12-1-2021

DEVELOPMENT OF METHODOLOGY AND SOFTWARE TO ENSURE THE QUALITY PERFORMANCE OF THE DEEP UNDERGROUND NEUTRINO EXPERIMENT HIGH VOLTAGE FIELD CAGE

Archit Jaiswal

DEVELOPMENT OF METHODOLOGY AND SOFTWARE TO ENSURE THE QUALITY PERFORMANCE OF THE DEEP UNDERGROUND NEUTRINO EXPERIMENT HIGH VOLTAGE FIELD CAGE

by

ARCHIT JAISWAL

Presented to the Faculty of the Honors College of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

HONORS BACHELOR OF SCIENCE IN PHYSICS

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2021

ACKNOWLEDGMENTS

I would like to thank Dr. Jaehoon Yu for providing guidance during this research and allowing me to participate in his research group. I am grateful to Cristobal Garces, Cristian Garces, and Hector Carranza for helping me throughout the process of research. I am thankful to the staff of the McNair Scholars Program at UT Arlington for all kinds of support and helpful advice during the research. My family's virtues and sacrifices will always remain the eternal reason for my success.

December 04, 2020

ABSTRACT

DEVELOPMENT OF METHODOLOGY AND SOFTWARE TO ENSURE THE QUALITY PERFORMANCE OF THE DEEP UNDERGROUND NEUTRINO EXPERIMENT HIGH VOLTAGE FIELD CAGE

Archit Kalpeshkumar Jaiswal, B.S. Physics

The University of Texas at Arlington, 2020

Faculty Mentor: Jaehoon Yu

The Deep Underground Neutrino Experiment (DUNE) is the U.S. flagship experiment being designed to study the characteristics of neutrinos which make up a quarter of the fundamental particle map. These subatomic particles help physicists understand the fundamental constituents of matter in the universe. In DUNE, neutrino interactions will be captured inside the active volume of the time projection chamber (TPC) using liquid argon (LAr) as the medium. The DUNE detector will be built using various parts assembled according to the proposed design which enables accomplishing the precision measurements for the underlying physics goals. The detector is constructed by assembling various basic components from the hardware manufacturers. It is crucial to

ensure the quality of every component before it is assembled into such a gigantic structure because a minor imperfection could compromise the precision and cause significant uncertainties to the scientific measurements. In this research, a methodology and a software tool to conduct the quality assessment of the DUNE High Voltage (HV) components, which get delivered from the manufacturer are developed. The outcomes of this research will be used to conduct quality assurance and quality control for all the HV parts before using them to build the components of the detector.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The Deep Underground Neutrino Experiment (DUNE) is the U.S. flagship experiment being designed to study the characteristics of neutrinos which make up a quarter of the fundamental particle map. These subatomic particles help physicists understand the fundamental constituents of matter in the universe, the fundamental forces between them, and the origin of the universe itself. In DUNE, neutrino interactions will be detected inside the active volume of the time projection chamber (TPC) using about 70,000 tonnes of liquid argon (LAr) as the medium. The DUNE detector will be built using various parts assembled according to the proposed design which enables accomplishing the precision measurements for the underlying physics goals. The detector is constructed by assembling various basic components from the hardware manufacturers. It is crucial to ensure the quality of every component before it is assembled into such a gigantic structure because a minor imperfection could compromise the precision and cause significant uncertainties to the scientific measurements. In this research, a methodology and a software tool to conduct the quality assessment of the DUNE High Voltage (HV) components, which get delivered from the manufacturer are developed. The outcomes of this research will be used to conduct quality assurance and quality control for all the HV parts before using them to build the components of the detector.

CHAPTER 2

LITERATURE REVIEW

Neutrinos undergo a weak interaction with a nucleon or an electron and have an extremely low probability of interacting with the matter. Therefore, in order to detect as many neutrinos as possible, a vast mass is necessary in the detector. DUNE is an international experiment aimed to provide a deeper insight into the neutrino property and proton decay studies. DUNE comprises of two neutrino detectors: a near detector and a far detector. The world's most powerful and intense neutrino beam will be projected towards these detectors from the Fermilab Long Baseline Neutrino Facility (LBNF) as shown in Figure 2.1. The near neutrino detector will be constructed at the Fermi National Laboratory in Batavia, Illinois, near Chicago. The far detector will be constructed 1.5 kilometers below the Earth's surface at the Sanford Underground Research Facility (SURF) in South Dakota, approximately 1,300 kilometers away from the neutrino beam source at Fermilab [1]. These detectors will enable the physicists to potentially redefine the current understanding of neutrinos and explore for new subatomic phenomena.

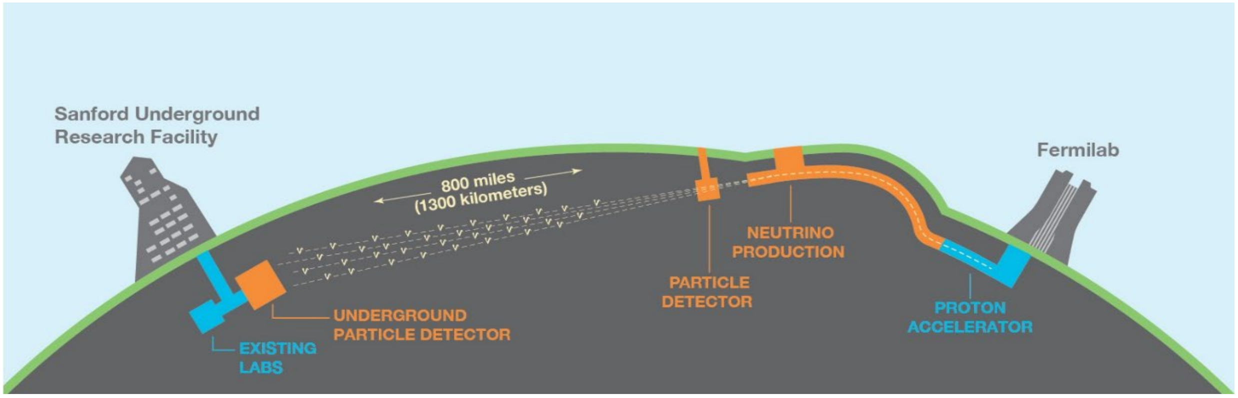


Figure 2.1: DUNE design for long baseline neutrino experiment [1]

As a neutrino gets detected in the TPC, it interacts with an Ar atom, producing multiple secondary charged particles from the breakup of the nucleus. These charged particles traverse through LAr and ionize the Ar atoms, creating ionization electrons and the positive Ar ions. These ionization electrons drift through LAr under the uniform electric field provided by a device called field cage and get collected in the anode, the detection device. The field cage is constructed inside the cryostat for the detection of these charged particles. The field cage will provide a uniform electric field to the active volume of LAr to observe the neutrino interactions. The far detectors of DUNE will hold a total of about 70 kt of LAr mass distributed into the four cryostats. Each detector holds 17.5 kt of LAr, in which 10 kt of LAr is in the active volume of the TPC [1].

