Jefferson National Laboratory Charged Pion Polarizability Experiment: Monte Carlo Simulation Using GEANT4 and Construction of Multi-Wire Proportional Chambers

A Senior Honors Thesis Presented

By

Jordan Scott Kornfeld

University of Massachusetts

Amherst, Massachusetts

May 2017

Professor Rory Miskimen, Physics

Professor David Kawall, Physics

Abstract

The Theory of Quantum Chromodynamics attempts to predict the interactions fundamental particles, quarks and gluons, through the strong force. There exists several discrepancies between QCD's theoretical predictions and experimental observations, one of which is the Charged Pion Polarizability. The Charged Pion Polarizability Experiment at Jefferson National Laboratory aims to re-measure this quantity and verify the agreement between theory and observation. The Medium-Energy Nuclear Physics Laboratory in the Leaderle Graduate Research Tower at UMass Amherst contributes to the completion of this project with the design and construction of Multi-Wire Proportional Chambers. The objective of this paper aims to provide the reader with a guide as to how design is optimized within the GEANT4 program and give details to productive MWPC wire stringing construction.

Table of Contents

1. Introduction....................4

2. Background....................5

i. The Standard Model....................5

ii. Pions and Polarizability....................7

iii. Measuring the CPP....................8

3. CPP Simulation in GEANT4....................10

i. Introduction....................10

ii. How GEANT4 Works....................10

iii. Linux Basics....................11

iv. What is the CPPSIM....................13

v. Contents of 'CPPSIM' and 'Events'....................14

vi. How to Run CPPSIM/How it Works....................15

vii. Adjusting Simulation Parameters....................19

viii. Analyzing Results....................22

ix. Useful Commands....................25

4. MWPC Construction....................26

i. Introduction....................26

ii. General Stringing Procedure....................28

iii. Details....................29

iv. Refined Stringing Process....................43

v. Stringing the Central Region....................46

5. Conclusions....................53

i. Synopsis of Work Completed....................53

ii. Current Status and Remaining Work....................54

6. References....................54

1. Introduction

The objective of physics is to come up with theoretical models that can predict experimental results. A famous example of this is Newton's Theory of Gravity. This theory can predict the motion of objects falling to earth, their speed, position, how long it will take to fall, where it will hit the ground, things of this nature. Even more importantly, this theory can predict these measurements extremely accurately, at least at relatively small velocities and weak gravitational fields. Quantum Chromodyanmics, also known as QCD, is a theory just like gravity except its objective is to predict the interaction of fundamental particles called quarks and gluons through the strong force. However, unlike gravity, the problem with QCD is that in some instances it is not accurate and its predictions do not match what is experimentally observed. This is a major flaw; after all, predicting experimental results is the entire purpose of a theory. One specific instance where the theory fails is in measuring the charged pion polarizability, also called the CPP. Re-measuring this quantity to find agreement between theory and observation is the objective of the Charged Pion Polarizability Experiment at Jefferson National Laboratory. The CPP has been measured several times previously with differing results seen in Figure 1, some showing agreement and some not. The CPP is one of the last major discrepancies of QCD and it is very important that it be resolved to verify the accuracy of the theory. Hopefully, new data will accomplish just this. The Medium-Energy Nuclear Physics Lab in Leaderle Graduate Research Tower focuses on simulating the CPP Experiment to optimize various parameters of Multi-Wire Proportional Chambers necessary for the measurement as well as the construction of these devices. [8]

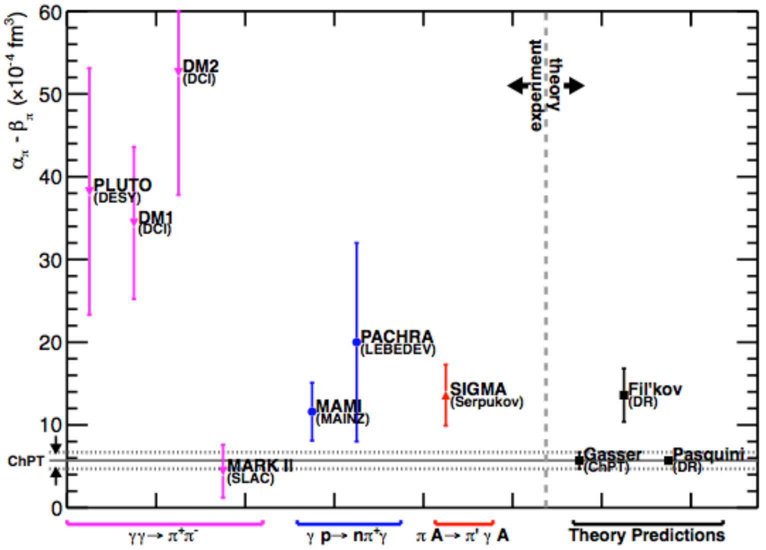


Figure 1. Different experimental results of the CPP, mathematically denoted by $\alpha_\pi - \beta_\pi$, compared to theoretical predictions and grouped by experiment type. [8]

2. Background

i. The Standard Model

The current Standard Model of Particle Physics describes interactions between three of the four fundamental forces: Electromagnetic, Weak, and Strong, with the exclusion of Gravity. In order of increasing strength these forces are ranked: Gravity, Weak, Electromagnetic, Strong. In addition, the Standard Model also categorizes and organizes the most fundamental particles: Quarks, Leptons, and Bosons, seen in Figure 2.

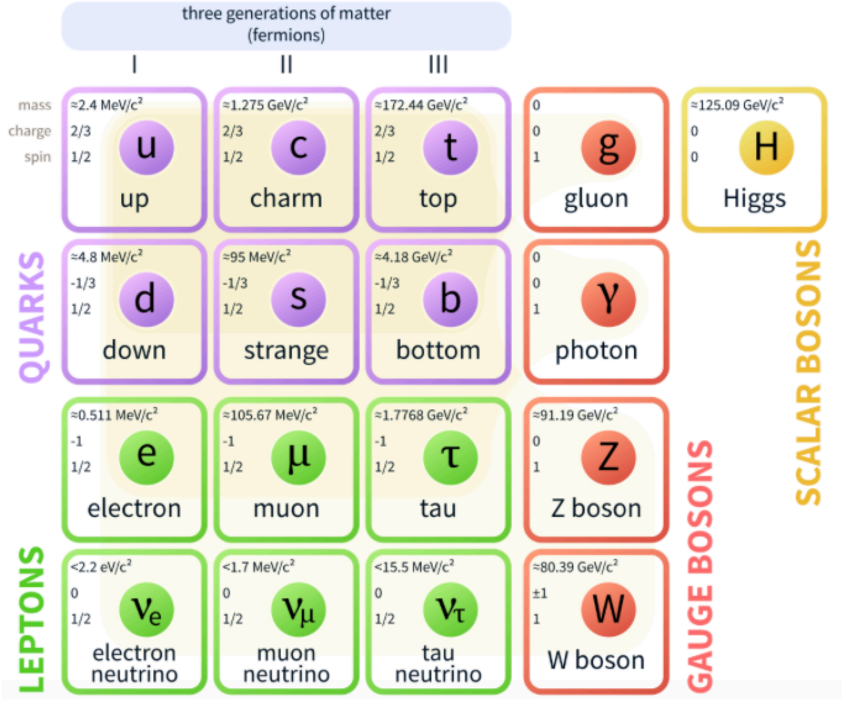


Figure 2. The current Standard Model of Particle Physics showing the generations of quarks and leptons and the boson force carriers. [5]

Bosons are known as force carriers and their exchange between quarks and leptons is responsible for producing the three fundamental forces described by the standard model. The gluon mediates the strong force, as does the photon to the electromagnetic force, and the W and Z bosons mediate the weak. It is theorized that the graviton mediates gravity but is has not been discovered yet. The idea of a particle being responsible for generating the force between other particles is a fairly abstract concept, much more so than the simple role played by quarks and leptons. These particles and various combinations of them generate all of the observed matter in the universe. Quarks and leptons are further categorized into pairs by generation as seen in Figure 2. With each increase in generation particle mass increases and stability decreases. Therefore, all stable matter in the universe is made from quarks and leptons of the first generation. In addition, each quark and lepton has their own corresponding antiparticle. An antiparticle has all of the same properties as the original particle but with opposite charge. For instance, the electron antiparticle is the positron and is the same in every way except that it has a positive charge. [2]

From here, composite particles made up of quarks are further categorized. Particles consisting of two quarks are mesons and particles made from three quarks are baryons. For example, the proton is a baryon made from two up quarks and a down quark. Likewise, a neutron is made from two down quarks and an up quark. Together with electrons these particles make atoms. The reason why composite particles are restricted to combinations of two and three quarks is because of an additional property quarks have called color. Color is similar to charge in that it is a unit a quark possesses. However, where charge can be plus or minus and can have various magnitudes described by any real number, color is strictly just a singular unit and a quark cannot have more or less than one unit. There are three colors: blue, green, and red, and any quark can have any one of these colors. If the quark happens to be an antiquark then the colors are also anti and it could be assigned: anti-blue, anti-green, or anti-red. The rule with composite particles is that they must be color neutral. Color neutrality can be achieved in two ways: by combining color with anti-color or combining one of each of the three colors together. For example, color neutrality is achieved from the combination of red and anti-red. It is also achieved from the combination red, blue, green. In fact, anti-red, anti-blue, anti-green is also neutral and would produce some type of antiparticle. This explains why composite particles can only be made up of either two or three quarks. [6]

ii. Pions and Polarizability

There are many different combinations of quarks and therefore many different mesons exist: the pion, the eta meson, the kaon, and the D meson are just a few. For the purposes of this paper and the CPP Experiment the pion is the most important of all mesons and its properties must be understood. Like any meson, the pion is composed of two quarks but there are actually three different types of pions made up of different combinations of quarks. There is the neutral pion and there are two charged pions. The neutral pion can be made from either an up and an anti-up quark or a down and an anti-down quark. In either case, the overall charge is neutral. On the other hand, the two charged pions are composed of either an up and an anti-down quark or a down and an anti-up quark. These are appropriately named the pi-plus and the pi-minus due to their respective charges. [6]

The CPP Experiment seeks to measure the electromagnetic polarizability of the charged pi-plus and pi-minus. The polarizability of these particles can be thought of in the classical sense as the tendency to have their charges displaced by an external electric field. Essentially, the CPP measures how easily the charged pion structure deforms in the presence of an electric field. If the composite quarks are bound together at some distance in equilibrium where the attractive strong force and repulsive electromagnetic force are equal in magnitude, the CPP measures how this distance changes in the presence of an electric field. Polarization is also effected by the presence of a magnetic field called the magnetic polarizability. The electric and magnetic polarizabilities are often denoted by α and β respectively, therefore the electromagnetic polarizability can be given as $\alpha \pm \beta$. Measuring the CPP is important because polarizabilities are fundamental properties of composite systems that provide information about the excited states of the systems. Specifically, in the case of mesons, polarizabilities provide an important test point for effective field theories and dispersion and lattice calculations. Among the meson polarizabilities, the CPP ranks of paramount importance because it tests fundamental symmetries at leading order. [8]

iii. Measuring the CPP

Jefferson National Laboratory plans to generate a beam of pions among other particles through the Primakoff effect, seen in Figure 3, using a 6 GeV photon beam and lead target. Basically, in this process, when a high energy photon comes in close enough proximity to a lead nucleus the two interact through the weak force to yield pions and muons. This produces the particle beam, which will then pass through a series of detectors called GlueX, also in Figure 3, to measure the CPP. The GlueX detector has many components with the main ones being the Forward Drift Chamber, FDC, the Time of Flight, TOF, and the Forward Calorimeter, Fcal. Additionally, Multi-Wire Proportional Chambers, MWPCs, will be implemented but are not pictured because technically they are not part of GlueX. Each one of these detectors has a special function to measure information about the particles passing through GlueX that will ultimately produce a measurement of the CPP. The purpose of the Forward Drift Chamber is to measure the momenta of the incoming particles. Next is the Time of Flight, its function is exactly what the name suggests, to measure the time taken for the particles to reach the detector. The Forward Calorimeter measures the energies of the particles as they pass the detector. Finally, there are the MWPCs, which measure the positions of particles. The CPP Experiment aims to measure the polarizability of pions, not muons. As such, the two need to be differentiated between during data analysis. Measurements from the MWPCs provide a distinction between pions and muons reveling muon data to be discarded. [1]

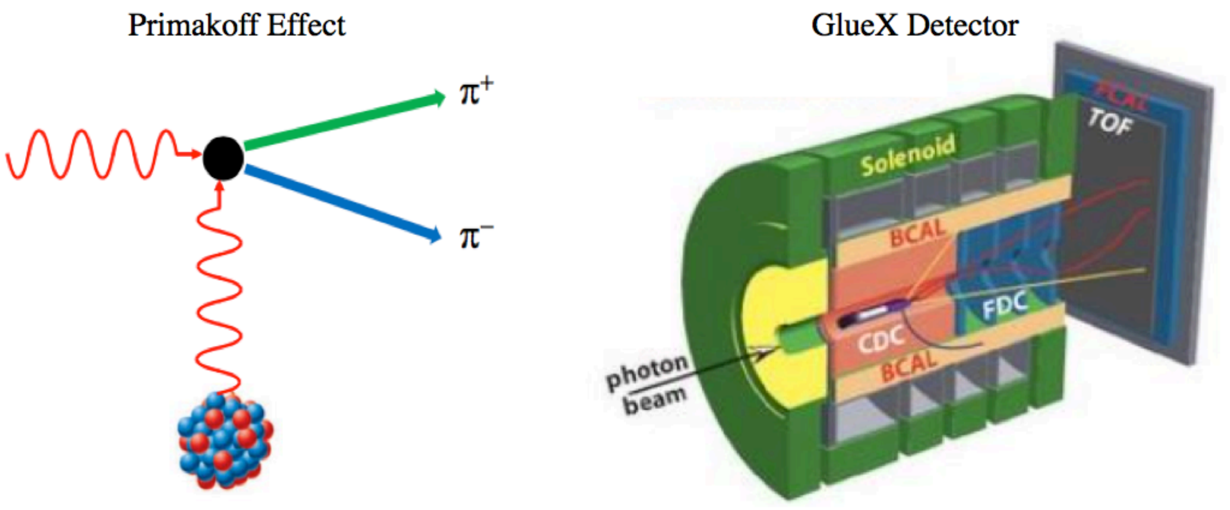


Figure 3. An example of the Primakoff Effect with a photon and lead nuclei interacting weakly to produce charged pions as well as GlueX and its various components. [4] [3]

3. CPP Simulation in GEANT4

i. Introduction

Modern physics experiments have grown to be enormously complex with billion dollar research facilities and collaboration between hundreds of scientists. As experimentation has evolved, the demand for sophisticated simulation technology has grown with it. This led to the development of the Geometry and Tracking, or GEANT, simulation tool using Monte Carlo methods by CERN in 1974. Since then, the program has been modified and upgraded continually, resulting in a series of four GEANT programs with GEANT4 being the most recent version. GEANT4 offers applications in various fields of physics such as nuclear, medical, and space studies and is used in experiments run by the world’s biggest research institutions such as CERN, Fermilab, SLAC, and JLab. [9]

ii. How GEANT4 Works

GEANT4 has a very extensive hierarchical structure that allows for virtually any experimental configuration and implements all known physics. The structure of GEANT4 can be seen in Figure 4. However, for the purposes of optimizing MWPC geometries, knowing the structure of GEANT4 is not necessary. The CPP Experiment Simulation, CPPSIM, has already been constructed within GEANT4 and is open source. GEANT4 and the CPPSIM have already been installed and downloaded onto the Linux machine within the Medium-Energy Nuclear Physics Lab. Therefore, all that is needed is an understanding of how the simulation is run, how geometries are edited, and how results are analyzed, along with some basic Linux instruction.

