Advancements in Online Monitoring and Visualization for SpinQuest in Experimental Nuclear Physics

Jordan Daniel Roberts
St.Pauls North Carolina

A Thesis presented to the Graduate Faculty of the University of Virginia in
Candidacy for the Degree of
Master of Science

Department of Physics
University of Virginia
Charlottesville
United States of America
August 11, 2023

Abstract

SpinQuest will measure the sea quarks Sivers asymmetry, with a target transversely polarized with respect to the incoming 120 GeV proton beam and the hall floor. The Sivers asymmetry describes the distribution of the unpolarized quarks inside a polarized nucleon. SpinQuest will explore the properties of the nucleon's spin composition by investigating the correlation of the light antiquarks' motion relative to the nucleon spin. A software package has been developed to display particle tracks at the hit level of the detector, breaking up the reconstruction process for the use of online monitoring. Online monitoring of the target, the detector, and the reconstruction will play a vital role in ensuring optimal performance to obtain the highest figure of merit possible given the experimental circumstances. This novel monitoring system will enhance the debugging process during commissioning and data acquisition through the ability to make use of machine learning pattern recognition techniques and anomaly detection. This scheme promises to aid target operators and shift takers during the two-year-long production runs when it begins in the Fall of 2023 at Fermilab. We were able to create a demonstration of organizing the tracklets that can be streamed into a machine-learning model. These results depict stations two and three, where the muon from the target is easier to identify in the reconstruction process.

Acknowledgement

I would like to give thanks to my team and advisors who have supported me through this academic journey such as Dustin Keller and Donal Day. I am thankful for the polarized target group for taking me in. I am thankful to the University of Virginia to give me the opportunity to do this research and be a part of the physics community which I would also like to thank. I would like to thank the community around me for coming to my practice talks. I would like to thank my parents for believing in me. Lastly, I would like to thank my wife, Haley, for being there next to me. Thank you.

Contents

1 Introduction

The fundamental quest to explore the nature of matter and the interactions governing it has driven human curiosity since ancient times. From the discovery of fire to the exploration of subatomic particles, the pursuit of knowledge has been an innate characteristic of sentient beings. In the realm of nuclear physics, the journey to comprehend the proton's intricacies has been a significant undertaking, leading us to the present-day SpinQuest experiment.

The primary focus of this thesis is to showcase the ongoing efforts of online monitoring and visualization within the context of SpinQuest. Additionally, it serves as a guide for future students, helping them navigate the intricacies of experimental setups for future work. Over time, our research group has accumulated invaluable knowledge, which has been continuously updated, passed down, and embedded within our experimental systems. As a result, this work aims to satisfy the reader's curiosity and provide comprehensive answers to pertinent questions about the SpinQuest experiment and the need for online monitoring.

At its core, the pursuit of understanding the proton's nature began with William Prout's hypothesis in 1815, suggesting that all atoms were composed of multiples of hydrogen [28]. This initial insight eventually led to Rutherford's groundbreaking gold foil experiment in the early 1900s, which unveiled the nucleus and paved the way for deeper exploration of the proton's composition.

Through the course of numerous advancements and experiments, we have come to understand that the proton comprises point-like particles called quarks, governed by the strong force mediated through gluons both discovered by the Stanford Linear Collider(SLAC)[19]. Despite these remarkable discoveries, we encountered a mystery known as the "proton spin crisis" in 1987, where it was revealed that the valence quarks alone did not account for the proton's complete spin by the European Muon Collaboration. They concluded that the proton quarks were responsible for $(14 \pm 9 \pm 21)\%$ of the total spin, which is substantially below the parton model prediction [6]. This was the kickstart to spin physics, the subset of physics concerned with the dynamics that occur inside the proton to create its spin, leading to other experiments to understand the spin structure of the hadron such as SLAC E155 [5]. To resolve this enigma, we embarked on a shared goal among nuclear physicists today – to delve deeper and uncover the missing spin. With the STAR experiment, The Relativistic Heavy Ion Collider (RHIC) determined that the valence quarks hold around 30% of the spin [8]. We discovered more accurate ways to describe the total spin of the hadron using sum rules summing over the quarks and gluon spin plus orbital angular momentum. This endeavor lies at the heart of the SpinQuest experiment. SpinQuest will probe the sea quarks with the Drell-Yan process to prove that the sea quarks orbit around the spin axis of the nucleon [4]. Sea quarks are the quarks that appear from the gluon field that, through the strong interaction, hold the valence quarks together. This process is called gluon splitting where a gluon decays into a quark-antiquark pair. These quarks exist for short time spans before turning back into gluons, however, occur frequently enough to have an effect on the dynamics of the hadron. Experiments such as COMPAS have done similar experiments on the valence quarks suggesting that the asymmetry for the sea quarks will be small and therefore will require precise measurement [25]. This asymmetry studied is the Sivers Asymmetry, a left-right asymmetry caused by the interaction during the collision originally used to explain large single-spin asymmetries [22]. SpinQuest will produce the first measurement of Drell-Yan proton-proton Sivers asymmetry [4]. A challenge SpinQuest faces is the possibility of false asymmetries caused by effects outside the collision. As the analysis of the asymmetries depends on the reconstruction, therefore, software built to display reconstruction is vital to prevent false asymmetries. By displaying the reconstruction at every step we can better localize the sources of the false asymmetry.

To study the Sivers asymmetry we employ the fundamental principle of deconstruction and reconstruction. By colliding hadrons – the particle class to which the proton belongs – within accelerators, we generate millions of explosive events. These events are captured and analyzed by various detectors, generating electrical signals that are recorded as crucial data points. Software packages are then utilized to reconstruct the particles of interest, eliminating irrelevant information based on initial data allowing us to obtain the four momenta. The four-momentum allows us to obtain the cross-section and from there we can use the Sivers function. The Sivers function is a correlation of the transverse momentum of an unpolarized parton with the spin of a transversely polarized nucleon [4]. As described in the background section, a non-zero Sivers function will lead to a non-zero orbital angular momentum of the quark sea.

An essential aspect of this research is the need for real-time reconstruction and monitoring, enabling us to optimize the experimental performance and achieve the highest possible figure of merit under the given circumstances. The ability to visualize the health of reconstruction during the time interval between event production and data recording is invaluable, aiding operators and shift takers in addressing issues during experiment runs. As one of SpinQuests goals is to study the Sivers asymmetry, a challenge is detecting false asymmetries that could occur during the experiment. These false asymmetries can come from internal consequences such as the cooling of the target malfunctioning or external consequences such as the time of day. Therefore, having the ability to analyze the data from the detector to quickly detect these asymmetries is of great importance. Consequently, this thesis seeks to delve into this critical challenge and develop a software package that can efficiently display relevant details such as the location of the particle within each station. To aid in the speed of the reconstruction we plan to utilize machine learning to make predictions of the slopes and intercepts of the particle path through the detector. It is the goal that the combination of pattern recognition to find issues from the reconstruction with AI, and visualization of reconstruction displayed for shift workers, challenges that could occur can be quickly managed.

As we embark on this scientific journey, we endeavor to shed light on the enigmatic spin properties of the proton and contribute to the collective knowledge of nuclear physics. Through the advancements in online monitoring and visualization, we hope to catch anomalies within the experimental run, such as false asymmetries. Ideally, this process would be done quickly to display real-time reconstruction to aid in optimizing the performance of the experiment while achieving the highest figure of merit possible given the experimental circumstances. Being able to display the health of the reconstruction at different locations of the detector in the time between the production and detection of the particle would solve the challenges that SpinQuest faces. It is the time between the production of the event and the recording of the data that this thesis strives to analyze and lead to the creation of a software package to quickly perform analysis.

2 Background

2.1 Particle and Spin Physics

Quantum Chromodynamics (QCD) serves as the cornerstone of modern nuclear physics and constitutes the foundation of the standard model. The standard model characterizes fundamental particles as point-like entities interacting with the electromagnetic, weak, and strong forces through the different gauge bosons. QCD contributes a color charge of red, green, or blue and their anti-variant to each quark. Every quark and lepton has its corresponding antiparticle. Combining a pair of quarks/antiquarks will form mesons or grouping three quarks/antiquarks will create hadrons, such as the proton. Particle spin, a fundamental intrinsic property, can be classified into three states: spin 0 for particles like the Higgs, half-spin for fermions such as the quark and leptons that create matter, and integer-spin for bosons such as the gluon which mediates the strong interaction. The photon is the mediator of the electromagnetic force and the Z and W bosons are mediators of the weak force. The standard model, represented below, embodies these fundamental interactions. Fig. 1 shows the 6 quarks and 6 leptons with the 4 gauge bosons and Higgs [27]. More information on the standard model and QCD can be found in the chapters of Griffiths, Martin, and Thomson [10][15][23].

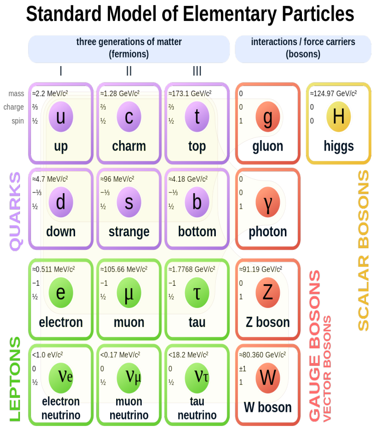


Figure 1: The Standard Model of Elementary Particles. Showing the Quarks in purple in the upper left box, the leptons in green inside the lower left box, and the gauge bosons in red and yellow to the right.

The property of spin is a type of quantum number intrinsic to all particles, combined with another quantum number, orbital angular momentum (OAM), the sum creates the total angular momentum of the hadron. To describe properties of the hadron we use sum rules which relate dynamic quantities such as the structure of the hadron to static quantities such as the angular momentum [18]. We will discuss two sum rules Jaffe-Manohar Sum Rule and Ji's Sum Rule that both connects the spin of the quarks and gluons to the OAM of both. Jaffe-Manhar connected the spin contribution summed over the quarks $\Delta\Sigma$ and gluon ΔG and their OAM components [12].

$$\Delta S = \frac{1}{2}\Delta\Sigma + \Delta G + L_q + L_g$$

Ji's decomposition is a gauge-invariant decomposition of the nucleon spin into the quark helicity $\Delta\Sigma$, the quark OAM L_q^Z, and the gluon contribution J_g^Z [13].

$$\Delta S = \frac{1}{2}\Delta\Sigma + L_q^Z + J_g^Z$$

Spin follows the same algebra as angular momentum, therefore adding two particles' spins together will result in a total spin. A natural assumption is that since hadrons have a spin of $\frac{\hbar}{2}$ the constitute spin of the hadron would sum up to equal $\frac{\hbar}{2}$. However, pioneering work by the EMC showed otherwise [6]. As mentioned in the introduction, the spin of valence quarks accounts for only a maximum of 30% [8]. Lattice QCD, a non-perturbative QCD model describing nuclear physics, estimates that 60% of the quark spin originates from the quarks themselves, while the remaining 40% comes from orbital angular momentum [7]. Nevertheless, spin physics is a relatively young field with upcoming experiments poised to make significant impacts. SpinQuest proposes that some of the missing spin resides within the quark sea, making it imperative to measure the spin structure at the quark sea rather than the valence quarks.

The hypotheses of the sea quarks having a contribution to the total spin of hadron is evident from QCD lattice calculations predicting sea quarks having a large contribution [2]. Experiments such as STAR have studied the spin of the gluon ΔG showing it to be near zero. However, the OAM contribution will not be known until the Electron Ion Collider (EIC) probes the gluon via Semi Inclusive Deep Inelastic Scattering or SpinQuest probing via $\frac{J}{\Psi}$ production.

To measure the specific properties of a hadron, it must first be probed to initiate an interaction. The output of this interaction can then be detected, and reconstruction allows us to reconstruct the particle tracks to obtain their four-momentum. The probe is what creates the interaction and the probe scale is how deep the probe can go. For instance, probing the valence quarks of the proton requires a probe scale small enough to interact with them. The probe scale is based on the size of the virtuality Q. The De Broglie wavelength tells us that a larger Q will give us a smaller probe scale λ. In processes like Deep Inelastic Scattering (DIS) or Semi-Inclusive Deep Inelastic Scattering (SIDIS),

$Q > 1$ allows the probe to enter the hadron. A SIDIS example in Fig.2 can be shown below where a hadrons quark collides with a lepton exchanging a virtual photon in the process [26].

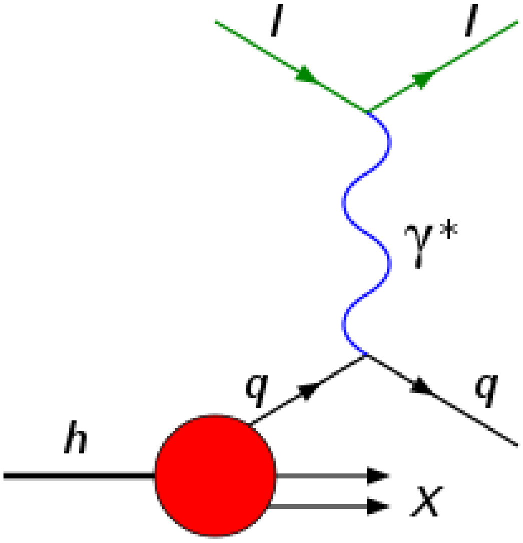


Figure 2: The SIDIS process of quark, q, from a hadron, h, colliding from a lepton, l, and exchanging a virtual photon γ^*. X is the fragmentation from the hadron.

The four-momentum resulting from probing the hadron provides insights into the parton structure and their dynamics within the hadron. The Parton Distribution Function (PDF) describes the longitudinal momentum probability distribution of partons inside the nucleon. PDF's give the number density of partons of type q inside a proton p [1]. Additionally, the Transverse Momentum Distribution (TMD) incorporates the partons' transverse momentum, carrying essential information about transverse spin and momentum inside the nucleus. TMDs unlike PDFs can link the parton spin,s_q to the hadron spin,S, and the transverse motion inside,$k_\perp$. The spin-dependant TMDs denoted as$f_q/p(x, k_\perp; s_q, S; Q^2)$ have 8 variants at leading order $1/Q$ of which exists a term $S(P \times K)$ which involves the Sivers function [1]. x is the fraction of the nucleon momentum carried by the parton [1].

SIDIS is a space-like process meaning that the space-time interval shown below, behaves like space $\Delta S^2 > 0$.

$$\Delta S^2 = (c\Delta t)^2 + \Delta x^2 + \Delta y^2 + \Delta z^2$$

This means that the initial state can not be correlated to the final state such as a time-like process, $\Delta S^2 < 0$. This paired with the fact that SIDIS undergoes fragmentation makes the process not ideal for quark sea probing. However, there has been substantial research conducted on probing the valence quark structure of the proton via SIDIS, such as the Common Muon and Proton Apparatus for Structure and Spectroscopy (COMPASS), or sea quarks such as Jefferson Lab, and HERMES [25][21][14]. Time-like processes, like the Drell-Yan process, have received less attention with current experiments only being that of STAR and COMPASS which contain low statics [9][24]. The Drell-Yan process involves the interaction of a quark and an antiquark, with the antiquark originating from the proton target. As depicted in Fig.3, this process is sensitive to sea quark production and is being probed for analysis [4].

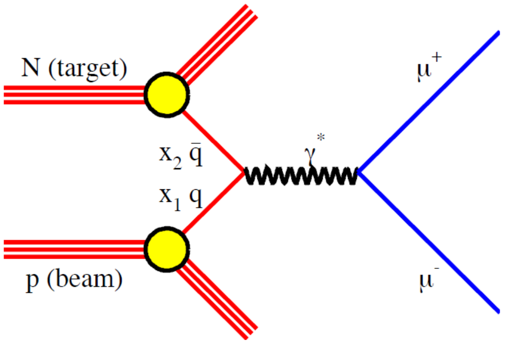


Figure 3: The Drell-Yan process is a time-like process where the photon determines the probe scale, but the probe itself is the proton beam, p. Antiquark $\overline{q}$ from the target N collides with quark, q from the beam. This creates a virtual photon, γ^* that decays into a lepton-antilepton pair μ^+ and μ^-. X_1 is the kinematics from the beam and X_2 is the kinematics from the target.