The field cage is constructed in modules made of aluminum profile electrodes which are separated by 6 cm and held by insulating Fiber Reinforced Plastic (FRP) I-beams. High Voltage Divider Boards (HVDBs) are mounted onto the aluminum profiles as shown in Figure 2.2. The HVDBs produce the voltage difference of 3 kV between the two neighboring aluminum electrodes, providing the uniform 0.5 kV/cm electric field in the active volume and force the ionization electrons to drift up towards the detection plane in the active volume surrounded by the field cage [2]. The charge readout anode present at the top of the detector to collect the data of the interaction in the active volume of the detector through capturing the ionization electrons. Due to the electric field applied across the active volume of the TPC, the positive ions are drifted towards the cathode while the negatively charged ionization electrons drift towards the anode located at the top of the detector.

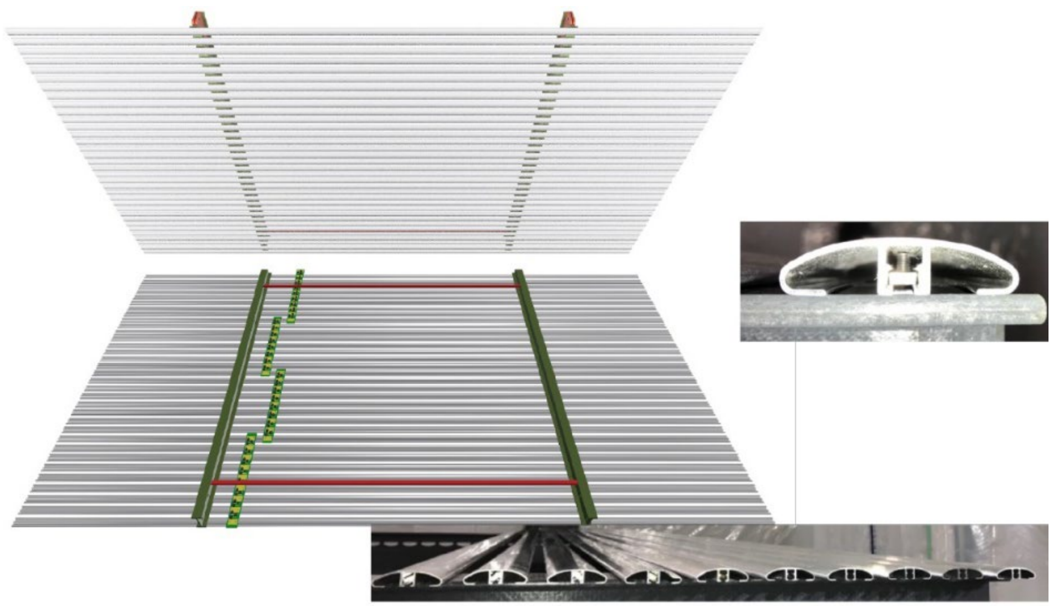


Figure 2.2: Sub-module assembly showing the aluminum profiles mounted on the FRP I-Beams [Credit: Brookhaven National Laboratory (BNL) US national laboratory in Upton, NY]

The aluminum profiles, along with HVDBs, are attached to 2-meter-long FRP pultruded structural elements, also known as I-beams [3]. The entire field cage in a cryostat is made up of 12 super-modules. Each of these super-modules has a dimension of 12 m × 12 m. Super-modules are made of submodules, which are 2-meter-high and 4 meter wide. Each submodule comprises of two I-beams, which support the aluminum profiles. The whole assembly of field cage will be submerged in the ultra-pure LAr and exposed to the high voltage inside a cryostat with the outer dimensions of 65.8 m (L) × 18.9 m (W) × 17.8 m (H) [3].

CHAPTER 3

SIGNIFICANCE OF RESEARCH

The field cage is made up of super-modules, which in turn are made up of smaller submodules. Any error in the preparation of the parts will directly affect the performance of the field cage. During the process of measuring the parts, several types of errors can influence the results and become the source of systematic uncertainty. The significant contributors to errors are faulty equipment, level of experience of the experimenter, and other random causes [5]. Therefore, it is essential to ensure the accuracy in preparing the field cage parts. I-beams and latch beams are the supporting structures for a submodule. Minute defects in I-beams and latch beams will proliferate the uncertainties as the submodules are assembled to produce a super-module, resulting in a more significant uncertainties in the whole field cage structure and its performance.

The uniform electric field provided by the field cage is responsible for creating uniform drift velocity of the charged particles. The position of charged particles is determined by the measurement of time which is inversely proportional to the drift velocity. The accumulation of uncertainties in the field cage parts preparation can eventually result in non-uniformity of the electric field, which affects the drift velocity of the ionization electron and thus degrades the performance of the detector. In addition, any sharp points or fibers from the FRP beams in the FC structure could be the source for an electron or ion plasma discharge, which temporarily blinds the detector or generates an ambiguous signal.

The components of the field cage arriving from various manufacturers undergo a quality control test to ensure that the quality requirements are satisfied according to the scientific design. The purpose of this research is to develop a methodology to perform a quality control examination of the components received from the manufacturers. The secondary purpose is to create the mechanics to determine the quality of each component by analyzing the data collected by measuring various attributes of a particular element. The outcomes of this research create an algorithm for the researchers to successfully conduct the quality control assessment in a timely and consistent fashion and reduce the risk of non-uniformities in the electric field. The ROOT based data analysis software will provide a data visualization feature to the researchers, which enables them to accurately determine the defects in the parts provided by the manufacturers. The software also aids in maintaining the standards of components that will be used to build the field cage and other supporting structures.

CHAPTER 4

METHODOLOGY FOR CONDUCTING MEASUREMENTS ON FRP BEAMS

The purpose of this research is to minimize the uncertainties in FC parts by ensuring the quality of materials received from the manufacturers. Several geometries of the parts like length, width, hole diameter, hole-to-hole distances, etc. will be physically measured using the scientific tools. To successfully complete the task of measuring field cage parts for quality control, a well descriptive algorithm has been developed and described in this section to ensure that the measurements are free from biases.

The initial step of conducting the quality inspection begins with labeling the parts using the appropriate color label with the part identification number printed on it. While labeling, visual inspection needs to be conducted for all the FRP beams for physical damages, and any kind of inconsistencies and defects should be noted. Visual inspection can help researchers to save time and effort in measuring the parts which are damaged due to mishandling during transportation. The technical design drawing can be used to obtain the details about structure while performing a visual inspection.

After the FRP beams pass the visual inspection for flatness and physical damages, they undergo dimensional inspection according to the detailed technical design drawing. It is crucial that a precise measuring tool, such as calipers, is used to perform the measurements of the FRP beams to maintain accuracy. Therefore, selecting a measuring tool based on the units and the level of precision specified in the technical design drawing will allow useful measurement readings for the data analysis task.

The lower jaws and upper jaws of the calipers can be used to measure various internal geometries in the FRP beams. The upper jaws are smaller in size and can be used to measure features like the diameter of the hole of the cuts through the FRP beams. The upper and lower jaws of the calipers are shown in Figure 4.1. In contrast, the lower jaws can be used to measure the external geometry, such as the thickness of the flange, or the shortest distance between the holes, etc.