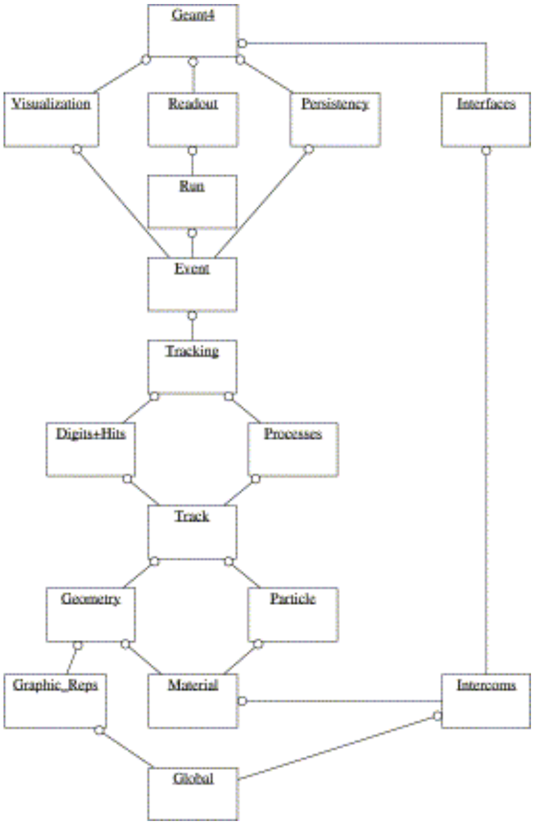
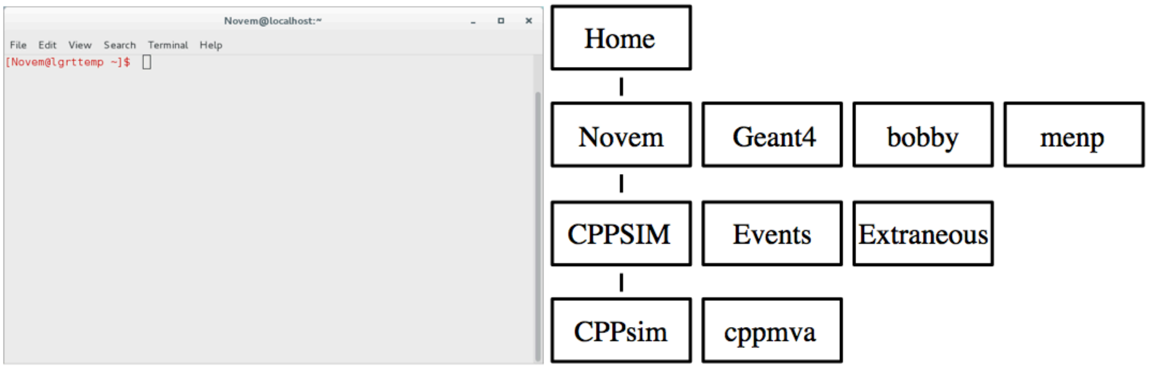


Figure 4. The hierarchical structure of GEANT4. [9]

iii. Linux Basics

Linux is just another type of computer operating system like Mac or Windows, only it is much more simplistic. Nearly everything involved in running CPPSIM will be done in the terminal seen in Figure 5. Open the terminal by right clicking on the desktop and selecting ‘Open in Terminal'. Note, there can be multiple terminals open at once allowing the computer to do multiple tasks at the same time. The terminal provides the interface for the user to navigate within the computer. By imputing commands into the terminal the user can move in and out of different directories and manipulate files. Just like GEANT4, Linux runs using a hierarchical structure. At the top of this structure is the 'home' directory. This directory contains all of the users on the computer: 'bobby', 'Geant4', 'menp', and 'Novem'. Contrary to expectation, the CPPSIM is actually located within the 'Novem' user with the password '10physics'. Within 'Novem' there are several dozen directories and files but really the only relevant ones are the directories 'CPPSIM' and 'Events'. This structure continues down to further levels such as 'CPPsim' and 'cppvma' where directories still continue within these.



```
[Novem@lgrttemp ~]$
```

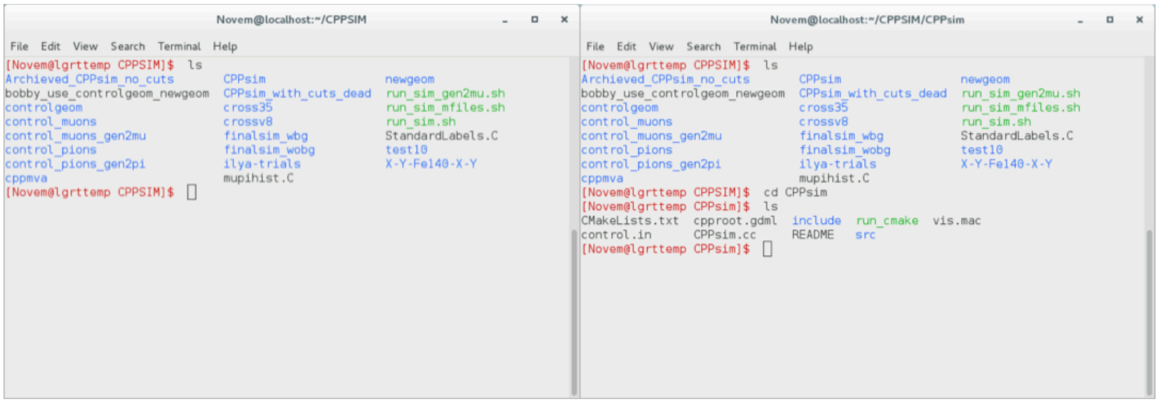
Figure 5. The Linux terminal and structure

Navigation within the terminal requires the use of a few basic command as well as knowledge about several types of objects: directories, files, and executable. Figure 6 shows the contents of the 'CPPSIM' directory.

Objects in blue are directories, which can be accessed with the command 'cd name_of_directory'. Here, 'cd' stands for change directory. There are several other usages of the 'cd' command such as, 'cd ..', which moves the user one directory higher in the hierarchy. Additionally, entering 'cd ~' brings the user from any directory to the 'Novem' directory. If the user knows the directory path they may enter that as well, ‘cd path/to/directory’. For example, in the ‘Novem’ directory, entering ‘cd CPPSIM/CPPsim’ brings the user into the ‘CPPsim’ directory. To see the contents of a directory enter ‘ls’, which stands for list.

Black objects are files and can be viewed and edited with the command ‘gedit name_of_file’.

Objects in green are executables and can also be viewed and edited with ‘gedit name_of_executable’. In addition, they can be executed with the command ‘./name_of_excecutable’. If the user wishes to execute from a directory separate from where the executable is located they may do so by entering the path to the executable. For example, ‘run_sim.sh’ can be executed within ‘CPPsim’ by entering ‘../run_sim.sh’.


Novem@localhost:~/CPPSIM
File Edit View Search Terminal Help
Novem@localhost:~/CPPSIM/CPPsim
File Edit View Search Terminal Help

```
[Novem@lgrttemp CPPSIM]$  ls
Archieved_CPPsim_no_cuts       CPPsim                   newgeom
bobby_use_controlgeom_newgeom  CPPsim_with_cuts_dead    run_sim_gen2mu.sh
controlgeom                    cross35                  run_sim_mfiles.sh
control_muons                  crossv8                  run_sim.sh
control_muons_gen2mu           finalsim_wbg             StandardLabels.C
control_pions                  finalsim_wobg            test10
control_pions_gen2pi           ilya-trials              X-Y-Fe140-X-Y
cppmva                         mupihist.C
[Novem@lgrttemp CPPSIM]$
```

```
[Novem@lgrttemp CPPSIM]$  ls
Archieved_CPPsim_no_cuts       CPPsim                   newgeom
bobby_use_controlgeom_newgeom  CPPsim_with_cuts_dead    run_sim_gen2mu.sh
controlgeom                    cross35                  run_sim_mfiles.sh
control_muons                  crossv8                  run_sim.sh
control_muons_gen2mu           finalsim_wbg             StandardLabels.C
control_pions                  finalsim_wobg            test10
control_pions_gen2pi           ilya-trials              X-Y-Fe140-X-Y
cppmva                         mupihist.C
[Novem@lgrttemp CPPSIM]$  cd CPPsim
[Novem@lgrttemp CPPsim]$  ls
CMakeLists.txt  cpproot.gdml  include  run_cmake  vis.mac
control.in      CPPsim.cc     README   src
[Novem@lgrttemp CPPsim]$
```

Figure 6. The contents of the ‘CPPSIM’ directory as well as an example of changing into the ‘CPPsim’ directory from ‘CPPSIM’ and listing its contents.

iv. What is the CPPSIM

CPPSIM is a monte carlo simulation of the exact CPP Experiment to be performed at Jlab. The simulation contains all the same equipment and conditions as the real experiment. A beam of particles is generated, the particles pass through GlueX and its various detectors, they pass through the MWPCs, and data is collected. Once all the particles pass through the simulation it outputs the results. The benefit of the simulation is that, unlike the real experiment, it can be run over and over again and its parameters are adjustable. These properties allow the user to optimize the parameters by repeatedly running the simulation to yield the best results.

v. Contents of ‘CPPSIM’ and ‘Events’

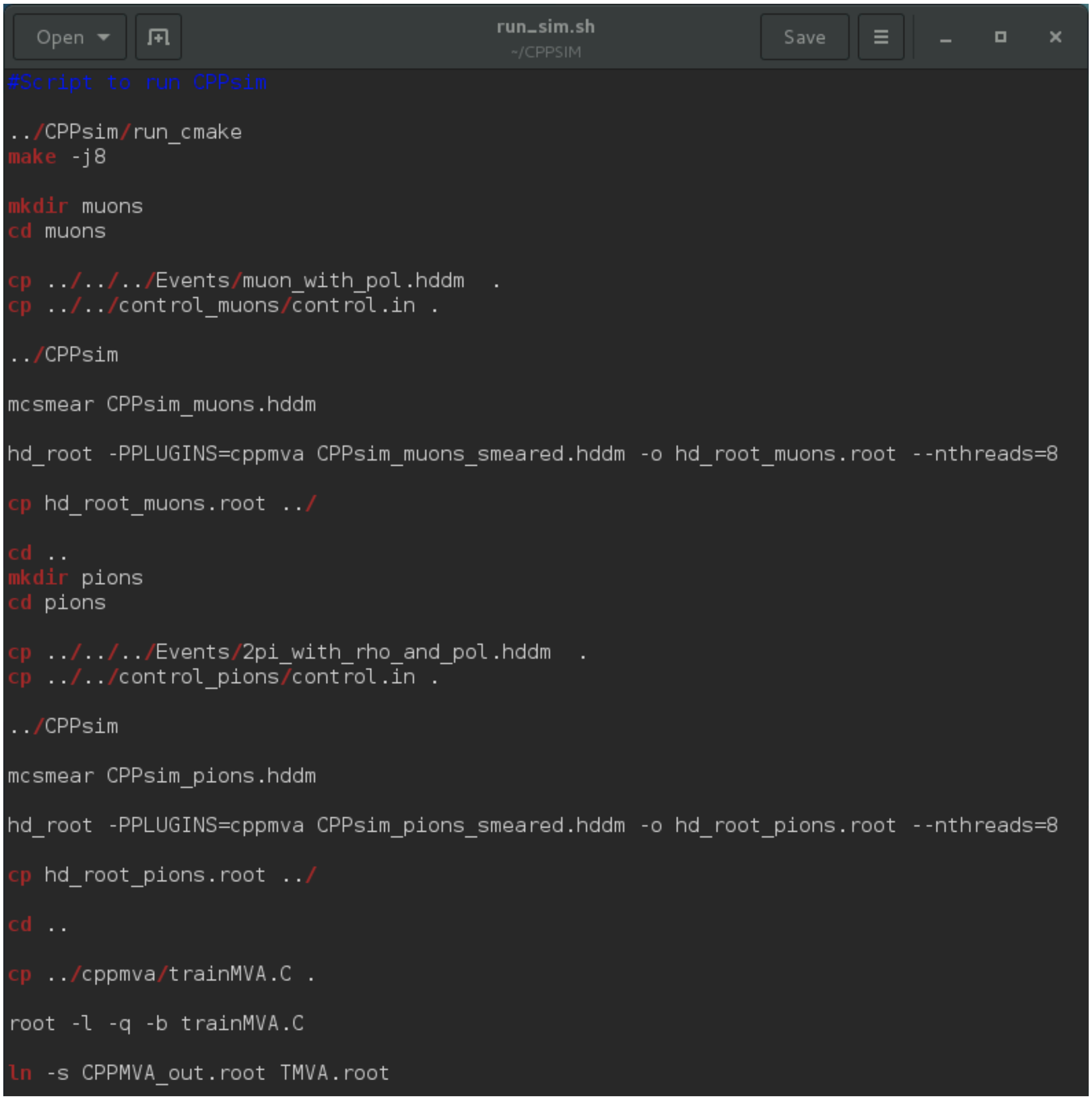
As mentioned previously the two main directories necessary to run CPPSIM are ‘CPPSIM’ and ‘Events’. There is one other hidden directory that is important but its role is small and it will be introduced later. Knowing the contents of ‘CPPSIM’ and ‘Events’ is crucial to understand how the simulation runs.

The contents of ‘Events’ are simple; it just holds event files. Event files are data files and contain the information about the particles that will be sent through the simulation. Such information could be, the number of particles, the energies and momenta of the particles, and the type of particles. Since pions and muons are the primary particles generated during the actual CPP Experiment, there are two types of event files: muon files and pion files. Muon files can vary widely between one another and likewise with pion files. Although their particle content may be the same, they can still have different energies and momenta, number of particles, angular distributions, and other qualities such as added background. Professor Miskimen and members of Jlab continually update these details to make the files as realistic possible. Event files are incredibly complex and large. Do not attempt to open them; they are intended purely for the simulation.

‘CPPSIM’ has more content and is slightly more complicated as seen in Figure 6. There are five main objects within ‘CPPSIM’: ‘CPPsim’, ‘cppmva’, build directories, control directories, and run simulation scripts. ‘CPPsim’ contains the information about the simulation itself and defines the experimental set up. The files within ‘CPPsim’ can be edited to adjust various parameters. Next, ‘cppmva’ contains the information necessary to run trainMVA, a multi-vitiate analysis, on the data to produce meaningful results. The build directories are where the actual simulation runs. Each time the simulation is run it is in a new build directory. The build directory is typically named according to the parameter that particular run is testing. Examples seen in Figure 6 are ‘cross35’, ‘newgeom’, and ‘test10’. Build directories use a lot of disk space because they contain the data of an entire simulation of potentially millions of particles. Therefore, build directories are often deleted or moved to an external disk once the results are produced and they are no longer needed. Additionally, there are control directories, which contain control files. Control files specify the initial conditions going into the simulation. These conditions are: the name of the event file, the number of particles, the name of the output file, and the number of cores in the computer. More about control directories and files will be explained later. Finally, there are run simulation scripts. Running the CPPSIM requires the input of a series of commands into the terminal. A command is entered, which the computer then performs, and is followed by the next command until the simulation is finished. Depending on the size of the event files the simulation can take hours. It is extremely time consuming to sit and wait for the computer to complete each step just to enter the next command and wait again. Often, the user already knows the order of commands to be entered into the terminal. A run simulation script is a script of all these commands in the correct order. Instead of entering a command and waiting for the computer to finish to enter the next, the script has all the commands in order waiting on deck to be entered automatically as soon as the computer finishes its current task. This way, the user just executes the script and the entire simulation runs.

vi. How to run CPPSIM/How it works

Run simulation scripts are executed within the build directory, where the simulation is processed. Simulation scripts vary depending on the initial conditions being used but they do just slightly. Typically, the script can be edited from simulation to simulation to accommodate the changing conditions. An example of a run simulation script is ‘run_sim.sh’ in the ‘CPPSIM’ directory. This script can be seen in Figure 7.



```
#Script to run CPPsim

../CPPsim/run_cmake
make -j8

mkdir muons
cd muons

cp ../../../Events/muon_with_pol.hddm  .
cp ../../control_muons/control.in .

../CPPsim

mcsmear CPPsim_muons.hddm

hd_root -PPLUGINS=cppmva CPPsim_muons_smeared.hddm -o hd_root_muons.root --nthreads=8

cp hd_root_muons.root ../

cd ..
mkdir pions
cd pions

cp ../../../Events/2pi_with_rho_and_pol.hddm  .
cp ../../control_pions/control.in .

../CPPsim

mcsmear CPPsim_pions.hddm

hd_root -PPLUGINS=cppmva CPPsim_pions_smeared.hddm -o hd_root_pions.root --nthreads=8

cp hd_root_pions.root ../

cd ..

cp ../cppmva/trainMVA.C .

root -l -q -b trainMVA.C

ln -s CPPMVA_out.root TMVA.root
```

Figure 7. The contents of ‘run_sim.sh’ runs the CPPSIM within a given build directory.