The Drell-Yan process involves the creation of a short-range virtual photon γ^* with virtuality Q, which quickly decays into a lepton-antilepton pair. Unlike SIDIS, the proton beam acts as the probe, while the probe scale is precisely determined by the virtuality. Q^2 is the square of the four-momentum transfer from the initial to the final lepton. The larger Q^2 is, the smaller the spatial region of size 1/Q that we are exploring.

Below shows the differential cross section σ of the Drell-Yan process. The differential cross section allows us to determine the probability that an interaction will occur. Due to the kinematics of the experiment, the Bjorken of the target x_t and the Bjorken of the beam x_b. The first term dominates, ensuring antiquark $\overline{q}$ will come from the target quark sea, and quark q will come from the beam.

$$\frac{d^2\sigma}{dx_b dx_t} = \frac{4\pi\alpha^2}{x_b x_t s}\Sigma e_q^2[\overline{q_t}(x_t)q_b(x_b) + q_t(x_b)\overline{q_b}(x_b)$$

Using the Sivers function combined with the Drell-Yan process, the resulting azimuthal angle can be linked with the rotation of the sea quarks. Measuring the Sivers function from Drell-Yan using the sensitivity of SpinQuest, we aim to discover the magnitude and sign of the sea quark Sivers function to determine if the quark sea has a non-zero OAM.

2.2 Sivers function

Significant transverse single-spin asymmetries have been observed during hadron-hadron collisions. D. Sivers proposed that these asymmetries could be explained by the azimuthal asymmetry of the transverse momentum when the hadron is polarized [22]. The Sivers function, A, represents the correlation of the transverse momentum of an unpolarized parton with the spin of a transversely polarized nucleon [4]. It is the ratio of the difference in the cross sections and the sum of cross sections from two different sides of the collision, left and right. As shown below, the UP arrow shows the spin polarization.

$$A = \frac{d\sigma \uparrow_L - d\sigma \uparrow_R}{d\sigma \uparrow_L + d\sigma \uparrow_R}$$

The denominator of this function is for normalization, it represents the unpolarized cross section. If the numerator is nonzero, it would be an indicator that there are some phenomena associated with the polarization direction of the hadron. This phenomenon can be concluded to be caused by a non-zero orbital angular momentum of the sea quarks. A visual representation of this would be picturing the sea quarks orbiting and a quark from the beam colliding with the quark sea. If the sea quarks are rotating in the same direction as the incoming quark it will be accelerated on interacting. It will decelerate if the rotation is opposite to the incoming quark. This would create a difference between the left and right sides. SpinQuest also wants to use the Sivers Function to test a fundamental prediction of QCD, which is that the Sivers function will undergo a sign change when switching from SIDIS to Drell-Yan[4].

$$f^{\perp qDY}(x, p_T^2) = -f^{\perp qSIDIS}(x, p_T^2)$$

SpinQuest aims to measure the left-right asymmetry of the parton distribution by probing the antiquarks in the quark sea. If SpinQuest confirms this hypothesis, the asymmetry can be linked to the quark sea's angular momentum around the proton's axis. A challenge of SpinQuest is the occurrence of false asymmetries, asymmetries that are not caused by the dynamics of the hadron. False asymmetries can be caused by changes in the time of day preferring some polarization states of the target over others during day or night. Hardware can cause false asymmetries, such as the alignment of the target arm or the cooling system of the target. Detecting false asymmetries is critical, making this work a vital contribution to the field of nuclear physics and spin physics. It is important to be able to detect and visualize these asymmetries when they occur as soon as possible. The ability to quality check the experiment in real-time is vital to ensure the data being collected has integrity. Therefore a fast online monitoring scheme is crucial to the success of the experiment.

2.3 The Experiment

For context, I will briefly discuss the SpinQuest experiment. SpinQuest was proposed to address the gaps left by previous experiments in attempts to solve the spin crisis. The predecessor to SpinQuest(E1039), called SeaQuest(E906), attempted to analyze the asymmetry of the concentration of the antiparticles up and down in the quark sea [16]. SpinQuest aims to probe the asymmetry of the sea quarks using a polarized fix target and Fermilab's unpolarized proton beam. The collision will produce a Drell-Yan event which produces detectable dimuons.

2.3.1 Fermilab beam

The importance of this work lies in the method and significance of understanding physics through collisions. Colliding particles allow us to probe nucleons and create a picture of what is happening at smaller scales, down to the femtometer range. We can obtain a greater resolution to investigate physics at said scales with a smaller probe. To obtain the proper resolution SpinQuest will utilize Fermilab's proton "Linac beam" which after being boosted enters the Main injector to go from 8 GeV to up to 120 GeV to enter the "Tevatron," a loop of 1000 superconducting magnets, all the way to the experiment's polarized fixed target. The collision of the proton with the polarized target generates a chaotic stream of particles that decay and scatter from the event vertex. This eventually results in detectable Dimuons that can be reconstructed. Reconstruction, in this context, refers to the process of organizing this chaotic data to analyze observables and gain a better understanding of the inner processes of the proton. During the event, the majority of the information obtained cannot be fully reconstructed back to the target such as partial tracks. These tracks are made of background particles not originating from the target and must be filtered. The approach taken in this thesis is to minimize the un-reconstructable particles by utilizing techniques from E1039's Fun4All software, K-Tracker, and accelerating computations through Python's Numba. Numba is a library that allows the use of the graphic processor to speed up computation. This software can also employ Artificial Intelligence fitting procedures to create a display of organized tracklets (particle paths through one station of the detector). This display will allow for quick extraction of information such as slope and intercept. The primary goal is to use this software to assist shift takers during experimental runs in identifying anomalies and resolving issues promptly. To reconstruct a complete particle track, data from different parts of the detector must be processed.

2.3.2 The Target System

The target system is a polarized fixed target contained within the LANL-UVA system which is shown in Fig.4. The target system is a 5T superconducting split coil magnet and the polarized targets are within the target insert attached to the target arm [4]. The polarized target beads contained within the target cups are NH_3 for the proton target, and ND_3 for the neutron target [4].

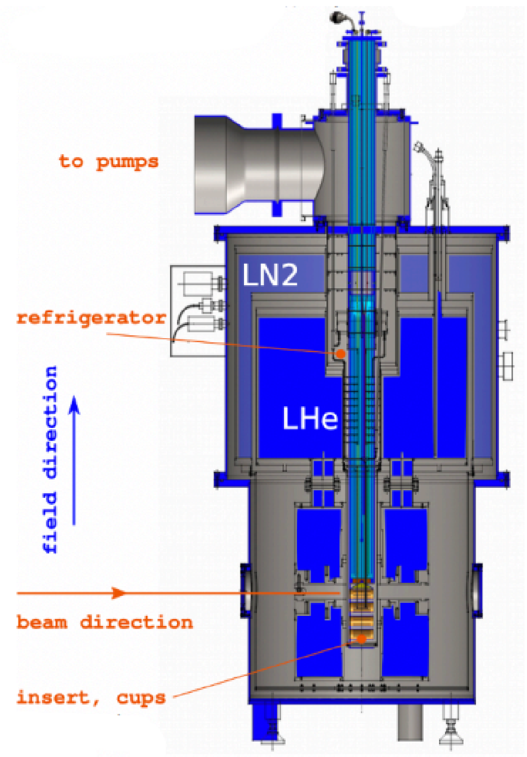


Figure 4: The cross sectional drawing of the polarized target system for SpinQuest. The beam comes from the left hitting the target insert.

2.3.3 SpinQuest Spectrometer

The spectrometer that will be used for SpinQuest was repurposed from the SeaQuest experiment with modifications made to the hodoscope to enable dark-photon detection. The complete spectrometer can be seen in Fig.5. While the goal is to provide detailed online monitoring systems throughout the entire detector, in this thesis our focus is on the drift chamber and hodoscope, which are integral to the reconstruction process.

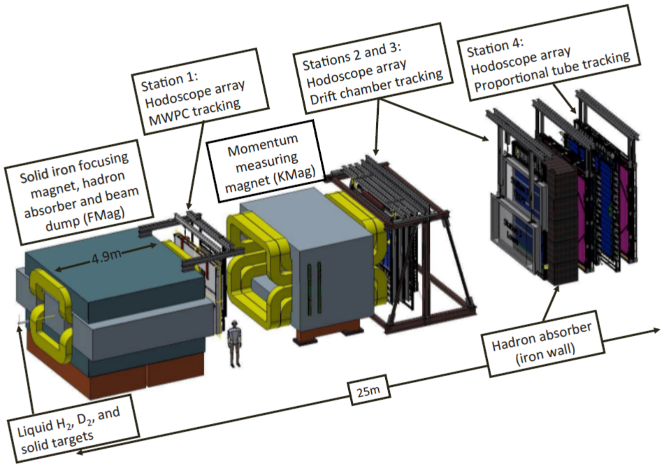

Station 1:
Hodoscope array
MWPC tracking
Stations 2 and 3:
Hodoscope array
Drift chamber tracking
Station 4:
Hodoscope array
Proportional tube tracking
Solid iron focusing magnet, hadron absorber and beam dump (FMag)
Momentum measuring magnet (KMag)
4.9m
Hadron absorber (iron wall)
25m
Liquid H_2, D_2, and solid targets

Figure 5: The SeaQuest Spectrometer. The proton beam comes along the Z axis from the left hitting the target and going into the F-Mag which acts as a dump. The next step is station 1 hodoscope and drift chamber. The K-mag is next which is used to find the muons momenta. Stations 2 and 3 come next with their hodoscopes and drift chambers. Last is the absorber and station 4.

2.3.4 The Magnets

The SpinQuest detector has two large dipole magnets. The upstream magnet is the F-Mag that generates a magnetic field of 1.8 Tesla. The purpose of the F-Mag is to create the acceptance of the detector allowing only high-momentum particles through [16]. The low-momentum particles are deflected out of the detector. The F-Mag also acts as the beam dump stopping the proton beam from entering the detector. Only muons and particles that decay into muons, such as pions, make it through the F-Mag. The muons from the pions are the main background source in the experiment. The K-Mag is the downstream magnet that creates a central magnetic field of 0.4 Tesla that is used to determine the momentum of the muon based on the deflection angle [16]. Both of the fields are oriented vertically so that the bend plane is horizontal [16].

2.3.5 The Drift Chamber and Hodoscope

The drift chamber, abbreviated to DC, is a matrix of wires sectioned into six planes as displayed in Fig.6. Its purpose is to determine the location of the muon with respect to the location of the DC station [16]. There are four stations within the detector with station one being downstream of the F-MAG and stations two and three being downstream of the K-Mag before the absorber. Beyond the absorber is station four which contains the proportion tubes. Fig.7 shows the top view of the detector where the black is the DC and the pink is the hodoscope. For reconstruction and this thesis stations two and three are the priority, as station one receives too much background from the dump, and station 4 is not needed to produce an accurate track.

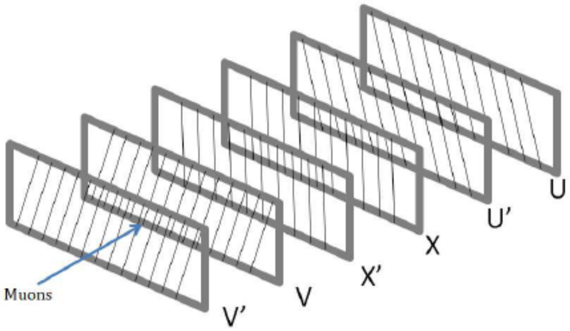


Figure 6: Displays that there are three pairs of planes for the prime and unprime detector planes. Muons enter in the horizontal plane.

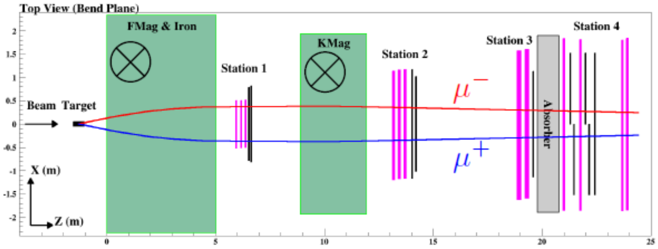


Figure 7: Displays the top view of the detector. From left to right: the target, F-Mag, Station 1, the K-mag, Stations 2 and 3, and lastly the absorber and station 4.

Each station consists of four types of wires which are the grounded wire, sensing wire, cathode wire, and field wire, all are shown in Fig.8. The DC takes advantage of the muon's charge by grounding the sensing wire and surrounding it with a cell of charged wires at -2400 volts. Additionally, -1500 volts are applied to the guard wires, creating an electric field between the potential wires and the sense wire. When a muon ionizes gas within the DC, the ions drift to the nearest sense wire, producing a readable voltage. The station is divided into three pairs of planes, the prime and unprime planes of

V, X, and U. The X plane consists of vertical wires with U and V tilted at a ±14° angle. This setup is because the x measurement is more important as the K-MAG bends the muon in the horizontal plane [16]. The prime plane is shifted by half a cell, which along with the combination of the U and V planes, allows the muon's position to be determined as left or right of the sense wire. The guard wire separates the electric fields of the prime and unprime planes. The DC determines the position via the drift time of the muon which it converts to drift distance. Each plane of the DC is labeled by a detector ID and each sense wire is labeled by an element ID. The hodoscope shares this labeling system with each detector ID having between 50-200 element IDs. This setup is used for each station in the DC and only the order of the planes change from station to station, which is described in the geometry section of the appendix.

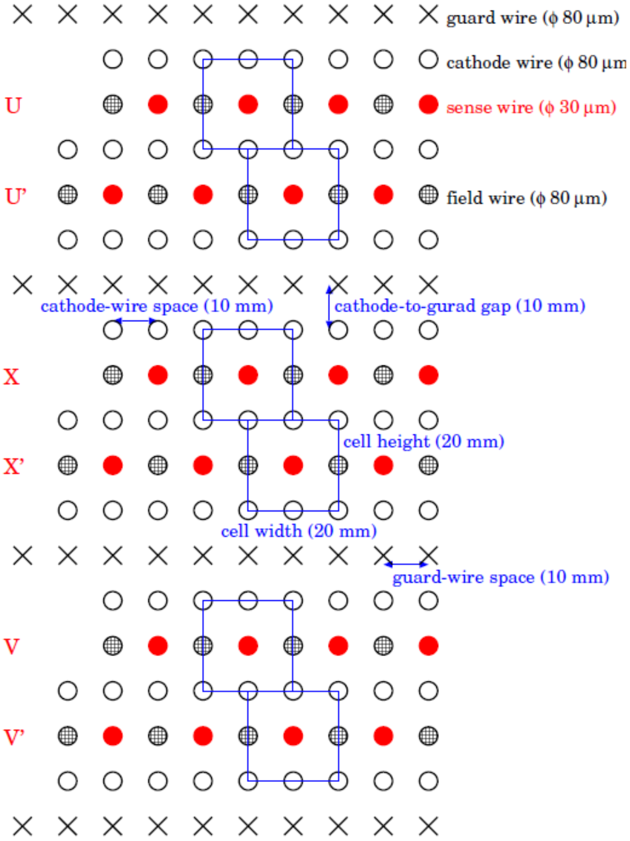


Figure 8: Displays that there are 3 pairs of planes for the prime and unprime detector planes and the different types of wires that make up the drift chamber.

After passing through the DC, the muon hits the array of hodoscopes. A hodoscope(Hodo) consists of paddles made of plastic scintillating tiles. When a particle interacts with the material, it produces a photon, which is collected and transported by a photomultiplier to create a readable voltage. The Hodo covers the X and Y planes of the detector, with the X-plane Hodo paddles aligned vertically and the Y-plane Hodo paddles aligned horizontally. The Hodo is designed to cover sections of the detector in the XY plane. Stations one, two, and four contain both X and Y Hodo planes. However, station three only contains X-plane Hodos due to the setup of station four. The purpose of the Hodo is to aid in finding the muon's position based on the start time of the ion drift. Together, the Hodo and DC create a picture of the muon's path through the detector, which can be reconstructed for analysis.

2.4 Software

2.4.1 The Monte-Carlo

As SpinQuest prepares to run, the work in this thesis used events generated via the Monte-Carlo method, a method of random sampling. This simulation was carried out using several software packages, such as PYTHIA, GEANT4, and ROOT, to produce a large set of events [3]. The Monte-Carlo simulations were run via the UVA Rivanna computing cluster and can be found on E1039's GitHub repository [3]. All of these packages worked together via a software package called Fun4All which allows modular capabilities to streamline the analysis. A part of this software package is the reconstruction software known as K-Tracker, which was used during SeaQuests experimental run to produce tracking information [3].