While conducting measurements, it is crucial to ensure that the jaws of a caliper are correctly aligned and parallel to the edges where distance is being measured [2]. Having jaws perpendicular to the object will increase the accuracy of the measurement. A minute gap between the surface of the FRP beam and the caliper jaws will add measurable errors into the measurements, which will increase the difficulty in determining the quality of the parts.

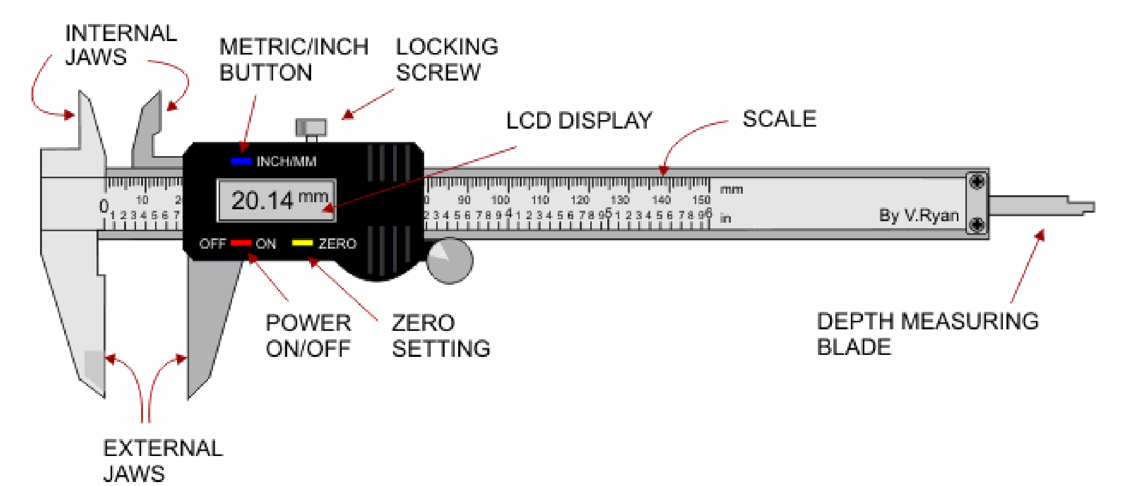


Figure 4.1: A digital calipers [9]

After measuring the field cage parts, the measurement readings will be analyzed using a software tool to generate plots and help determine the quality of parts under examination. A ROOT [8] macro file has been developed as a part of the research, which allows a user to feed a data file containing the experimental data of measurements and the theoretical values of the corresponding feature of the FRP beams from the design issued by DUNE collaborators. The ROOT platform-based software file will execute and generate the graphical results to aid the researchers in determining if the tested part meets the quality requirements, which were detailed in the technical design drawing of the parts.

CHAPTER 5

ROOT BASED DATA ANALYSIS PROGRAM

ROOT is a data analysis framework created by CERN (European Organization for Nuclear Research, Geneva, Switzerland) [8] to provide a tool for statistical data analysis to the High-energy Physics (HEP) community. The TTree object container is capable of efficiently conducting statistical data analysis over a vast data set, allowing a user to choose a comprehensive set of mathematical and statistical functions along with various algorithms for generating regression analysis to facilitate the process of data analysis [6].

During the development process, to efficiently craft a user-friendly computer program and automate most of the user inputs, the measurements data analysis program was divided into five components as follows:

I. Functionality disclosure

II. Input file processing

III. Calculation of deviations

IV. Generation of the histogram plot

V. Generating a regression analysis (fit) on a histogram plot.

5.1 Functionality Disclosure

An important software developer practice is to inform the user about the functionalities of the software and guide the user through the features offered by the software. After executing the dataAnalysisProgram.C file on the ROOT framework, the program starts executing. The ROOT framework must be launched from the same directory where the dataAnalysisProgram.C file is stored on the computer. The program is designed such that it instructs the user about its execution and provides a description of its functionalities after execution in the ROOT environment.

The lines 45 – 57 of the dataAnalysisProgram.C (Appendix A) demonstrated the code which is responsible for informing about the functionalities to its user. The numbers are associated with the corresponding functionalities to ease the functionality selection for the user. After printing the instructions on the terminal window, the program will wait for the user input. The user can enter the number associated with the functionality after the command prompt represented by >> or the user can exit the program and return to the ROOT environment by entering "quit" after the command prompt. If the program receives "1" as user input, it will generate a histogram along with a Gaussian fit without performing any calculation on the data present in the input file. If the program receives "2" as user input, then it will perform the calculation on the input data from the file and generate a histogram along with a Gaussian fit on the distribution. The functionalities mentioned above are illustrated with figures in section 6 of this paper.

5.2 Input File Processing

After informing the user about the possible options to select in the program, the user will be asked to enter the name of the input file. This input file should contain the experimental values of measurements conducted using the measuring tool for the quality control of the field cage parts. The input file should follow the format specified in Appendix B, and the input file should be stored in the same location as the dataAnalysisProgram.C (Appendix A) file. Following the specified format for the input file will allow the program to interpret the input data without the user's intervention. The file name is case sensitive and entering the wrong file name will gently stop the program execution while bringing the awareness to the user about the issue.

Lines 60 – 92 of the dataAnalysisProgram.C (Appendix A) shows the code which fulfills the task of segregating the title of a plot, axis labels, experimental measurement values, and the theoretical measurement value from the data of the input file. The software continues to read in the data from the input file until the end of the file is reached. All the rows of the input file are read as a string object, and by using the exception handling mechanism provided by C++ and the characteristics of generating errors by a C++ function, it became possible to process the data from input file and store it according to its actual data type. The numerical values from the input file are stored in a data type known as vector<double>, which stores the data in a sequence similar to a stack or an array. Similarly, axis labels and titles are stored as a vector of string. All the string objects will be passed into a function that converts them into a numeric value but generates an error if the object is not convertible to a numeric value. Using this fact, objects which do not generate an error while passed into the function are the numeric value, and those objects which create an error are left as a string and stored into a vector<string>.

5.3 Calculation of Deviations

After processing the input file, if the user opts to obtain the visualization of the differences of experimental values from the theoretical values, the calculation of deviations needs to be performed by the program. Lines 164 – 168 shown in the code of dataAnalysisProgram.C (Appendix A) are responsible for calculating the differences between experimental values and the theoretical value. The theoretical value is subtracted from the experimental value to obtain a positive deviation if the measured value is higher than the desired value. A histogram will plot the deviation values to facilitate the researchers in determining the difference of FRP beams from the suggested design.

5.4 Generating a Histogram Plot

After completing the calculations and preparing the data points to plot a histogram, the program will create a histogram object and fill the data points to the histogram. ROOT data analysis framework has a histogram class that supports three types of histograms: 1D, 2D, and 3D. The data analysis can be performed using a 1D histogram with one float per channel. Therefore, the dataAnalysisProgram.C (Appendix A) creates an object of the TH1F class from the ROOT library using the constructor TH1F(). Lines 171-210 of the code generates a histogram plot using the ROOT library. A constructor is a member function of a class which initializes objects of a class. The constructor of TH1F class requires five parameters to create a TH1F object which is described as follows:

a) const char* name: A string that will be displayed on top of the histogram plot
b) const char* title: A string containing the title on the histogram, x-axis label, and y-axis label
c) Int_t nbinsx: An integer value which will serve as the number of bins in the histogram
d) Double_t xlow: The minimum value of the plot in the histogram window
e) Double_t xup: The maximum value of the plot in the histogram window

The variables need to be computed and processed by the program before calling the constructor to generate a histogram. The dataAnalysisProgram.C (Appendix A) is designed to tolerate specific user faults like missing the titles and axis labels in the input file. A common mistake of not having axis labels in the input file will not cause the program to crash. In case of such user mistakes, the program will skip labeling both the axis and generate the histogram as it will do in the ordinary circumstances.