A break down of ‘run_sim.sh’ is as follows:

- ‘../CPPsim/run_cmake’ moves the simulation information from ‘CPPsim’ to the build directory.

- ‘make -j8’ configures the simulation within the build directory. The flag ‘-j8’ indicates the number of cores in the computer, in this case eight.
- ‘mkdir muons’ makes a directory called ‘muons’ within the build directory.
- ‘cd muons’ moves the user into the ‘muons’ directory.
- ‘cp ../../../Events/muon_with_pol.hddm .’ copies the event file, ‘muon_with_pol.hddm’, into the ‘muons’ directory.
- ‘cp ../../control_muons/control.in .’ copies the desired ‘control.in’ file into the ‘muons’ directory.
- ‘../CPPsim’ runs ‘muon_with_pol.hddm’ through the simulation and outputs a file ‘CPPsim_muons.hddm’
- ‘mcsmear CPPsim_muons.hddm’ smears the data within ‘CPPsim_muons.hddm’ to make the output file, ‘CPPsim_muons_smeared.hddm’, more realistic.
- ‘hd_root -PPLUGINS=cppmva CPPsim_muons_smeared.hddm -o hd_root_muons.root - nthreads=8’ converts the file ‘CPPsim_muons_smeared.hddm’ to a root file, ‘hd_root_muons.root’.
- ‘cp hd_root_muons.root ../’ copies ‘hd_root_muons.root’ back one directory into the build directory.
- ‘cd ..’ moves the user back one directory into the build directory.
- ‘mkdir pions’ makes a directory called ‘pions’ within the build directory.
- ‘cd pions’ moves the user into the ‘pions’ directory.
- ‘cp ../../../Events/2pi_with_rho_and_pol.hddm .’ copies the event file, ‘2pi_with_rho_and_pol.hddm’, into the ‘pions’ directory.

- ‘cp ../../control_pions/control.in .’ copies the desired ‘control.in’ file into the ‘pions’ directory.
- ‘../CPPsim’ runs ‘2pi_with_rho_and_pol.hddm’ through the simulation and outputs a file ‘CPPsim_pions.hddm’
- ‘mcsmear CPPsim_pions.hddm’ smears the data within ‘CPPsim_pions.hddm’ to make the output file, ‘CPPsim_pions_smeared.hddm’, more realistic.
- ‘hd_root -PPLUGINS=cppmva CPPsim_pions_smeared.hddm -o hd_root_pions.root -nthreads=8’ converts the file ‘CPPsim_pions_smeared.hddm’ to a root file, ‘hd_root_pions.root’.
- ‘cp hd_root_pions.root ../’ copies ‘hd_root_pions.root’ back one directory into the build directory.
- ‘cd ..’ moves the user back one directory into the build directory.
- ‘cp ../cppmva/trainMVA.C’ copies ‘trainMVA.C’ into to build directory.
- ‘root -l -q -b trainMVA.C’ processes ‘hd_root_muons.root’ and ‘hd_root_pions.root’ together in ROOT to yield the results of the simulation
- ‘ln -s CPPMVA_out.root TMVA.root’ links ‘CPPMVA_out.root’ with ‘TMVA.root’

The referencing system for the simulation should be kept constant throughout the script for ease of use. The only references that need to be edited are the names of the event files and the location of the control files. Every other file name should be kept consistent from run to run. This consistency starts with ‘control.in’. Although the name of the directory containing this file can be different, the name of the control file should always be ‘control.in’. Even though control files will have different contents their name should always be ‘control.in’. The reason for this is because CPPSIM reference a control file named ‘control.in’. If the control file is name anything else it will not be referenced and the simulation will not run. Even within each ‘control.in’ file there should be consistency. Control files can be edited in four ways by the: name of the event file, number of particles, name of the output file, and number of cores in the computer. The number of cores in the computer is set constant at eight. The name of the output file should always be either ‘CPPsim_muons.hddm’ or ‘CPPsim_pions.hddm’ depending on the type of event file that is being sent through. However, the name of the event file and the number of particles will need to be updated. The output files are kept consistent so that the file name that ‘mcsmear’ references does not need to be changed. Consequently, ‘mcsmear’ always outputs either ‘CPPsim_muons_smeared.hddm’ or ‘CPPsim_pions_smeared.hddm’. And these files are always converted into either ‘hd_root_muons.root’ of ‘hd_root_pions.root’. The user could choose different names for these references and could change them from run to run but they will just be making extra work for themselves.

vii. Adjusting Simulation Parameters

So far it has been shown that two files need to be edited before each simulation to ensure that it runs correctly. These files are the control file, where the number of particles and event file name must be updated, and the run simulation script where the name of the event file and control directory must be updated. Additionally, there are other files that must be updated as well in order to vary simulation parameters. After all, the main purpose of running the CPPSIM is to vary different parameters within the simulation to determine which configuration yields the best results. Therefore, it is necessary to know how to vary these parameters. For the purposes of MWPC optimization the three main parameters are the: size of the MWPC dead region, the degree of angular beam cuts, and the MWPC geometries.

As one would expect, the beam axis is most concentrated with particles at its center. The concentration of particles in this region is so great that if the MWPC's were to detect these particles they would be overloaded. Therefore, a dead region within each MWPC must be implemented where no particles are detected. This region can be thought of as a long tube concentric with the centers of each MWPC. A two dimensional schematic of this region can be seen in Figure 8. In the CPPSIM this region can be turned on and off and its size can be adjusted. This is done in the file 'CPPSensitiveDetectorFMWPC.hh' with directory path CPPSIM/CPPsim/include/CPPSensitiveDetectorFMWPC.hh. In this file, line 125 defines the dead region size. The line reads 'int cutradius = 85', where 'cutradius' is defined to be radius of the dead region in millimeters for each MWPC.

Next, angular cuts can be implemented and adjusted to various degrees. Angular cuts are defined to be conical surfaces centered on the beam line and originate with the beam. There are two cuts, one defines the outer boundary and the other defines the inner boundary where particles can only exist within the region between the two surfaces. Again, this region can be seen in Figure 8. A perpendicular cross section of these angular cuts yields two concentric circles, between which the particles exist. Implementing these cuts is necessary to avoid the extreme regions of the radial particle distribution. As mentioned before, the concentration of particles near the beam line is extremely high and likewise, the concentration far way from the beam line is extremely low. Implementing angular cuts allows for the removal of these regions. Angular cuts are defined in the file CPPSIM/CPPsim/include/CPPSensitiveDetectorTOF.hh. In lines 138 and 139 two radii are specified by 'double outer_rad2 = 316339' and 'double inner_rad2 = 7167'. These are the radii of the concentric circles produced from a cross section at the position of the Time of Flight, where 'outer_rad2' defines the outer radius squared in millimeters and ‘inner_rad2’ defines the inner radius squared in millimeters. Knowing that the TOF is located 606 centimeters down the beam line, the radii can be adjusted to vary the angles of the angular cuts using simple geometry. Currently, the radii are set to correspond to angles of 0.8 degrees and 5.3 degrees.

Finally, the MWPC geometries can be adjusted within two files ‘cpp_HDDS.xml’ and ‘ForwardMWPC_HDDS.xml’. These files are located in a hidden directory ‘$HDDS_HOME’. Collectively, all of the MWPCs together make up the mother MWPC. The mother has additional components such as slabs of iron and aluminum used to construct the MWPCs. The region of space filled by the mother’s contents is called the mother volume. The position of the mother volume is taken to be at its center and can be adjusted along the beam line within the file ‘cpp_HDDS.xml’. Line 84 in this file reads ‘<posXYZ volume=”ForwardMWPC” X_Y_Z=”0.0 0.0 924” />’, where 924 is the position of the center of the mother volume in centimeters from the beam line origin. As the contents and size of the mother volume are varied, this parameter should be adjusted accordingly to ensure that the face of the first MWPC is always set at a position of 825 centimeters down the beam line. The contents of the mother volume are defined in the file ‘ForwardMWPC_HDDS.xml’. This file is much more extensive than ‘cpp_HDDS.xml’ but it is straightforward and can be learned easily. The file defines the dimensions of different structures such as the: mother volume size, iron absorber size, MWPC size, and aluminum plate size. The MWPC and aluminum plate sizes are constant so only the mother volume and iron absorber will need to be adjusted. Be sure that the mother volume is set to be big enough to fit all its contents. Also, keep in mind that a ‘box’ is defined with Cartesian coordinates and a ‘tub’ is defined with cylindrical coordinates. Next, the file ‘cpp_HDDS.xml’ defines the positions of all the structures. The positions are taken to be at the center of each structure. Be careful not to define the positions in such a way that the structures overlap.

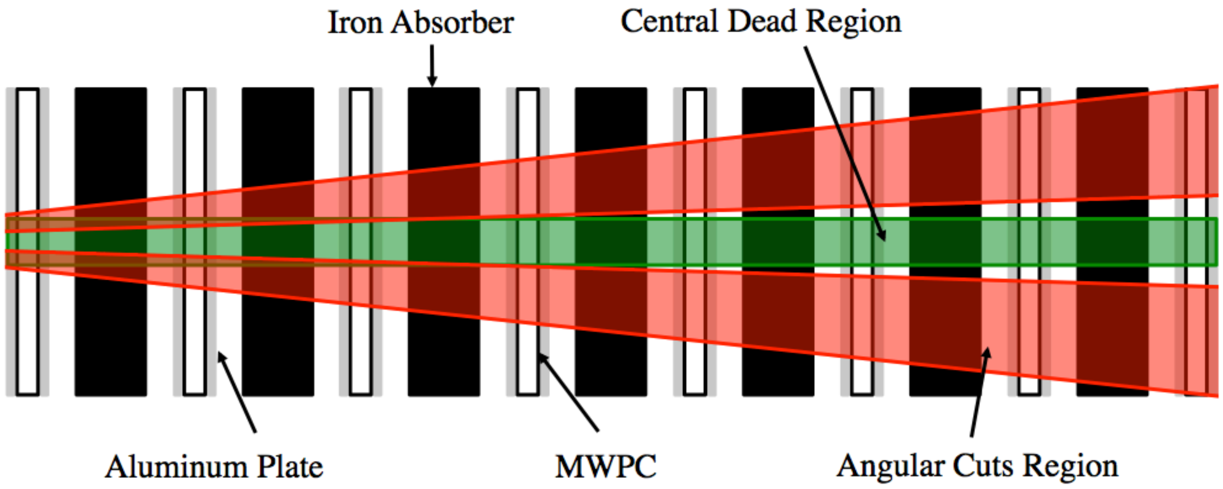


Figure 8. A mother MWPC configured to have eight alternating MWPCs and iron absorbers with indicated dead and angular cut regions.

viii. Analyzing Results

It is not enough to just run the simulation and produce an output file for muons and pions. These files need to be processed to display meaningful results in the form of interpretable graphs. This is done with the macro file ‘trainMVA.C’ and can be viewed in the ROOT Data Analysis Framework. Additionally, event files can also be analyzed in ROOT. Often this is handy because these files are too large to be opened manually. It can be helpful to view various properties of the particles in the event files before they are sent through the simulation to get a better idea of their initial conditions.

Event files come in the form .hddm but they can only be viewed by ROOT if they are in the form .root. Therefore, they need to be converted. This is done with the command ‘hd_ana -PPLUGINS=janaroot -PAUTOACTIVATE=DMCThrown name_of_file.hddm’. This command outputs a file named ‘janaroot.root’. This file can be viewed in ROOT with the command ‘root -l janaroot.root’. Within ROOT, the command ‘.ls’ will list the different trees available for viewing. In this case, DMCThrown is the only tree. The command ‘DMCThrown->Print()’ shows all of the different properties of the event file available for viewing such as the particles distribution of: energy, momentum, and angle, as well as many others. These distributions can be viewed with the command ‘DMCThrown->Draw(“name_of_property”)’. This command will display the desired graph, which can then be saved.

The process for analyzing the simulation output files is very similar. In fact, this process is actually already built into the run simulation script. Observe in ‘run_sim.sh’ that two files are produced by ‘mcsmear’: ‘CPPsim_muons_smeared.hddm’ and ‘CPPsim_pions_smeared.hddm’. These files can be thought of as the output files of CPPSIM. Just as the event files need to be converted from a .hddm file to a .root file, the output files need to be converted in a similar way. For example, the pion file is converted with the command ‘hd_root -PPLUGINS=cppmva CPPsim_pions_smeared.hddm -o hd_root_pions.root -nthreads=8’. In this case, the command outputs the file ‘hd_root_pions.root’. This process is identical in the case of muons, only in the syntax replace ‘pions’ with ‘muons’. These .root files can then be viewed in ROOT with the command ‘root -l hd_root_pions.root’. Again, the command ‘.ls’ shows the trees available and the only available should be ‘cppmva’. The command ‘cppmva->Print()’ shows all of the different properties of the output file available for viewing such as the number of hits in the: TOF, Fcal, and MWPC, as well as many others. These histograms can be viewed with the command ‘cppmva->Draw(“name_of_property”)’. This command will display the desired graph, which can then be saved.

Additionally, the run simulation script takes the .root converted output files and processes them even further. Up until this point the particles have been sent through the simulation as if they were independent of one another. However, in the real experiment this will obviously not be the case; muons and pions travel down the beam line together. This part of the analysis crosses the two output files to yield the results of their interaction. Notice that both ‘hd_root_muons.root’ and ‘hd_root_pions.root’ are copied back into the same build directory. Additionally, the file ‘trainMVA.C’ is copied from the directory ‘cppmva’ into the build directory. The following command, ‘root -l -q -b trainMVA.C’, prompts ‘trainMVA.C’ to cross the two .root files within ROOT. Once this is done, ROOT can be opened with ‘root -l' and the results can be viewed with ‘TMVA::TMVAGui()’. This command yields the interactable seen in Figure 9.

This interactable can produce many graphs but the one of most importance is the (5b) Classifier Background Rejection vs Signal Efficiency curve. This curve is often referred to as the Figure of Merit, FOM, and it shows the muon rejection as a function of pion acceptance. The CPP Experiment wishes to measure as many pions as possible, however, muons should not be included in this data. It is very important that the muon rejection percentage be as high as possible while also keeping the pion acceptance percentage high. It can be seen from Figure 9 that this curve suddenly falls as the pion acceptance approaches ninety percent. One of the main purposes of running the CPPSIM is to find parameters that optimize this curve in a way that maximizes the muon rejection for maximum pion acceptances.

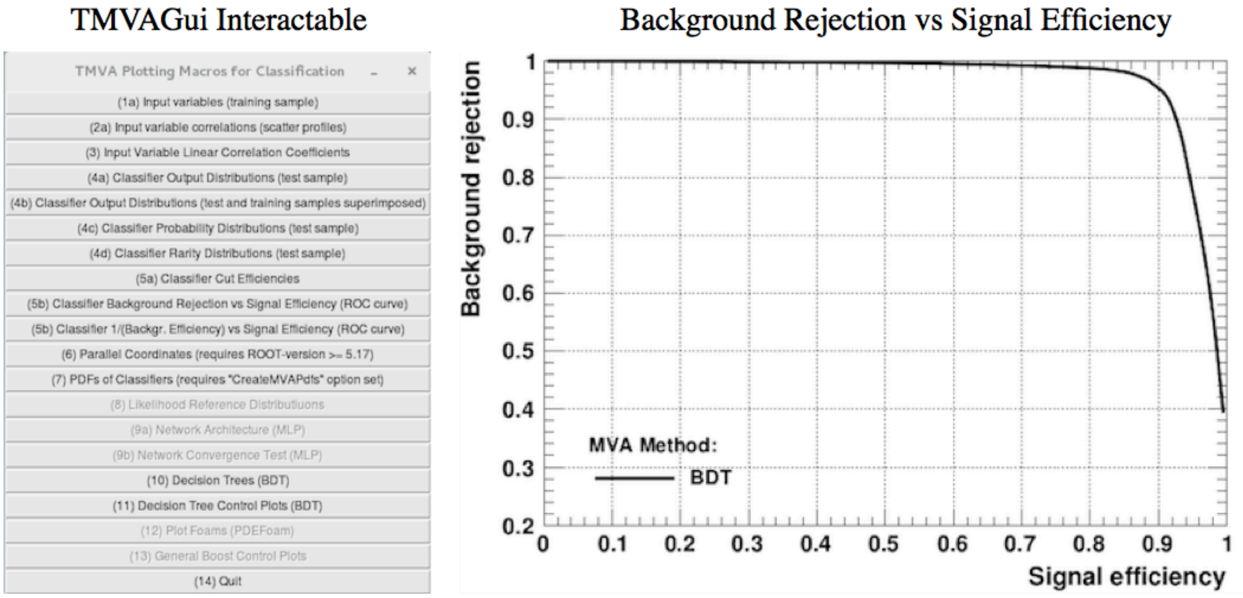


Figure 9. The TMVAGui interactable and the various figures it can produce, one of which is the Background Rejection vs Signal Efficiency curve.

ix. Useful Commands

A good portion of the time spent running the CPPSIM can be trouble shooting. Also, keep in mind that the way the simulation is run can be flexible. It does not have to be run using the exact methods described. The following are useful commands and their functions that will assist in trouble shooting or running the simulation in general.