2.4.2 Reconstruction

The primary objective of the reconstruction process in nuclear physics is to investigate hidden observables within particles. To achieve this analysis, the reconstruction extracts crucial information, such as the four-momenta of particles directly resulting from collisions. However, it must also effectively identify and filter out the four-momenta originating from background events, unrelated to the collision of the target. The reconstruction plays a vital role in studying the Sivers asymmetry, and any inaccuracies during reconstruction, such as those caused by the cooling system which can lead to false asymmetries. Hence, it is imperative for the software used in reconstruction to analyze each phase rapidly and accurately.

The reconstruction process is based on the K-Tracker library from the Fun4All software package. K-Tracker, named after the Kalman-Filter method it employs to find the vertex of complete tracks, was initially used for reconstruction in the previous SeaQuest experiment. Originally written in Fortran, K-Tracker was later re-implemented in C++.

During usage, it became evident that this reconstruction approach was not suitable for the real-time display of complete tracks required for online monitoring. To address this, an optimization strategy was adopted. The logic of K-Tracker was analyzed, and the components most suitable for online monitoring were identified. The slower components were replaced, resulting in a more efficient reconstruction process.

K-Tracker achieves this optimization through a system of loops and cuts based on the detector's geometry and tracklets. Tracklets represent incomplete tracks constrained to one of the drift chamber stations. The key objective of K-Tracker is to iterate through individual hits within a station and retain only those hits that feasibly exist within a defined window, based on the detector's geometry.

For reference, the geometry used in the reconstruction is listed in the appendix of this thesis, along with an explanation of each component. Fig.9 illustrates the logic loop of K-Tracker utilized in Fun4All.

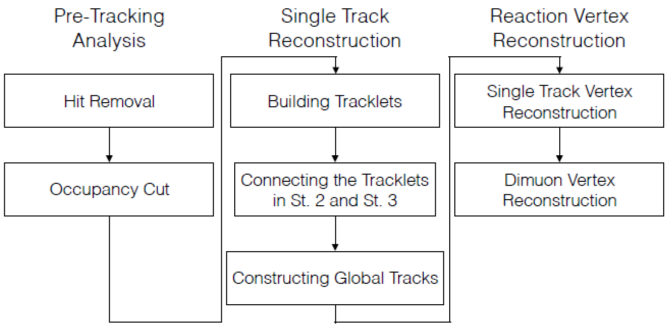

Pre-Tracking Analysis
Hit Removal
Occupancy Cut
Single Track Reconstruction
Building Tracklets
Connecting the Tracklets in St. 2 and St. 3
Constructing Global Tracks
Reaction Vertex Reconstruction
Single Track Vertex Reconstruction
Dimuon Vertex Reconstruction

Figure 9: The logic of K-Tracker

Once the selection of good hits is completed, K-Tracker combines them into tracklets at the station level. A tracklet consists of 6 hits that create the particle path through the station. It then merges tracklets from stations 2 and 3 to form the back tracklet. Subsequently, the vertex is determined by combining tracklets created in Station 1 using the sagitta ratio of Station 1 to Station 2. This results in a complete track known as the global track. At each level (station, backtrack, and global track), the slope, intercept, and momentum of the track are computed through several fits. This process is repeated for each hit that meets the initial occupancy cut of a single event, performed iteratively for every event in a spill, which typically consists of approximately 30,000 events. Unfortunately, in its current state, K-Tracker takes several days to process the reconstruction.

The significance of efficient reconstruction software cannot be overstated, especially in nuclear physics experiments where vast amounts of data need to be processed promptly. As SpinQuest studies asymmetries from reconstruction, false asymmetries are a challenge in this experiment. Efforts to further optimize K-Tracker or explore alternative reconstruction methods could greatly enhance the analysis speed and accuracy. This will bring us closer to more insightful discoveries in the field of nuclear physics.

2.4.3 Machine Learning

Machine learning, a prominent subfield of artificial intelligence (AI), has many applications that span diverse fields, including natural language processing, computer vision, recommendation systems, fraud detection, healthcare diagnostics, and autonomous vehicles. The fusion of nuclear physics and AI heralds the direction of the future, as it revolutionizes how computers handle complex tasks, leading to groundbreaking advancements in AI technologies. Machine learning, holds great significance in the domain of nuclear physics, enabling the development of algorithms and statistical models that facilitate computers to learn from data and make predictions without explicit programming. The primary objective of machine learning in this context is to create systems that continually enhance their performance as they encounter more data.

The process of applying machine learning to nuclear physics involves several essential steps:

1. Data Collection: To train the machine learning model effectively, relevant and representative data is gathered. In our research, we utilized Fun4Sim to generate simulated events devoid of any background. Subsequently, these events were modified to incorporate background tracks, simulating accurate events observed in experiments. The resulting data is then transformed into a hit matrix—an array of detectors and their elements—explained in detail in the following chapter.

2. Data Preprocessing: This crucial step involves cleansing, transforming, and preparing the data for model training. In our software, we detect and remove the background added during the simulation phase based solely on the detector's geometry. This process is discussed in more detail in the next section.

3. Feature Extraction: We identify and select pertinent features/patterns from the data to serve as inputs for the machine learning algorithm. In our study, we focus on the slopes and intercepts of complete tracks, derived from the truth values of the simulation. These features act as the answer key to the questions provided to the model. This way it can validate its predictions.

4. Model Selection: To address the specific problem and data nature, we carefully choose an appropriate machine learning algorithm. Our research primarily employs supervised learning, wherein the model is trained to predict answers based on provided ground truth values, answers. We define the number of layers and type of activation functions to accurately create a function that best fits our data. We define an optimizer, Adam, that carefully adjusts the weights of the function.

5. Training: By feeding the prepared data into the model, it learns patterns and relationships inherent in the data. During this training process, the model adjusts its internal parameters using the optimizer to minimize errors and enhance its performance on the given task.

6. Evaluation: To gauge the model's performance, we assess it using separate data, referred to as the validation or test set. This evaluation helps us estimate how well the model generalizes to new, unseen data.

7. Model Tuning: In this step, we fine-tune the model by adjusting hyperparameters or modifying its architecture to optimize performance further, aiming to achieve the best results for our specific task.

8. Deployment: Once the trained model demonstrates satisfactory performance, it is deployed for real-world use, allowing it to make predictions or decisions on new incoming data. The output is directed to a display package like Python's dearpygui.

3 Organization of the Hit Matrix for the Drift Chamber for SpinQuest

All of the software that was developed is located on my personal GitHub repository which can be found here: [20]. A collection of the simulated data utilized to create the background can be found on UVA's Rivanna Path:/project/ptgroup/Jay. From following the procedure detailed below you should be able to recreate the results shown in this thesis. This procedure will replicate some of the logical steps taken during the online monitoring process done during the SpinQuest run.

3.1 Simulating a realistic spill

In order to test and confirm the software's accuracy, we needed to create a spill with data that would resemble what is expected to be seen during SpinQuest's run. A robust simulation of the event for SpinQuest was built on the University of Virginia computing cluster known as Rivanna. This simulation is built with Fun4All and uses Fun4Sim, modified to reproduce E1039 events. The core software is updated to contain the updated Geant4 target files, which create the geometry of the target system and detector. The production mode specified in Fun4Sim is for Single muons and dimuons. K-Tracker is also modified to print out values useful for debugging the code. We produced 10 million events with a certain percentage making it through the detector acceptance, built into Fun4sim, around 60,000 events. We used 20,000 events for this specific demonstration.

From Fun4Sim, we produced events that contained what is called a complete track. A complete track is when a particle track goes completely through each station, meaning there are always six hits in each station. By itself, this does not accurately portray a real event. During an event, background from the dump and outside the detector is produced. One of the main challenges of this software package is to be able to find complete tracks and discard partial background tracks. The background that is expected is based on SeaQuest's hit matrices. A hit matrix is a format used by UVA that displays the individual hits by comparing element ID on the y-axis and detector ID on the x-axis as shown in Fig.10. The hit matrix proved to be useful in displaying the reconstruction of the event and determining the efficiency of the software. The hit matrix gives us a reference for the hit position in proximity to the wire elements in the chamber.

The sources of background are the random hits that appear occasionally and more frequently than the partial tracks. Partial tracks are defined as tracks that do not completely make it through the detector. The main source of partial tracks originates from pions that enter the dump and decay into muons. A complete track will have six hits within each station that are related to each other, and have several hits that match the hodoscope hits. Only muons that come directly from the target are reconstructable; therefore the background partial tracks must be filtered. To simulate an accurate background we used several events with complete tracks and inserted fragments of them into a single event. This script is built to be tuned to the different percentages of partial tracks seen in different stations. As an example, the station with the most partial tracks is the first station and the lowest is the final station. The partial track's depth into the detector is also randomized to be between two to six planes. A random number generator is also used to insert random hits into the stations to mimic cosmic muons. With the background built, we injected a complete track randomly pulled from the events to serve as our track candidate during the reconstruction. It is crucial to identify the track's location in order to quantify the success of the reconstruction during testing. With the spill created we studied the occupancy of the stations from SeaQuest to determine how accurate our simulation is. The work for the occupancy was done previously by Kenichi Nakano [17].

3.2 Organizing the Tracking information

We reproduced what we deemed as the most crucial and useful parts of K-Tracker in the programming language, Python3. We chose Python as it is the quickest to script in, is very accessible, and has a huge community that frequently updates it. With libraries such as Numba, the speed of the program becomes easily comparable with languages closer to Assembly, such as C. As explained in the background section, the motivation for studying K-Tracker is the requirement that the analysis can be done quickly for online monitoring. K-Tracker entangled in Fun4all is not fast enough to be suitable for online monitoring and visualization purposes. The answer to this challenge is optimizing the reconstruction

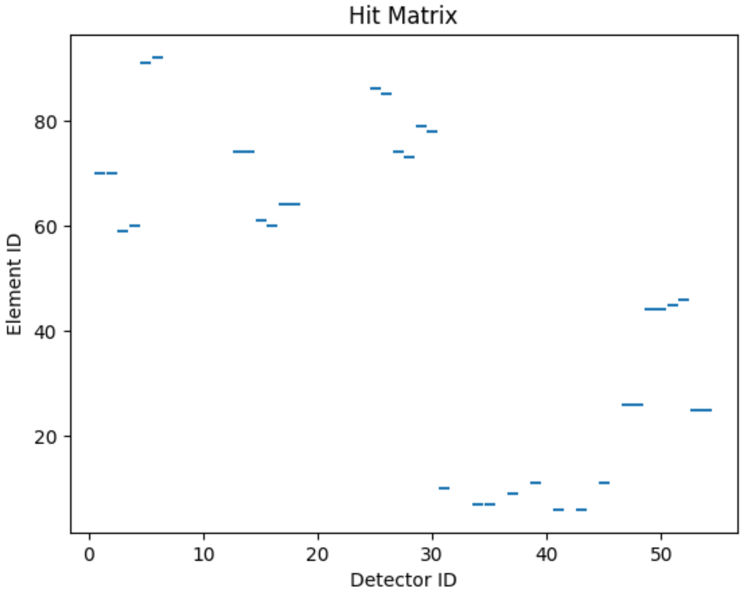


Figure 10: An event of a complete track for a single μ^- created by Fun4Sim. The Y-axis is the element ID which represents the section of the plane, and the X-axis is the detector ID, of the detector. Each station for a Single muon has six hits in the drift chamber. The drift chamber starts at detector ID 1 and stops at 30. The hodoScope goes from detector ID 31 to 46. For the reconstruction, only hodoscopes 31-40 are required. The proportional tube covers the remainder of the detector and is not used in this software.

with Numba GPU acceleration. The software package built for this work can be found on my GitHub [20]. The logic diagram, Fig. 11, details the setup and loop of the software to organize the tracks with the ability to optimize for GPU acceleration and interface display.

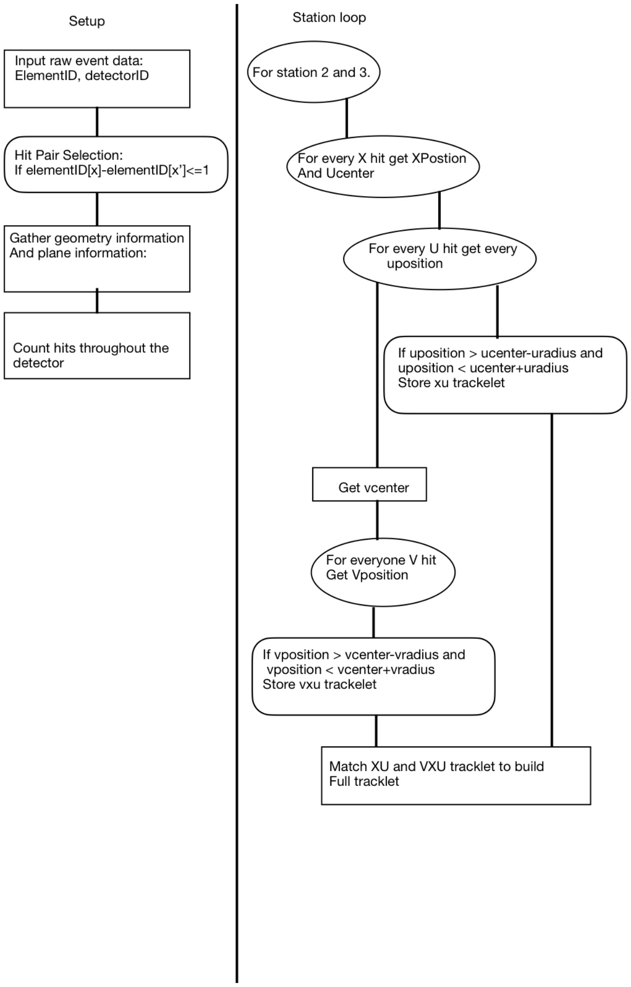

Setup
Input raw event data:
ElementID, detectorID
Hit Pair Selection:
If elementID[x]-elementID[x']<=1
Gather geometry information
And plane information:
Count hits throughout the
detector
Station loop
For station 2 and 3.
For every X hit get XPostion
And Ucenter
For every U hit get every
uposition
If uposition > ucenter-uradius and
uposition < ucenter+uradius
Store xu trackelet
Get vcenter
For everyone V hit
Get Vposition
If vposition > vcenter-vradius and
vposition < vcenter+vradius
Store vxu trackelet
Match XU and VXU tracklet to build
Full tracklet

Figure 11: The logic of the background minimizer. The data is imported and sorted into element id and detector ID and the hit pairs are found in each station. Then the station-by-station loop starts to find and construct tracklets. The station loop consists of creating windows based on the X plane hit position onto the U and V planes. Hits within these windows are kept.

As the Diagram shows, the program begins with the input being read in the form of the hit matrix, detector ID vs element ID. This information is raw data straight from the DAQ system of the detector. Once the hits are found the corresponding prime and unprime planes in the station are combined into hit pairs if their element ID does not differ by one. This step gets rid of single random hits from the background not coming from a real track as the offset of the prime plane provides insurance that the prime hit comes from its partner's unprime hit. Once the hit pairs are found the wire information of the station is determined. The plane ID is determined based on the detector ID and used to find the constants from the detector geometry. To sort the background from the hits used in the complete track, a window is projected from the x-hit pair candidate onto the U plane. This window has a Radius for the U plane based on the plane detector ID and geometry which formula is this:

$$URadius = \mid \frac{1}{2} * X_{WireSpan} * \sin(\theta_U) \mid +$$
$$T_X^{Max} \mid (Z_U - Z_X) \mid \cos(\theta_U) +$$
$$T_Y^{Max} \mid (Z_u - Z_X) \mid \sin(\theta_U) + 2 * Spacing + \delta$$

Where the $X_{WireSpan}$ is the length of the wire on the X plane. The θ_U corresponds to the angle of the U and V planes. The spacing is the distance between the wires, the Z position of the planes is labeled with Z, and δ is a term used for tuning. The maximum slopes T_X^{Max} and T_Y^{Max} are the maximum allowed slopes in the station. For the V plane, the Radius is different and the formula is

similar but without the δ:

$$VRadius = Spacing * 2 * \cos(\theta_U) +$$
$$\mid (Z_U + Z_V - 2 * Z_X) * \cos(\theta_U) * T_X^{Max} \mid +$$
$$\mid (Z_V - Z_U) * \sin(\theta_U) * T_Y^{Max} \mid + 2 * Spacing$$

The center of the window is based on the position of the wire of the unprime hit element ID and the detector ID, as well as the geometry of the detector. The formula utilized for the position of hit on the wire is:

$$WirePosition = (elementID - \frac{(N_{Elements} + 1)}{2}) * Spacing +$$
$$X_{OffSet} + X_0 * \cos(\theta_U) + y_0 * \sin(\theta_U) + \delta$$

Here the element ID is subtracted by the number of elements added by one and divided by 2. The X offset is the offset of the prime planes to the unprime planes. x_0 and y_0 are the plane intercepts.

