JOINT INSTITUTE FOR NUCLEAR RESEARCH

Dzelepov Laboratory of Nuclear Problems

FINAL REPORT

ON THE SUMMER STUDENT PROGRAM

Study of single pion production in the charged-current interactions of neutrinos in a NOνA Near Detector and enhancement of detecting galactic supernova with NOνA Far Detector

Supervisors:

Oleg Samoylov

Andrey Sheshukov

Student:

Khatbullina Leylya, Russian Federation

St.Petersburg Electrotechnical University

Participation period:

July 18 – September 2

Dubna, 2016

CONTENTS

ACKNOWLEDGEMENTS

I would like to express my appreciation to Oleg Samoylov and Andrey Sheshukov, my practice supervisors and director of University Centre of Joint Institute for Nuclear Research, S. Z. Pakuliak, for giving me an opportunity to participate in Student Summer Program 2016.

I also wish to acknowledge the help provided by Alexander Olshevskiy, Anastasia Bolshakova, Vladimir Allakhverdian and Irina Sidorkina for consulting me in my work and help me with work organization.

I would like to thank to Elena Karpova for excellent organize our excursions.

Finally, I feel all the gratitude for all NOνA JINR team. Thanks for your help!

Part I

Study of single pion production in the charged-current interactions of neutrinos in a NOνA Near Detector

1. Introduction

NOνA is a long-baseline neutrino experiment designed to observe the neutrino oscillation phenomenon of muon neutrinos oscillating into electron neutrinos. The NuMI beam line located at Fermilab produces the muon neutrinos for this observation. The Near and Far Detectors are constructed off-axis of the NuMI beam line and located at Fermilab and northern Minnesota (electron neutrino appearance maximum), respectively. NOνA's primary goals are to measure the neutrino mixing angle Θ_{13}, determine the mass hierarchy (the sign of Δm_{13}^2), and provide insight into the CP violating phase term, δ_{cp}.

An important aspect of long-baseline neutrino experiments is their usage of two detectors in order to make more precise measurements. The placement of one detector near the source of the neutrinos, (for NOνA it is the NuMI beam at Fermilab), allows to monitor and study that original neutrino flux from the source. This information can then be extrapolated to estimate the expected numbers and energies of oscillated neutrinos seen in the Far Detector (FD). Properties of neutrinos without oscillations are studied in Near Detector (ND) furthermore beam is controlled in the NOνA ND.

The physical dimensions of the Near Detector are 4.1 m wide and 4.1 m tall and 14.5 m length, the weight is 330 tons. For the Far Detector, it is 15.5 m width and height, and 59.8 m length, weight is over 14 ktons. [1]

The goal of this part of work is studing of single pion production in the muon neutrino charged-current interactions with nucleons. The quality of event detection are important for us, as well as estimation of efficiency of the NOνA ND.

2. Neutrino interactions with matter

Neutrino-nucleus interactions are currently a topic of great interest as these offer unique opportunities to explore some fundamental questions in physics leading to explain the various thought provoking phenomena in nature and also in determining the structure of matter. Neutrinos are the cleanest probe of nuclear matter as they are light and electrically neutral particles and hence do not interact through the strong nuclear force. When encountering nuclear matter, these penetrate deeply into a nucleon before occasioning a weak interaction after which these either escape unchanged, retaining their flavour or change into their associated charged lepton partners (μ,e,τ).

Weak interactions can proceed via charged-current (CC) or neutral-current (NC) channels. In charged-current interactions, a W+/- boson is emitted as the neutrino converts into its charged lepton partner. Neutral-current interactions are facilitated by the exchange of a Z^0 boson that leaves the neutrino flavor unchanged. As the weak force maximally violates parity, the handedness of the neutrino is fixed and they are all left handed. [2]

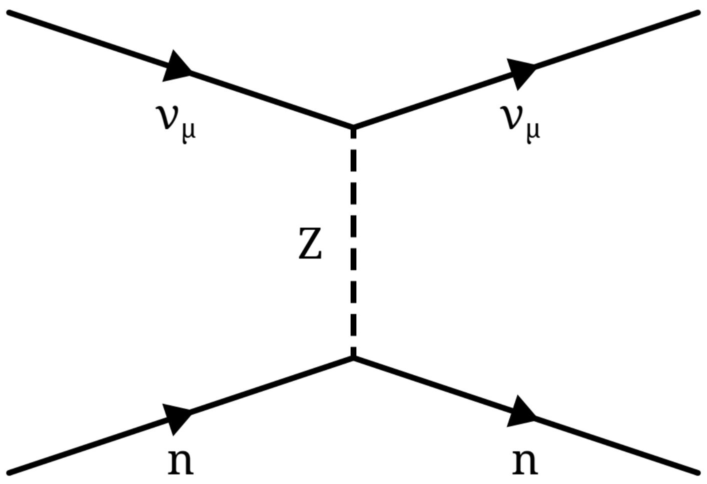

Figure 1: Feynman diagram of NC elastic scattering by a ν_μ on a neutron.

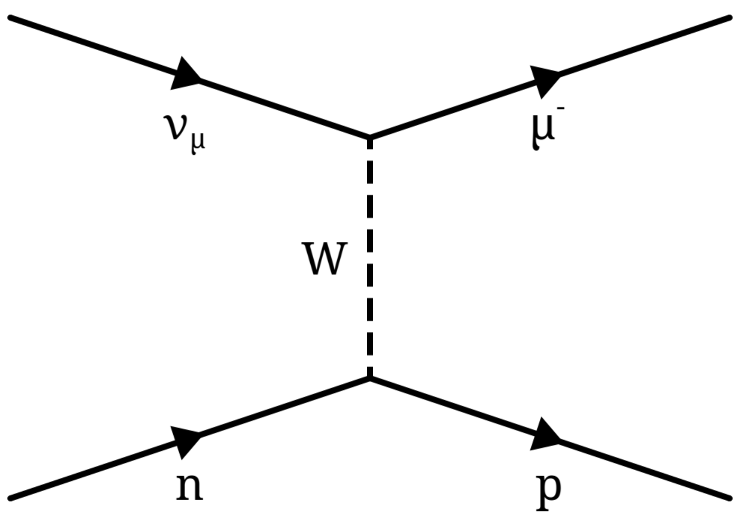

Figure 2: Feynman diagram for ν_μ CC QE scattering.

Beginning at the lowest neutrino energies the first nucleon interactions available to neutrinos are ones in which the nucleon recoils intact (Figure 1). When this occurs via the NC, all neutrinos and anti-neutrinos can scatter off both neutrons and protons in what is referred to as "NC elastic" scattering: $\nu + N \rightarrow \nu + N$.

Once neutrinos acquire sufficient energy they can also undergo the analogous CC interactions. Because of the need to create the charged lepton's mass this is referred to as "quasi-elastic" scattering (CC QE). For ν_μ with $E_\nu < 1$ GeV CC QE is the dominant interaction, however the cross-section plateaus at higher E_ν as the available Q^2 increases and it becomes increasingly unlikely for the nucleon to remain intact (Figure 13).

At higher energies, with more Q^2 available, neutrinos gain access to inelastic scattering processes. Although the lepton side of these interactions looks the same, on the hadronic side the target nucleon is "knocked" into a baryonic resonance, for example an N* or Δ, the available resonances being determined by the neutrino's energy (e.g. Figure 3). [2]

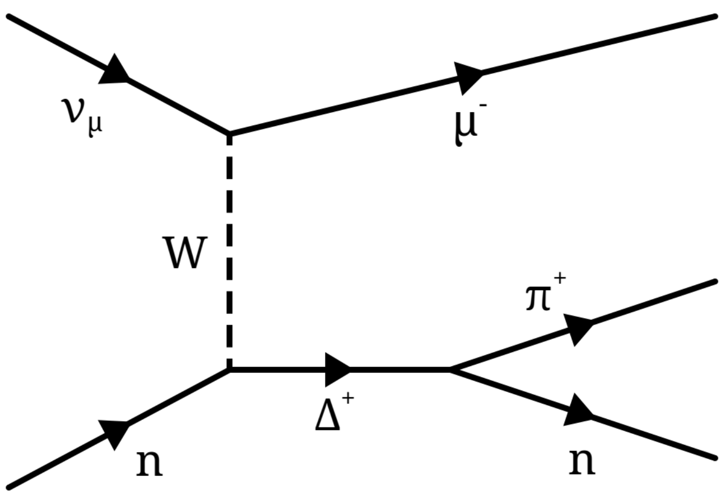

Figure 3: Feynman diagram for an example of resonance: ν_μ CC resonance single-pion interaction.

These resonances then decay back down to a nucleon, most often accompanied by a single pion. However a variety of final-states can result depending on the resonance and can include multiple pions, kaons or a radiative photon. All combinations of neutrinos and anti-neutrinos, scattering off neutrons and protons, via charged- or neutral-current, which obey charge conservation can occur. Resonance production is most significant in the transition region between CC QE and DIS dominance, 0.5 GeV $< E_\nu < 10$ GeV, above which it plateaus like CC QE.

At even higher energies the neutrino is able to transfer sufficient momentum that the internal structure of the nucleon can be probed. Now neutrinos can scatter directly off the quarks inside in a process known as deep inelastic scattering (DIS). The neutrino can scatter off any of the quarks that appear inside the nucleon, including those which form the “sea” of quarks and anti-quarks that are constantly popping in and out of existence. Which of these the neutrino can see depends on the four-momentum transfer available: at lower values the nucleons contain mostly up, down and some strange quarks, but higher values can access the higher-mass and shorter-lived quarks too.

The most visible consequence of DIS is the break up of the nucleon containing the struck quark. As the struck quark recoils the nucleon fragments, and the strong force between the quarks results in “hadronisation”. In an experiment this appears as a jet of strongly interacting particles. DIS is the dominant process for $E_\nu > 10$ GeV. (Figure 4)

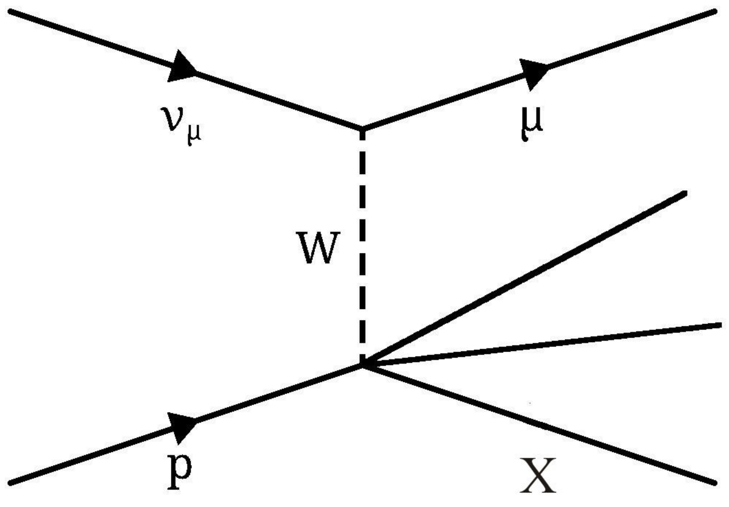

Figure 4: Feynman diagram for an example deep inelastic scattering.