- ‘rm -r name_of_directory_or_file’ deletes a directory or file.
- ‘sudo command’ performs a command as the root user if the ‘Novem’ user does not have access.
- ‘df -h' views the disk space available.
- ‘chmod +x name_of_file.sh’ converts a .sh file into an executable.
- ‘scons install’ must be executed following any changes to files within ‘$HDDS_HOME’ to save them.
- ‘mv name_of_file location/of/file’ moves a file from one directory to another.

- ‘mv name_of_file new_name_of_file’ changes the name of a file.

Finally, when all else fails, commands often have help flags, ‘-h’. Executing a command with a help flag will display all the information pertaining to that command and hopefully will be the key to resolving any persisting issues.

4. MWPC Construction

i. Introduction

MWPCs are housed in an aluminum frame and are composed of many parts - electronics, a gas mixture, and gold plated wires are just a few. MWPCs in the CPP Experiment contain three hundred wires strung parallel across the inside of the detector. The objective of this section is to provide optimal strategies for stringing such a large quantity of these wires. The MWPCs must remain pristine throughout the duration of their construction to avoid contamination from dust, dirt, and such. This greatly adds to the difficulty of the stringing procedure as it must be performed within the close quarters of a clean room along with other precautions that slow productivity.

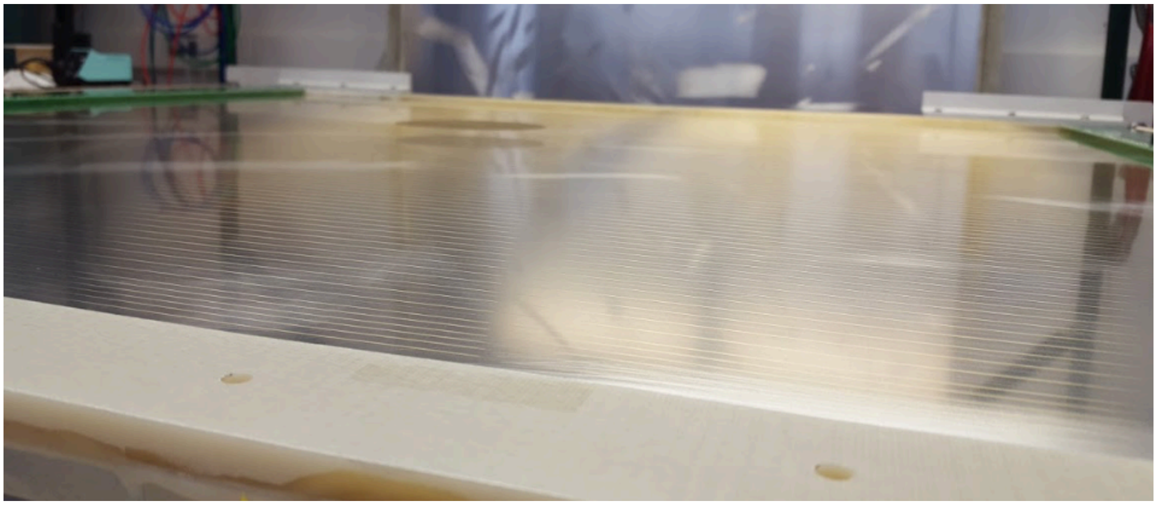


Figure 10. A completely strung MWPC detector.

The clean room and an MWPC can be seen in Figure 11 below. To string a wire it must be specifically tensioned across the MWPC plate while supported by the acme rods and positioned on the correct electronics pads. Then, the wire must be soldered to the pads and the excess wire cut off. This method is simple and straightforward in principle but because of the wires delicacy the process becomes much more difficult in actuality. Stringing and soldering a single wire does not take too long, just several minutes usually, but with three hundred wires total, stringing to completion is extremely tedious. As such, one should take care to optimize the stringing procedure and accustom themselves to the wire and detector sensitivities to reduce the number of working hours necessary to finalize the detector.

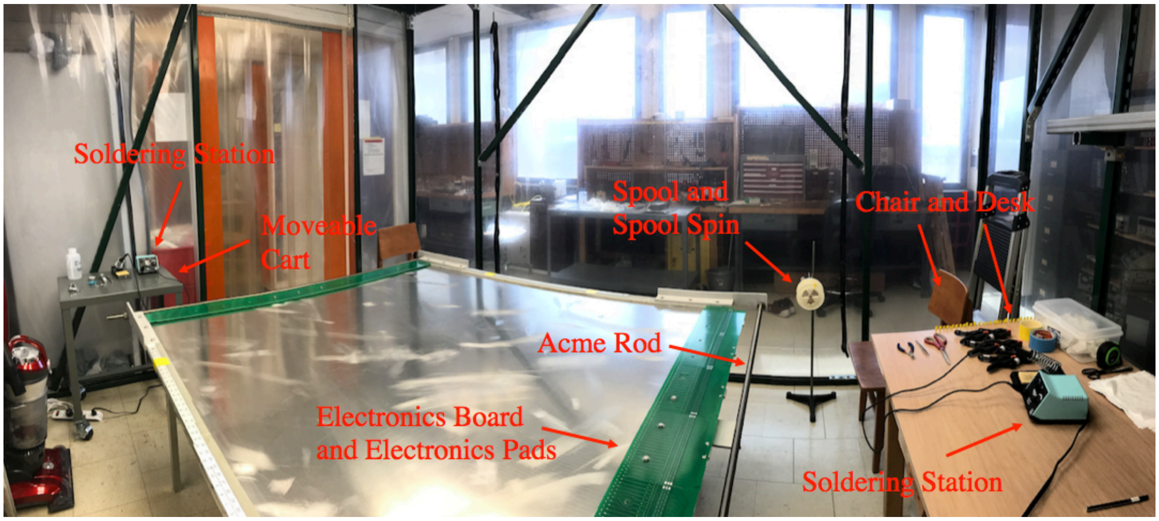


Figure 11. The clean room and MWPC with several key pieces of equipment indicated.

The clean room has been arranged to facilitate the stringing process with several key features. Firstly, the MWPC is accessible on all sides. The second is the spool pin located at one corner of the MWPC. Here, a spool of wire may be loaded allowing a segment of wire to be easily unwound alongside the MWPC. Thirdly, there is a chair and desk next to the spool pin which provides a convenient tool bench and workstation. Finally, there are two soldering irons on either side of the MWPC - one on the desk and one on a moveable cart which provides another workstation. Along with tools and material, this is everything that is needed to completely string the detector efficiently.

ii. General Stringing Procedure

The basic concepts of the stringing process are straightforward in general but there are many subtleties to the actual process that are not obvious at first glance. Having an understanding of the entire process at a basic level will be helpful when trying to grasp the finer details explained later. To start, wire segments are generated by unraveling the spool alongside the MWPC. The wire is cut from the spool and a clamp is attached to each end. The wire is lowered into position over the MWPC where it is supported by the acme rods on either side. The clamps dangle below the acme rods and provide tension across the wire. Once in position, either end of the wire will fall within a groove of the acme rod threads and rest over an electronics pad. The wire is then soldered onto the pads and the excess wire is cut off. Similar variations of this procedure are repeated to string the detector to completion.

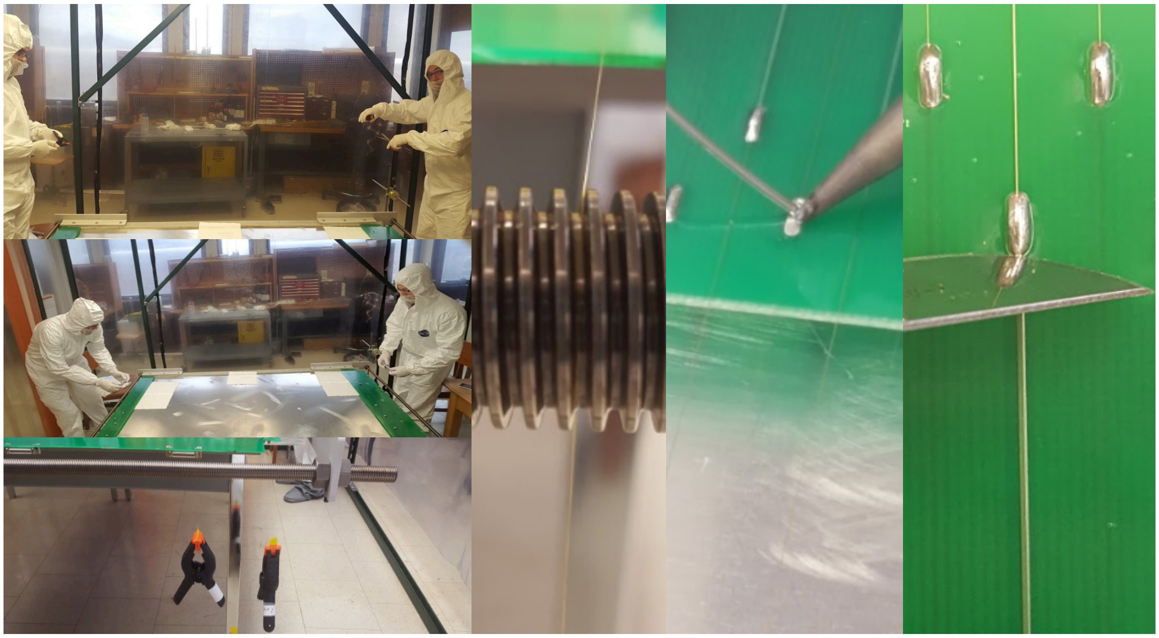


Figure 12. Several major steps of the wire stringing process: unraveling, cutting, and clamping a wire segment, lowering the segment into position, soldering and cutting the wire.

iii. Details

Preparation: The first task in preparing for the stringing process is to wipe down the MWPC plate. Wipe the plate thoroughly using the polyester rags provided in the lab and acetone. This should remove any impurities that may have accumulated on its surface. Once cleaned, avoid letting anything touch the plate, it should remain pristine for the rest of the stringing process. However, there is one exception to this regarding the polyester rags to be explained later.

Cleanliness Precautions: To keep the MWPC clean for the duration of its construction there are several cleanliness precautions implemented. The main line of defense is the clean room within which the detector is assembled. The room is isolated from the rest of the laboratory by clear plastic walls that help prevent the entrance of dust and limit turbulent airflow. The room is constructed such that the plastic walls do not fully reach the floor, leaving a gap of several inches around the perimeter. This exposure is intentional as it provides an exhaust for any dust that does manage to enter the clean room. This is achieved by the filtered ventilation system atop the room. It blows air down creating a pressure gradient to force dust out the bottom. The airflow out the bottom also helps limit dust from entering this region. Another preventative measure is the breezeway entrance before the clean room. Similarly to how a traditional breezeway reduces the transfer of heat from inside to the outside of a building, the clean room breezeway helps prevent the transfer of dust from outside into the clean room. Essentially, the breezeway is just another space that the dust can get stuck in before it potentially makes it into the clean room. The dust in the air is not all that is concerning. There is dust and dirt on the floor as well. To prevent this from being tracked into the clean room there are large sticky pads on the floor just before and after the clean room entrance. Ideally, the pads trap the dust and dirt preventing them from becoming airborne. One should intentionally step on each pad several times before entering the clean room to remove any loose particles that may be on their feet. Similarly, there is double sided tape in the clean room. Several strips of this can be stuck to a workstation's surface with the other sticky side also exposed. The tape can then be used as a safe way to dispose of excess wire segments and other small particles that could easily become airborne if they are not accounted for. The final precautions have to do with one's person as people are dirtier than any item that will enter the room. Shoes are not to be worn in the clean room. The reason for this is pretty obvious; shoes track all kinds of particulates and are really dirty in general. What will be worn is a clean suit. The clean suit provides a barrier between the clean room environment and one's body. People are continuously shedding hair and skin that should not fall onto the detector. The clean suit is one piece with a hood and covers the entire body except for the hands and face. These two areas are protected by latex gloves and, if necessary, a beard cover for the face. With all of these measures in place the clean room is well protected and dust is rarely seen within it.

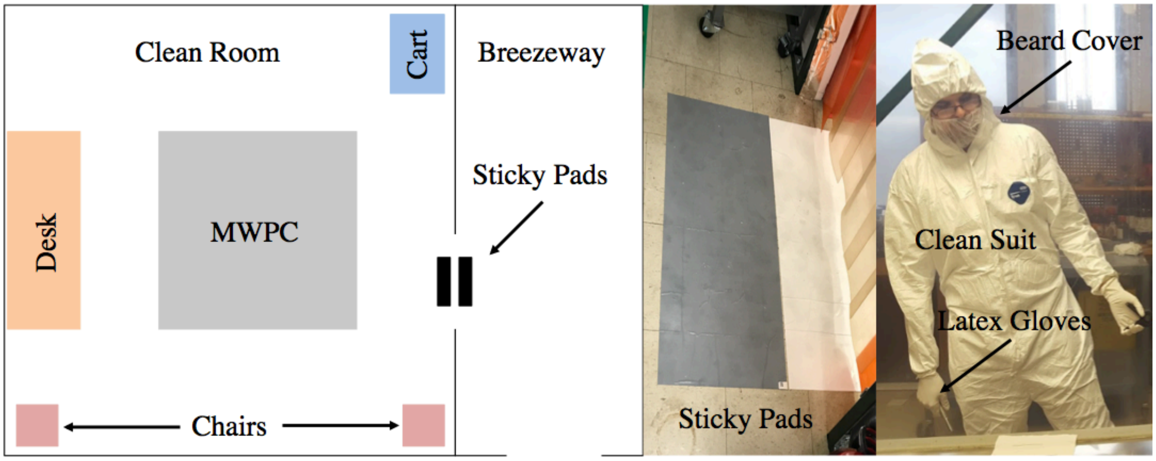


Figure 13. A diagram of the clean room and breezeway entrance as well as examples of cleanliness precautions.

Types of Wires: There are two types of wires to string - sense wires and field wires. The sense wire gauge is 0.0008" and the field wire gauge is 0.00314". As such, the sense wires are much more delicate and require great care when handling. The sense wires can only withstand tension up to the order of fifty grams before snapping, meaning that even the subtlest jerk will likely cause them to break. On the other hand, the field wires are relatively robust and will not require as much care. One need not be concerned with the tension load placed on field wire, however, they should be cautions of jerking for the sake of maintaining slow and controlled movements around delicate sense wires. In the case of the field wire, attaching the clamps is easy - one just clamps them to the wire. However, attaching them to the sense wire is not as simple due to its relatively small gauge. If one tries to clamp a bare sense wire it will simply slide through the jaws of the clamp when tension is applied. To resolve this, one must provide a surface on the wire suitable for the clamp to grasp. We have found that the simplest solution to this is to fold a piece of tape over each end of the wire and then clamp each piece of tape. The one caveat is that the tape adhesive must be strong enough to adhere to the wire. There are several variations of tape within the lab; we have found that simple Scotch Tape will suffice. In addition, each of these wires has their own corresponding electronics pad type. The pads closer to the detector plate are sense pads and ones further from the plate are field pads.