With the hit pairs found, the number of hits is counted for each plane. The reconstruction loop starts with station two with the X plane, iterating over the number of X hits. The wire position of the X hit is found and is used to determine the center of a U window with the formula:

$$UCenter = XWirePosition * \cos(\theta_U)$$

With each U center, the X position is carried with it inside the array to match the hits together. The number of U hits is iterated over and the wire position of the U hit is determined. From this point, the UX tracklet is created by determining if the U-hit position is inside the window. This tracklet is a piece of a complete track that goes between plane X and U. This is done by comparing the hit position to the bottom of the window and the top of the window. If the hit is found to be inside the window the UX tracklet is saved with the corresponding U center and U position. In the same loop, V Center is created as it depends on U and X. The U center and U position is is saved with the V center and the formula is:

$$VCenter = 2 * UCenter - UWirePosition$$

The next stage of the loop iterates over the number of V hits. The position of the V hit is found and the comparison is made similar to the U plane. Once the V hits are selected, they are saved with the V, X, and U information. The final stage of the loop is that the tracklet is built based on its station with all six hits being combined. The whole loop repeats for station three and is saved.

With the tracklets organized in stations two and three, the hodoscope is used to remove tracks that are not found within the hodoscope range. This is done by masking the area of possible hits from the drift chamber onto the area of the hodoscope. The final result would be all the possible tracks in station two and three that makes sense based on the geometry of the detector. The final step would be to use AI to connect the tracklets into complete tracks.

With this procedure, the following results are obtained.

4 Results

Below are described the result obtained during this work. This software package is not finalized and the most updated version can be found on my GitHub page [20].

4.1 The Background

The events for this thesis were generated with Fun4Sim on Rivanna for single μ^-. A pool of 60,000 events was created with the vertex being throughout the entire detector. This means that the vertex could be from the target, the dump, or the gap between. The selection of events used in the background were selected at random from the event pool. We created only events for testing; Fig.12 and Fig.13 are examples of the background generation process. The first figure shows the partial tracks constructed randomly from the event pool. The depth of each partial track was also randomly selected for each track. The second figure displays the background after complete track injection with the partial tracks represented by the black horizontal lines and the injected complete track represented by the vertical red lines.

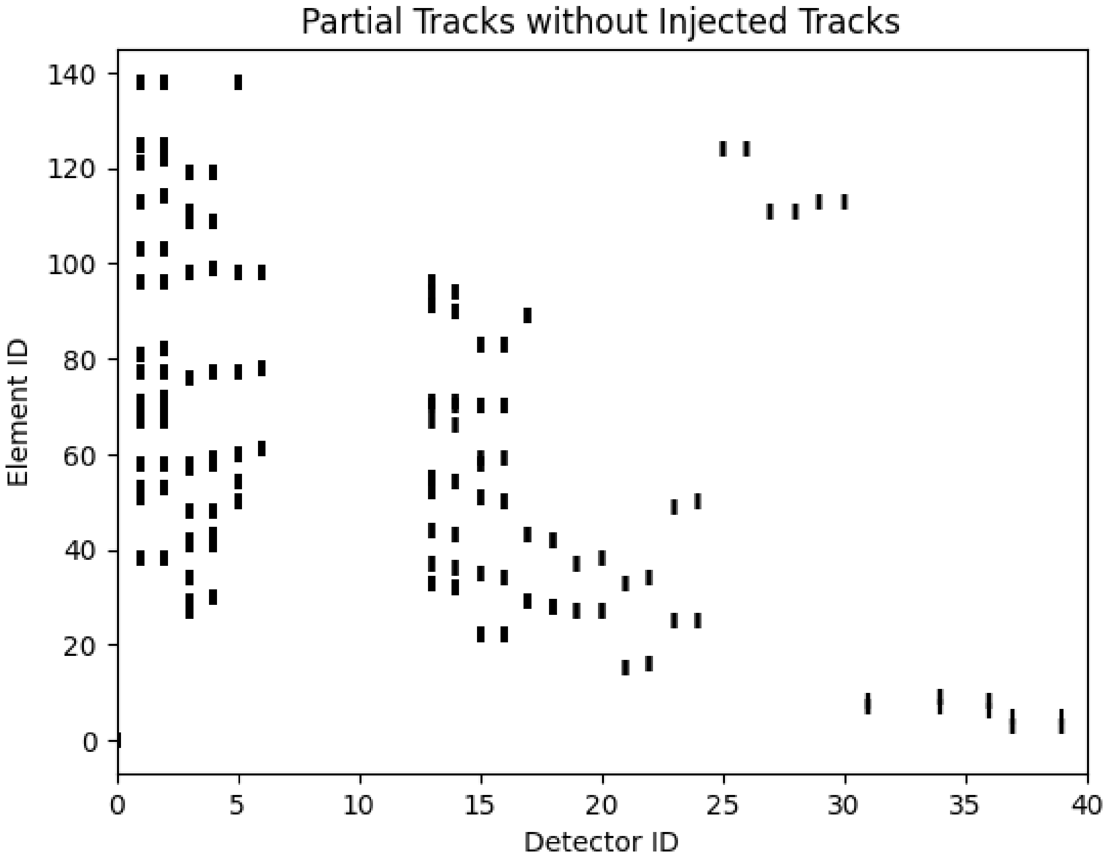


Figure 12: Displays the hits randomly taken from several complete tracks allowed to go to randomly selected depths of the detector. The hit matrix is an array of element ID's (y-axis) and detector ID's (x-axis). This makes up the background of the event as these tracks are not allowed to go through the whole station. The Drift chamber is from 1-30 detector ID.

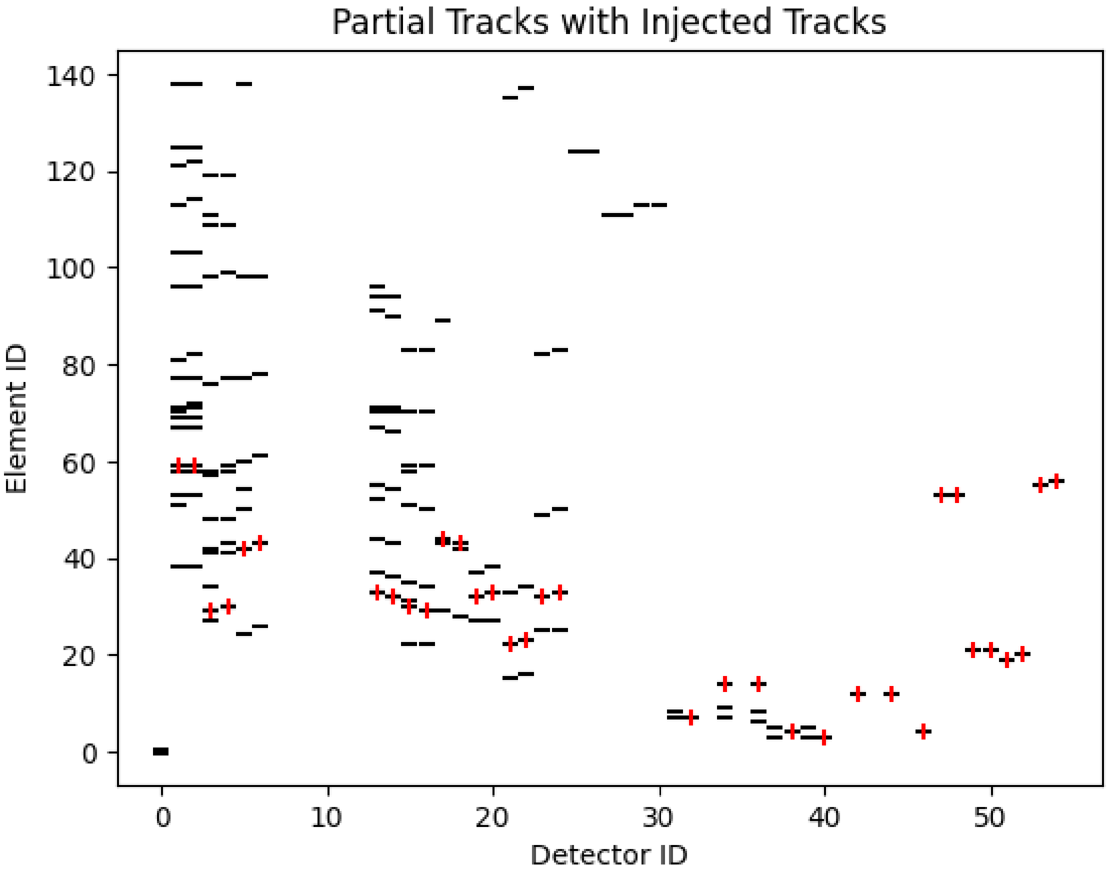


Figure 13: Displays horizontal black lines which represent the above partial tracks and random sparse hits. As well as red vertical lines representing the injected track that completely goes throughout the entire detector. The hit matrix is an array of element ID's (y-axis) and detector ID's (x-axis). The total number of hits is 147 excluding the first station has 80 hits. The injected track, hits in red, is the only complete track in the event. The Drift chamber is from 1-30 detector ID.

A zoom-in of the drift chamber can be seen in Fig.14 with the density of hits in each station based on the occupancy study done by Kenichi Nakano, for Seaquest Run six [17]. From the study, it was found that the highest occupancy was in station one and decreased for stations two and three. To simulate this a random range of hits between 15 and 35 started in station one and decreased by 50% to 25%. The hodoscope was also simulated as shown in Fig.15 where the number of background hits ranged between one and three hits.

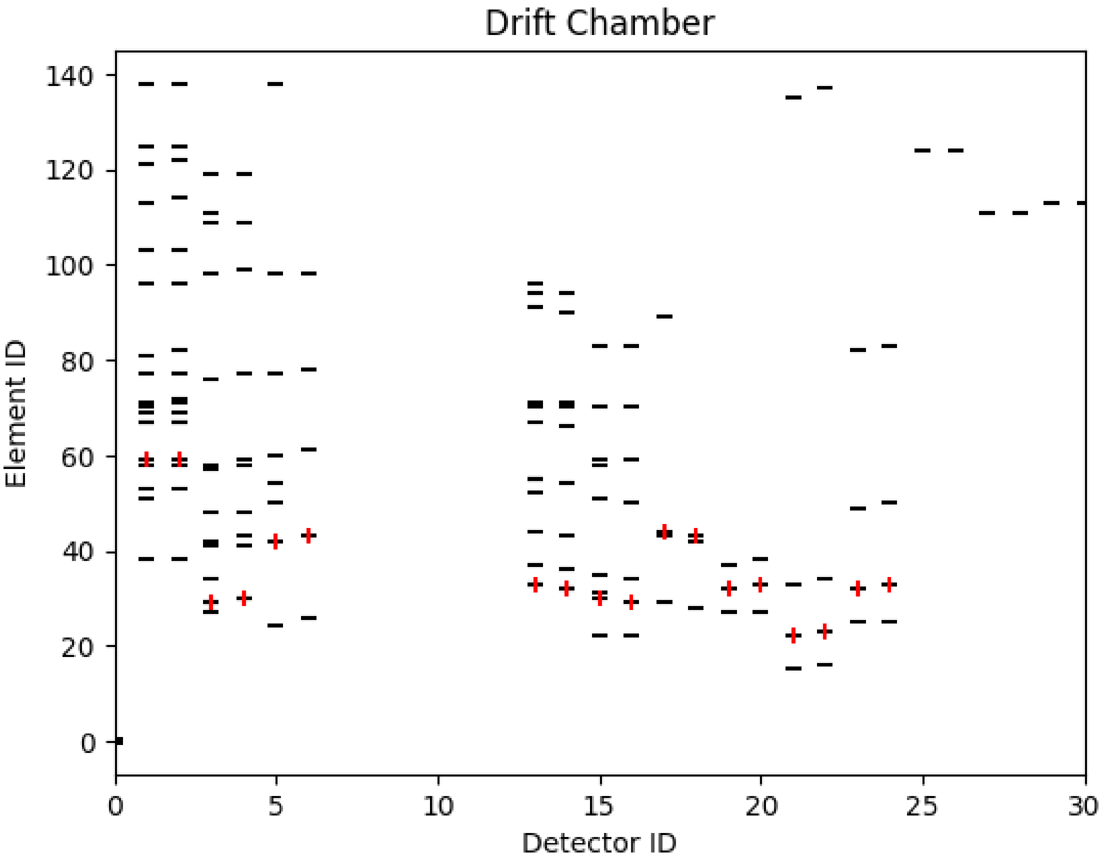


Figure 14: The drift chamber for one of the background events where the horizontal black lines represent the above partial tracks and random sparse hits represent cosmic muons. The hit matrix is an array of element ID's (y-axis) and detector ID's (x-axis). These hits make up the Background. The red vertical lines represent the injected track that completely goes throughout the entire detector, the Drift chamber is from 1-30 detector ID. This is the track we want to keep.

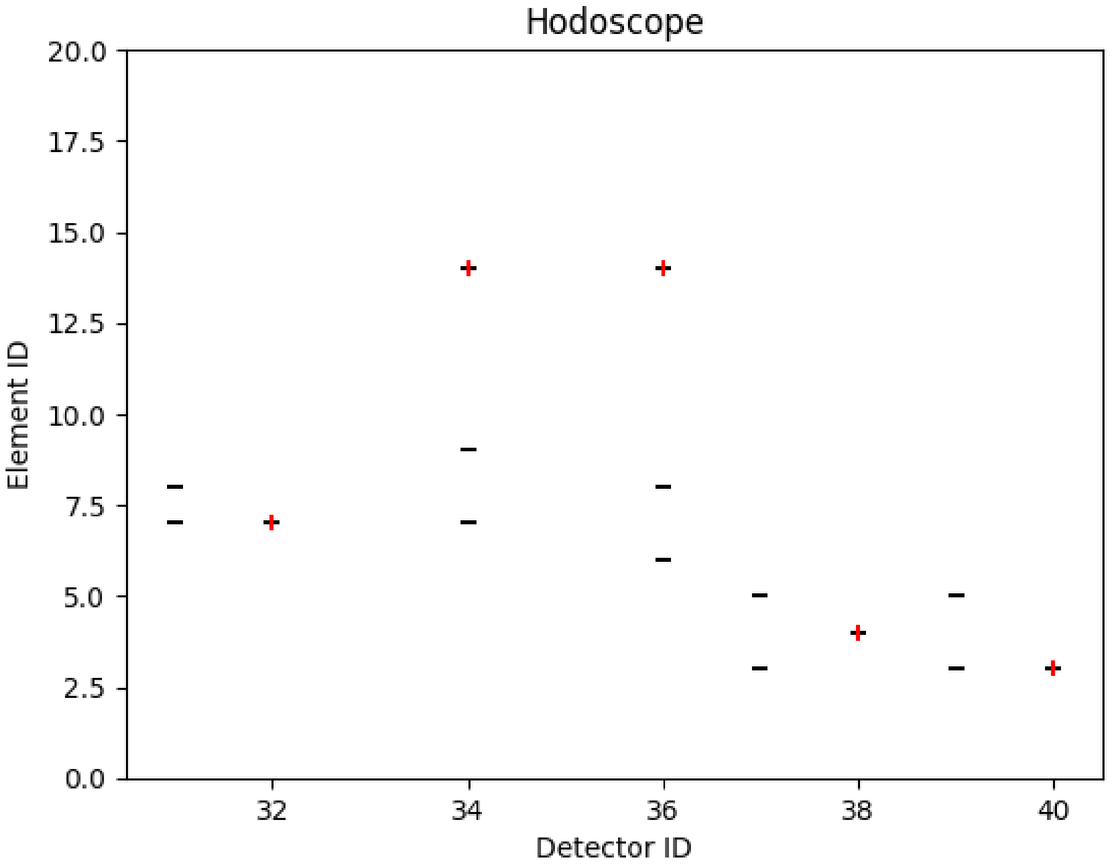


Figure 15: The hodoscope for one of the background events where the horizontal black lines represent the above partial tracks. The red vertical lines represent the injected track that completely goes throughout the entire detector. The hit matrix is an array of element ID's (y-axis) and detector ID's (x-axis).