One advantage of using nuclear targets is that it makes available an additional interaction mode known as “coherent” scattering. The defining feature of coherent scattering is that the nucleus recoils as a whole, unfragmented, in the same state as when the neutrino arrived. This can only be achieved if the four-momentum transfer to the nucleus is kept small.

At low E_ν, a neutrino can undergo NC coherent scattering, resulting only in the slight recoil of the struck nucleus. At higher neutrino energies, both CC and NC coherent scattering becomes possible, which also results in the creation of an additional final-state particle such as a π, ρ or K meson. Figure 5 shows the example of CC coherent π+ production. [2]

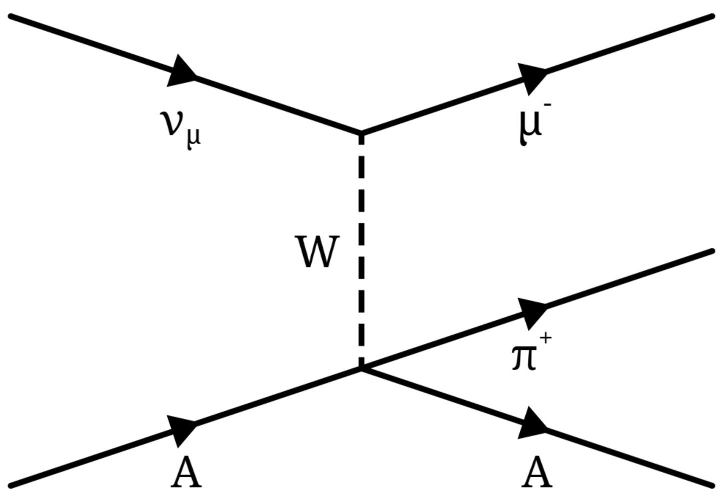

Figure 5: Diagram of CC coherent π+ production.

The requirement that the four-momentum transfer to the nucleus is kept small, strongly constrains the kinematics of coherent scattering such that the final-state lepton, and any additional particles created, are produced at small-scattering angles with respect to the incoming neutrino. It is also this constrained kinematics which results in the coherent cross-sections being relatively small.

3. Neutrino interaction simulations

Neutrino interaction simulations play an important role in the development of experimental analyses, and the interpretation of the resulting measurements.

Nova Experiment uses GENIE simulation. GENIE (Generates Events for Neutrino Interaction Experiments) is a suite of products (Generator, Comparisons, Tuning) for the experimental neutrino community.[3] It is supported and developed by an international collaboration of scientists. The well-known Generator implements a modern framework for neutrino Monte Carlos and includes state-of-the-art physics modules. The GENIE physics model is universal and comprehensive: It handles all neutrinos and targets, and all processes relevant from MeV to PeV energy scales. The Generator includes several tools (flux drivers, detector geometry navigators, specialized event generation apps, event reweighting engines) to supported generator-related analysis tasks developed with special NOvA software. [4]

Monte Carlo methods are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. Their essential idea is using randomness to solve problems that might be deterministic in principle. Monte Carlo methods are mainly used in three distinct problem classes: optimization, numerical integration, and generating draws from a probability distribution.

The beam at the near detector is about 94% pure ν_μ with a mean energy of 2 GeV. The rest of the beam is mainly ν_e and anti-neutrinos. (Figure 8) Data from all the near detectors are used to fine-tune the simulated neutrino energy spectrum. There is the different reaction channels, but in this survey we have focussed on neutrino induced single pion production on nucleons in the charged-current interactions of muon neutrino. (Figure 9) Data was obtained by GENIE ver. 2.10.4 [4]

Simulated events:

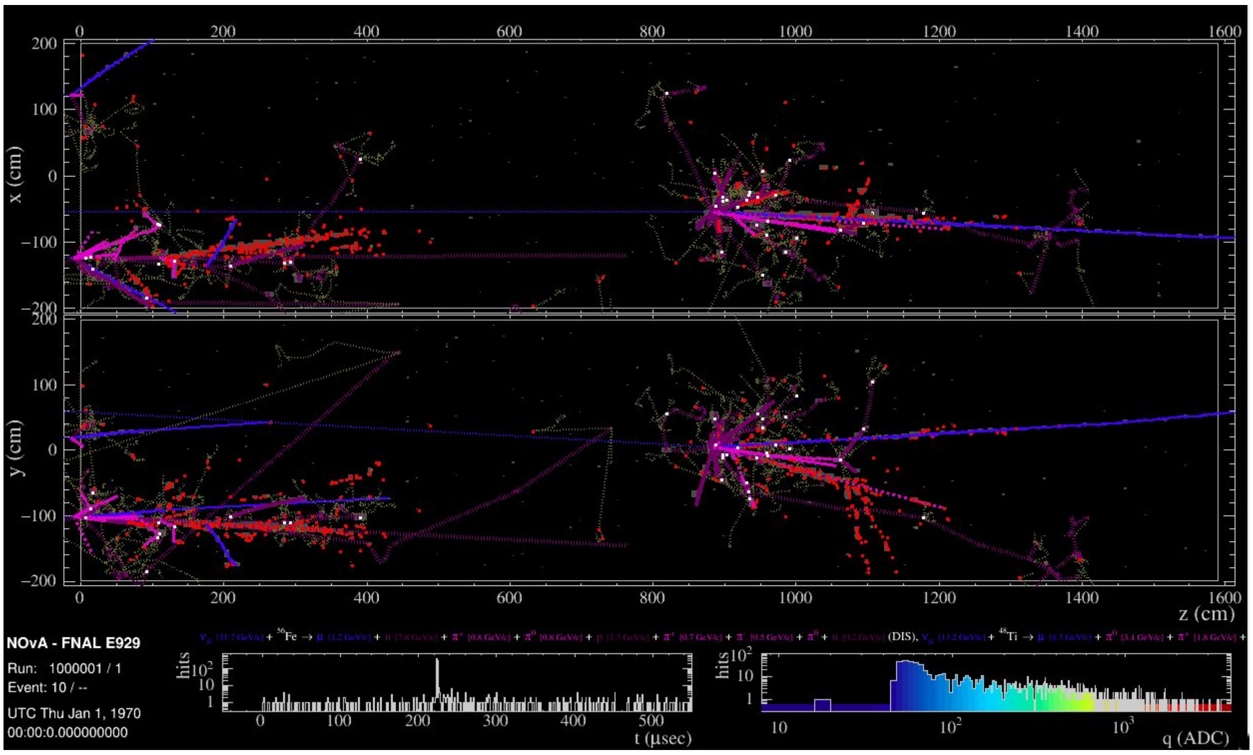

Figure 6: Both RES CC interactions with single pion production

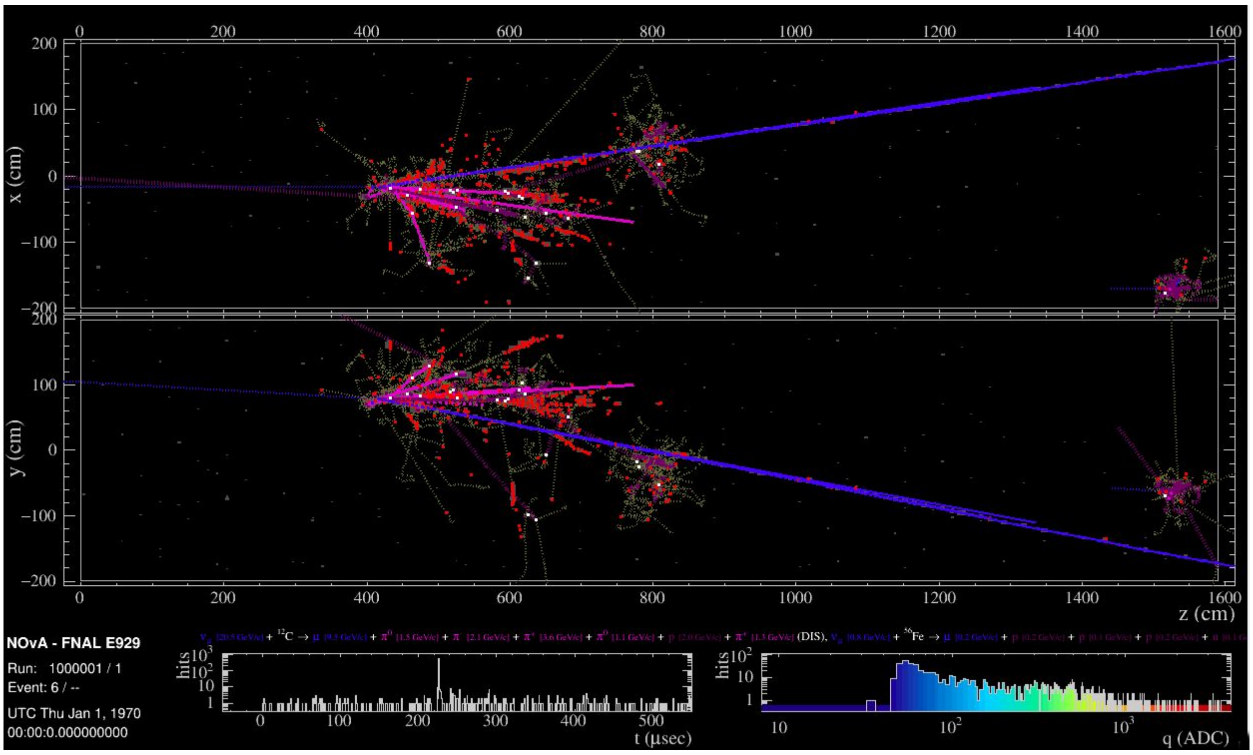

Figure 7: Event with two pion production in RES CC interaction

4. Analysis

Data analysis was performed with Art tools [5].

The Art Framework consists of a collection of libraries and a few runtime-configurable main programs, all written in the C++ programming language. In addition to the main program that users execute directly, art includes facilities to:

- define a variety of experiment-written modules that perform the steps in a workflow
- configure the coordinated execution of these modules
- handling experiment-defined descriptions of experimental data
- read and write files containing these data
- managing ancillary data (e.g. detector geometry descriptions, calibration data)
- keeping track of the provenance of the data generated during execution of the program

The art framework is supported by a wide variety of other software products, all of which are provided to users of art along with the framework. These include the configuration language used to determine the workflow for an art job, commonly-used physics libraries, and even the compilers used to build art. NOvA software uses this Fermilab Art framework. [6]

Plots were build with CERN ROOT facilities. ROOT is an object-oriented program and library developed by CERN. It was originally designed for particle physics statistical analysis, visualisation and storage and contains several features specific to this field. It is mainly written in C++ but integrated with other languages such as Python and R. [7]

Image below (Fig. 8) shows number particles generated in beam. Green - number of ν_μ, orange - number of ν_e, red - number of $\overline{\nu_e}$ and blue - number of $\overline{\nu_\mu}$.

Particle type can be identified by PDG Code [8] :

PDG Code	Particle
14	muon neutrino
12	electron neutrino
-14	muon antineutrino
-12	electron antineutrino

Table 1: Elements PDG codes.