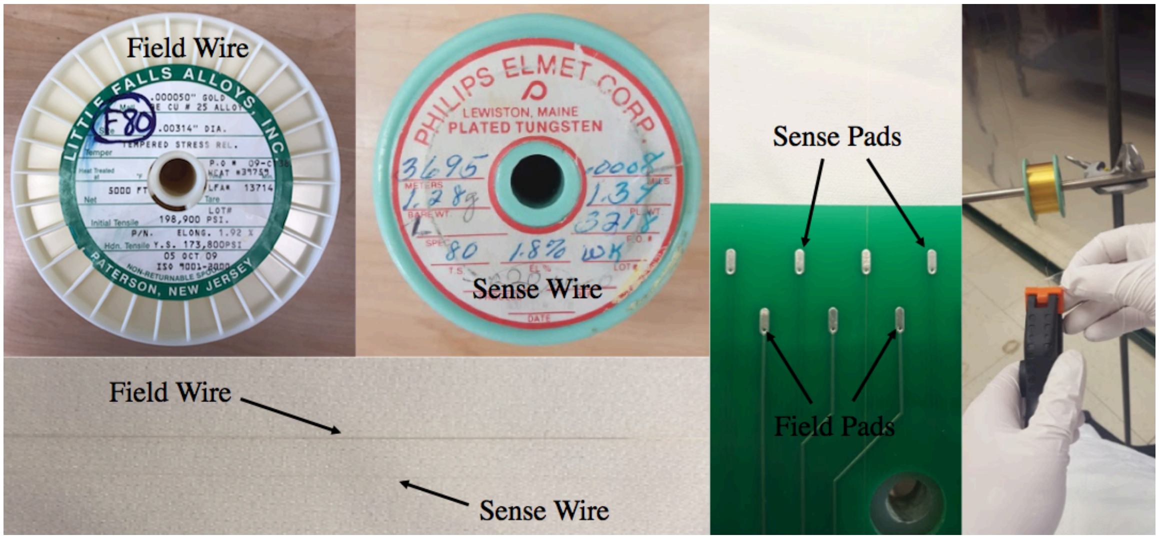


Figure 14. A spool of each wire type, field and sense, along with their relative gauges and pads, as well as the clamp-tape technique.

Detector Regions: In addition to the two types of wires there are also two regions of the detector - the center region and the outer region. The only difference between the two is that within the central region there are sixteen carbon tubes each strung across a sense wire forming a circular pattern. Stringing the central region is much more difficult than the outer region and should be done first. However, the procedure for stringing this region is the same as that of the outer region with just a few extra steps and as such will be presented second.

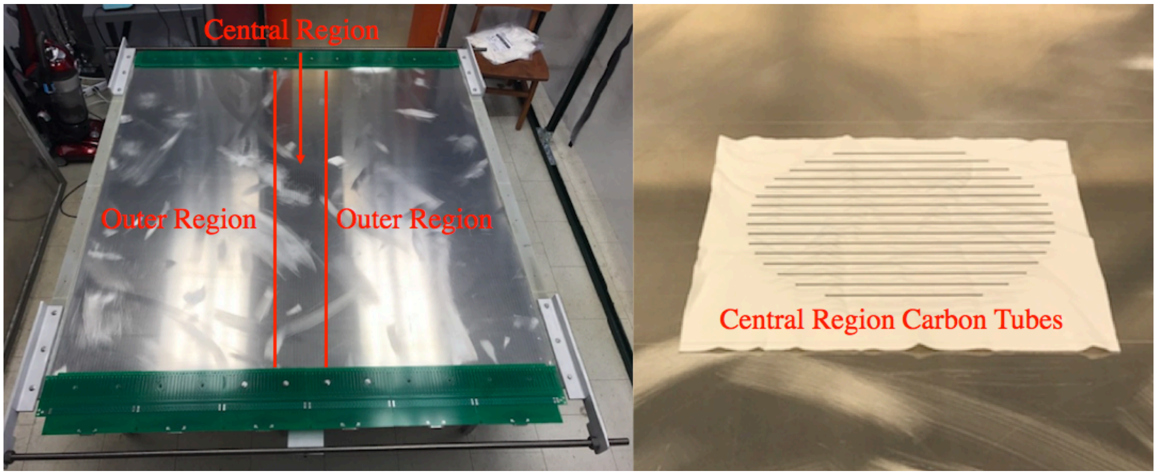


Figure 15. An MWPC with central and outer regions indicated. Also, a completely strung central region with circular carbon tube pattern.

Purpose of the Carbon Tubes: In the CPP experiment each MWPC detector is centered along the beam line of particles. Intuitively, the center of this beam has the highest concentration of particles and it decays radially outward. The concentration of particles in the center is so high that if the detector were to measure them it would be overloaded with data. To prevent this from happening the central region needs to be shut off. This is accomplished with the placement of the carbon tubes. Detection is inversely related to the gauge of the sense wire. By inserting the carbon tubes into the central region they effectively thicken the wire and shut off detection in that region.

Acme Rods and their Calibration: Before any wire stringing can be done the acme rods must be calibrated. These rods are essentially just giant screws manufactured with very precise thread spacing. The idea is that the precise spacing of the threads can be used to space the wires across the MWPC with equal accuracy. Due to manufacturing tolerances of the electronics boards, the exact center of each electronics pad actually varies somewhat and is not accurate enough to guide the placement of each wire. Instead, the acme rods are used to position each wire. To do this, the rods are mounted on either side of the MWPC by aluminum braces which attach to the G10 frame. The rods are threaded within the aluminum braces such that spinning them will cause a shift in their position one way or the other. To calibrate them, string four field wires with fifty gram clamps but do not solder. String two over the fourth field pads in from either end and the other two over the twelfth field pads from either side of the center. Then, spin the acme rods until the placement of all four wires is optimized over the center of each pad. This is somewhat of an iterative procedure since adjusting the position on one side slightly affects the other. Adjustments must be done several times alternating either side to fine tune the wire positioning. This is best done with two people but can be done with one just as well. Note, the wires may not fall exactly over the center of all the pads, this is fine, trust that the thread spacing is more accurate that the pad spacing, after all, this is the rods very purpose. Now that the rod positions are finalized they must be locked into place to prevent them from accidentally moving during the long stringing procedure. To do this, just screw the large nuts up against both sides of each brace. Be careful not to spin the acme rod as you spin the nuts. Now, there is just one final step needed to complete the calibration process. This is to ensure that each rod is level with the MWPC itself. There are three support screws underneath either rod, one on each end and one in the middle. These screws can be used to slightly adjust the height of the rods at either one of these positions. Adjust the screws such that each of the four field wires just barely grazes the electronics pads they fall over. After this, calibration is complete and stringing may begin.

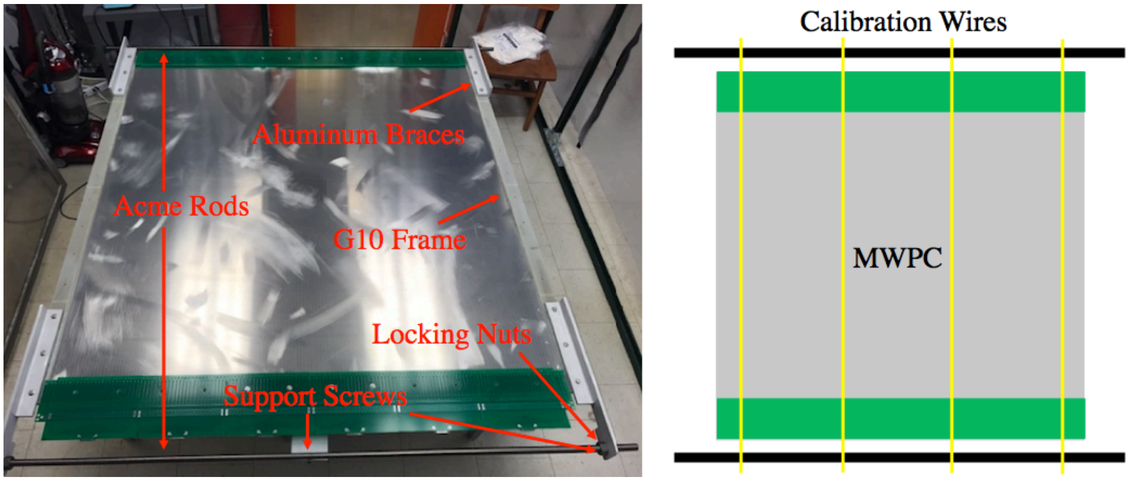


Figure 16. Acme rod configuration diagram and calibration wire schematic.

In addition, there is a small caveat that comes with stringing using the acme rods. These rods have square bottomed threads, not triangular ones as typical screws have. This means that there is a slight degree of freedom to the wires position within any given thread. Before any stringing begins, a side to which to push each wire after placement must be established. For example, wires in the past detector were always pushed towards the side nearest the window in the lab. This will ensure consistency in the positioning of every wire. Otherwise, the wire may fall anywhere within the square bottom leaving a large error in wire position. The diagram below gives a visual explanation of this procedure. Notice in the diagram that the orientation of the screws relative to each other is anti-parallel. The acme rods should be set this way when they are locked into the aluminum frame. It is easier to observe than to explain this phenomenon but essentially this orientation helps prevent the wire positioning from sliding before it is soldered into position. Basically, as the wire runs off the screw stretching over the detector it should press up against the side of one thread. If the screws are not anti-parallel then on one rod the wire will not be against the thread as it runs towards the detector but rather as it runs towards the ground.

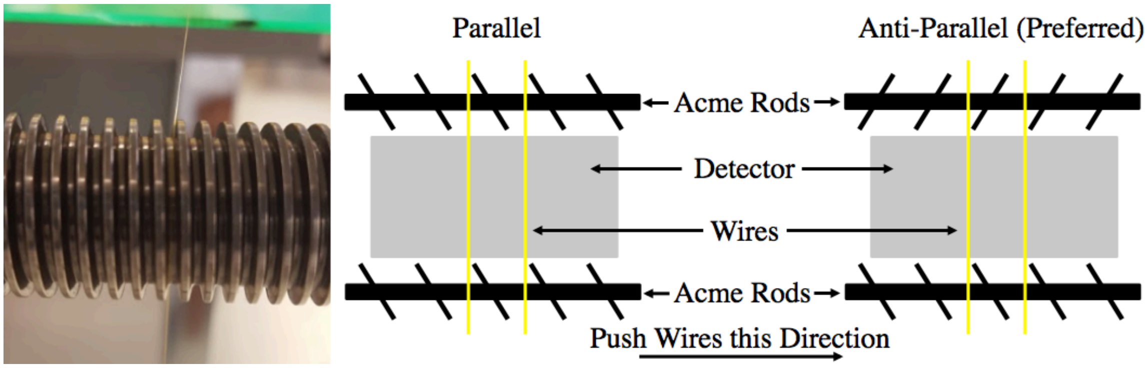


Figure 17. An example of a sense wire within an acme rod thread as well as a diagram of parallel and anti-parallel acme rods indicating which direction to push the placed wire.

Wire tensions: Any wire that is anchored at two points will have some amount of gravitational sag in-between. Although the wires strung across the MWPC are very light, they are no exception and there is still a slight sag to them. The amount of sag depends on the type of wire being strung and the region being strung. Field wires are thicker and heavier than the sense wires and therefore have greater sag. Sense wires in the central region have the added weight of carbon tubes and therefore have greater sag than sense wires in the outer region. As such, different wires will need to be tensioned specifically based on their wire type and position to maintain consistent sag throughout the detector. Field wires do not change throughout the detector, so the tension across field wires is the same for each. For the past detector this tension was provided by fifty gram clamps attached at each end. Sense wire tension depends on region. In the outer region the tension is the same for each wire and historically was set by twenty-five gram clamps attached at each end. The tension of the sense wires in the central region actually varies from wire to wire due to the different carbon tube lengths on each. Intuitively, wires with tubes of greater length will need to be tensioned with a greater load to account for the added mass. The tension of each of these wires varies between twenty-five and fifty grams and is provided in the Tension Position Map by Professor Miskimen. A past map can be viewed below. Variation of clamp masses is created by attaching additional weight to twenty-five gram clamps until the desired weight is reached. A successful strategy for this has been to tape nuts to the clamps to add mass.

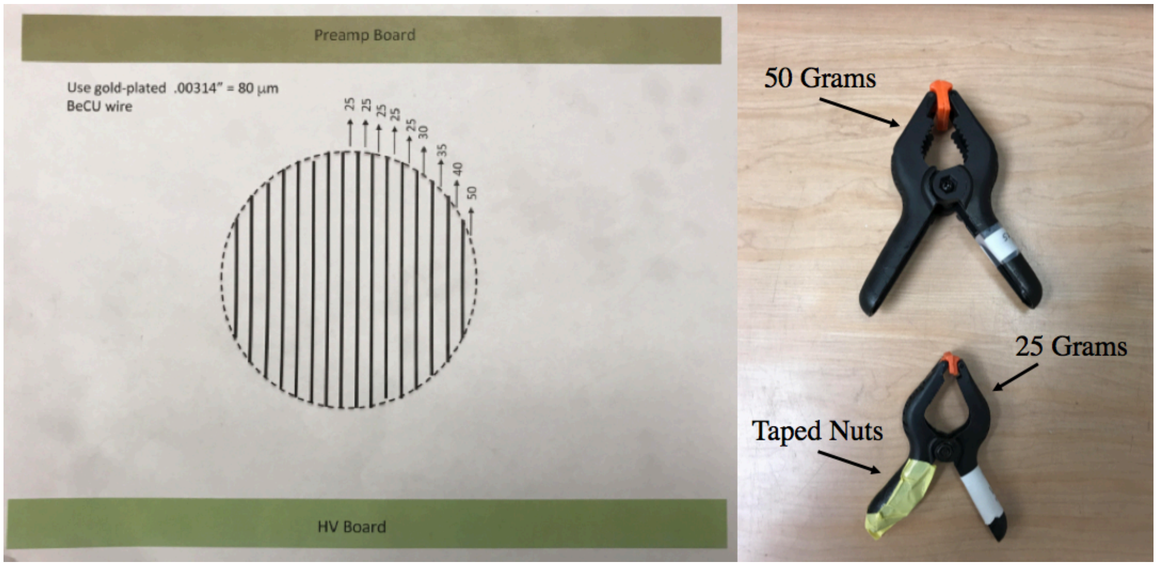


Figure 18. Wire tension map as well as clamps of various weights.

Wire Transportation: Handling wires can be very unwieldy - they are difficult to see and can easily be snagged or jerked causing them to break. A convenient method for transporting clamped wire segments is to use the transport bolts located on the desk and cart. Two students handling the wire segment each hold the threads of a bolt and gently drape their wire ends over the smooth shank. They lower the clamps until the wire is tensioned and can be released without jerking. The students may then transport the wire by simultaneously moving their bolts in a given direction. The convenience of this practice is that it maintains constant tension in the wire, meaning that, even if one cannot see it, they have a pretty good idea of where it is in space. Also, it comes in handy when positioning the wires because it drastically increases the accuracy of its placement and the tension prevents the wire from sagging and touching the MWPC plate. This method is not restricted to using bolts, any smooth cylindrical surface will suffice. However, the bolts do seem to work especially well due their head preventing the wire from sliding off.

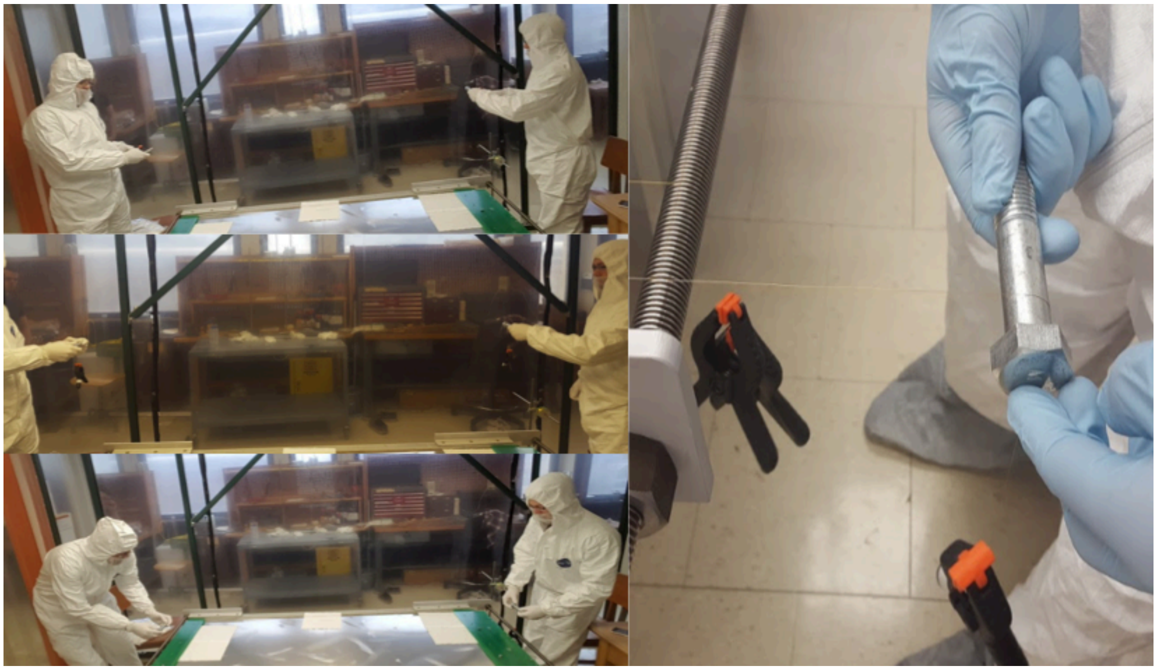


Figure 19. Several key steps in manipulating a wire segment using the transport bolts: transfer the wire to the bolts and gently tension, translate the wire over the MWPC, lower the wire into position.