In figures 131215, the red hits are a reference for the eye to display the actual complete track that we want the software to keep. The black lines are the background, muons not directly from the target, that can not be reconstructed. In order to get the highest merit of reconstruction we must first filter these background muons. In order to do that we utilize the geometry of the detector to create windows, projections of hits from the X plane onto the other planes.

4.1.1 Window Comparison

To confirm the window size and placement, a study was done to display the actual window being created by K-Tracker. This test was done with a dimuon track as shown in Fig. 16. Fig.17 and Fig.18 shows the window size and center overlay on top of the window produced by K-Tracker.

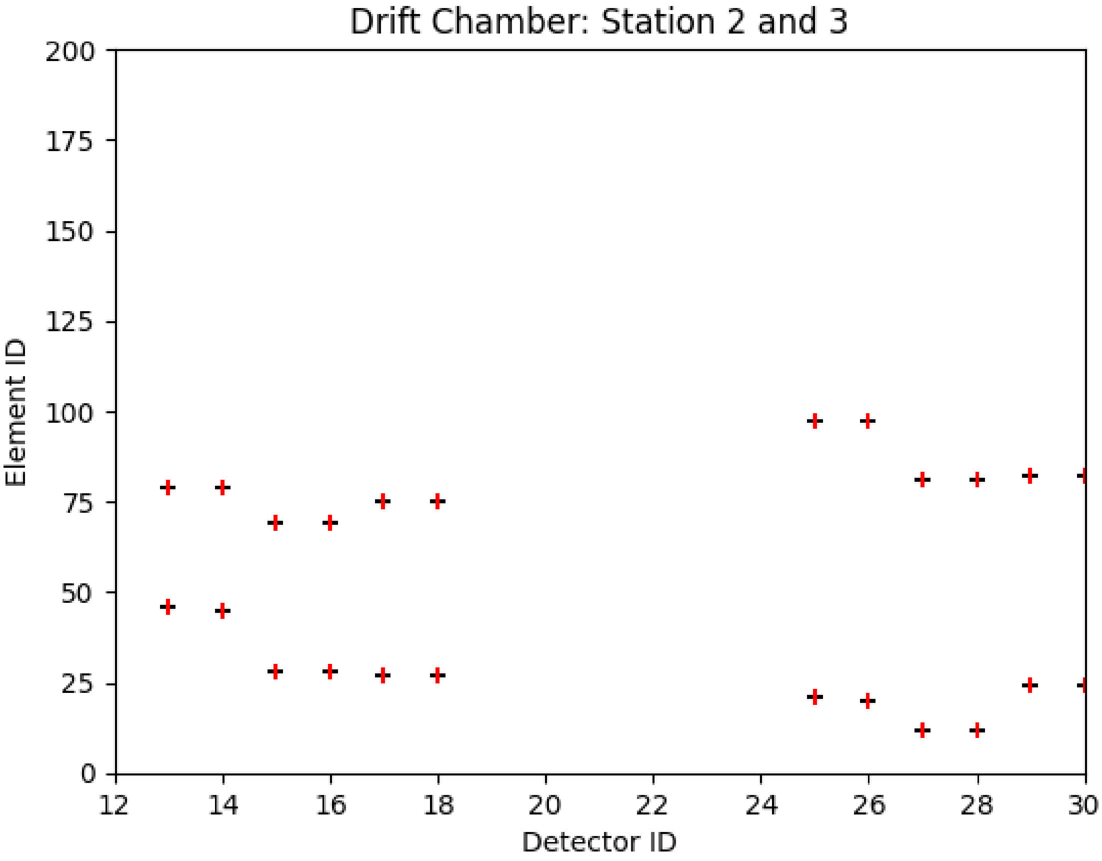


Figure 16: A dimuon track used to test the window size and position. The horizontal black lines are the background hits and the red vertical lines are the injected tracks. The hit matrix is an array of element ID's (y-axis) and detector ID's (x-axis). This event is devoid of background for window testing. This is for station 2 and station 3 of the drift chamber.

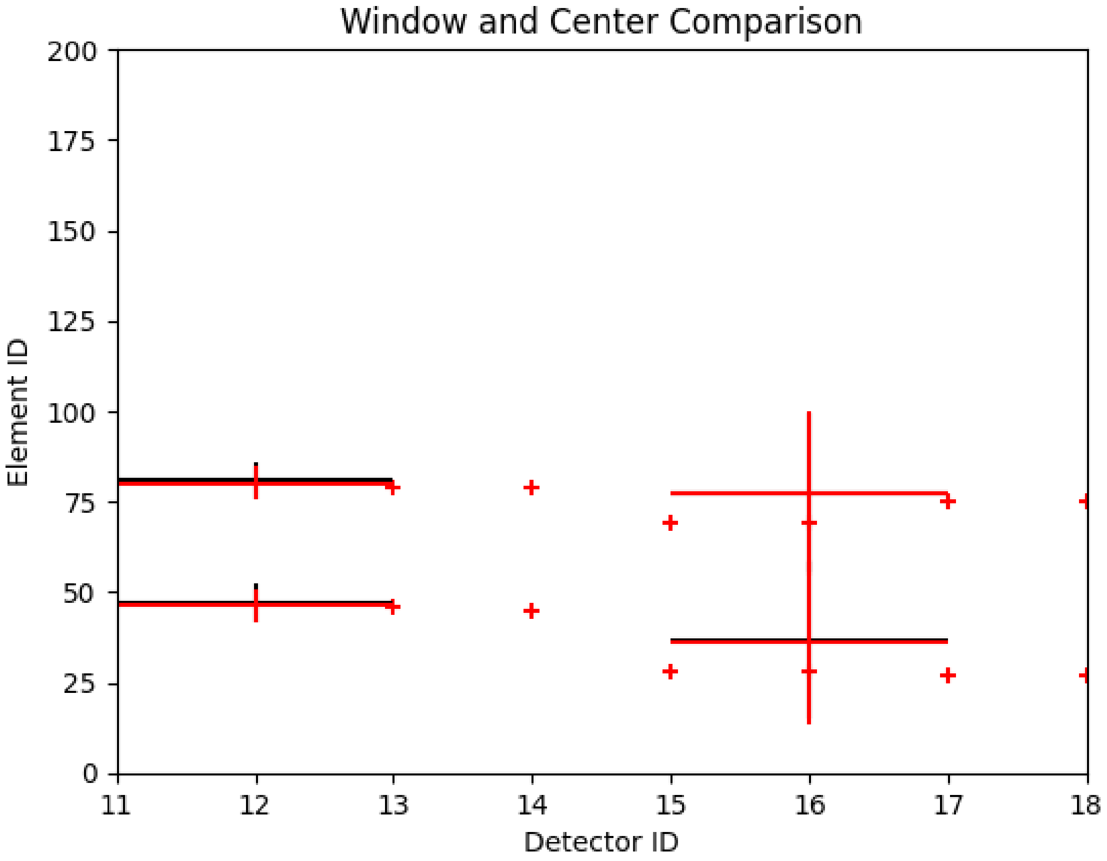


Figure 17: The U plane and V plane windows, vertical bar, and the U and V center, horizontal bar, for station 2. The red marker is what K-Tracker produced and the black marker is what the software produced. The hit matrix is an array of element ID's (y-axis) and detector ID's (x-axis).

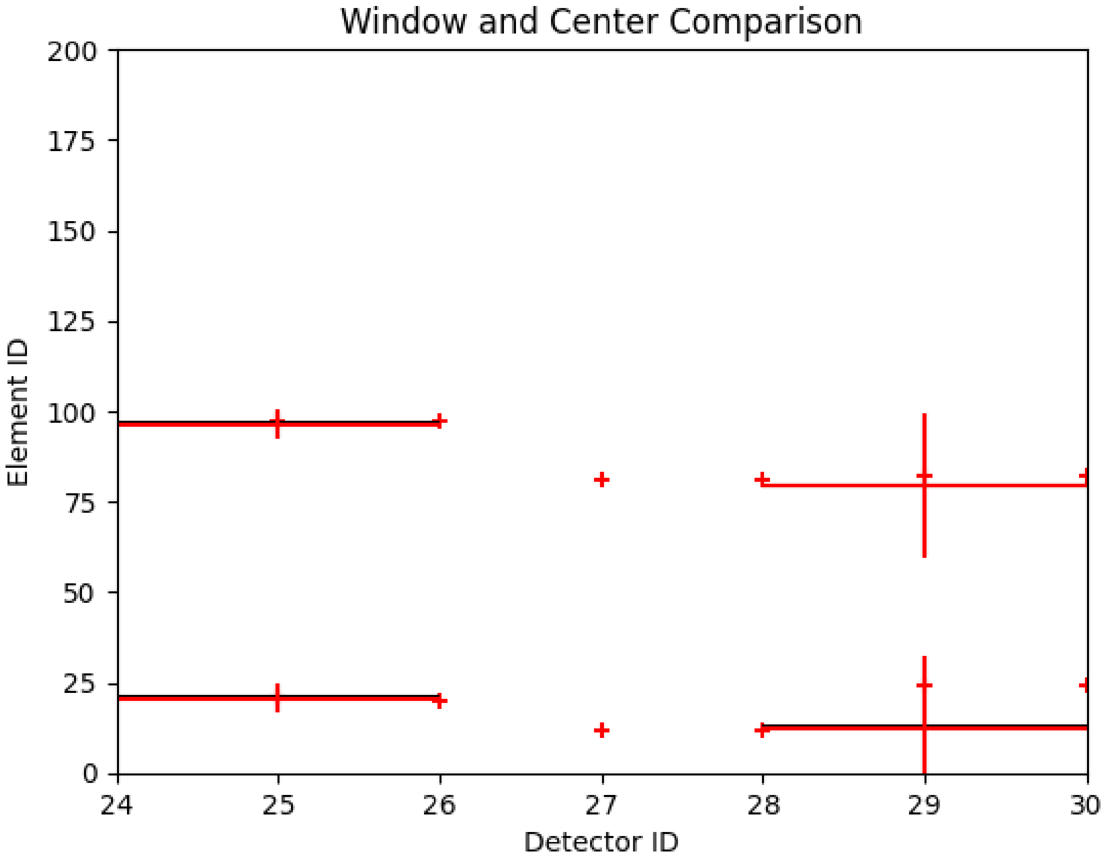


Figure 18: The U plane and V plane windows, vertical bar, and the U and V center, horizontal, bar for station 3. The red marker is what K-Tracker produced and the black marker is what the software produced. The hit matrix is an array of element ID's (y-axis) and detector ID's (x-axis).

With these windows, we were able to select and organize the tracklets as shown in Fig.19.

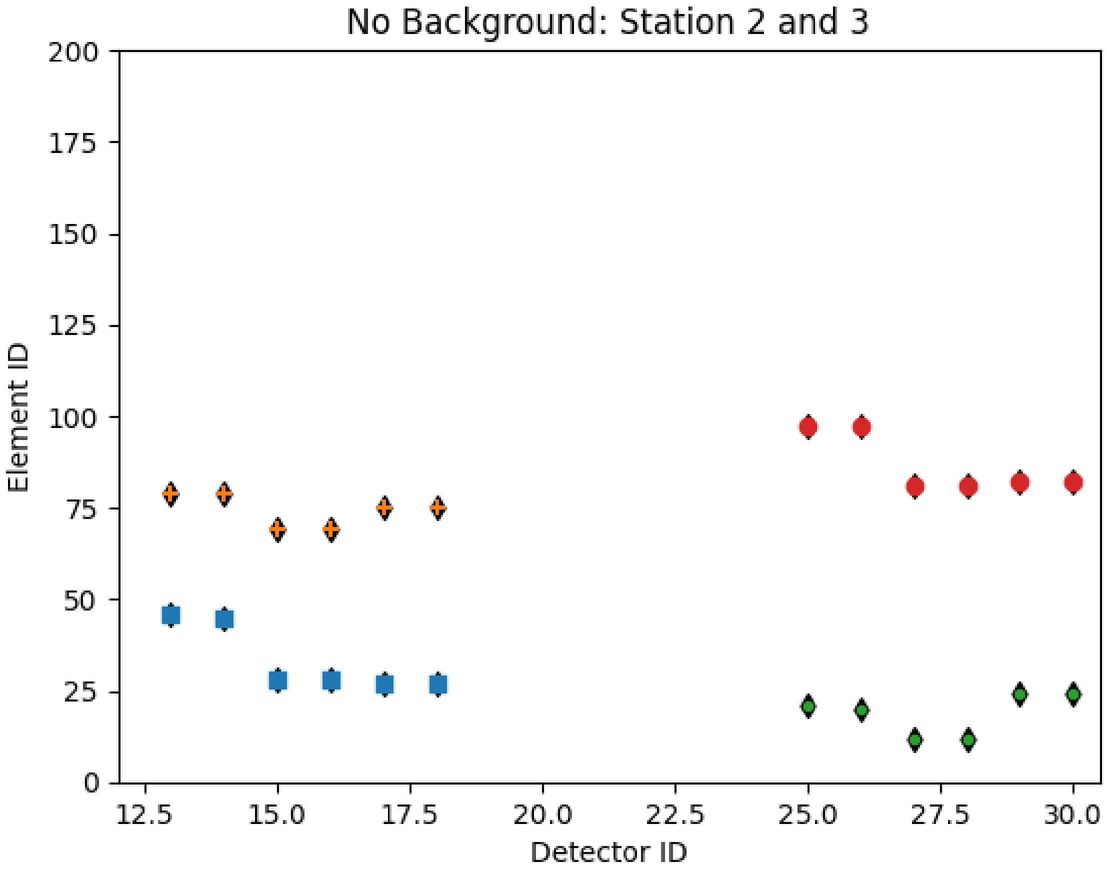


Figure 19: The tracks organized by the software based on the windows created. This test was done to make comparisons with windows created from K-Tracker. The hit matrix is an array of element ID's (y-axis) and detector ID's (x-axis).

4.2 Hit to Track Cleaning

With the Windows tested, the background was added to fully test the software. The hit pairs are selected as detailed in the methods section. Fig.20 shows stations 2 and 3 of the drift chamber after hit pair selection. As shown the hit pair selection manages to remove sparse hits such as in the upper element id region.

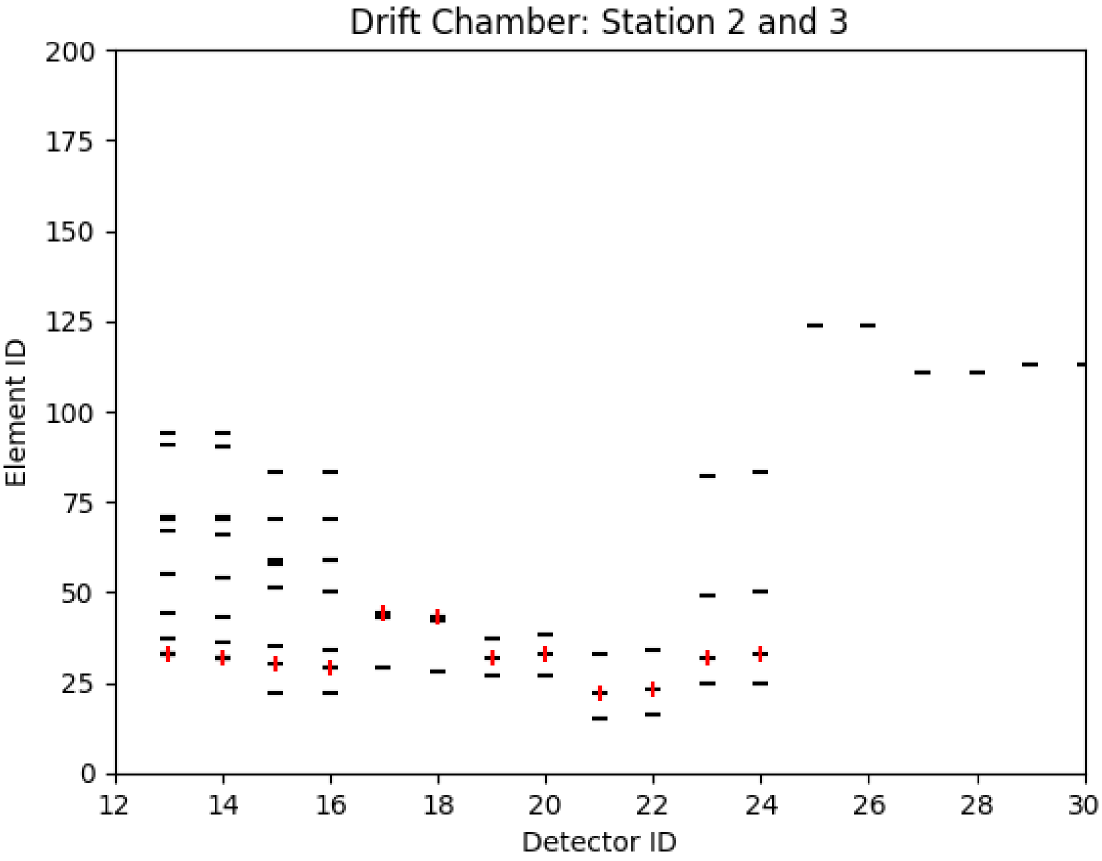


Figure 20: The hit pairs of stations 2 and 3. The horizontal black lines are the background hits and the red vertical lines are the injected tracks. The hit matrix is an array of element ID's (y-axis) and detector ID's (x-axis).