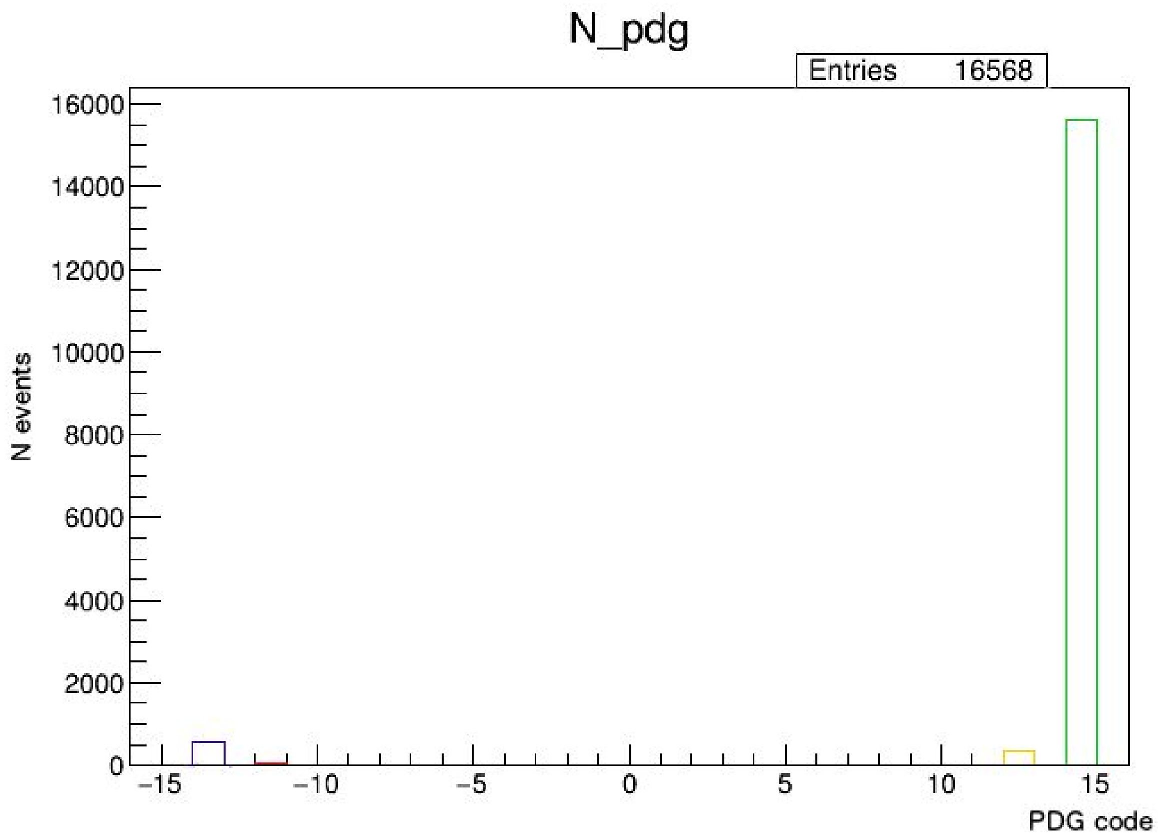

Figure 8: Neutrino beam compositions as defined in Table 1.

Plot below shows number of interactions with specific number of produced pions.

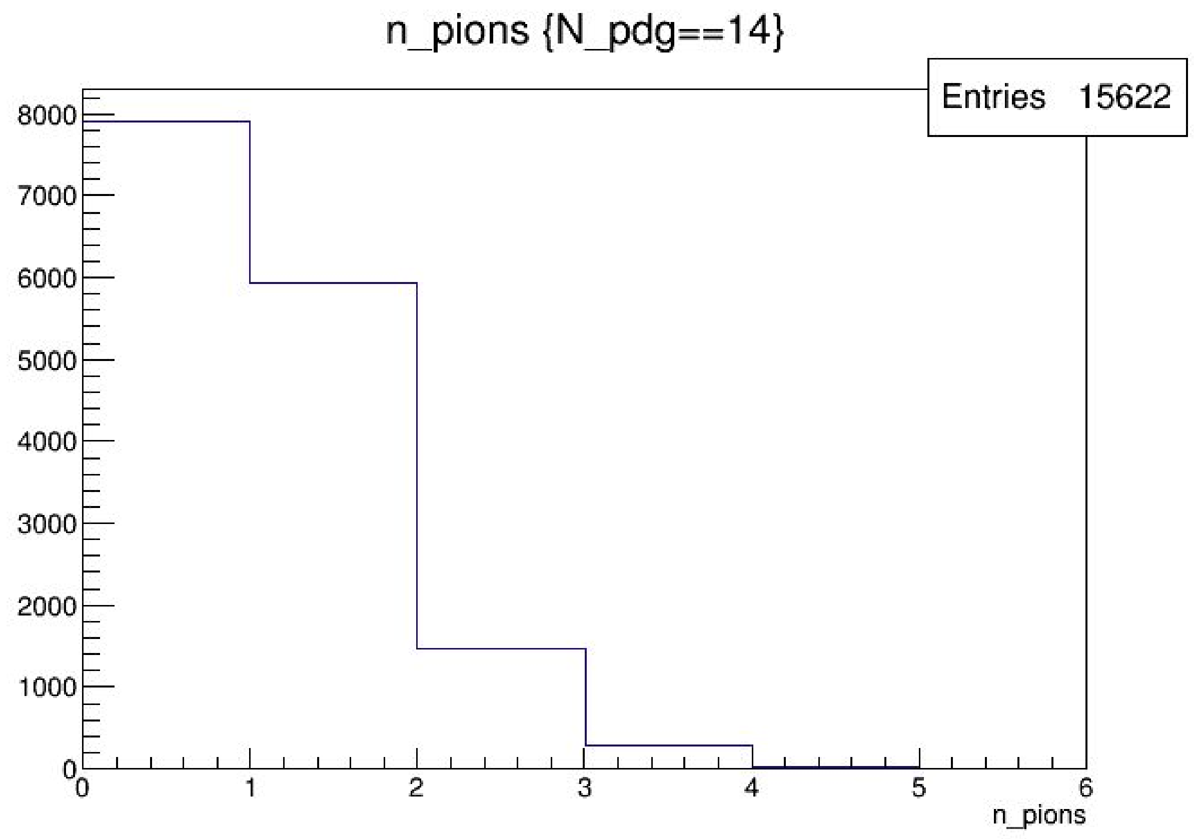

Figure 9: Pion multiplicity of the neutrino interactions in NOvA ND.

# pions	# interactions
0	7908
1	5929
2	1476
3	284
4	22
5	3

Table 2: Pion multiplicity of the neutrino interactions

Next histogram presents number of events divided by interaction type: charged-current or neutral-current.

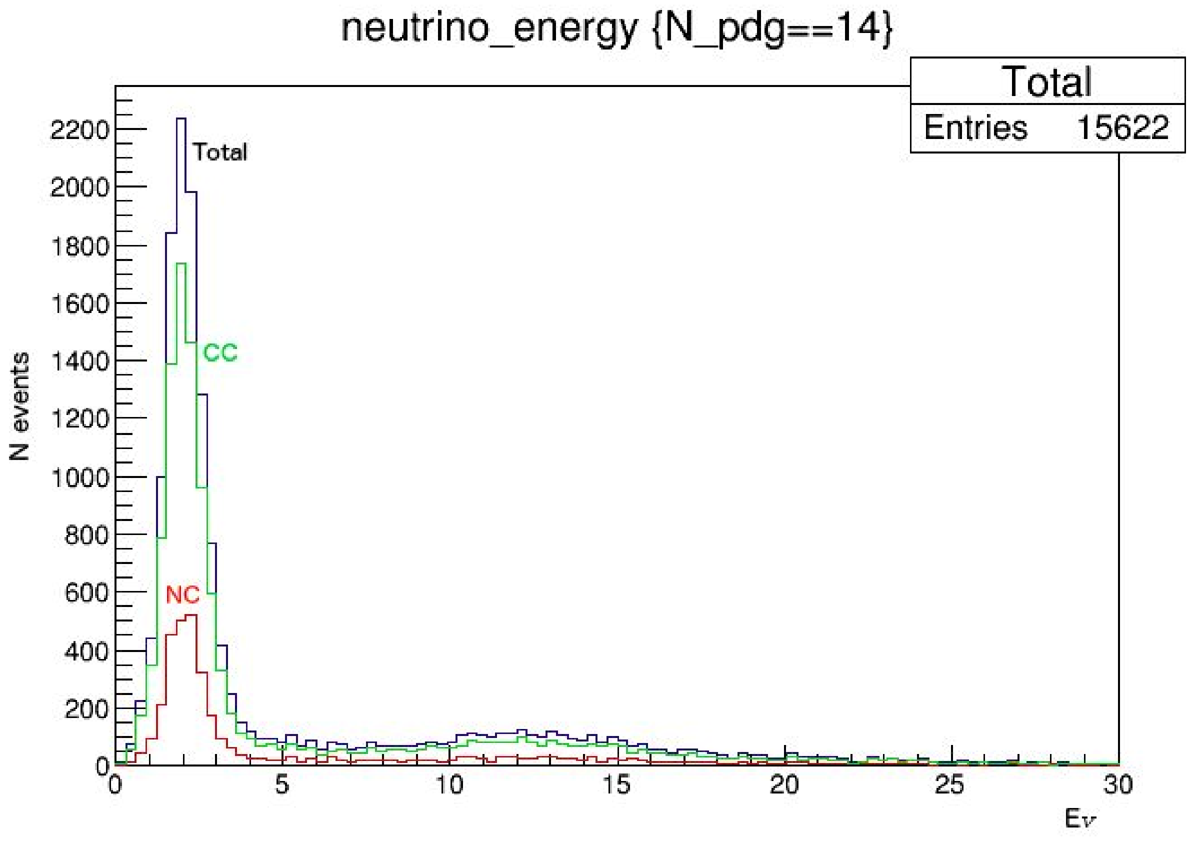

Figure 10: Number of interactions depending on neutrino energy for different channels.

We simulated all possible kinds of neutrinos scatterings in NOνA ND. We are interested of pion multiplicities for all the types of neutrino-nucleon interactions. As presented on image below (Figure 11).

As we can see those interactions make up a large part of all interactions, they can be well simulated and detected by NOνA ND. Further among this interactions we will select only events with single pion.