Soldering: Soldering is a significant portion of the stringing process and has two purposes. The first is that it conducts between each wire and the electronics pads. The second is that it serves as the adhesive connecting the wire to the detector to prevent it from moving and to maintain tension. The soldering procedure requires two things - a soldering iron and solder itself. The solder is a metal alloy and comes wound around a spool similar to wire and the iron provides a hot metal tip used to melt and apply the solder. The best way to solder is to touch the iron, the solder, and the pad all together at the same time. The iron will melt the solder and it will flow onto the pad. Melt a sufficient amount of solder, about a grain of uncooked rice, and then remove the iron and the solder simultaneously. What should result is a small, smooth, and round bead of solder encasing the wire and completely covering the pad. Be warned, of all the skills needed to string the detector, soldering is the most difficult and takes the most practice to master. There are several things that can go wrong when soldering. One is that too much solder is melted and the solder joint is too big. The joints should be relatively small and even more importantly they should be consistent. If the joint is noticeably bigger or different from the others in any way it should be removed and replaced. A second problem that can occur is that the joint has a sharp spike or deformity. This is an issue because charge builds up at these sites leaving them prone to arcing and additionally produces inconsistent electric fields. Typically, a deformity occurs if the pad is not completely covered with solder and a spike will sometimes form when removing the iron from the joint. It is possible to fix these issues by re-melting the joint and adding just a little extra solder to help smooth it over. The joint becomes bigger and more difficult to fix with each time the solder is re-melted. If this technique does not work on the first attempt, remove the joint and re-solder completely.

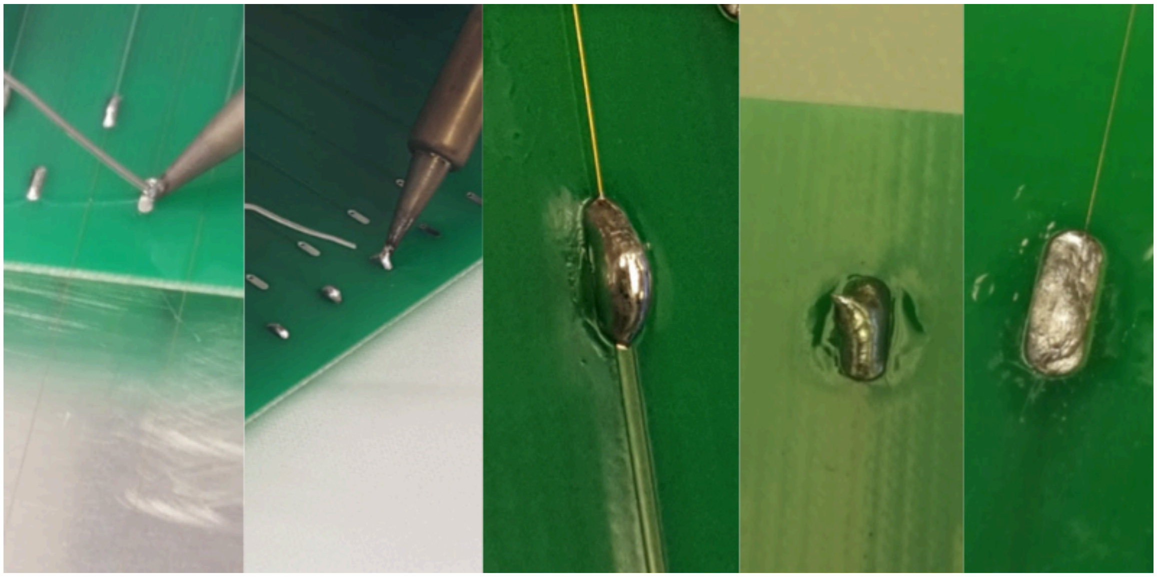


Figure 20. The soldering technique as well a successful joint, a spikey joint, and a dull joint that has been re-melted too many times. Notice the tail being pulled in the second image.

Beginners will find that it is pretty easy to get the solder to completely coat each pad; it simply flows to the boundaries, stops, and pools. However, they will find it much more difficult to avoid producing spikes. Removing the iron tends to pull a tail of solder with it. When the iron detaches from the tail it cools and becomes a solid spike. There is a part of the solder called flux which helps prevent this. The flux is essentially a lubricant that helps the solder flow. If you form the joint before all the flux is burnt away it will help pull the tail back into a bead when the iron is removed. Another way to help avoid forming a spike is simply to practice. Soldering is just like playing a sport or crafting a piece of art; there is a special touch to it that cannot be explained but rather is learned from experience. Given this, it is a good idea to practice soldering before the actual stringing of wires. There are spare PCB boards with electronics pads in the lab. Practice soldering about one hundred pads to get a comfortable feel for the skill.

Soldering Iron Tip Protection: The soldering iron is hot, typically set at seven hundred degrees Fahrenheit. As a result, the metal tip can oxidize quite easily. When this happens the tip struggles to conduct heat easily and is basically useless. There is a way to fix it if this happens but the more it oxidizes the easier it will in the future so it is best to prevent it in the first place. The tip oxidizes because of heat, therefore the best way to limit oxidation is to limit the amount of time that the tip stays hot. There are several ways to do this; one way is to turn the irons on and off in between soldering wires. The second method is to position several wires and solder them back-to-back and then turn the irons off to prepare the next set. Either way, this keeps the tips cool when they are not being used. Secondly, keeping the tip clean will help prevent oxidation. Under each iron holder is a sponge. Wet the sponge and use it to wipe the tip before and after soldering each joint. This will wipe off any solder residue stuck on the tip that would assist oxidation. Finally, there is a product called tip tinner that can be used to coat the tip with tin to keep it clean. Some people recommend using this between each joint but we have found that it is usually unnecessary and only needs to be used when the tip is especially struggling to melt the solder. As long as the tip stays clean by using the sponge the tip tinner should not be needed. Now, if the tip is already oxidized and needs to be fixed this can be done with brass wool. Rub the brass on the tip to scrape off the oxidation but be careful. There is an iron coating on the tip that can also be scraped off. If this happens the tip will oxidize much more easily in the future.

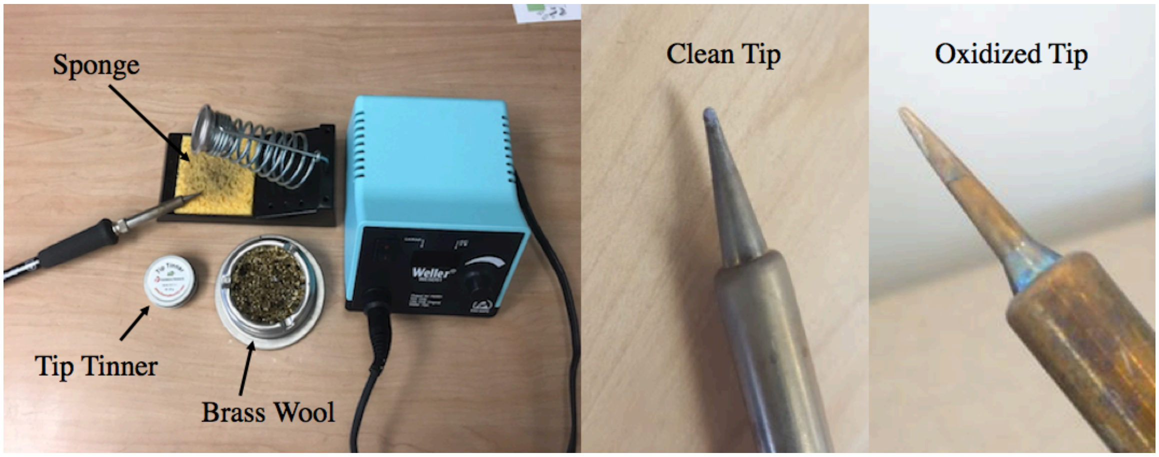


Figure 21. Tools used to protect the iron's tip: a wet sponge, tip tinner, and brass wool. Also, comparison between a clean tip and an oxidized tip.

Solder Removal: Given that soldering can be temperamental at times it will be necessary to know how to remove solder when joints are not satisfactory. This is done using solder remover, which is just a woven cord of copper wire. The copper looks like a wick and acts as such to draw the solder off the pad. Be sure that the excess wire has not been cut from the joint and the clamp is still dangling. If not, the entire wire will need to be re-strung. Removal is simple, re-heat the joint into a fluid with the iron and touch the end of the copper to the joint. As long as the solder stays liquid, it will be drawn from the pad up the length of the cord. This means that the iron needs to be touching the pad or the copper just above the pad the whole time to conduct heat. If not, the solder will solidify and the copper will be stuck to the pad. To fix this, just re-apply the iron to the pad and the solder will liquefy again; do not try to rip the copper off the pad if it is stuck. Depending on how much solder needs to be removed, you may need to adjust the copper as more and more of it is used. Just slide fresh copper along over the pad as it absorbs the solder like a paper towel wiping up a spill. As the copper slides across the pad, burned flux tends to smear onto the PCB leaving a dark residue. This is not a big deal but remember to wipe it up with acetone and a polyester rag after all the solder is removed. The majority of the time solder needs to be removed there will be a wire encased in the joint. Solder remover absorbs most of the solder but not all of it. There will be a small amount leftover coating the pad and the wire meaning that typically the wire will still be stuck to the pad. The wire must be repositioned because during the removal process it may have shifted. Touch the iron to the pad to liquefy the solder coating and lift the wire off the pad. Remove the iron and the solder will harden so the wire can be repositioned. Much of the stress of soldering is relieved knowing that there is always another chance if the joint is not sufficient, especially when removal is easy and the miniscule amount of solder leftover actually makes soldering the second time much easier.

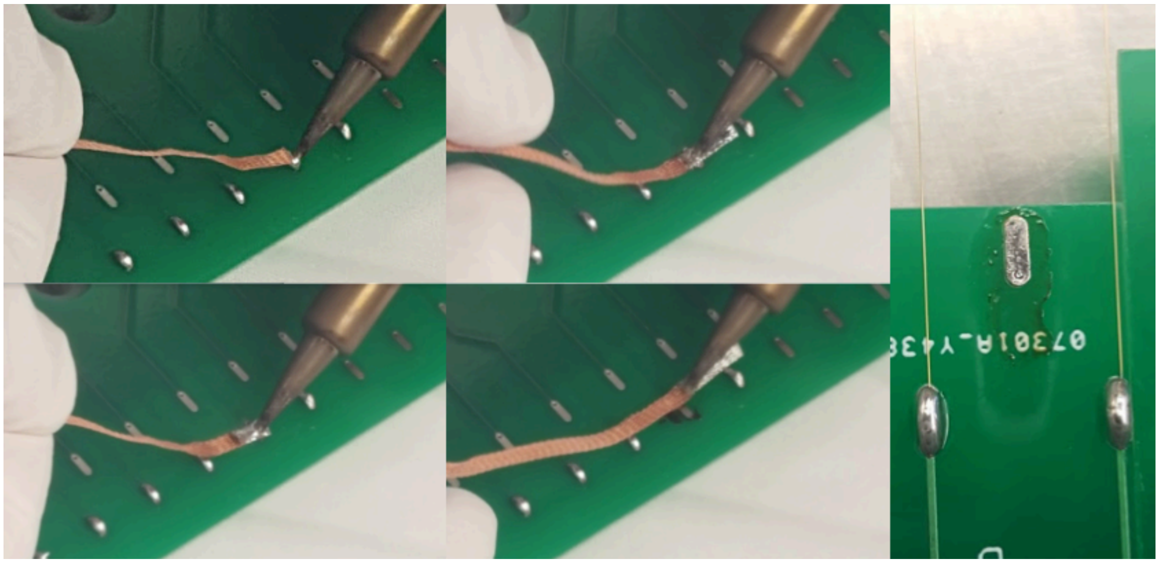


Figure 22. Highlights of the solder removal process. Slowly slide the copper wick and iron tip over the joint to absorb the solder. Notice the flux residue after removal.

Flux Spatter: The flux in the solder assists with the application process greatly. Unfortunately, little pellets of flux tend to jump from the solder as it is melted and spatter onto the MWPC plate where they harden and stick. The plate is supposed to stay pristine throughout the entire string process so this must be avoided. The pellets can be saved from contacting the plate by laying a polyester cloth around the area being soldered to catch them. The polyester cloths are the one exception to anything touching the plate. Still, use of the cloths should be limited; no more than four or five cloths at a time is necessary. Be sure to slide the cloths along as new wires are strung and soldered. It is easy to forget to move the cloth and after several wires it will be trapped beneath them. If this happens just use tweezers cautiously between wires to gently slide the cloth until it is accessible. Occasionally flux spatter will cling to a strung wire. There are two options to fix this - scrape it off with tweezers or re-string the wire. It is easy to scrape off but there may be a chance that there is still invisible residue left over. The safest procedure is to replace the wire entirely. These events happen very infrequently so wires will not need to be replaced often.

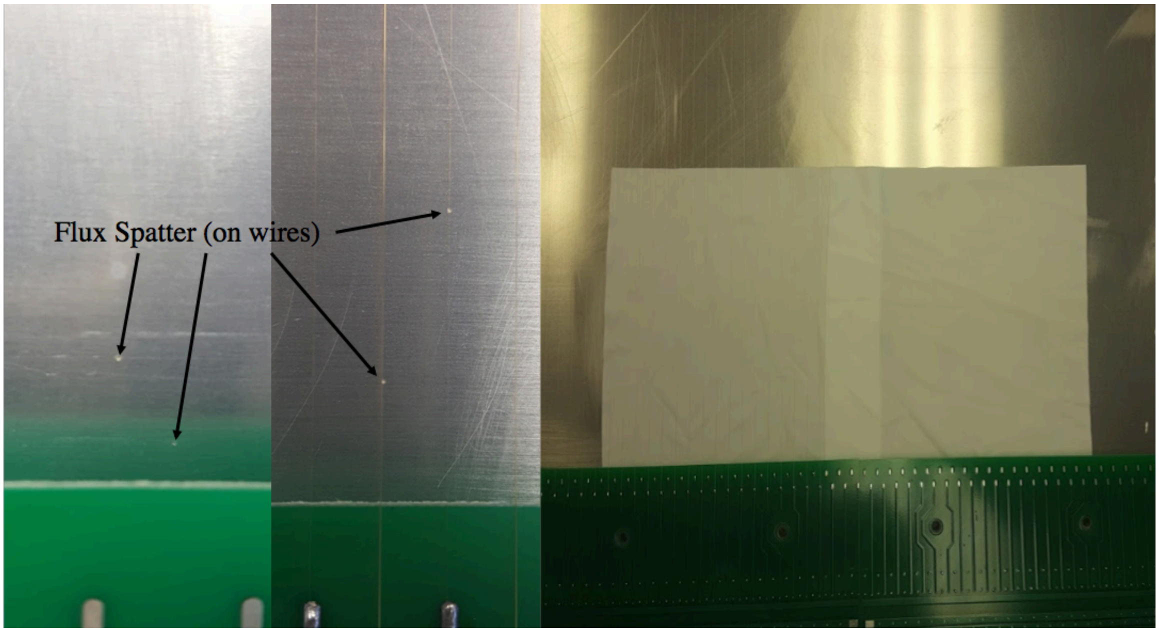


Figure 23. Flux spatter on the plate and on wires. Place polyester cloths on the plate in the soldering region to catch flux spatter.