5.5 Generating a Regression Analysis on a Histogram Plot

After the histogram creation process, the program will define an object from the TF1 class in the ROOT library. The TF1 object can be referred to as a 1-dimensional function defined from a lower bound to an upper bound. A constructor from the TF1 class is used to create a TF1 object by passing the parameters link name of the TF1 object, name of the function, the lower bound value, and the upper bound value.

After defining a Gaussian function from TF1 class, TH1::Fit() is used to generate a Gaussian curve on the plotted histogram. TH1::Fit() is given an input of predefined function created by an object from the TF1 class. The code in lines 206 – 210 is responsible for producing a Gaussian curve on the histogram.

CHAPTER 6

DATA ANALYSIS PROGRAM TESTING

The Data Analysis Program developed during the research was tested to ensure that it is fully functional and capable of visualizing the data collected from the quality test. DUNE far detector field cage comprises of various parts, but due to the unavailability of the parts that will be used in the DUNE far detector field cage, the older version of a FRP latch beam will be used to perform a quality test using the methodology and software described in this paper. The DUNE collaborators provide the technical design drawing for the FRP beams, which consists of every technical detail that is required to manufacture the part. A technical design drawing for the latch beam used for the quality test is shown in Appendix C of this paper.

For the convenience of data analysis, various geometries on the latch beam are assigned using an alphanumeric labeling system, as shown in Figure 6.1. The holes are assigned a letter starting from A, whereas the outer edges of the latch beams are assigned a number. For the purpose of testing the software, only the measurement data such as the diameter of holes A, length of slot D, and the breadth of the latch beam are analyzed and discussed in the paper. Eight independent measurements are taken for each parameter using a digital caliper with a precision of 10 micrometers.

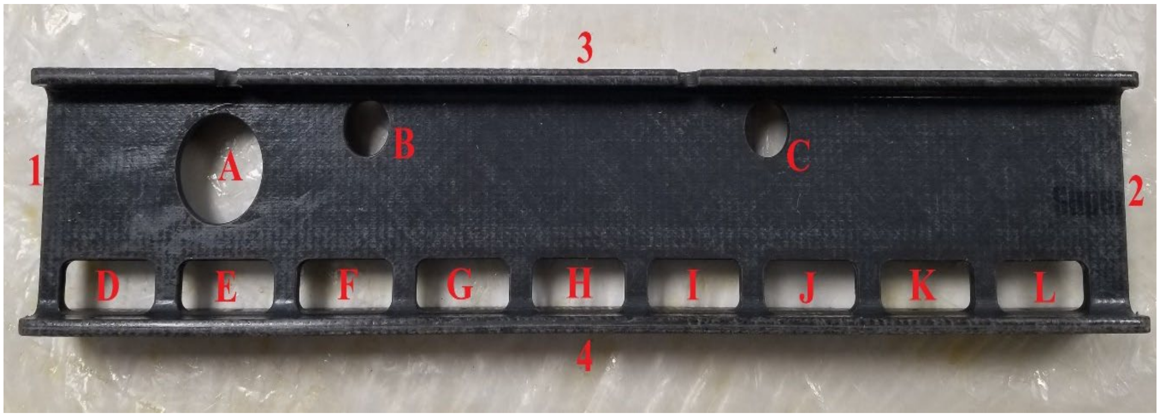


Figure 6.1: Nomenclature of the latch beam for the purpose of quality testing

The experimental measurement data of hole A is shown in Table 6.1. The theoretical measurement value for the diameter of hole A is 43.4 millimeters, which is obtained from the technical design drawing. The input file created for the dataAnalysisProgram.C to obtain the data analysis is shown in Figure 6.2.

Table 6.1: Experimental measurement values of the diameter of A

Measurements of the diameter of A (mm)
43.30
43.29
43.33
43.21
43.16
43.19
43.31
43.21

The left-most column is showing the number of lines in the text file. The first line in the input file is the title of the plot, followed by the axis labels in the next two lines. The experimental measurement values are entered form lines 4 – 11, and the line 12 includes the theoretical measurement value from the technical design drawing. The input file should not have any extra line to prevent the program from generating an inaccurate histogram plot.

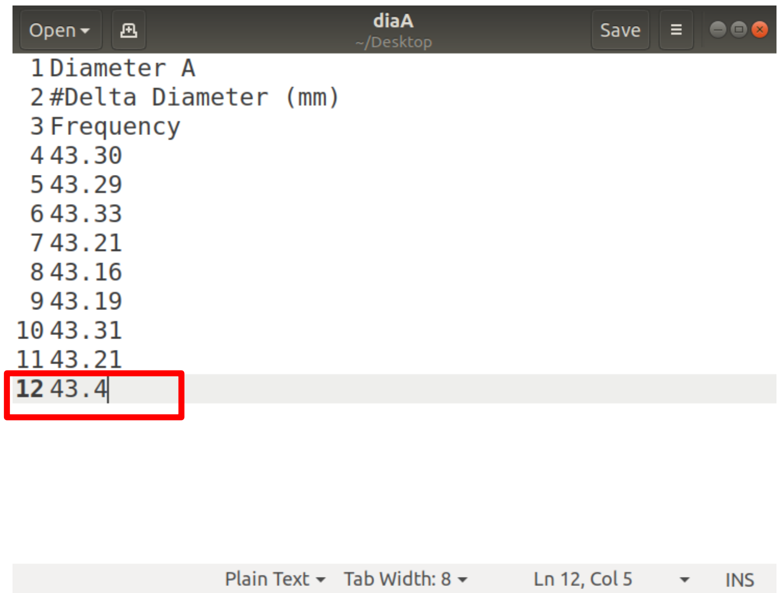


Figure 6.2: Input file of the measurement values of A created for the data analysis through the program (theoretical value from the technical design drawing is highlighted with a red box)

After the program is executed, it informs the user and navigates through the functionality of the program as shown in Figure 6.3. The program is executed in the ROOT data analysis framework (Version 6.18/04) environment using a terminal window on Ubuntu 18.04.4 LTS operating system. After printing the message on the screen, the program waits for the user to enter the number for using the corresponding functionality or enter "quit" to return to the ROOT command prompt.