With the hit pairs selected the tracks are organized within their stations as shown in Fig.21 and Fig.22. The track combination is shown with different marker styles and colors with the actual injected track represented as a black vertical line. The total amount of hits between the two stations shown is 58 with 45 different tracklet combinations.

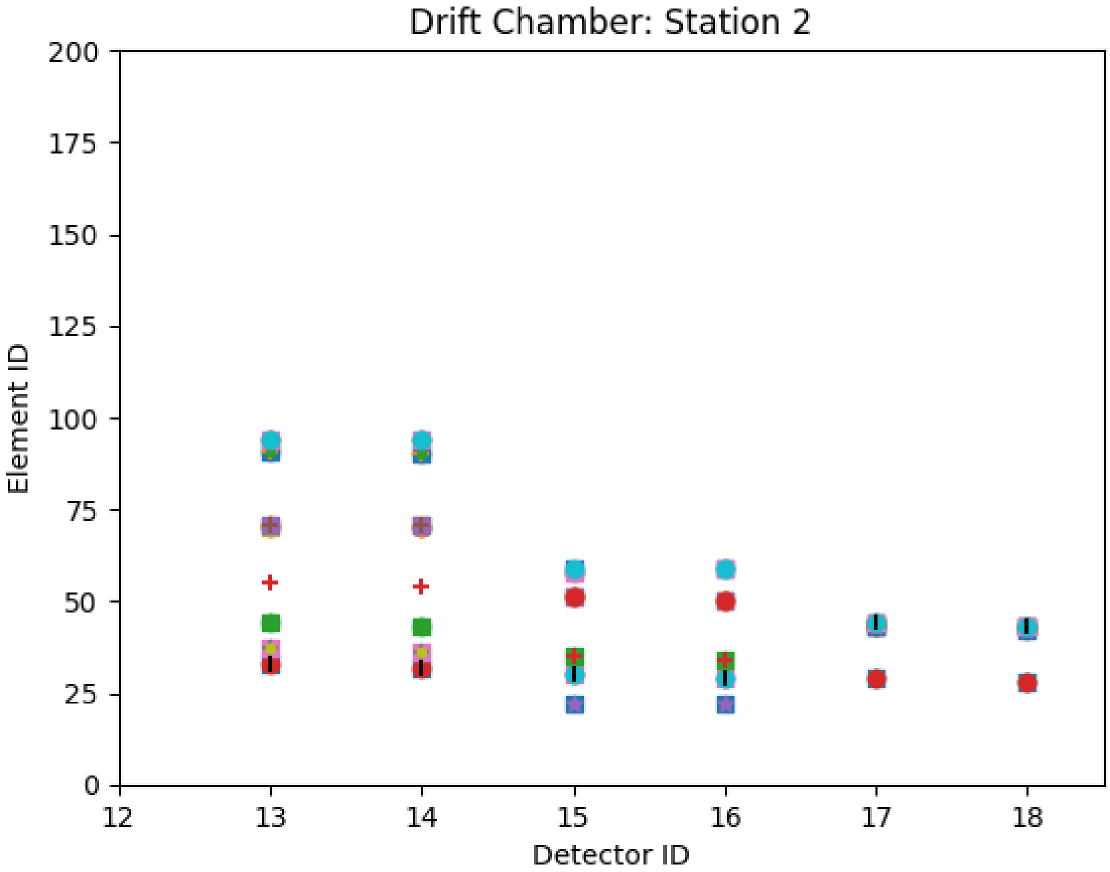


Figure 21: Station 2 track combinations. The hit matrix is an array of element ID's (y-axis) and detector ID's (x-axis). Each color and marker style represents a different tracklet. Here there are 40 tracklets. The black diamond is a reference to know where the actual track is.

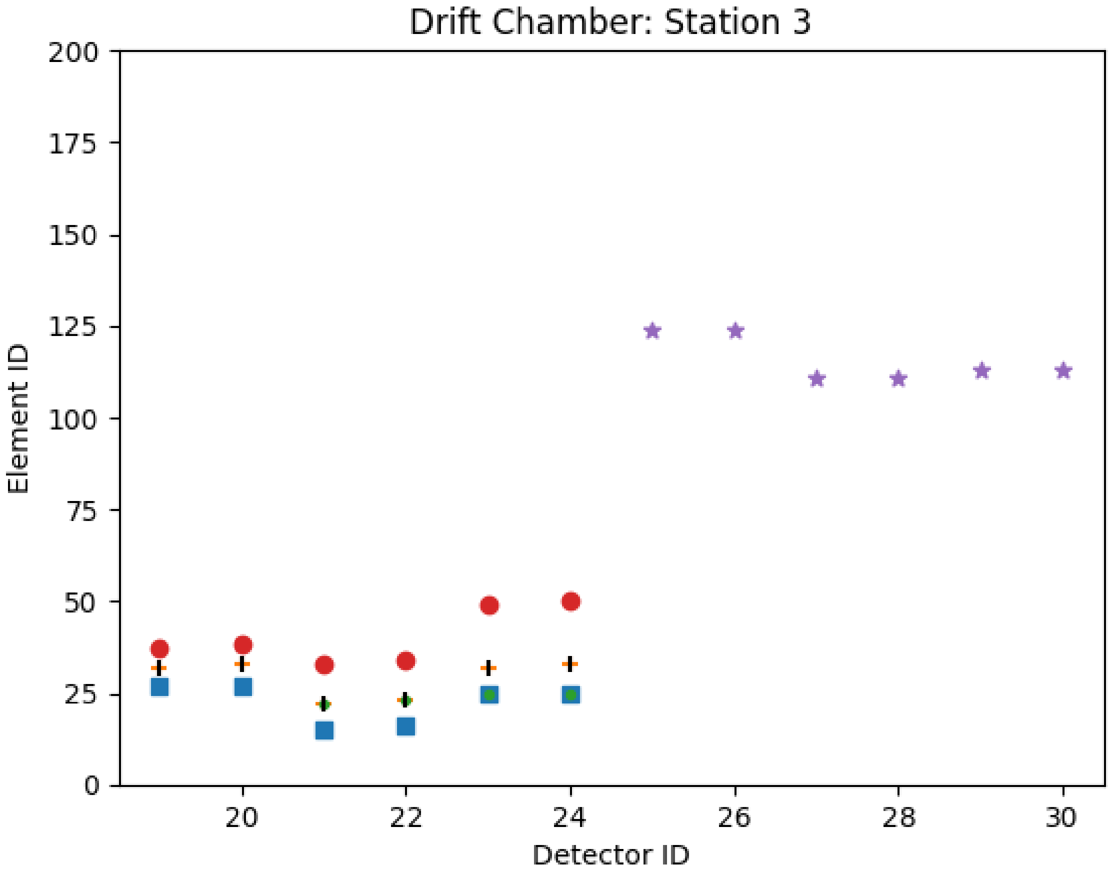


Figure 22: Station 3 track combinations. The hit matrix is an array of element ID's (y-axis) and detector ID's (x-axis). Each color and marker style represents a different tracklet. Here there are five tracklets. The black diamond is a reference to know where the actual track is.

The hodoscope hits are then matched to the drift chamber hits as described in the methods section based on the overlap of the boxes in Fig.23. This box is created by using the hit positions from the drift chamber and the area of the hodoscope. Where two overlap is where the hits are allowed to be from the drift chamber. With the hits matched successfully with the hodoscope we finish with the selected tracks in Fig.24. These hits are now able to continue to the fitting section of the software. This filters out more background hits leaving us with the final 36 hits with 6 tracklet combinations.

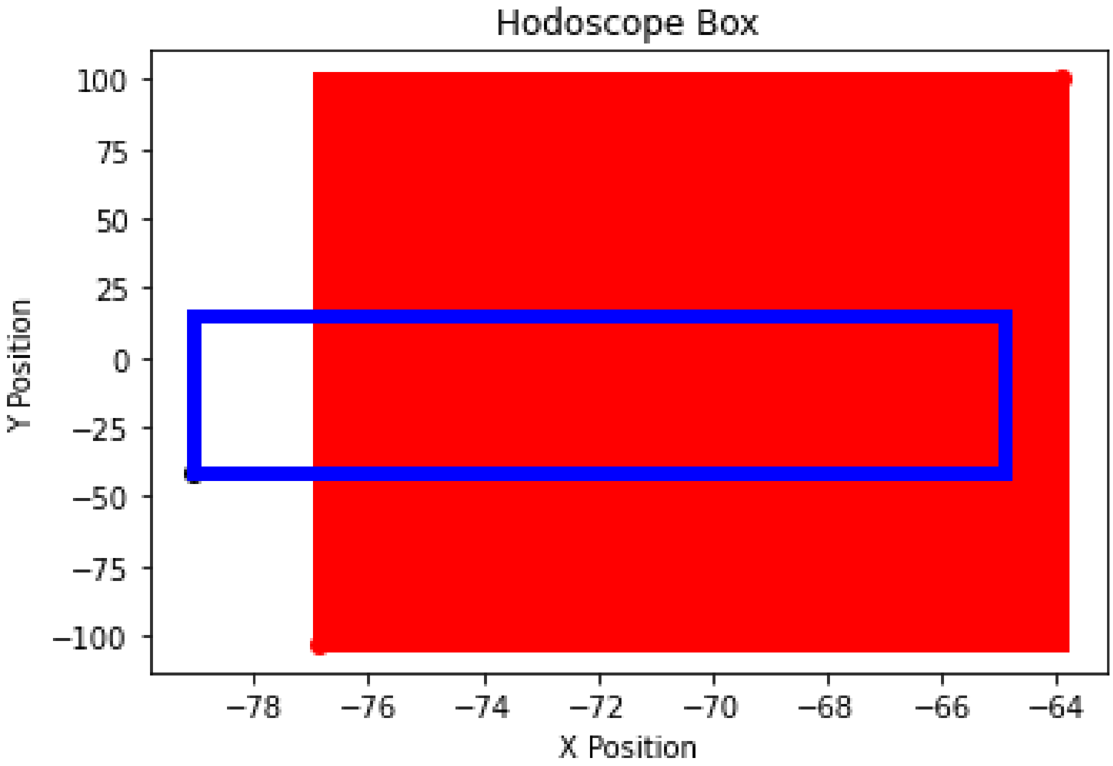


Figure 23: The areas of the drift chamber, blue box, and the solid red box, the hodoscope. Where they overlap is where the hits are allowed. The axes are X and Y positions.

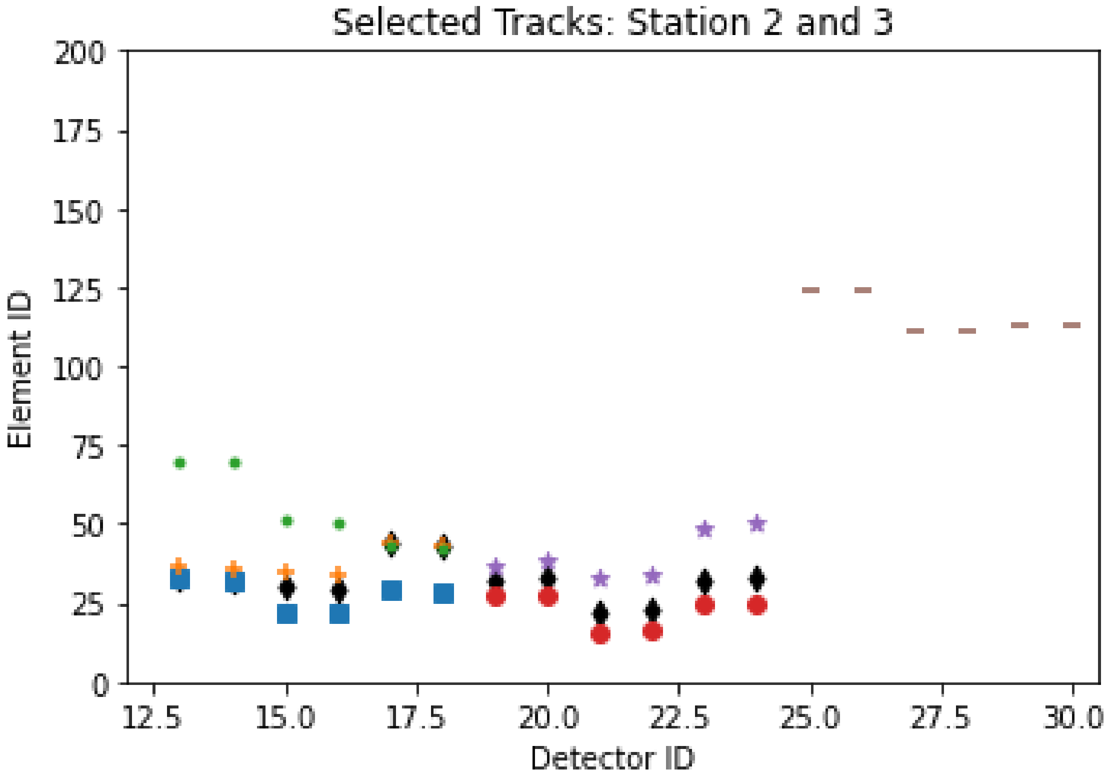


Figure 24: A figure of the Selected Tracks after the hodoscope masking. The hit matrix is an array of element ID's (y-axis) and detector ID's (x-axis). Here there are six, three in both stations. The black diamond is a reference to know where the actual track is.

4.3 Preliminary Result: The Training

To correctly determine which tracks from station 3 correctly match up with station 2, a fitting procedure was utilized to obtain the slopes and intercepts. To do this we used an AI fit to the tracks sampled from a pool of simulated events. The fitting procedure is done with TensorFlow and Keras [**Tensorflow**][11]. The model was trained with 10,000 events, with 30% of those events being used to test the two models, one for slope and one for intercept. The optimizer is Keras Adam and the loss/metric function is Mean Square Error[11]. Each hidden layer used Rectified Linear Unit, relu, activation[11]. The relu activation function is described as $f(x) = max(0, x)$ which only allows positive numbers. This function defines the output of the node given an input.

The loss of the intercept model was: 13.8 and the loss of the slope model was: .1. The architecture is shown below in Fig.25 which displays the model for the intercepts and Fig.26 which displays the model for the slopes. The boxes represent layers and the hidden layers are dense with several nodes.