Cleaning Flux Spatter: If the polyester cloths are forgotten and flux spatter manages to get on the MWPC plate, it must be removed. The easiest way to do this is with acetone and a polyester cloth. However, the plate is only accessible to this method if wires have not been strung above the spatter. Typically this will not be the case because spatter arises from the very act of stringing wires. There is still an effective method for removing spatter stuck below wires. This can be done with the vacuum in the clean room and a specially configured soda straw attached to its end. The straw is just thin enough that it can slip between wires. With the vacuum on, the straw can be used to scrape the spatter off the plate and suck it up once its detached.

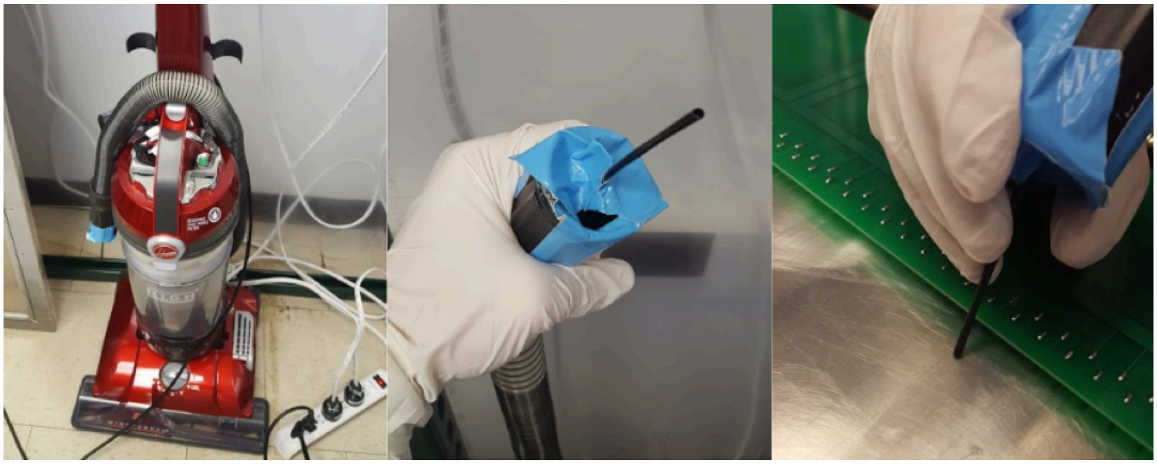


Figure 24. Customized vacuum with soda straw attachment. Removing flux with the vacuum-straw in-between wires.

iv. Refined Stringing Process

The procedure used to string wires will depend on the region and the number of students available to do so. We have developed several procedures for the cases when there are two, three, or four students. It is possible to string with just one student, however, this is not recommended; for a guide to this procedure see Johnson's Thesis. [7] The major difference between procedures is the allocation of who unravels and clamps each segment of wire; aside from this, wire positioning and soldering is identical. Below will give the details to each stringing procedure for field wires. The same procedures may be applied when stringing sense wires with the additional step of applying tape to the wire ends before clamping. Additional procedures for stringing the sixteen central sense wires will be explained after.

Two students: Student A works on the desk side of the MWPC and student B works on the cart side. Student B clamps the wire end at the spool and unravels a segment by walking back to the cart side. Student A clamps and cuts the wire at the spool and the two tension the wire over the transport bolts. The segment is then cautiously translated over the MWPC and lowered into the correct position. During this step, prevent the dangling clamps from swinging and hitting the acme rods or other wires. Repeat this procedure for several other segments and place them at arbitrary positions before soldering, they will be moved to their correct positions later. The number of additional segments will be limited by the clamps available and by the amount of extra space on the MWPC to place them. When placing the extra segments be sure to space them far enough apart that the dangling clamps do not bump each other. Also, try to locate them in a way that will make transporting them to their correct position easy. Both students simultaneously solder and cut their ends of the correctly positioned wire. Then, they transfer a wire segment to the next correct position, solder, cut, and repeat for the remaining wires. The students then continue to repeat this entire process.

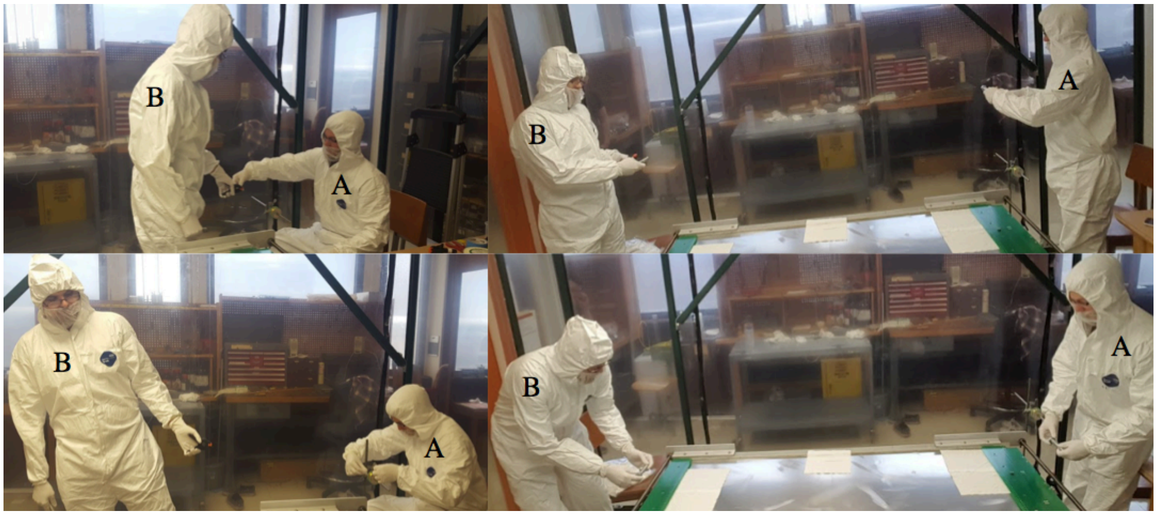


Figure 25. Allocation of people in the two-person string procedure.

Three Students: Student A works on the desk side, student B works on the cart side, and student C sits in the chair behind the spool pin. Student C clamps the wire end at the spool and hands it to student B who unravels the segment by walking back to the cart side. Student C clamps and cuts the wire end at the spool and hands it to Student A. Students A and B tension the wire over the transfer bolts and correctly place it while student C clamps one end of the next wire. Student B unravels the next wire and student C clamps and cuts it at the spool to give to student A. Students A and B positions the wire arbitrarily with the transfer bolts while student C clamps the next wire. This process is repeated for as many wires as wanted or possible. Students A and B simultaneously solder and cut the correctly placed wire. Students A and B then transfer a wire segment to the next correct position, solder, cut, and repeat for the remaining wire segments. The three students then continue to repeat this process. The main difference with this procedure is the use of Student C to save time wire clamping for Students A and B.

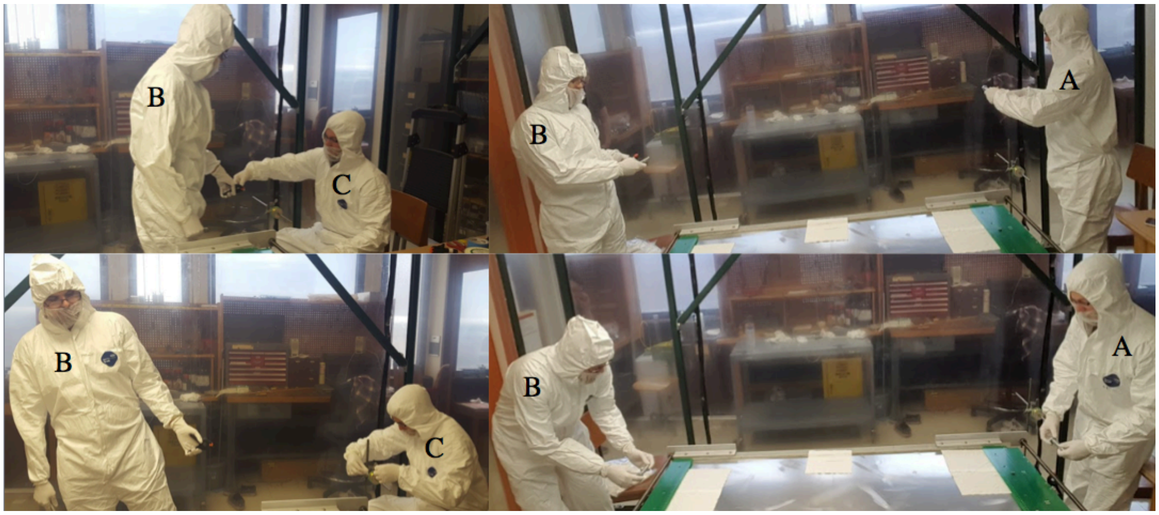


Figure 26. Allocation of people in the three-person stringing procedure.

Four Students: Student A works on the desk side, student B works on the cart side, student C sits in the chair behind the spool pin, and student D works alongside the MWPC unraveling wire segments. Student D clamps the wire end at the spool and unravels a segment by walking back to the cart side. Student C clamps and cuts the wire at the spool and the two tension the wire over the transport bolts. Students A and B receive the wire and transport bolts from students C and D and move it into position. Students A and B solder and cut the wire as students C and D prepare the next wire. Once finished, students A and B receive the next wire from students C and D and the process continues. This is the quickest and smoothest of all the processes. It takes just about the same amount of time to position, solder, and cut a wire as it does to clamp, unravel, cut, and tension over the bolts. This procedure is most efficient because no time is spent waiting at any point.

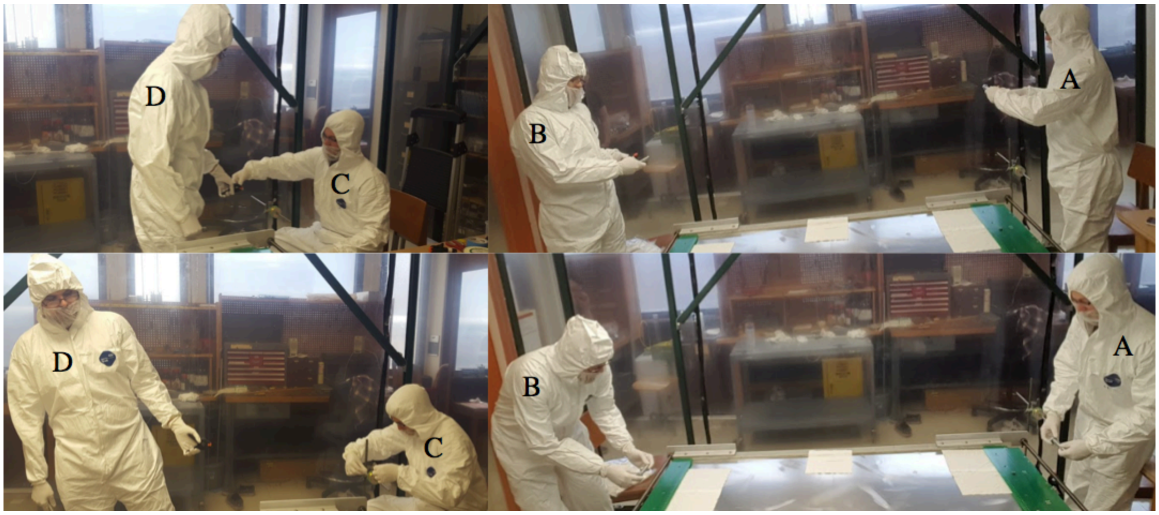


Figure 27. Allocation of people in the four-person stringing procedure.

v. Stringing the Central Region

Carbon tubes are strung across the central sense wires to create a dead region in the detector. There are several steps added to the general wire stringing procedure to achieve this. Wire stringing is already a very delicate procedure and these additional steps make it even more so, luckily there are only sixteen carbon tubes. It should be reiterated again, especially within this section, the central region should be strung first. This region is the most difficult to string and having the rest of the detector open makes it much easier.

Cutting and Sanding Tubes: Each of the sixteen carbon tubes needs to be cut to a specific length to form the circular pattern in the central region. This is done with a precise ruler and an xacto-knife in the lab. The lengths of each carbon tube will be provided by professor Miskimen. In addition, the ends of the tubes need to be rounded with sand paper. This is due to the same reason that the solder joints cannot be spikey; charge builds up at edges and causes arcing and irregular electric fields. Be sure to sand the tubes in the prep room and not in the clean room to avoid creating airborne dust. For more detail about this procedure refer to Johnson's Thesis. [7]

Stringing Tubes: Since the sense wire is so thin it is difficult to see and manipulate. Because of this, it can be extremely challenging to thread the sense wire through the entire carbon tube, especially for longer tubes. Just like soldering, the best way to improve at this is simply to practice. There are several strategies to threading but for the most part this technique will be unique for each person and will be developed from experience. With that being said however, from the author's experience, the best way to string a tube is to hold the tube in one hand and the end of the wire in the other. While sitting down, rest your forearms on your thighs and string the wire just above your lap; this position allows for very fine dexterity. Hold both the tube and the wire about a centimeter from their ends. Gently insert the wire into the tube and keep pushing it through about a half-centimeter at a time. Do not move the tube, just push the wire through it. Push the wire from a half-centimeter away from the carbon tube all the way to its end. Do not push from farther than a half-centimeter away. At this distance, if the wire resists the push it can bend and kink easily. A kink in the wire ruins the process completely because it cannot pass though the tube and one must start over. Once the wire emerges from the other side of the tube, pinch it and lift it up several feet to slide the tube down the wire. Then, tape and clamp the wire end. In general, the best advice is to go slow and be gentle. Stringing the tubes require patience more than anything.

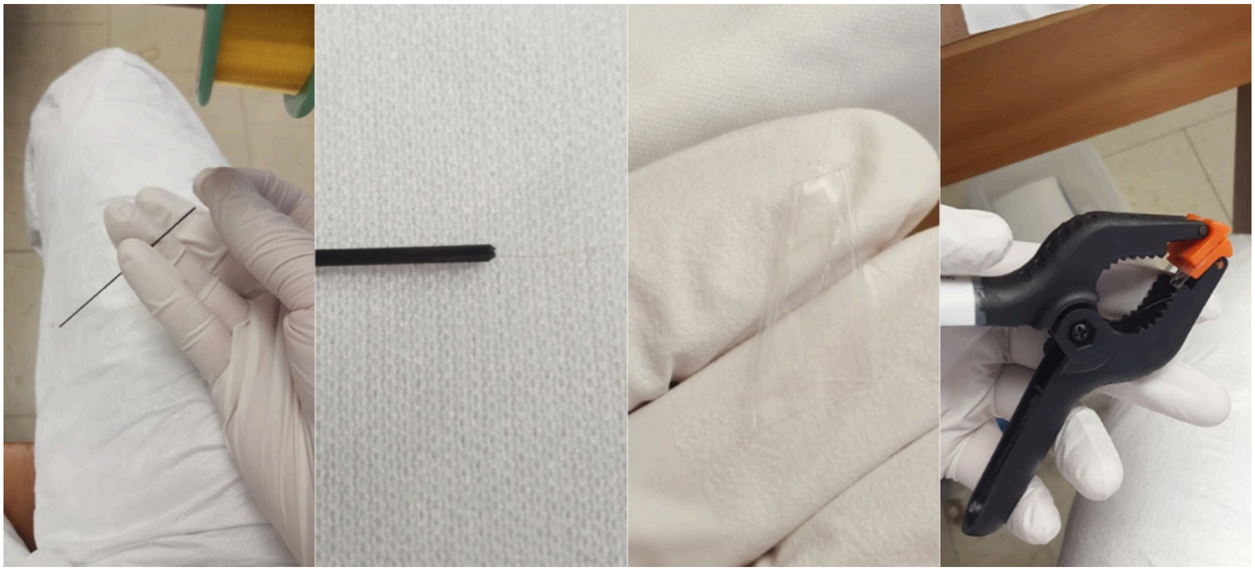


Figure 28. Several steps of the carbon tube stringing procedure.