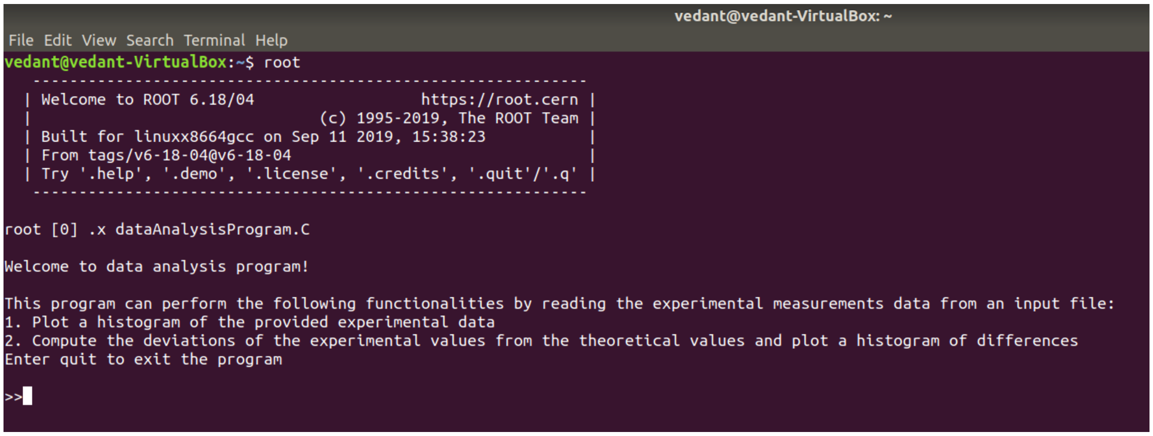


Figure 6.3: Welcome message and functionality disclosure by dataAnalysisProgram.C

If "2" is input by the user, the program prompts the user to enter the name of the input file. As the user enters the file name, the program starts searching for the file in the same directory from where the program is executed. The program will gently inform the user about the unavailability of the file without abruptly crashing the program. Figure shows the output during a failure to open the file. On successfully opening the file, the program outputs the data present in the input file along with the theoretical value segregated from the experimental values, allowing the user an opportunity to corroborate. The program will calculate the differences from the theoretical value, and it will display the deviations calculated using the experimental values and the theoretical value as shown in Figure .

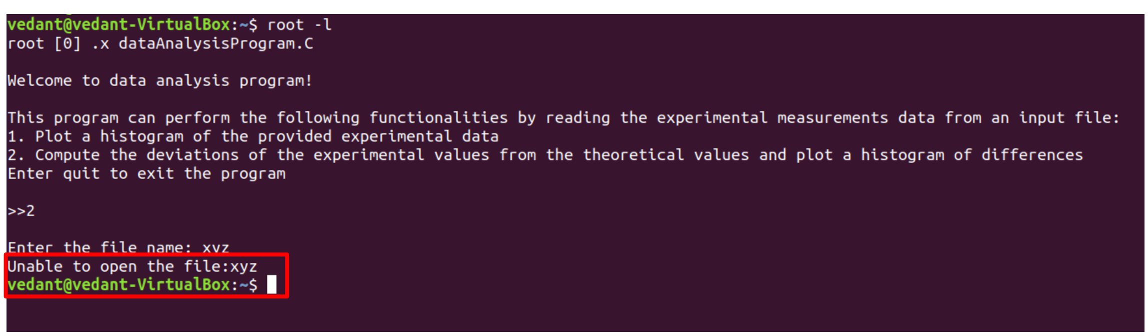


Figure 6.4: Input file not found, informing the user about the issue with the input file

The program will ask the user to enter the number of bins before generating the histogram, which provides the user a flexibility to easily change the bin size until the histogram plot with the required precision is obtained. On providing the input of "7" for the number of bins, the program will generate a histogram plot on a new window as shown in Figure.

```
vedant@vedant-VirtualBox: ~
File  Edit  View  Search  Terminal  Help
Welcome to data analysis program!

This program can perform the following functionalities by reading the experimental measurements data from an input file:
1. Plot a histogram of the provided experimental data
2. Compute the deviations of the experimental values from the theoretical values and plot a histogram of differences
Enter quit to exit the program

>>2

Enter the file name: diaA
File Opened Successfully!

Data:
43.3
43.29
43.33
43.21
43.16
43.19
43.31
43.21
Theoretical measurement value: 43.4

Deviations:
-0.1
-0.11
-0.07
-0.19
-0.24
-0.21
-0.09
-0.19
Total number of data points to plot: 8

Recommended bin size is 3

Enter the number of bins:
```

Figure 6.5: Successfully found the input file and processing the measurements data

The plot also contains a statistical box to facilitate the data analysis for the user. The program also provides a statistical output to the terminal window (as shown in Figure) after generating a histogram plot with a Gaussian fit.

```
 FCN=1.36467e-09 FROM MIGRAD    STATUS=CONVERGED      59 CALLS          60 TOTAL
                     EDM=2.73031e-09    STRATEGY= 1      ERROR MATRIX ACCURATE
  EXT PARAMETER                                   STEP         FIRST
  NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
   1  Constant     5.12039e+00   2.28527e+00   8.84052e-04   3.46848e-05
   2  Mean        -1.39907e-01   2.57543e-02   1.25419e-05  -1.33704e-03
   3  Sigma        6.92146e-02   2.07488e-02   3.96885e-05   2.94474e-04
```

Figure 6.6: Statistical analysis produced by the dataAnalysisProgram.C in the terminal window while conducting a quality test for the diameter of hole A

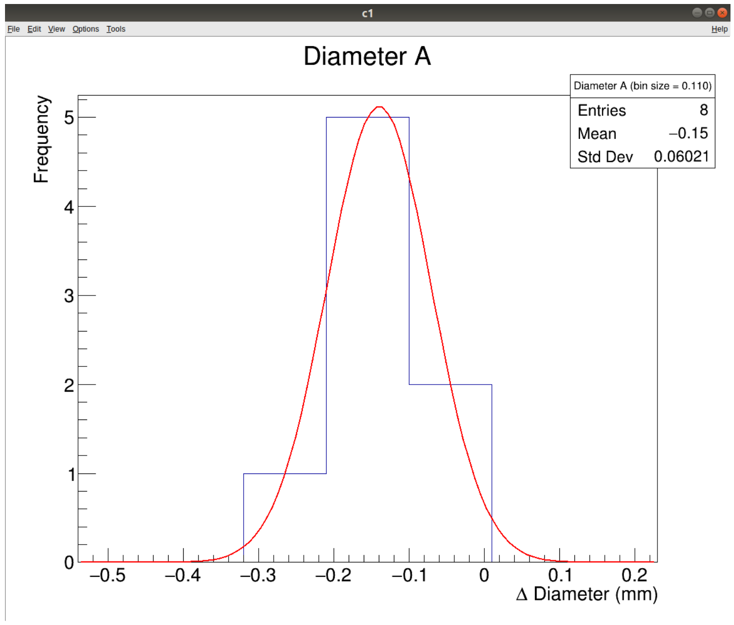


Figure 6.7: The histogram plot generated by the program
for deviations in diameter of hole A

The histogram plot in Figure posits that the range of maximum deviation is 0.1 mm to 0.2 mm for the diameter of hole A. The negative sign in the deviations states the diameter of A is shorter than the value mentioned in the technical design drawing. According to the tolerance mentioned in the technical design drawing, the maximum acceptable deviation can be 0.254 mm. Thus, the diameter of A is in the acceptable tolerance range and qualifies in the quality test.

