEE Transactions on Signal Processing **48**, 5. (2000)

[34] H. M. Moya-Cessa and F. Soto-Eguibar. *Discrete fractional Fourier transform: Vandermonde approach.* arXiv:1604.06686. (2016)

[35] J. Garcia et al. *Fractional-Fourier-transform calculation through the fast-Fourier-transform algorithm.* Applied Optics **35**, 35. (1996)

[36] J. W. Cooley and J. W. Tukey. *An algorithm for the machine calculation of complex Fourier series.* Mathematics of Computation **19**, pp. 297-301. (1965)