Placement Strategy: Once the wire is completely threaded through the tube the whole wire needs to be strung. For reasons to be explained shortly, unravel slightly more wire than usual, about a foot and a half of wire should dangle from each of the acme rods. Then, clamp the wire ends with the appropriately weighted clamps, cut the wire, and use the transfer bolts to lower the segment into position. Although each carbon tube has a specific position in the central region, the wires should first be positioned as close to the edge of the detector as possible to facilitate the next procedure of painting the tubes. They will later be moved into their correct positions. They are placed at the edge of the detector because it is much easier to paint at the edge than it is at the center. Also, at the edge, the G10 frame can be used as a rest to stabilize ones hand during the painting process. Four wires should be painted at a time. This is the maximum amount that can be placed close enough to the edge to be painted controllably. The first wire can be positioned nearly at the edge but because of the dangling clamps the next three wires cannot be placed as close without bumping into each other. To prevent bumping they are typically placed at least a clamp's length apart. However, this distance can be shortened by about half by staggering the clamps as seen in Figure 29. This is why more wire should be unraveled than usual; the extra wire will give more freedom to stagger. Without it, there might not be enough length in vertical clamp positions to stagger effectively.

Now, the wires are positioned and the tubes are strung across them, but the tubes may not be in an optimal position. The tubes slide very easily along the wires and during the placement process they could have shifted to one side or the other. The tubes will need to be moved back to the center of each wire. Believe it or not, a good way of doing this is with an xacto-knife. The back edge of the knife is very fine but not quite sharp enough to cut the wire and can be used to push the tube along the wire. Notice that since each wire has been staggered, the center of each wire does not line up with the center of the detector. The wire's center is slightly to the left or the right of the detector center by a few inches depending on the direction of the stagger. When repositioning the tube, try to align its center with the wire's center, not the detector's center. Perfect alignment is not crucial because the wire and tube will be moved and adjusted later anyways, but this is still a good practice.

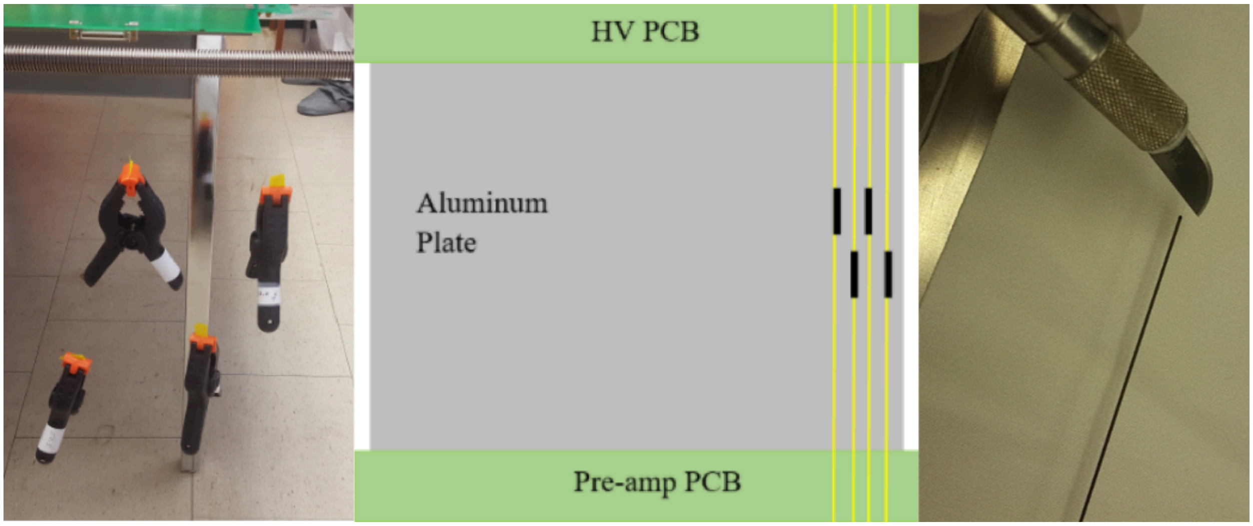


Figure 29. Staggered clamps and a diagram of strung carbon tube sense wires placed at the edge of the detect slightly off center as well as a demonstration of the carbon tube positioning technique using an xacto-knife.

Painting the Tubes: Once the tubes are strung across each wire they need to be painted at their ends. The purpose of the paint is to act as an adhesive and fix the tubes in position on each wire. Obviously, the tubes should not be able to slide along each wire once it is soldered. If this were the case then all of the tubes would just fall to one end of the detector once it was erected. Additionally, the paint also serves as a conductor between the wire and each tube. The type of paint used is carbon based and will conduct electricity. The carbon comes in the form of very fine grit and must be mixed thoroughly into the paint immediately before application. Be sure to place polyester rags beneath each tube to catch any paint incase it spills or drips.

It does not take a lot of paint to adhere the tubes, just a dab on the top and bottom of the wire at either end of the tube. Similar to solder joints, these dabs should be round and smooth but achieving this can be challenging. The paint is applied with the tip of a fine paintbrush, meaning that steady hands are crucial. The best technique is to rest ones wrist on the G10 frame for stability and hold the paintbrush as one would a pencil. Fill a small plastic cup with stirred paint and hold it with the other hand. Then, quickly but gently dip the paintbrush tip and dab the end of the tube. The paint must be applied quickly because it develops a film in less than five seconds. This film is the primary cause of irregularity in the paint dabs. Because of the film development and carbon settlement, one must reset after each dab and prepare for the next. The paint needs to be re-stirred before each dab, especially the very first one, and either the paintbrush needs to be cleaned with paint thinner or another fresh brush needs to be used. Frequently, the dab will be irregular or there will be an extension of paint along the wire. If either of these qualities are too extreme the entire wire and tube needs to be replaced because there is no way to fix them. Unfortunately, this means cutting and sanding a new tube and restringing it. Since each tube requires four dabs it is likely that at least one of them will be too imperfect and the wire and tube will need replacement. This is why constructing the central region is by far the most challenging portion of the wire stringing process. Although perfection should be the goal, keep in mind that no dab will be perfect, especially not all four of them; it is just too difficult to achieve. Do your best constructing this section, but realize that some quality standards may have to be sacrificed and it is okay. Once painted, give it a full twenty-four hours to dry. After this time, test that the adherence was successful, again, with the back edge of an xacto-knife. Push the tube with the same amount of force used to slide the tube along the wire before it was painted; this is not a lot of force. If the tube resists the push and does not move then the adherence was successful.

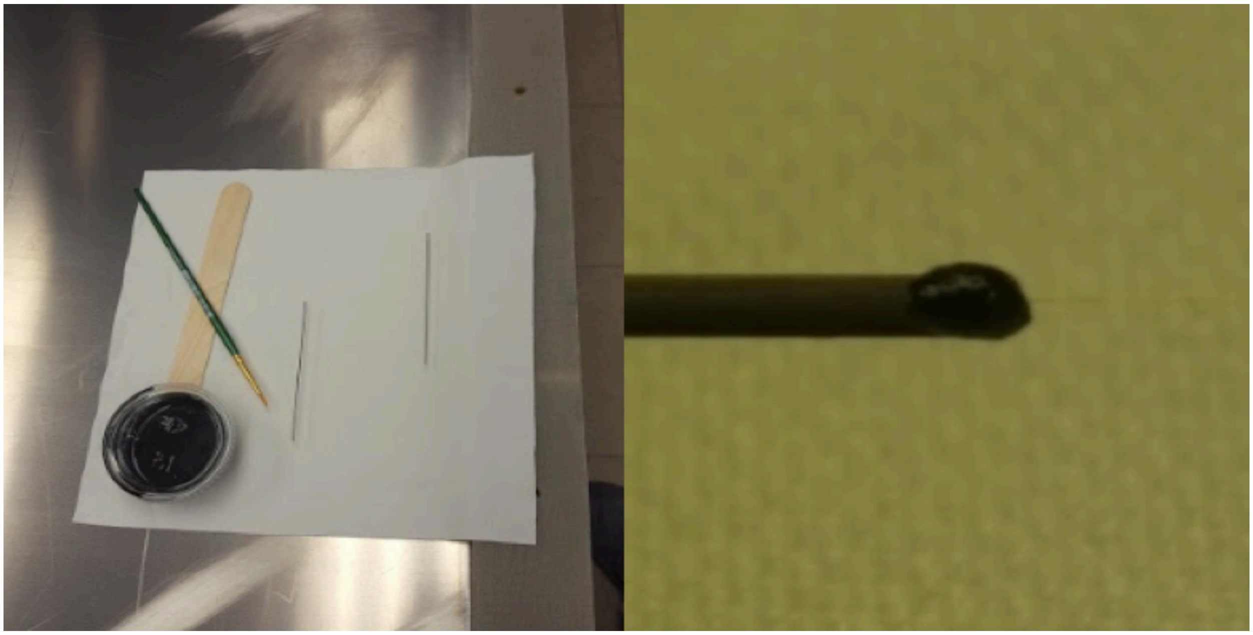


Figure 30. The setup designed to paint wires most efficiently with a fine tipped paintbrush and small container of carbon paint as well as an example of an optimal carbon tube paint job.

Transferring Tubed Wires: There is actually nothing different from using the transfer bolts to move a painted tubed wire than there is with just a regular wire. In fact, the idea of using the transfer bolts was developed from brainstorming a way to move painted tubed wire while maintain the tension across them. It was thought that reducing the tension of a wire could potentially break or loosen the paint joints adhering the tubes. Therefore constant tension was sought and transferring the wires with the transfer bolts provided a way to do this. This method was found to be so effective that it was adopted for the use of transferring all wires. For this method, as always, just lift the portions of dangling wire simultaneously with the bolts until they are parallel with the acme rods, raise the wire off the acme rods, translate the wire over the detector, and lower it into position. I suppose for this specific instance one should be even more cautious not to jerk the wire and break the paint joints but the mechanics of the transfer process is identical. Additionally, try to position the wires such that each carbon tube is centered with the detector to make the circular pattern.

Centering tubed wires: Once the tubed wires are in the correct acme rod thread the tubes will need to be precisely centered with the detector. This is accomplished by sliding the sense wire through the threads to move the tube in either direction perpendicular to the rods. The best way to do this is to gently push up on one of the dangling clamps and relieve some of the tension on that side. The weight from the clamp on the other side will pull the entire segment in that direction. Be extremely gently here, in many instances the tube will need to be moved less than a millimeter and the targeted position can easily be overshot. The correct position of each tube is given by the Carbon Tube Position Map seen in Figure 31. The map gives the distance in centimeters from either the Preamp Board or the HV Board to the end of each tube. The best way to physically mark this position on the detector is to use a guide wire. This is a clamped field wire that stretches across the detector perpendicular to the stringing direction and parallel with the Preamp and HV boards with clamps dangling from the G10 frame. The guide wire can be positioned a specific distance from either Board using the metal meter sticks taped to the G10

Frame. Then, the position of the guide wire can be used as a marker for the correct position of a specific carbon tube. Basically, the end of the carbon tubes should just barely line up with the guide wire when looking down the length of the wire. The position of the guide wire is then adjusted accordingly to the next tube to be placed. Since the guide wire rests on the metal meter stick, the guide wire hovers just about a millimeter above all the strung wires and does not interact with them in any way. Be sure that tension is maintained across the guide wire at all times so it does not sag and touch the strung wires. Once a tubed wire is correctly centered it can be soldered to the electronics pads and the excess wire cut off to finalize it.

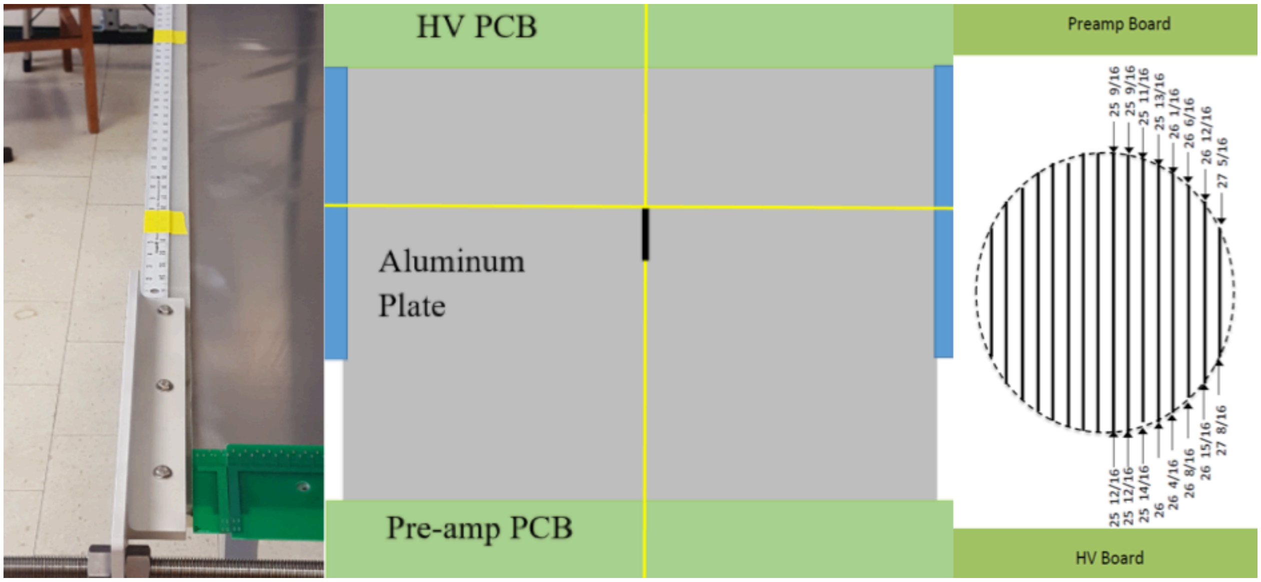


Figure 31. A diagram of the carbon tube positioning technique with guide wire as well as the carbon tube positioning map.

5. Conclusions

i. Synopsis of Work Completed

A lot has been accomplished in the Medium-Energy Nuclear Physics Lab in the past year pertaining to the CPP Experiment. Regarding the CPPSIM, a dead region and angular cuts were implemented as well as tests of many new MWPC mother volume geometries. Such tests varied the number of MWPCs and the thickness of interleaved iron absorbers. In addition, several event files were added background and run through the simulation in an attempt to determine the most realistic event files for muons and pions. As far as MWPC construction, the first prototype MWPC to be used in the CPP Experiment was completely strung. This was an incredible milestone as its construction posed a significant learning curve.

ii. Current Status and Remaining Work

In terms of the CPPSIM, new geometries still need to be tested and background needs to be established into realistic event files. It is believed that the current benchmark geometry can be outperformed; it is just a matter of determining what specific geometry will do this. As far as MWPC construction, the first prototype is still not complete. Its electronics need to be finished and then it can be filled with gas, sealed, and transported to Jlab. If the prototype is deemed successful then construction of an additional number of MWPCs satisfying the optimized geometry requirements will begin.

6. References

[1] A. Deur, T. Whitlatch, E. Smith, A. Somov, L. Pentchev, B. Zihlmann, M. Ito. “Summary of Hall D Subsystems and Parameters”, May 2014.

[2] CERN Document Saver. “The Standard Model: The Standard Model explains how the basic building blocks of matter interact, governed by the four forces”, 2017.

[3] C. Haughwout. “Design and Prototyping of MWPC Preamp Electronics for Use in the Charged Pion Polarizability Experiment at Jefferson Lab”, Honors Thesis, University of Massachusetts Amherst, May 2015.

[4] D. Lawrence, R. Miskimen, E. Smith. “Measuring the Charged Pion Polarizability in the $\gamma\gamma^* \to \pi^+\pi^-$ Reaction”, Presentation for JLab PAC40, 2013.

[5] Fermilab, Office of Science, United States Department of Energy. Particle Data Group. Image: Standard Model of Elementary Particles.

[6] Griffiths, David. "Introduction to Elementary Particle Physics". Weinheim: WILEY-VCH Verlag GmbH & Co. KGaA, 2010. Second Revised Edition.

[7] R. Johnson. "The Charged Pion Polarizability Experiment at Jefferson Lab: Detector Instrumentation and Simulation", Honors Thesis, University of Massachusetts Amherst, May 2017.

[8] R. Miskimen, et al. "Measuring the Charged Pion Polarizability in the $\gamma\gamma^{*} \rightarrow \pi^{+}\pi^{-}$ Reaction", Proposal for JLab PAC40, 2013.

[9] S. Agostinelli, J. Allison. "GEANT4 - A Simulation Toolkit", Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors, and Associated Equipment. July 2003.