A similar analysis was conducted on the length of hole D, and the obtained histogram plot is shown in **Error! Reference source not found.**. The histogram plot describes that the differences lie in the range from 0 to 0.1mm, and according to the technical design drawing, the acceptable tolerance range for the length of hole D is 0.254mm. Hence, the length of hole D remains in the acceptable tolerance range and qualifies in the quality test. The ROOT framework was not able to generate an appropriate Gaussian fit because the distribution is not Gaussian. In such cases, the ROOT framework assumes that it has to start fitting the values from the lower bound to the upper bound of the window, which results in the Gaussian fit going out of the histogram plot window.

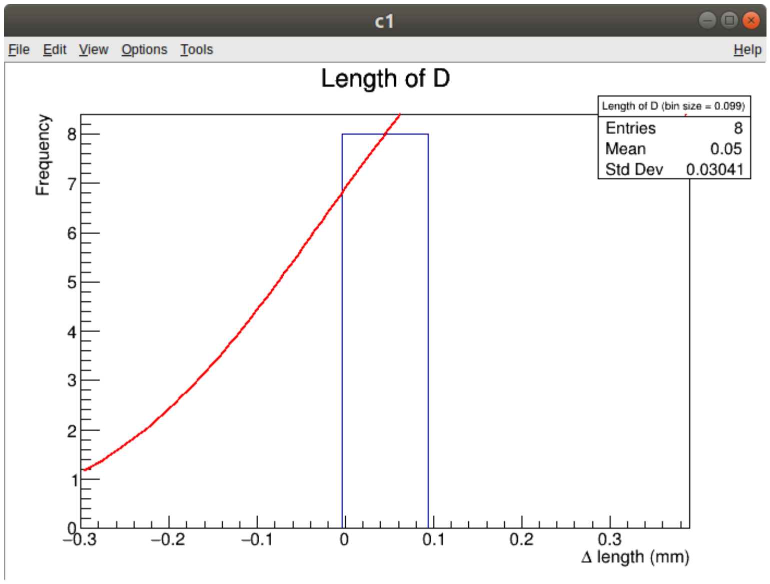


Figure 6.8: The histogram plot generated by the program for data analysis of deviations in the length of hole D

Further continuing the quality analysis on the width of the latch beam, eight independent measurements were conducted from the edge 3 to edge 4. The experimental measurement data were analyzed using the dataAnalysisProgram.C by following a similar procedure, as mentioned earlier in this paper. **Error! Reference source not found.** shows the histogram plot provided by the program. The histogram plot describes that maximum deviations fall in the range of 0.02mm to 0.12mm. According to the technical design drawing, the maximum acceptable tolerance range is 0.254mm. Hence, the deviations in the breadth of the latch beam come in the acceptable tolerance range and qualify in the quality test.

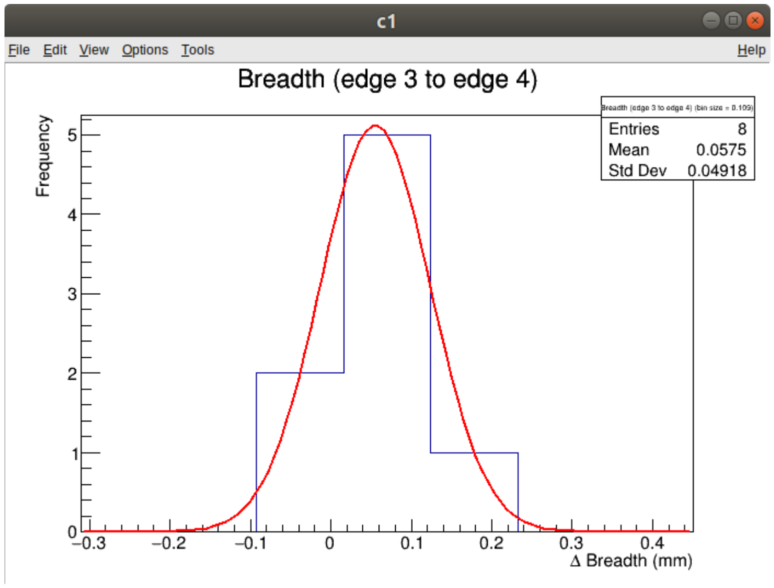


Figure 6.9: The histogram plot for deviations in the breadth of the latch beam

CHAPTER 7

CONCLUSION

The goals of this research are to develop a methodology and provide a software tool for the researchers to aid and expedite the process of quality control testing on the FRP beams for DUNE field cage parts in a consistent and timely fashion. DUNE is an international experiment utilizing LAr TPC technology to investigate the mysteries of neutrinos and their role in the universe. The large detectors required in DUNE require a robust structure free from error and deformities which will contribute to uncertainties in important neutrino physics measurements. As such, the quality of the field cage one of the vital factors to the outcome of the experiment.

The methodology for conducting the measurements on FRP beams serves as a benchmark for the researchers. It allows reducing the risk of failure at an inchoate state of the experiment. The developed methodology described in this paper provides guidelines that will allow new researchers to properly conduct quality tests of the FRP beams. It also gives an idea about the usage of measuring tools and the precisions obtained in the measurements.

The Data Analysis Program developed during this research increases the accuracy and productivity of the researcher and helps them to facilitate the use of statistical analysis features provided by the ROOT data analysis framework. The mechanics of the data analysis program are elaborated in this paper to provide a better insight into the functioning of the software in the ROOT framework environment. The software is also designed for ease of use, especially to researchers who only have an elementary understanding of computer programming. During the development phase, a significant focus was to create a program that is easy to operate and provides an abstraction from the complicated ROOT programming. The software development emphasized minimal user-to-program interaction, which prevents the user from being distracted in the errors produced during the process of the ROOT program.

Finally, the program was tested with various use cases to ensure its functionality and performance in the data analysis process. The procedure of conducting a quality test was replicated during the software testing phase, and the use of dataAnalysisProgram.C (Appendix A) is elaborated in the software testing section of this paper. The content of the input file used to feed the field cage part quality measurement data to the program is also described while explaining the task of data analysis using the developed software. An analysis was performed on the data visualization generated by the program to determine if the part passes the quality control test. In the future, the developed methodology and the software tool will be used to perform quality control testing on the parts before using them to build the DUNE far detector field cage.