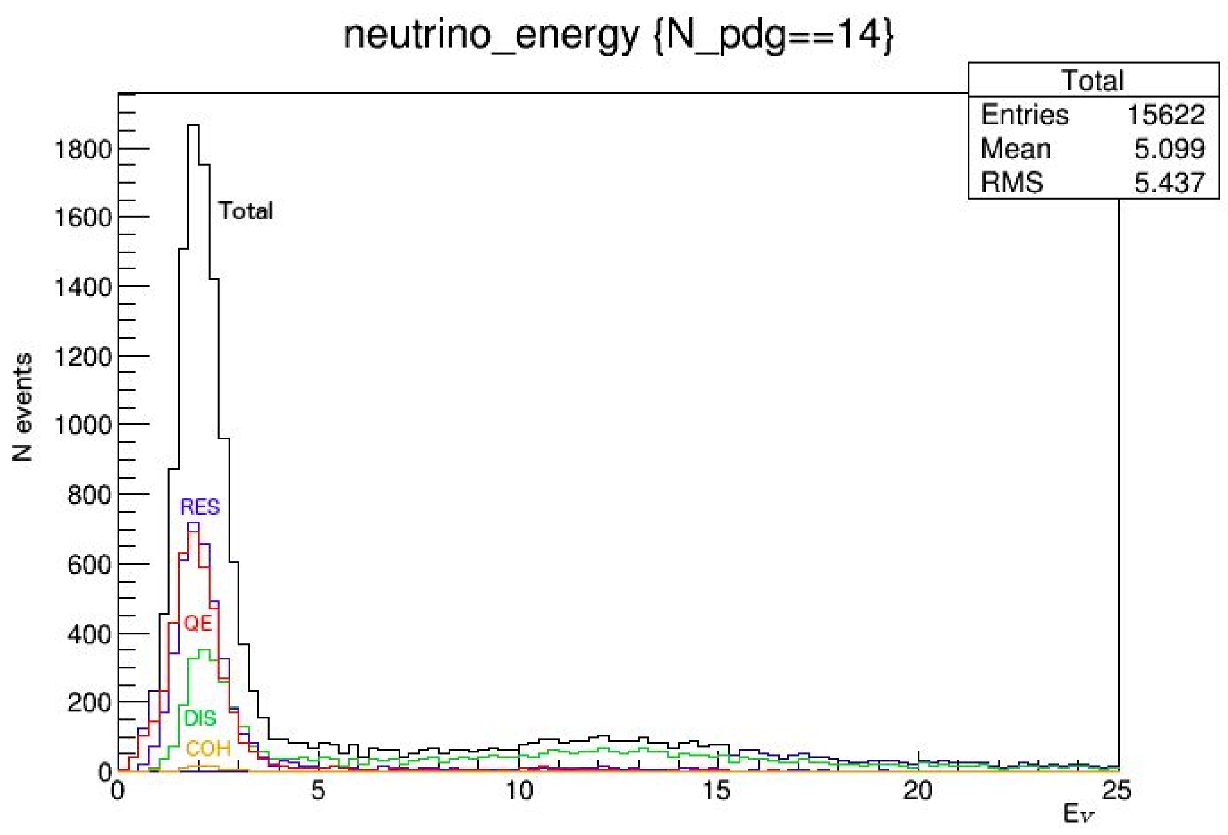

Figure 11: Neutrino energy spectra for different interactions types.

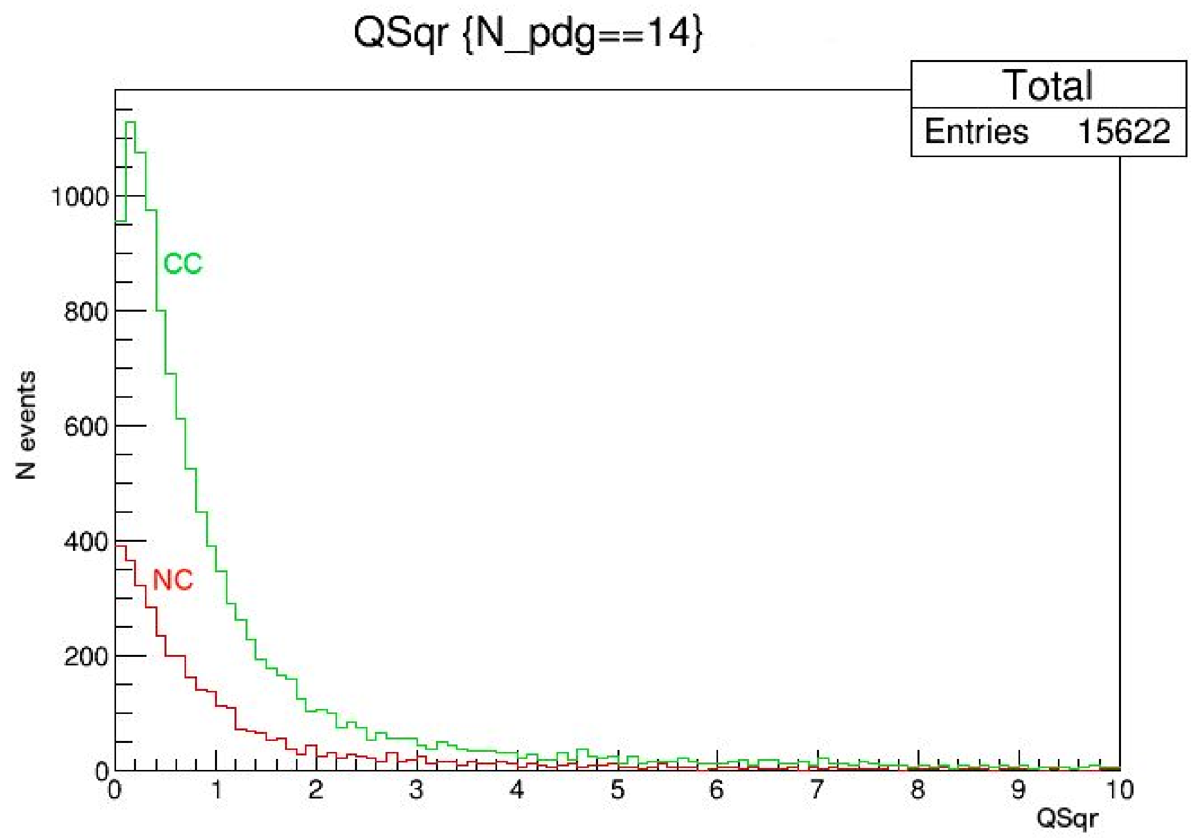

Figure 12: Q^2-distribution for CC and NC events.

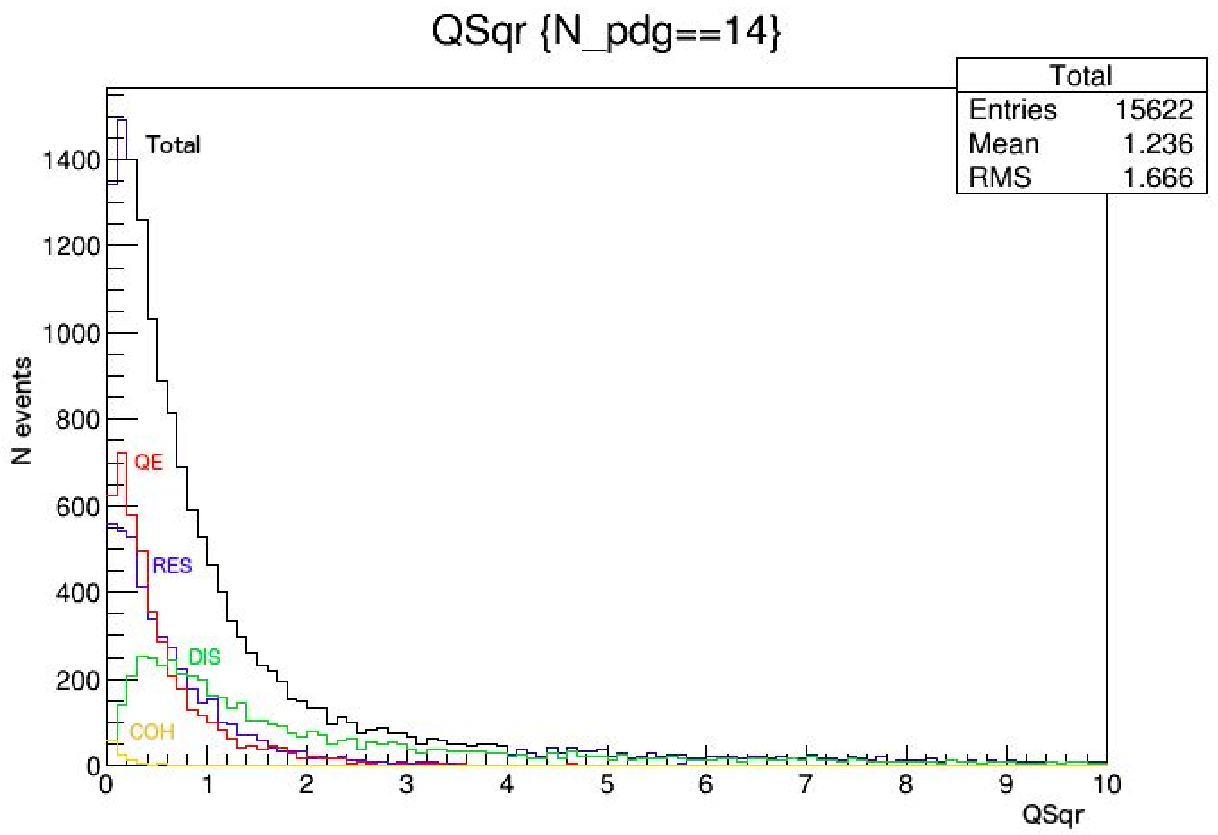

Figure 13: Q^2-distribution for different interactions types.

Consider the following table:

Interaction type	Percent of Total
Charged-current (CC)	**75.8%**
CC quasi-elastic	21.4%
CC single pion production	31.7%
CC deep inelastic scattering	20.9%
CC (other)	1.8%
Neutral-current (NC)	**24.2%**

Table 3: Neutrino Interactions in GENIE for NOνA beam composition.

As can be seen number of CC interactions with single pion dominates over the other interactions. Below histograms show distributions of single pion interactions.