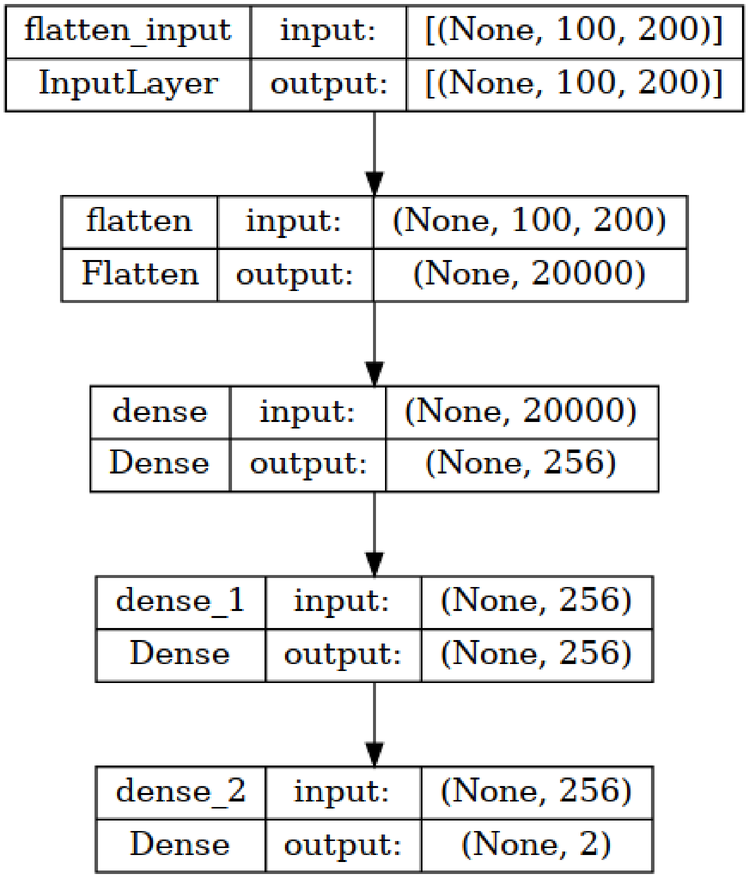


Figure 25: The architecture for the model that predicts the intercepts for the tracklets. The input is a hit matrix of zeros and one's size is 100 by 200. This input is then flattened. The output is a 2D array of X intercepts and Y intercepts. There are two hidden layers with activation relu.

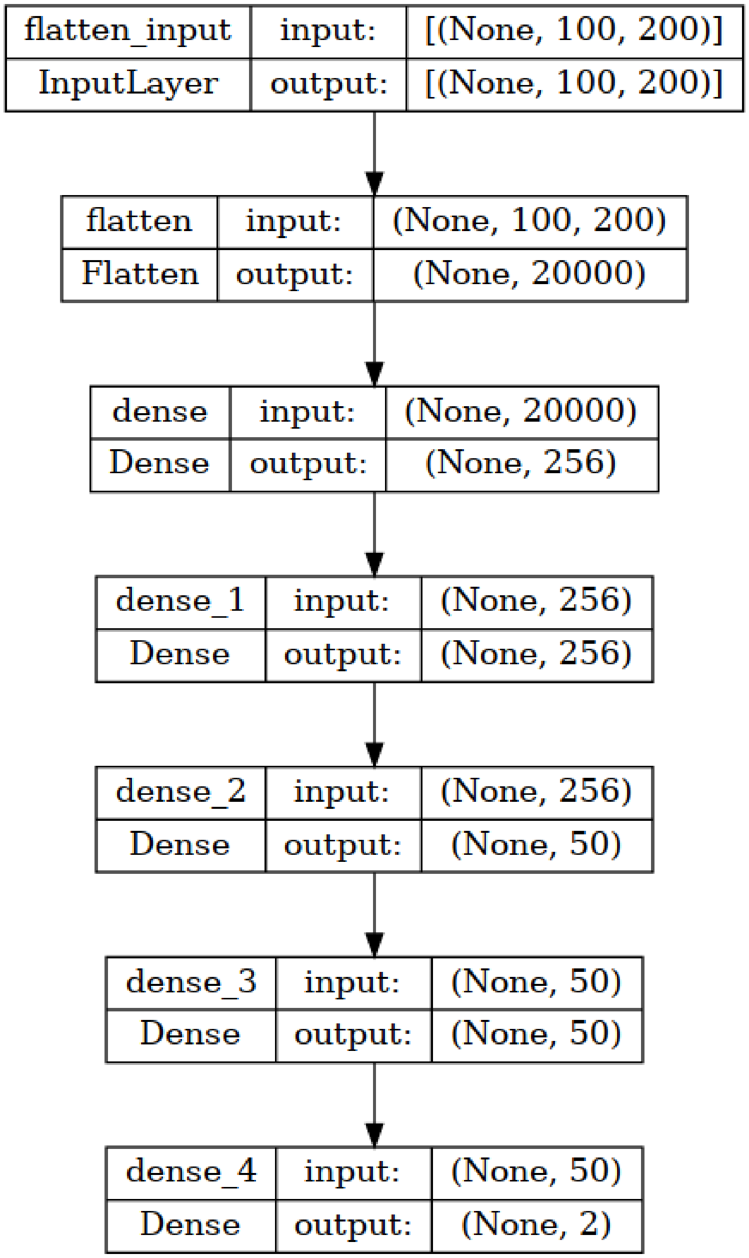

flatten_input | input: | [(None, 100, 200)]
InputLayer | output: | [(None, 100, 200)]
flatten | input: | (None, 100, 200)
Flatten | output: | (None, 20000)
dense | input: | (None, 20000)
Dense | output: | (None, 256)
dense_1 | input: | (None, 256)
Dense | output: | (None, 256)
dense_2 | input: | (None, 256)
Dense | output: | (None, 50)
dense_3 | input: | (None, 50)
Dense | output: | (None, 50)
dense_4 | input: | (None, 50)
Dense | output: | (None, 2)

Figure 26: The architecture for the model that predicts the slopes for the tracklets. The input is a hit matrix of zeros and one's size is 100 by 200. This input is then flattened. The output is a 2D array of X slopes and Y slopes. There are four hidden layers with activation relu.

We used a train and test split to do validation while training the model. Model loss is done on the training data set while validation loss is done on the testing data set. Fig.27 and Fig.28 display the model loss compared to the validation loss of the intercept model and the slope model. It is shown that both models are overfitting, however, no gain or loss is obtained from this overfit as the model stabilizes.

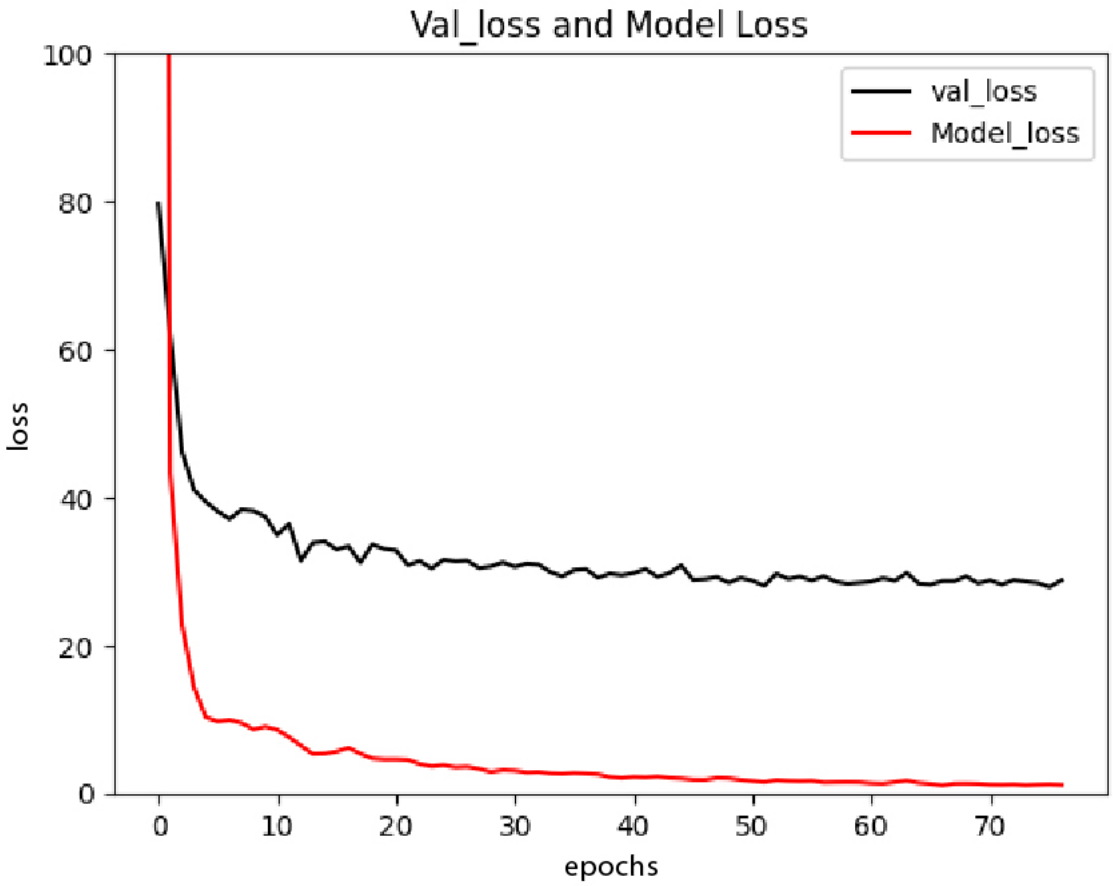


Figure 27: The loss of the model compared with the validation loss. This model was trained on the x and y intercepts. The x-axis is the number of epochs and the y-axis is the loss.

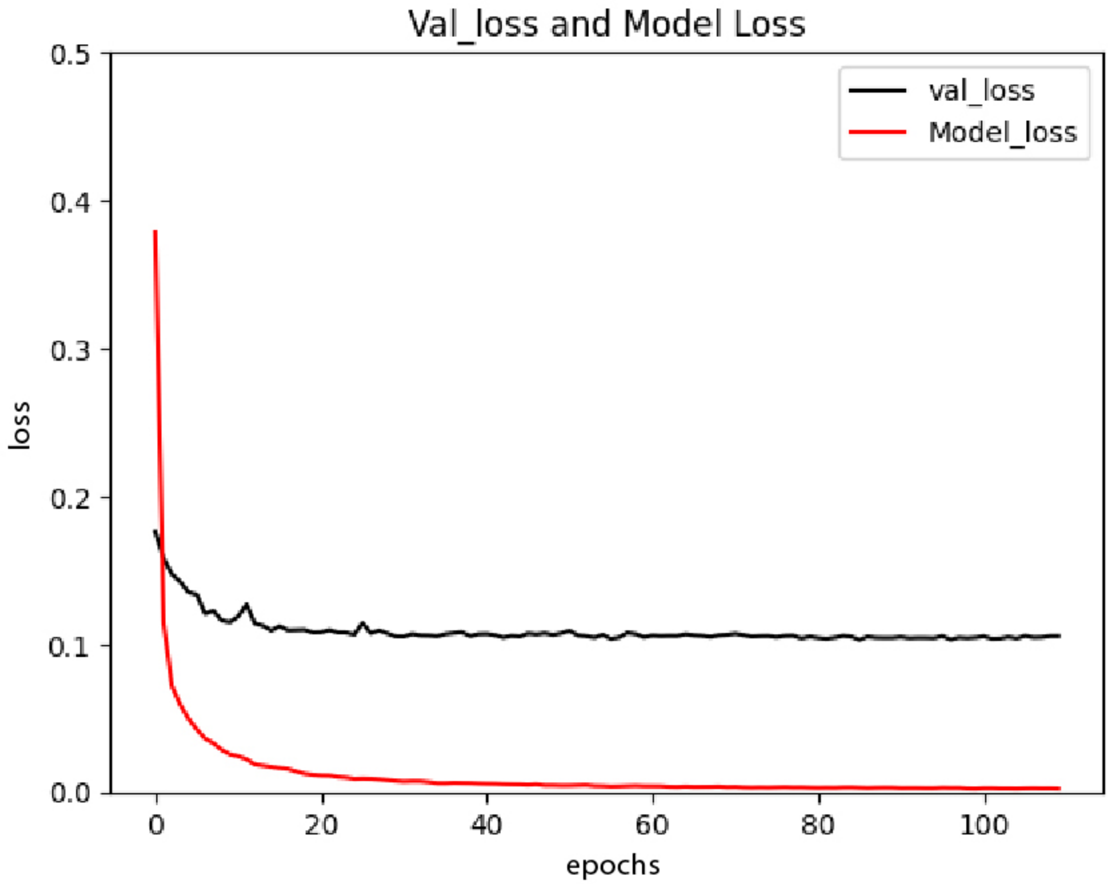


Figure 28: The loss of the model compared with the validation loss. This model was trained on the x and y slopes. The x-axis is the number of epochs and the y-axis is the loss.

With the model trained Fig.29 displays the approximate error between the model values and the truth values of the slope and intercepts. As can be seen, the standard deviation does not go beyond 1.

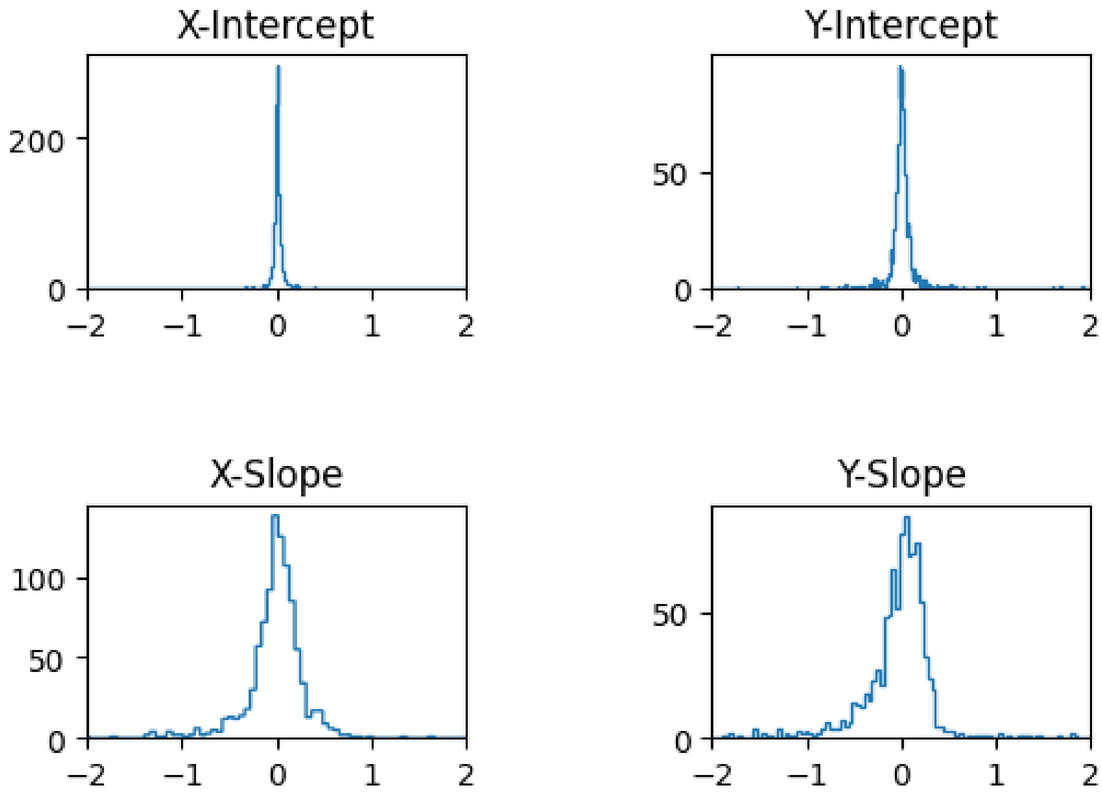


Figure 29: This figure shows the 1D histograms of approximate error of the X-Intercept, Y-Intercept, X-Slope, and Y-Slope. This is done by dividing the difference between the prediction and truth with the truth values.

5 Discussion of Results and Conclusion

When dealing with reconstruction and machine learning the question of accuracy and precision arises. Compared with K-Tracker, the hit selection at the station level matches the window selection. However, as the results show the accuracy of removing background tracks using the software after the hodoscope masking is 50%. We believe that the accuracy will improve to 80% after the AI fitting is finalized and the hodoscope masking is better tuned. We also obtained a precision of 81% background removal. The accuracy and precision were obtained by using the following standard methods. True positives; cases in which it is a background hit and is correctly removed. True negatives are cases in which it is a hit and is kept. False positives are cases in which it is a hit and removed. Lastly, false negatives, are cases in which it is a background hit and is kept. To get an accurate reconstruction, four out of the six hits must be determined within the wire cluster. We believe that the majority of the hits saved in the selected tracks are within that error except for the track in the positive section of station 3. We believe that tracks like this will be removed when combining stations 2 and 3. More tuning of the hodoscope masking could allow for a more accurate track selection as true hits were removed after hodoscope masking.