APPENDIX A

DATA ANALYSIS PROGRAM.C

```
/**

Developed by Archit Kalpeshkumar Jaiswal

This program will take an inputfile consisting of the values of measurements recorded by the user and plots the histogram

File must contain the theoretical value of parameter from the technical design drawing or report
Title, X axis and Y axis labels are optional but if present then the order should be Graph Title then X axis label followed by Y axis label
**/


#include <iostream>
#include <fstream>
#include <sstream>
#include <cstdlib>

// importing ROOT data analysis libraries
#include <TMath.h>
#include <TCanvas.h>
#include <TStyle.h>
#include <TSpectrum.h>
#include <string.h>



void dataAnalysisProgram(){

  int        numberOfBins;          // used to determine the bin size in histogram and holds the value from user input
  double     maxBound, minBound;    // variables for histogram constructor to define range of the plot

  std::string temp          = "";   // holds the initial selection input from the user
  std::string middleString  = "";   // holds the axis titles
  std::string titleWithBin  = "";   // holds the title and bin size
  std::string reader        = "";   // holds the input directly from data file as a string
  std::string title         = "";   // stores the title seperately to put on the plots/graphs
  std::string fileName      = "";   // holds the file name from user input

  std::stringstream titleStream;    // merges the bin size with the subtitle and have bin size printed till 3 decimal places
  std::vector<std::string> titleStr;  // holds all the titles (i.e., Plot title and both axis titles)

  std::vector<double> measurements;  // holds all the numeric form of inputs (dataType: double)
  std::vector<double> differences;   // this will hold all the numeric values of differences, this will have one entry less that the measurements because last is the expected value


  std::cout << "\nWelcome to data analysis program!\n" << endl;
  std::cout << "This program can perform the following functionalities by reading the experimental measurements data from an input file:" << endl;
  std::cout << "1. Plot a histogram of the provided experimental data" << endl;
```

```
  std::cout << "2. Compute the deviations of the experimental values from the theoretical values and plot a histogram of differences" << endl;
  std::cout << "Enter quit to exit the program" << endl;
  std::cout << "\n>>";
  std::getline( std::cin, temp );
  if (strcasecmp(temp.c_str(), "quit") == 0)
  {
    return;
  }
  std::cout<< "\nEnter the file name: "; // prompting to the user for entering the file name
  std::getline(std::cin, fileName); // reading the input from user
  ifstream inputFile(fileName, ios::in); // creates a file object, sets it as input and also instructs that if the file does not exist then do not create a new file
  if (inputFile.is_open()) // verifying the characteristics of file
  {
    std::cout << "File Opened Successfully!" << std::endl;
  }
  else
  {
    std::cout << "Unable to open the file:" << fileName << std::endl;
    exit(1); // stop the execution if the file is not openable
  }
  while(inputFile.peek() && !inputFile.eof())
  {
    std::getline(inputFile, reader);
    try
    {
      double tempVal = std::stod(reader);
      measurements.push_back(tempVal);
    }
    catch (const std::invalid_argument& )
    {
      title = reader;
      titleStr.push_back(title);
    }
  }
  if (std::stoi(temp) == 1)
  {
    // last entry in the measurements array will be a theoretical value of that perticular parameter
    // Last value of measurements array will not be used to determince the range of histogram
    maxBound = (ceil (*max_element(measurements.begin(), measurements.end() - 1))) + 1;
    minBound = (floor(*min_element(measurements.begin(), measurements.end() - 1))) - 1;
```

```
    // ask for the number of bins from user
    std::cout << "\nRecommended bin size is " << ceil(sqrt(measurements.size()))<< endl;
    std::cout << "\nEnter the number of bins: ";
    cin >> numberOfBins;

    // creating a string to label the title, x-axis and y-axis of the histogram
    if (titleStr.size() == 3)
    {
      middleString = titleStr[0] + "; " + titleStr[1] + "; " + titleStr[2];
    }
    else
    {
      middleString = titleStr[0] + "; ; ";
    }

    // calculating the bin size
    float binSize = ( (maxBound - minBound)/ numberOfBins );

    titleStream << std::fixed << titleStr[0] << " (bin size = " << std:: setprecision(3) << binSize << ")";
    titleWithBin = titleStream.str();
    TH1F *myHistogram = new TH1F (titleWithBin.c_str(), middleString.c_str() , numberOfBins, minBound, maxBound);



    std::cout << "\nTotal number of data points: "<< (measurements.size() - 1) << endl;
    std::cout << "Data: " << endl;

    // filling the data points in histogram
    for (int j = 0; j < measurements.size() - 1 ; j++) // minus 1 because last value is the theoretical value of a measurement
    {
      std::cout << measurements[j] << endl;
      myHistogram-> Fill(measurements[j]);
    }

    // calculating the parameters for a curve
    double binMax    = myHistogram-> GetMaximumBin();
    double x         = myHistogram-> GetXaxis()-> GetBinCenter(binMax);
    int maxBinVal    = myHistogram-> GetBinContent(binMax);

    // apply a chi-square curve on the histogram plot
    TF1 *curve = new TF1("curve", "gaus", minBound, maxBound);

    myHistogram -> Fit (curve);
    myHistogram -> Draw("hist");
    curve       -> Draw("SAME");

  }
  else if( std::stoi(temp) == 2)
  {

    std::cout << "\nData: " << endl;

```

```
for (int j = 0; j < measurements.size() - 1 ; j++) // minus 1 because last value is the theoretical value of a measurement
{
  std::cout << measurements[j] << endl;
}

// calculating the deviation from theoretical value
double actualValue = measurements.back();
std::cout << "Theoretical measurement value: "<< actualValue << endl ;

std::cout << "\nDeviations: "<< endl;
for (int i = 0; i < measurements.size() - 1; i++)
{
  differences.push_back(measurements[i] - actualValue);
  std::cout << differences[i] << endl;
}
std::cout << "Total number of data points to plot: " << differences.size() << endl;

maxBound = (*max_element(differences.begin(), differences.end())) + 0.3;
minBound = (*min_element(differences.begin(), differences.end())) - 0.3;

// ask for the number of bins from user
std::cout << "\nRecommended bin size is " << ceil(sqrt(differences.size()))<< endl;
std::cout << "\nEnter the number of bins: ";
cin >> numberOfBins;

// creating a string to label the title, x-axis and y-axis of the histogram
if (titleStr.size() == 3)
{
  middleString = titleStr[0] + "; " + titleStr[1] + "; " + titleStr[2];
}
else
{
  middleString = titleStr[0] + "; ; ";
}

// calculating the bin size
float binSize = ( (maxBound - minBound)/ numberOfBins );

titleStream << std::fixed << titleStr[0] << " (bin size = " << std:: setprecision(3) << binSize << ")";
titleWithBin = titleStream.str();
TH1F *myHistogram = new TH1F (titleWithBin.c_str(), middleString.c_str() , numberOfBins, minBound, maxBound);

// filling the data points in histogram
for (int i = 0; i < measurements.size() - 1; i++)
{
  myHistogram-> Fill(differences[i]);
}

double binMax   = myHistogram-> GetMaximumBin();                 // gets the bin number with maximum height
double x        = myHistogram-> GetXaxis()-> GetBinCenter(binMax); // gets the corrosponding x value for the bin with max height
```

```
    int maxBinVal   = myHistogram-> GetBinContent(binMax);            // gets the corrosponding y value of the tallest bin

    TF1 *curve = new TF1("curve", "gaus", minBound, maxBound);

    myHistogram-> Fit (curve);
    myHistogram-> Draw("hist");
    curve      -> Draw("SAME");

  }

  return;
}