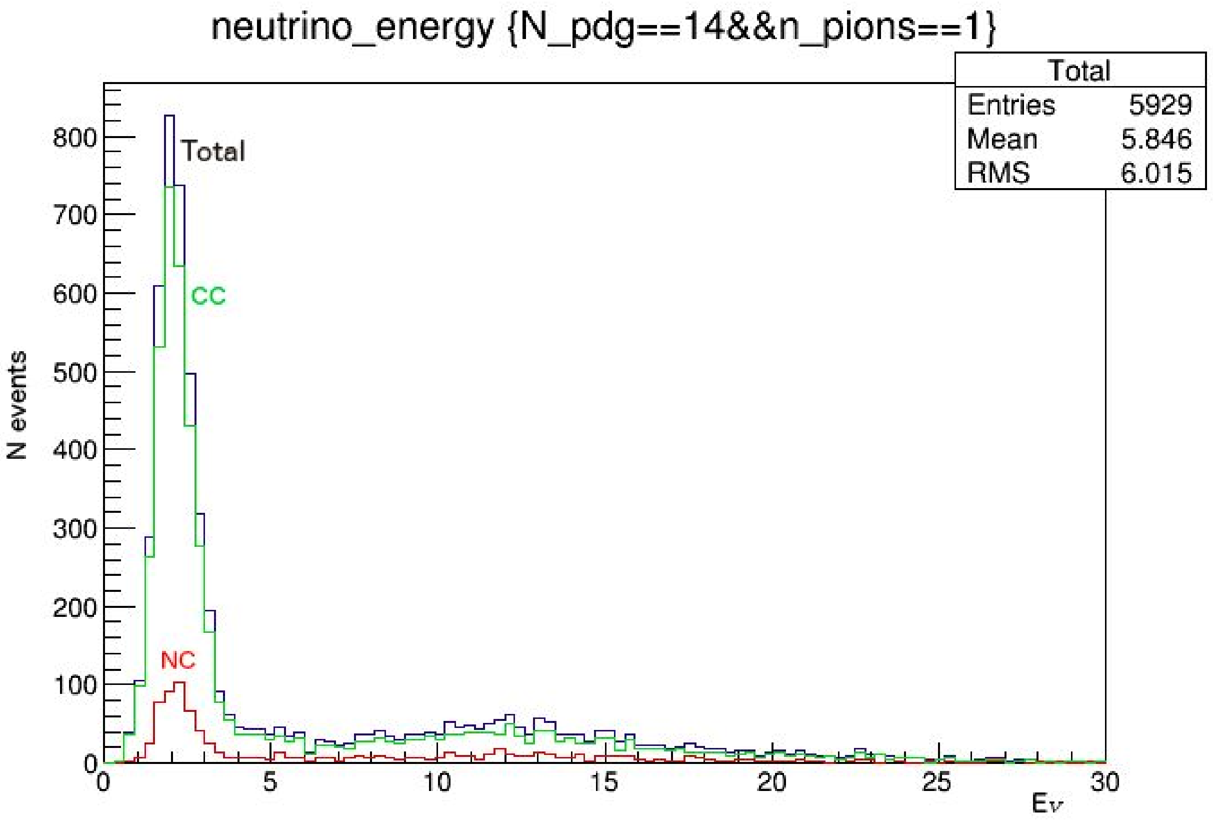

Figure 14: Neutrino energy spectra for both CC and NC channels.

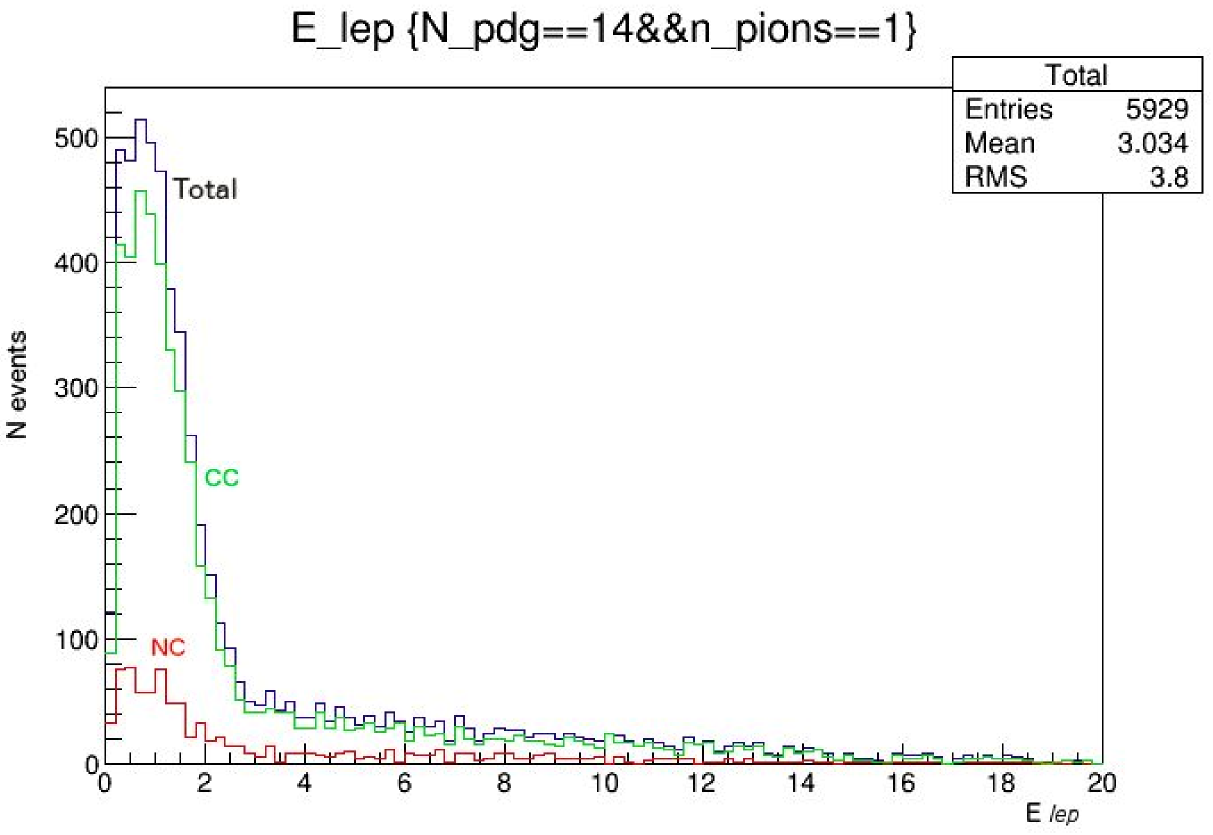

Figure 15: Distribution of scattered lepton energy.

Finally we can tell that Near Detector provide an opportunity to detect CC channels with single pion. Further we channels with two pions interaction can be studied. This information can be used for beam controlling and improving its detection efficiency. We can measure interaction kinematics from obtained data, that could be used for analysis of reconstructed tracks, f.e. with Kalman Track algorithm, and an energy estimator based on track's length. (Table 4)

Kinematic variables
$Q^2 = 2E_\nu(E_\nu - p_\mu^L) - m^2$
$\nu = E_\nu - E_\mu$
$W^2 = M^2 + 2M\nu - Q^2$
$x = Q^2 / 2M\nu$
$y = \nu / E_\nu$

Table 4: The neutrino interactions are often described by this kinematic variables.

Q^2 is the invariant 4-momentum transfer squared, ν is the neutrino energy transfer, *W* is the effective mass of all secondary hadrons (invariant hadronic mass), *x* is the Bjorken scale variable, *y* is the relative energy transfer, E_ν is the incident neutrino energy, E_μ and p_μ^L are the energy and longitudinal momentum of the muon, *M* is the nucleon mass and *m* is the muon mass.

For each hadron in the hadronic system, we define the variables $z = E_h / \nu$, $x_F = 2p_L^* / W$ and p_t where E_h is the energy in the laboratory frame, p_L^* is the longitudinal momentum in the hadronic c.m.s., and p_T is the transverse momentum with respect to the momentum transfer direction. [9]

5. KalmanTrack reconstruction

Several track reconstruction methods for the NOvA detectors have been developed to address the many tracking applications.(Figure 16) One of these methods, based on Kalman filters, will be describing. The Kalman filter approach to track reconstruction was chosen because the formalism allows for both the finding and fitting of tracks in one process. Additionally, the Kalman filter routine can be extended to allow for the proper handling of multiple scattering. Currently the reconstruction has been developed for straight tracks as an approximation to the true particle tracks which show nonlinear effects due to multiple scattering.

The reconstruction takes place in three steps. The first step applies a base level calibration of the hits recorded in the detector correcting for cell to cell differences in the detectors. The second step takes all the hits recorded in the full trigger window and clusters them into groups associated together in time by looking for a minimum level of activity in the detector without large time gaps between hits. This is the result of work by Slicer4D module. The final step of the reconstruction takes all the individual time groups of hits and applies a geometric pattern recognition routine to find tracks. The pattern recognition routine is made up of three major subroutines.

The first subroutine forms track seeds by assuming that adjacent hits in each time grouped cluster belong to the same track in each independent detector view. The second subroutine then uses a Kalman filter to propagate the track seeds plane by plane through the detector adding hits to the track that are consistent χ^2 with the track. The consistency of a hit is determined based on the change in of the track fit from the inclusion of the hit. The track fit is updated after the addition of any hit using the Kalman filter to perform a weighted average fit of the track to the hits. The third subroutine then takes all the tracks found in each independent view and matches them together forming a three dimensional reconstructed track. [10]

The reconstruction method requires that each track passes through 3 planes and has at least 4 hits in each view.

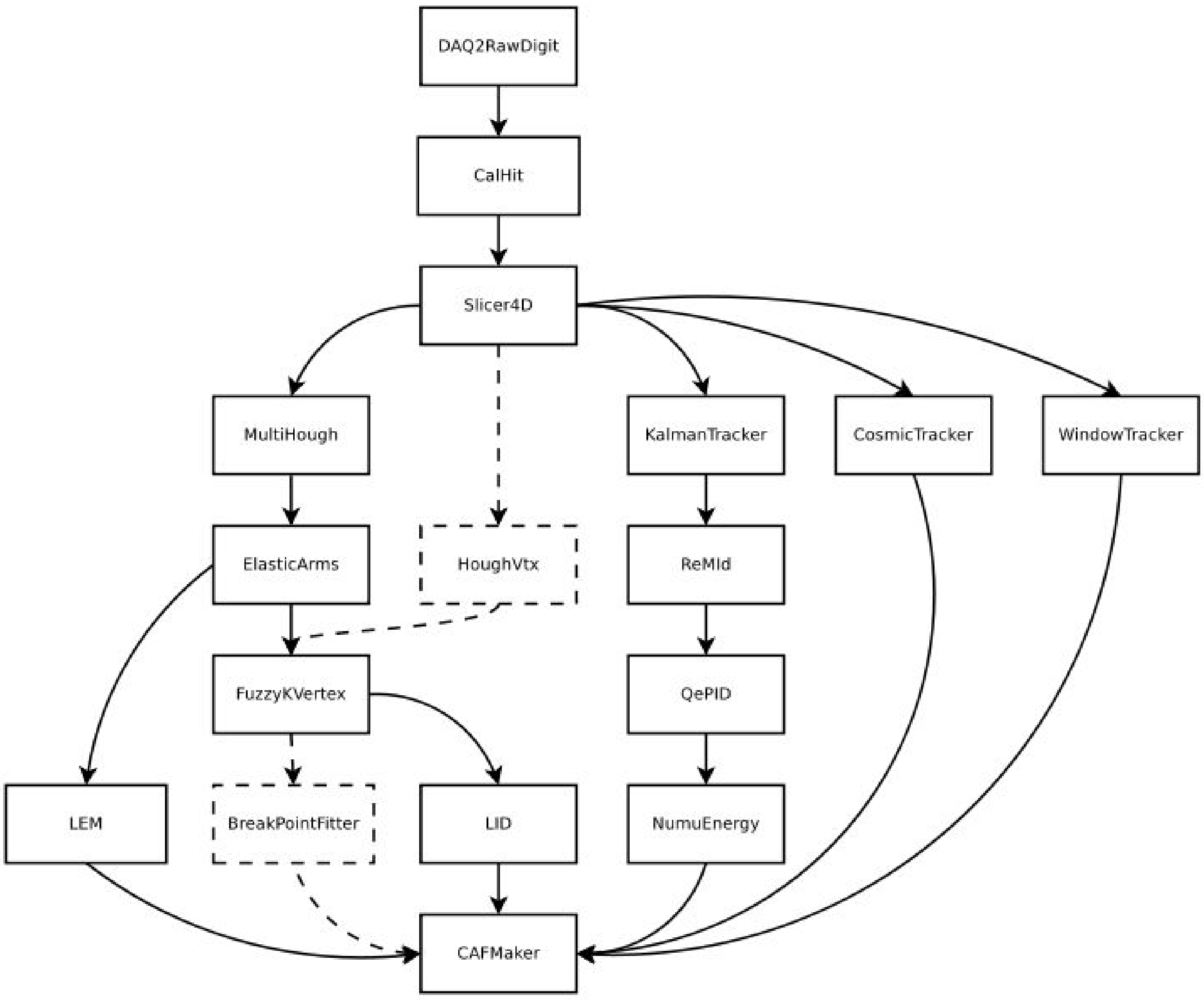

Figure 16: Full reconstruction chain for the neutrino events and background as well.