As the preliminary results show the AI fitting is starting and is looking promising with the widths of the approximate errors not being too large. The overfitting of the model could be prevented by utilizing a different activation function or tuning the hyperparameters of the model. It will be the goal preceding the launch of SpinQuest in September 2023 to tune the accuracy of the track selection to the highest merit possible and to incorporate displays that utilize the GPU such as dearpygui, a Python library that allows for the creation of interfaces. With dearpygui a simple interface can be built to display some of the useful plots shown in this thesis. The estimated workload of this integration is not high and the main concern is tuning the software. As shown it can be useful to visualize events from the collision at a station level. Given repeatable results we will be able to detect anomalies when they arise in the detector based on the accuracy of the reconstruction. We have shown that the tracks can be organized in a way that is comparable to how K-Tracker selects hits with accuracy and precision that is suitable for reconstruction when finalized. With the tracklets selected in stations 2 and 3, the reconstruction can be done by using the slopes and intercepts from the AI fit to connect back to station 1. The vertex can then be localized and with the aid of the K-Mag, the four-momenta can be determined finishing the reconstruction. With the reconstruction, the left-right asymmetry can be calculated and displayed to be used in online monitoring.

We have shown a background generator that creates realistic spills based on the occupancy of SeaQuest. We were able to show that K-Tracker can be rewritten with modern computing tools to show visually what is happening at each step of the reconstruction. This ability to display step-by-step speeds up quality checks and allows monitoring in real time. We were able to obtain a precision of 81% and an accuracy of 50% background removal. We showed a recall of 53% of the complete track. Preliminary we were able to show that AI can be paired with accelerated K-Tracker Track Selection to optimize for fast information integration and anomaly detection. This will provide a system that allows for asymmetry detection solving one of the challenges of the experiment.

6 References

[1] M. Anselmino. "The transverse structure of protons and neutrons: TMDs". In: *Scholarpedia* 6.2 (2011). revision #91873, p. 10209. DOI: 10.4249/scholarpedia.10209.

[2] *et al.* C. Alexandrou. "The nucleon spin and momentum decomposition using lattice QCD simulations". In: *arXiv:1706.02973v3 [hep-lat] 18 Nov 2017* (2017).

[3] SpinQuest Collaboration. *E1039-Collaboration*. URL: https://github.com/E1039-Collaboration. (accessed: 07.02.2023).

[4] SpinQuest Collaboration. *SeaQuest with a Transversely Polarized Target (E1039)*. URL: https://twist.phys.virginia.edu/work/E1039proposal_final.pdf.

[5] The E155 Collaboration. "Precision Measurement of the Proton and Deuteron Spin Structure Functions g2 and Asymmetries A2". In: *arXiv:hep-ex/0204028v1* (2002).

[6] The European Muon Collaboration. "AN INVESTIGATION OF THE SPIN STRUCTURE OF THE PROTON IN DEEP INELASTIC SCATTERING OF POLARISED MUONS ON POLARISED PROTONS". In: *Nuclear Physics B328* (1989).

[7] Arthur Conover. "Kinematic dependence of the dilution factor in the SpinQuest experiment E1039 at Fermilab". In: *Master Thesis: University of Virginia* (2020).

[8] *et al.* E. Aschenauer. "The RHIC Spin Program: Achievements and Future Opportunities". In: *arXiv:1304.0079 [nucl-ex]* (2012).

[9] J. C. Collins *et al.* "Sivers effect in Drell Yan at BNL RHIC". In: *arXiv:hep-ph/0511272v2* (2006).

[10] David Griffiths. *Introduction to Elementary Particles*. Wiley-VCH, 2008.

[11] Keras information and API.

[12] R.L. Jaffe and Aneesh Manohar. "The g1 Problem: Deep Inelastic Electron Scattering And The Spin Of The Proton". In: *Nuclear Physics B Volume 337, Issue 3, 25 June 1990, Pages 509-546* (1990).

[13] Xiangdon Ji. "Gauge-Invariant Decomposition of Nucleon Spin". In: *Phys. Rev. Lett. 78, 610 – Published 27 January 1997* (1997).

[14] *et al.* Marcin Bury. "Extraction of the Sivers function from SIDIS, Drell-Yan, and W±/Z boson production data with TMD evolution". In: *arXiv:2103.03270* (2021).

[15] B.R. Martin and G. Shaw. *Particle Physics*. Wiley, 2008.

[16] Kei Nagai. "Recent Measurement of Flavor Asymmetry of Antiquarks in the Proton by Drell–Yan Experiment SeaQuest at Fermilab". In: *Doctoral Thesis for Department of Physics, Tokyo Institute of Technology* (2017).

[17] Kenichi Nakano. *Plane Occupancy in Run 6*. URL: https://seaquest-docdb.fnal.gov/cgi-bin/sso/RetrieveFile?docid=9631&filename=slide.pdf&version=3. (accessed: 07.16.2023).

[18] Michael E. Peskin and Daniel V. Schroeder. *Quantum Field Theory*. Perseus Books, 1995.

[19] Michael Riordan. "The Discovery of Quarks*". In: *Science* (1992).

[20] Jordan Roberts. *SoftwarePackage*. URL: https://github.com/JayDanielsss/Jtracker. (accessed: 07.02.2023).

[21] *et al.* S.Arnold. "Sivers effect at Hermes, Compass and Clas12". In: *arXiv:0805.2137* (2008).

[22] Dennis Sivers. "Single-spin production asymmetries from the hard scattering of pointlike constituents". In: *Phys. Rev. D* 41 (1 Jan. 1990), pp. 83–90. DOI: 10.1103/PhysRevD.41.83. URL: https://link.aps.org/doi/10.1103/PhysRevD.41.83.

[23] Mark Thomson. *Moden Particle Physics*. Cambridge University Press, 2013.

[24] Xiaoyu Wang and Zhun Lu. "Sivers asymmetry in the pion induced Drell-Yan process at COMPASS within transverse momentum dependent factorization". In: *Phys. Rev. D 97, 054005* (2018).

[25] Xiaoyu Wang1 and Zhun Lu. "Sivers Asymmetry in the pion induced Drell-Yan process at COMPASS within TMD factorization". In: *Phys. Rev., D97(5):054005* (2018).

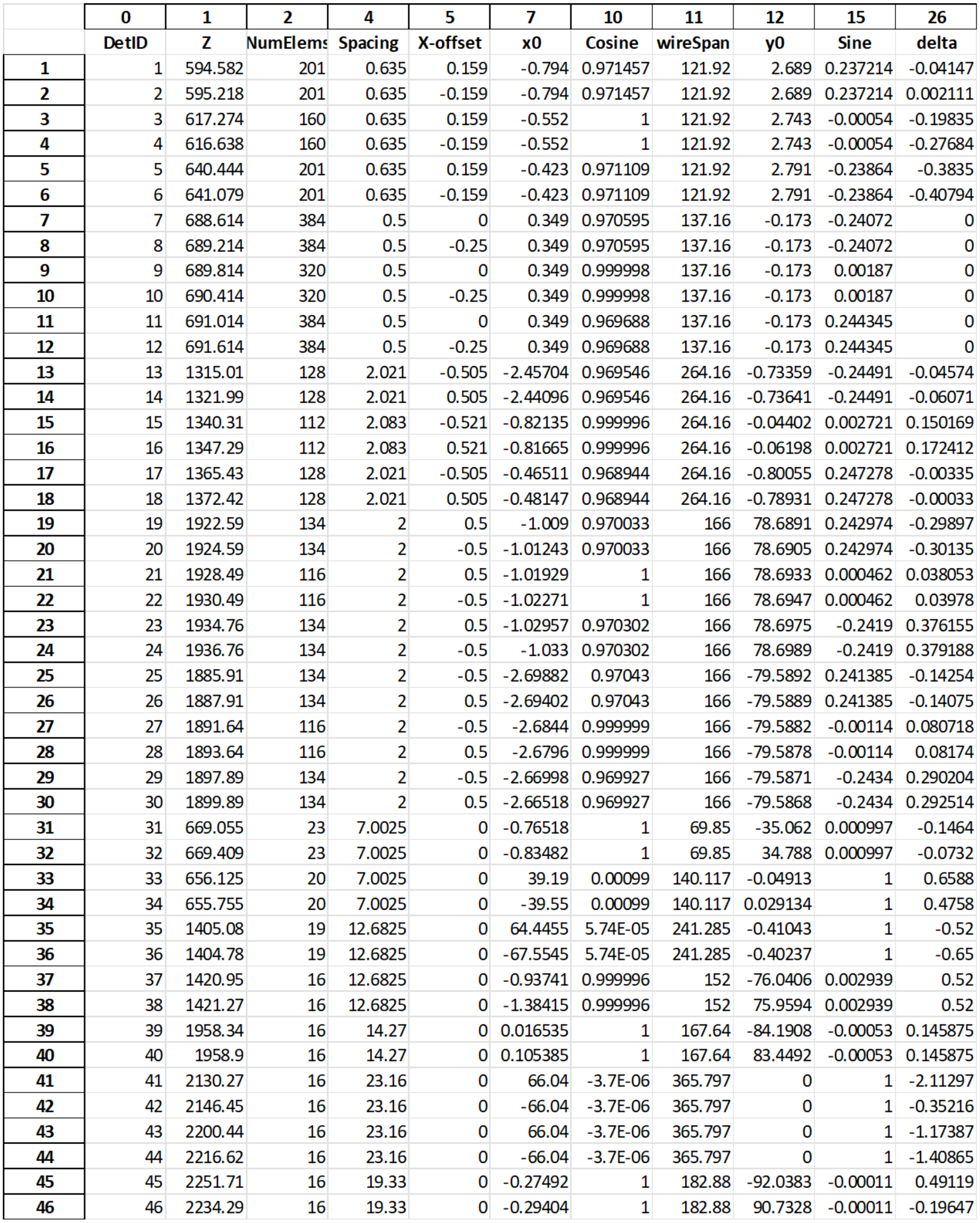
[26] Wikipedia contributors. *Deep inelastic scattering — Wikipedia, The Free Encyclopedia*. [Online; accessed 30-July-2023]. 2023. URL: https://en.wikipedia.org/w/index.php?title=Deep_inelastic_scattering&oldid=1167416734.

[27] Wikipedia contributors. *Standard Model — Wikipedia, The Free Encyclopedia*. [Online; accessed 30-July-2023]. 2023. URL: https://en.wikipedia.org/w/index.php?title=Standard_Model&oldid=1166898021.

[28] Jaime Wisniak. "William Prout". In: *Educación Química* 26.2 (2015), pp. 162–173.

7 Appendix

This section is for diagrams and tables that are useful for the software but not needed to understand the results. Below is Fig.30 which displays the geometry used in the software. This geometry was scraped from a project created by Eric Fuchey. DetID is the detector ID which is connected to the plane, V,X,U of the chambers. The Z position of the detector component is listed as Z. The number of elements relates to the number of individual detectors in one detector plane, such as the wires in the drift chamber. The spacing is space between these detectors. The X-offset is the offset of the prime planes. X0 and y0 are the detector x and y intercepts. Cosine and Sine are of the U wire angle, hence for the V plane the angle is flipped. The wireSpan is the length of the detector. Finally, the delta term is used to tune the U radius to account for every complete track.



	0	1	2	4	5	7	10	11	12	15	26
	DetID	Z	NumElems	Spacing	X-offset	x0	Cosine	wireSpan	y0	Sine	delta
1	1	594.582	201	0.635	0.159	-0.794	0.971457	121.92	2.689	0.237214	-0.04147
2	2	595.218	201	0.635	-0.159	-0.794	0.971457	121.92	2.689	0.237214	0.002111
3	3	617.274	160	0.635	0.159	-0.552	1	121.92	2.743	-0.00054	-0.19835
4	4	616.638	160	0.635	-0.159	-0.552	1	121.92	2.743	-0.00054	-0.27684
5	5	640.444	201	0.635	0.159	-0.423	0.971109	121.92	2.791	-0.23864	-0.3835
6	6	641.079	201	0.635	-0.159	-0.423	0.971109	121.92	2.791	-0.23864	-0.40794
7	7	688.614	384	0.5	0	0.349	0.970595	137.16	-0.173	-0.24072	0
8	8	689.214	384	0.5	-0.25	0.349	0.970595	137.16	-0.173	-0.24072	0
9	9	689.814	320	0.5	0	0.349	0.999998	137.16	-0.173	0.00187	0
10	10	690.414	320	0.5	-0.25	0.349	0.999998	137.16	-0.173	0.00187	0
11	11	691.014	384	0.5	0	0.349	0.969688	137.16	-0.173	0.244345	0
12	12	691.614	384	0.5	-0.25	0.349	0.969688	137.16	-0.173	0.244345	0
13	13	1315.01	128	2.021	-0.505	-2.45704	0.969546	264.16	-0.73359	-0.24491	-0.04574
14	14	1321.99	128	2.021	0.505	-2.44096	0.969546	264.16	-0.73641	-0.24491	-0.06071
15	15	1340.31	112	2.083	-0.521	-0.82135	0.999996	264.16	-0.04402	0.002721	0.150169
16	16	1347.29	112	2.083	0.521	-0.81665	0.999996	264.16	-0.06198	0.002721	0.172412
17	17	1365.43	128	2.021	-0.505	-0.46511	0.968944	264.16	-0.80055	0.247278	-0.00335
18	18	1372.42	128	2.021	0.505	-0.48147	0.968944	264.16	-0.78931	0.247278	-0.00033
19	19	1922.59	134	2	0.5	-1.009	0.970033	166	78.6891	0.242974	-0.29897
20	20	1924.59	134	2	-0.5	-1.01243	0.970033	166	78.6905	0.242974	-0.30135
21	21	1928.49	116	2	0.5	-1.01929	1	166	78.6933	0.000462	0.038053
22	22	1930.49	116	2	-0.5	-1.02271	1	166	78.6947	0.000462	0.03978
23	23	1934.76	134	2	0.5	-1.02957	0.970302	166	78.6975	-0.2419	0.376155
24	24	1936.76	134	2	-0.5	-1.033	0.970302	166	78.6989	-0.2419	0.379188
25	25	1885.91	134	2	-0.5	-2.69882	0.97043	166	-79.5892	0.241385	-0.14254
26	26	1887.91	134	2	0.5	-2.69402	0.97043	166	-79.5889	0.241385	-0.14075
27	27	1891.64	116	2	-0.5	-2.6844	0.999999	166	-79.5882	-0.00114	0.080718
28	28	1893.64	116	2	0.5	-2.6796	0.999999	166	-79.5878	-0.00114	0.08174
29	29	1897.89	134	2	-0.5	-2.66998	0.969927	166	-79.5871	-0.2434	0.290204
30	30	1899.89	134	2	0.5	-2.66518	0.969927	166	-79.5868	-0.2434	0.292514
31	31	669.055	23	7.0025	0	-0.76518	1	69.85	-35.062	0.000997	-0.1464
32	32	669.409	23	7.0025	0	-0.83482	1	69.85	34.788	0.000997	-0.0732
33	33	656.125	20	7.0025	0	39.19	0.00099	140.117	-0.04913	1	0.6588
34	34	655.755	20	7.0025	0	-39.55	0.00099	140.117	0.029134	1	0.4758
35	35	1405.08	19	12.6825	0	64.4455	5.74E-05	241.285	-0.41043	1	-0.52
36	36	1404.78	19	12.6825	0	-67.5545	5.74E-05	241.285	-0.40237	1	-0.65
37	37	1420.95	16	12.6825	0	-0.93741	0.999996	152	-76.0406	0.002939	0.52
38	38	1421.27	16	12.6825	0	-1.38415	0.999996	152	75.9594	0.002939	0.52
39	39	1958.34	16	14.27	0	0.016535	1	167.64	-84.1908	-0.00053	0.145875
40	40	1958.9	16	14.27	0	0.105385	1	167.64	83.4492	-0.00053	0.145875
41	41	2130.27	16	23.16	0	66.04	-3.7E-06	365.797	0	1	-2.11297
42	42	2146.45	16	23.16	0	-66.04	-3.7E-06	365.797	0	1	-0.35216
43	43	2200.44	16	23.16	0	66.04	-3.7E-06	365.797	0	1	-1.17387
44	44	2216.62	16	23.16	0	-66.04	-3.7E-06	365.797	0	1	-1.40865
45	45	2251.71	16	19.33	0	-0.27492	1	182.88	-92.0383	-0.00011	0.49119
46	46	2234.29	16	19.33	0	-0.29404	1	182.88	90.7328	-0.00011	-0.19647

Figure 30: A table of the spinquest geometry used in this software