```

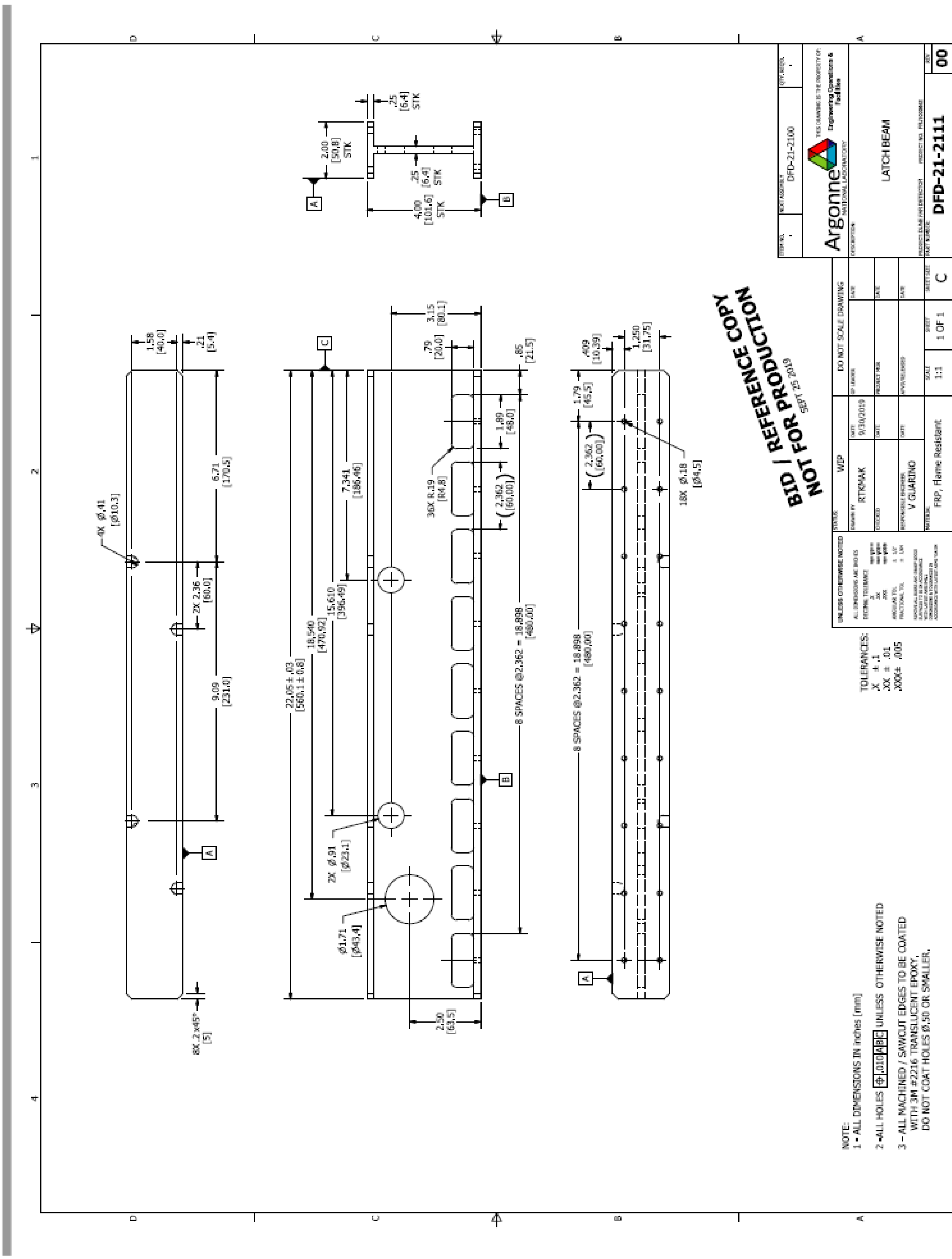
APPENDIX B

INPUT FILE FORMAT

Title of the Plot
X-axis title
Y-axis title
Experimental value 1
Experimental value 2
Experimental value 3
…
Theoretical value

APPENDIX C

TECHNICAL DESIGN DRAWING


BID / REFERENCE COPY
NOT FOR PRODUCTION
SEPT 25 2019
TOLERANCES:
.X ± .1
.XX ± .01
.XXX± .005
NOTE:
1 - ALL DIMENSIONS IN inches [mm]
2 -ALL HOLES ⌖ .010 A B C UNLESS OTHERWISE NOTED
3 - ALL MACHINED / SAWCUT EDGES TO BE COATED WITH 3M #2216 TRANSLUCENT EPOXY, DO NOT COAT HOLES Ø.50 OR SMALLER.
DFD-21-2100
Argonne NATIONAL LABORATORY
LATCH BEAM
DFD-21-2111
00
WIP
RTKMAK
9/30/2019
V GUARINO
FRP, Flame Resistant
DO NOT SCALE DRAWING
1:1
1 OF 1
C

REFERENCES

[1] Deep Underground Neutrino Experiment Website <https://www.dunescience.org/>

[2] ASTM International. A1087/A1087M-16 Standard Practice for Using Hand Calipers to Measure the Width of Steel Sheet. West Conshohocken, PA, 2016. Web. 14 July 2020. <https://doi-org.ezproxy.uta.edu/10.1520/A1087_A1087M-16>

[2] Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume III: DUNE Far Detector Technical Coordination, 2020.

[3] Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume IV: DUNE Far Detector Single-phase Technology, 25 March 2020.

[4] Deep Underground Neutrino Experiment (DUNE), Far Detector Interim Design Report, Volume I: Physics, Technology & Strategies, 2018.

[5] Verma, Harish C. Concepts of Physics. New Delhi: Bharati Bhawan, 2008, 23

[6] Antcheva, I., et al. "ROOT — A C++ Framework for Petabyte Data Storage, Statistical Analysis and Visualization." Computer Physics Communications, vol. 180, no. 12, 2009;2015;, pp. 2499-2512.

[7] Horton, Ivor. Ivor Horton's Beginning Visual C++ 2012. Indianapolis, IN: Wiley, 2012. Print.

[8] ROOT Reference Guide V6.22 <https://root.cern/doc/master/index.html>

[9] V.Ryan, A Digital Calipers <https://technologystudent.com/equip1/vernier1.htm>.

BIOGRAPHICAL INFORMATION

Archit Kalpeshkumar Jaiswal is an undergraduate student pursuing a dual Honors Bachelor of Science in computer engineering and physics. He is aiming for a Ph.D. in mechatronics and interested in devoting this career to developing advance defensive equipment and military technologies. His research interests are focused on robotics, embedded control systems, and quantum computing. He started participating in High Energy Physics research in his 2nd year as an undergraduate at the University of Texas at Arlington. He also excelled in several ABET-accredited courses that require rigorous critical thinking ability and group collaboration efforts to complete the project deliverables. Due to his contribution to physics research, he also had an opportunity to give a presentation at the American Physical Society (APS) conference held at Texas Tech University in Lubbock, Texas. After completing this McNair research, he had another opportunity to present at the 23rd Annual MKN McNair Heartland Research Conference. He was also appointed for the summer internship as a Research and Development (R&D) Intern at a multinational national technology consulting company, Infosys. During the internship, he contributed to developing a technique that can allow an Industrial Internet of Thing (IIoT) device to detect targets and perform tasks in co-ordination with other smart devices. The company further offered him an extended appointment and requested to work towards developing a similar solution that can run on a quantum computer. His passion for learning will enable him to accomplish the goal of making this planet a better and safe place for humanity.