After KalmanTrack reconstruction we get files in RECO format with information about reconstructed tracks.

Reconstruction of GENIE events:

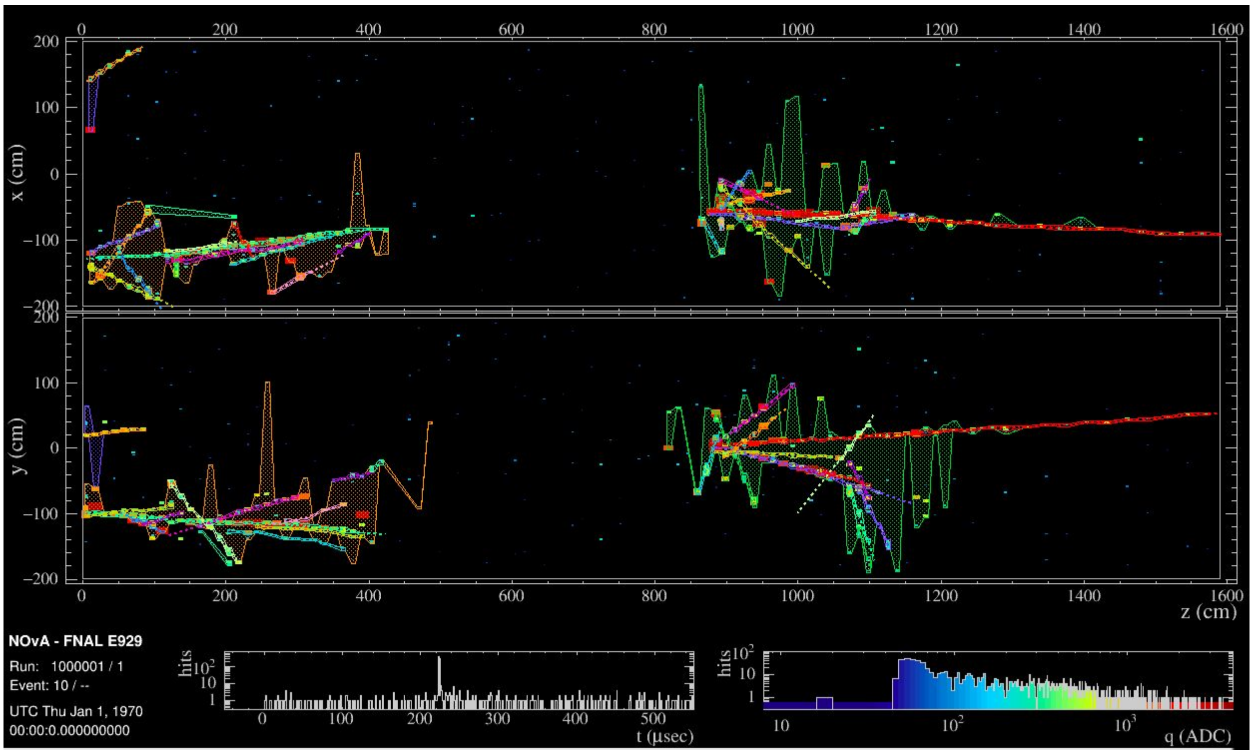

Figure 17: An example of reconstruction events simulated on Figure 6.

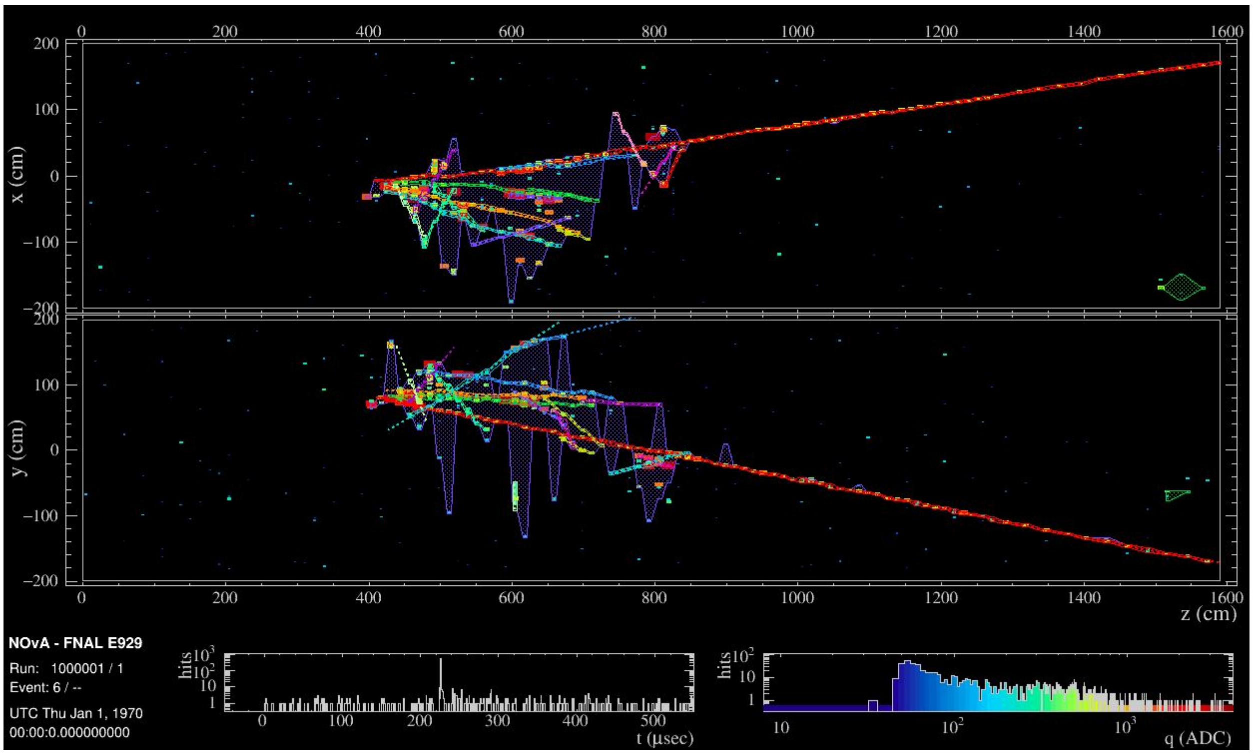

Figure 18: An example of reconstruction events simulated on Figure 7.

6. Results

In this work we studied neutrino interactions chanel ν_μ CC π+/- with NOvA software included neutrino generator GENIE. We calculated and sorted NOvA neutrino interaction compositions for different types of Interaction Currents and Kinematical regions. We estimated pion multiplicities of the channels and checked track reconstruction algorithm based on Kalman fitter. We are expected a good advantages to use it for single-pion cross-section measurements. Further neutrino interaction chanel ν_μ CC 2π can be studied.

7. References

1) NOvA Neutrino Experiment https://www-nova.fnal.gov/

2) Daniel I. Scully (2013) “Neutrino Induced Coherent Pion Production”. *University of Warwick, Department of Physics*

3) GENIE http://www.genie-mc.org

4) NovaSoft http://nusoft.fnal.gov/nova/novasoft/

5) Art (Event-processing framework) http://art.fnal.gov

6) Nova Art Wiki https://cdcvs.fnal.gov/redmine/projects/novaart

7) ROOT User's Guide https://root.cern.ch/guides/users-guide

8) L. Garren (2006) “Monte Carlo Particle Numbering Scheme”. *Fermilab*

9) Tingjun Yang (2009) “Study of Muon Neutrino to Electron Neutrino Oscillations in the MINOS Experiment”.

10) N. Raddatz (2011) “Track Reconstruction in the NOvA Experiment”. *School of Physics and Astronomy, University of Minnesota*

11) Mark Messier (2008) “Neutrino Detectors for future facilities”. *NUFACT Summer school*

Part II

Enhancement of detecting galactic supernova with NOνA Far Detector

1. Introduction

A supernova (SN) is an astronomical event that occurs during the last stellar evolutionary stages of a massive star's life, whose dramatic and catastrophic destruction is marked by one final titanic explosion. For a short time, this causes the sudden appearance of a 'new' bright star, before slowly fading from sight over several weeks or months. Due to the wide range of astrophysical consequences of these events, astronomers now deem supernovae research, across the fields of stellar and galactic evolution, as an especially important area for investigation. In core collapse supernova, the vast majority of the energy is directed into neutrino emission, and while some of this apparently powers the following main explosion 99%+ of the neutrinos escape the star in the first few minutes following the start of the collapse. [1]

Theoretically NOνA Far Detector can detect signal from SN. Though there are several solvable problems: NOνA FD is on the surface, NOνA detectors are designed for measuring ~2 GeV neutrino interactions, but SN neutrinos are ~10 MeV. NOνA DDT (Data Driven Trigger) system works with milliblocks: 5ms data chunks, but neutrino signal is extended in time for ~1s.

In this paper we will create dataset with SN signal and we will try to detect it with convolution function. Convolution kernel is an array of 200 values. First we will count number of interaction candidates, then we apply filter filter to the time sequence, to enhance signal shape. If we see a signal shape above BG level, then trigger signal we be send.

2. Dataset creating

Signal model of SN, which has the average distance from the Earth 10 kiloparsecs, can be simply described as follows: duration of block 1 sec., signal peak on 20ms. (Figure 1). Every second detectors produce 200 values - number of reacted clusters. There exists background signal which value is more then SN signal. (Figure 2)

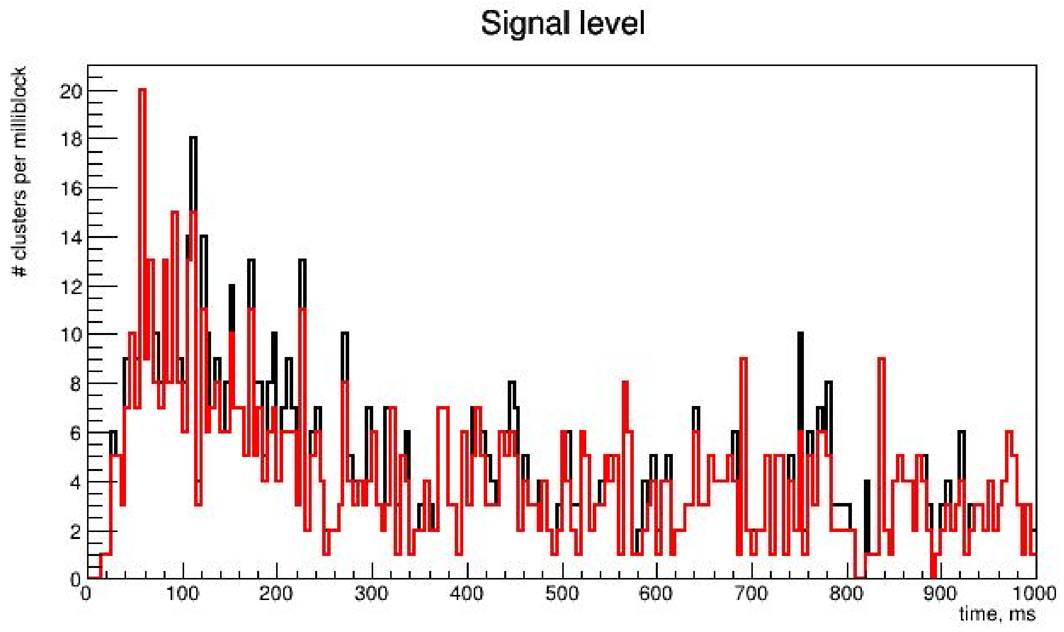

Figure 1: Signal level

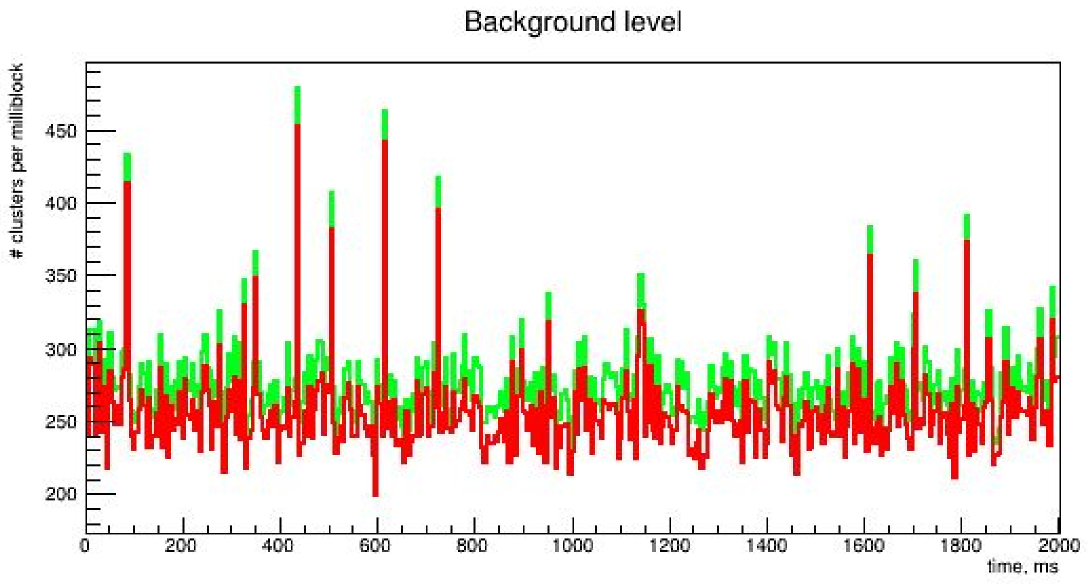

Figure 2: Background level

The signal model presented on Fig.1-2. It’s the result of detailed simulation of supernova neutrinos interacting in the detector, modeling of electronics response and reconstruction of obtained data.

Poisson distribution was used for dataset modelling. This is a discrete distribution and sum of two poisson distributions is poisson distribution too.

For the next parameters:

```
len = 3                # the number of seconds in the file
minimum = 100           # rough minimum signal value
peak = 106              # rough signal peak value
lambd = 100             # lambda of poisson distribution
```

We get this picture of simulated signal dataset duration of 3 seconds, where signal starts from 2 second.   .

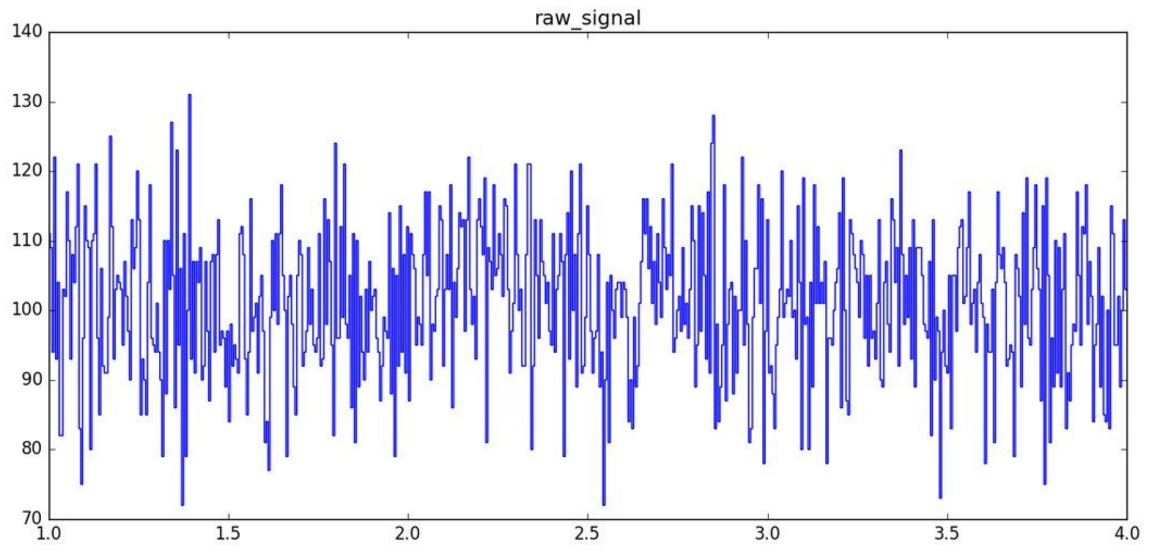

Figure 3: Simulated dataset.

On such dataset we will be training our program to learn best convolution kernel.

Convolution is a mathematical operation on two functions (f and g); it produces a third function, that is typically viewed as a modified version of one of the original functions. For complex-valued functions *f, g* defined on the set *Z* of integers, the discrete convolution of *f* and *g* is given by:

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]\, g[n-m]$$, where *g* - original series and *f* - kernel.

3. Algorithm

Initially for finding best kernel we will use genetic algorithm. Genetic algorithms (GA) are adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. As such they represent an intelligent exploitation of a random search used to solve optimization problems. Although randomised, GAs are by no means random, instead they exploit historical information to direct the search into the region of better performance within the search space. [2]

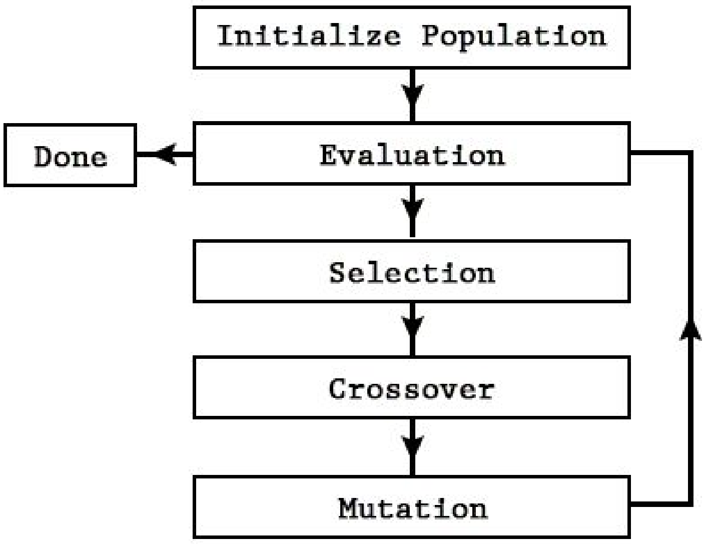

Figure 4: GA flow chart

Development steps:

- set a fitness function to evaluate the solution domain
- create the first generation
- set mutation and cross functions
- determine terminating conditions

4. Development

Programs were developed on Python 3 language, additionally IDE PyCharm and jupyter Notebook tools where used. [3,4,5]

Fitness function

A fitness function is a particular type of objective function that is used to summarise, as a single figure of merit, how close a given design solution is to achieving the set aims.

```python
def fitness_function(data, is_signal):
    if is_signal:
        return min(0, max(data) - 1)
    else:
        return min(0, 0.5 - max(data))
```

After filtration signal should be greater then 1 and noise should be less then 0.5.

Loss function

Loss function is used for parameter estimation, and the event in question is some function of the difference between estimated and true values for an instance of data.

```python
def calculate_loss(kern):
    scr = 0
    for track in train_data_sgn:
        fb = np.convolve(track, kern[:-1], mode='valid') + kern[-1]
        scr -= fitness_function(fb, True)
    for track in train_data_bkg:
        fb = np.convolve(track, kern[:-1], mode='valid') + kern[-1]
        scr -= fitness_function(fb, False)
    return scr
```

Our goal - minimize this function by kernel.

Mutation function

Mutation alters one or more values from its initial state.

```
def mutation(kernel):
    new_kernel = np.copy(kernel)
    index = random.randint(0, len(new_kernel) - 1)
    new_kernel[index] += random.gauss(0, 0.0001)
    return new_kernel
```

Cross function

Crossover is a genetic operator used to vary the programming of parameters from one generation to the next. It is analogous to reproduction and biological crossover. Cross over is a process of taking more than one parent solutions and producing a child solution from them.

```
def cross(kern1, kern2):
    a = random.random()
    new_kern = kern1 * a + (1 - a) * kern2
    return new_kern
```

First generation

We created first generation with random values.

```
kernel = np.random.rand(200) / 100
```

Terminating conditions

- a solution is found that satisfies minimum criteria (score < *eps*)
- fitness function does not more improve
- fixed number of generations reached (for example #generations = 100)

First results

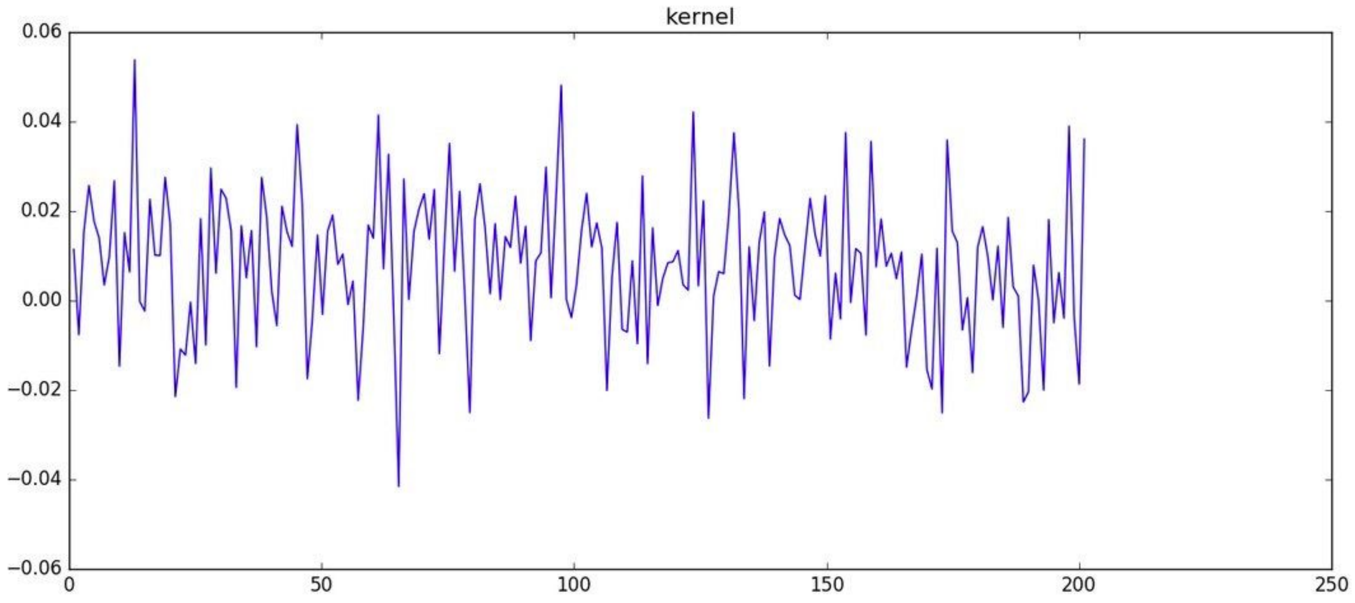

Figure 5: Kernel

Parameters:

```
minimum = 100           # rough minimum signal value
peak = 106              # rough signal peak value
lambd = 100             # lambda of poisson distribution
population_size = 20    # populations size, number of different kernels
n = 1000                # number of iterations
eps = 0.01              # terminating condition
```

Result looks not very good. Optimization is very slow and takes lot of time. As improvement we will try to use gradient descent and increase epochs number.

5. Gradient descent

Stochastic gradient descent is a stochastic approximation of the gradient descent optimization method for minimizing an objective function that is written as a sum of differentiable functions. In other words, SGD tries to find minimums or maximums by iteration. In stochastic gradient descent, the true gradient of $Q(\omega)$ is approximated by a gradient at a single example: $\omega = \omega - \eta \nabla Q_i(\omega)$ [3]

In pseudocode, stochastic gradient descent can be presented as follows:

```
● Choose an initial vector of parameters $\omega$ and learning rate $\eta$
● Repeat until an approximate minimum is obtained:
        - randomly shuffle examples in the training set.
        - $for\ i = 1, \ldots, n\ do : \omega = \omega - \eta \nabla Q_i(\omega)$
```

Gradient is calculated by function `def calculate_grad(kern)`.

Results:

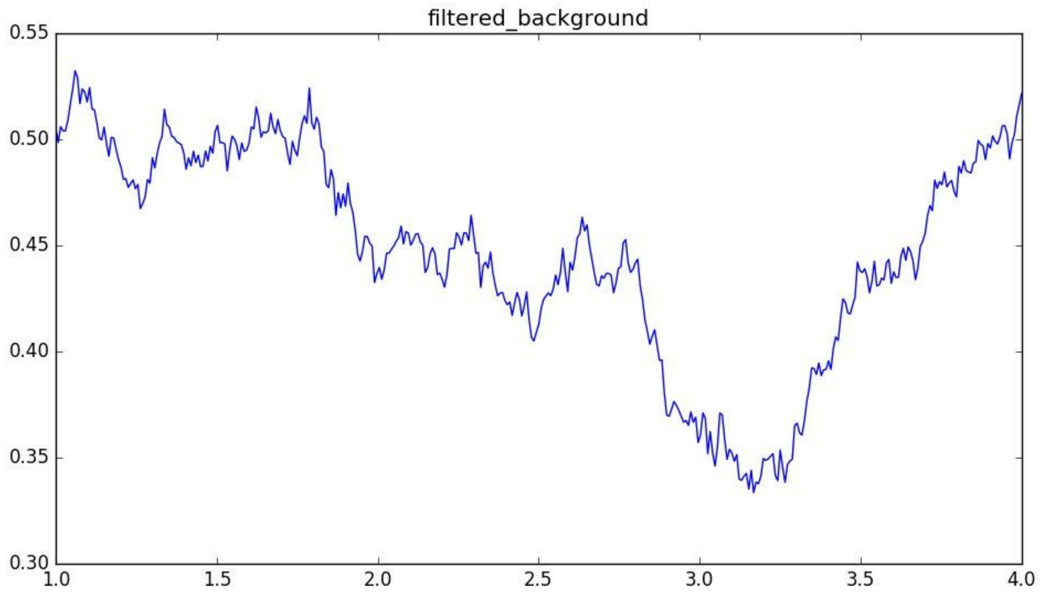

Figure 6: Filtered background.

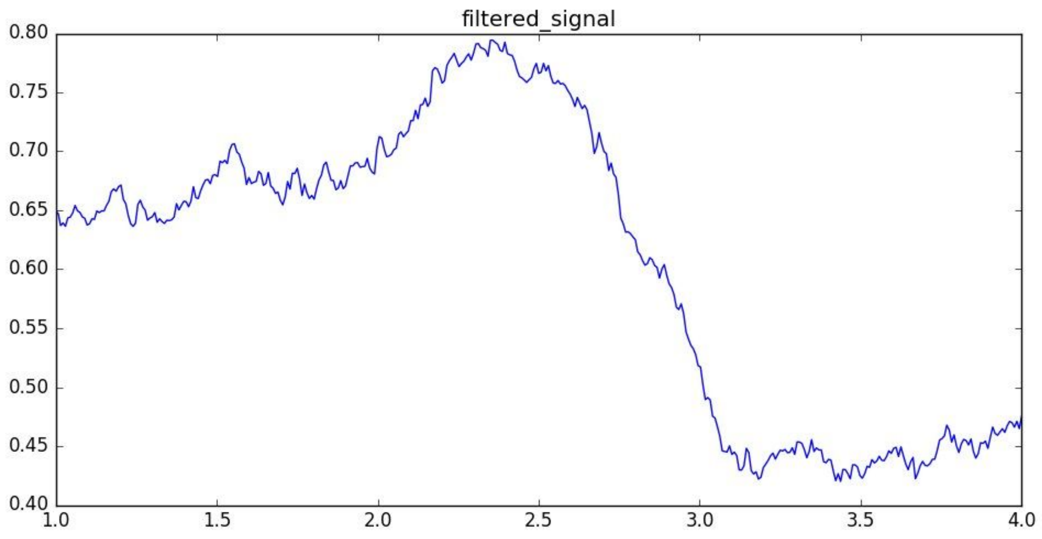

Figure 7: Filtered signal.

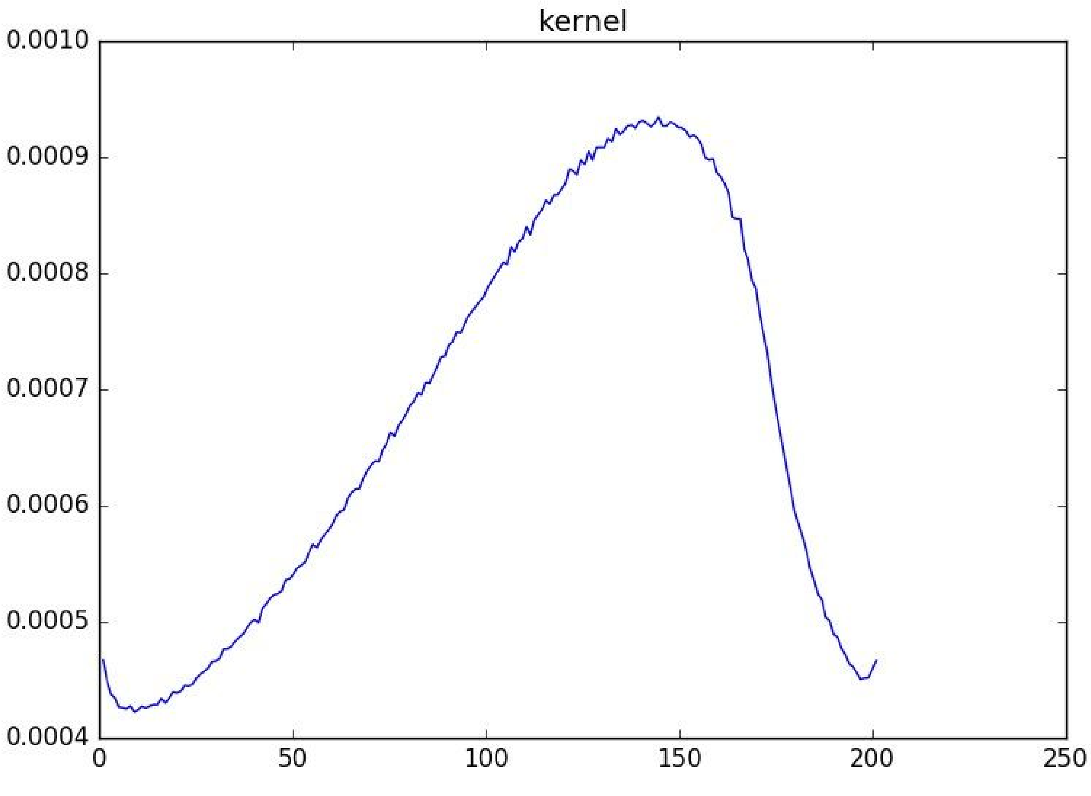

Figure 7: New kernel

Parameters:

```
minimum = 100       # rough minimum signal value
peak = 106          # rough signal peak value
lambd = 100         # lambda of poisson distribution
```

This kernel looks better and we can tell that gradient descent helps us improve results. Among the drawback can be noted slow work and among advantages is an easy implementation and applicability for wide range of optimization problems.

6. Results

In this work we solved the problem of finding the optimal kernel to detect SN. Despite the fact that signal to noise ratio is large (signal is poisson with mean 106, noise is poisson with mean 100) we can separate signal from noise with good precision.

On our simulated data we get type I error = 0.0004 and type II error = 0.0053.

Other method like neural networks can give some improvements, but their training will take more time and will require more effective algorithms.

7. References

1) Supernova https://en.wikipedia.org/wiki/Supernova

2) Genetic algorithm https://en.wikipedia.org/wiki/Genetic_algorithm

3) Python Software Foundation https://docs.python.org/3/

4) PyCharm IDE https://www.jetbrains.com/pycharm/

5) Jupyter Notebook http://jupyter.org/

6) Gradient descent https://en.wikipedia.org/wiki/Stochastic_gradient_descent

7) Repository https://github.com/HeyLey/supernova