

# Developing a Compact Multi-Wire Proportional Chamber for Cosmic Ray Detection

Sahar Khashayar; Ike Qian; Alexander Moscella<sup>1</sup>; Andrew Schick<sup>2</sup>; Prof. Rory Miskimen, PhD<sup>3</sup>

<sup>1</sup>University of Massachusetts, Amherst  
<sup>2</sup>University of Massachusetts, Amherst  
<sup>3</sup>University of Massachusetts, Amherst

## Introduction

An MWPC, or Multi-Wire Proportional Chamber, is a device used in particle physics that generates electric fields and can detect particles. The MWPC consists of several parallel wires, alternating between field wires (kept at ground) and sense wires (high voltage). It is filled with a gas, usually Argon, and when a charged particle travels through, it leaves a trail of dislodged electrons. The path of the particle can therefore be determined by the current running through the sense wires.

This research project aims to create a functional mini-MWPC that is accessible for civil applications, as well as create a program that can analyze the data collected by the detector. The program should be able to help differentiate between signals gathered by the detector, and analyze which signals are simply noise and which are actual data. If successful, the program will also be used in the lab's future.

## Building the Detector

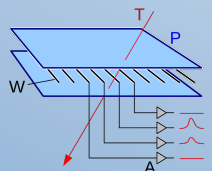


Figure 1. Diagram of the wire chamber

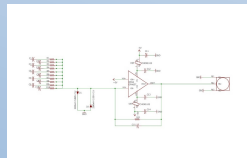


Figure 2. Schematic for the printed circuit board

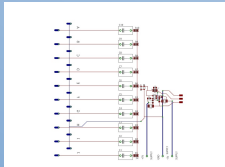


Figure 3. Board layout for the PCB

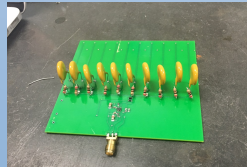


Figure 4. PCB with all components soldered on

The printed circuit board, or PCB, design is the most electronically complex component of the particle detector. Since the signals produced by particles are incredibly small, they must be amplified – we accomplish this through the use of an operational amplifier.



Figure 5. Stringing the wires across the detector

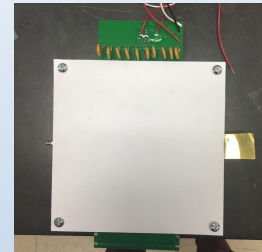
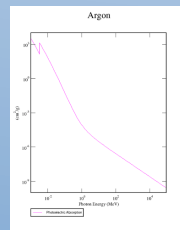


Figure 6. Sealed and completed detector

## Building the Detector (Cont.)

After calculating the attenuation of X-Rays through Styrene (62.64% of cosmic rays would penetrate), we decided to use this material to seal the detector. We strung the wires across from high voltage board to our PCB, painted the styrene with conductive paint, and sealed it up. Unfortunately, when hooked up to gas, it was discovered that the circuit was drawing current. From there, the detector went through lots of troubleshooting, and we found out that after printing, the ABS frame had absorbed charged ions from the solution meant to dissolve the 3D print support material. A new frame was printed and an entirely new detector built from scratch.

## Calculations



Graph 1: Photoelectric absorption of argon

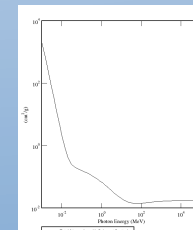
The formulas we use for the amount of the X-rays penetrating through a material is:

$$I_p(x) = I_0 e^{-\mu x}$$

$$\lambda = 1/A \rho$$

Efficiency = 39.9%

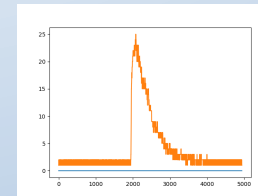
$I_0$  = number of x-rays  
 $x$  = thickness of the media  
 $\lambda$  = attenuation decay  
 $A$  = total attenuation with coherent scattering at certain energy level  
 $\rho$  = density of the media



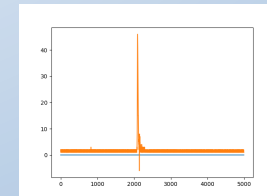
Graph 2: X-Ray attenuation of styrene

## Data Analysis Program

The code we created would have to be able to take in data, separate it into individual signals, and then determine which were noise and which were actual data.



Graph 3. Example of a Cosmic Ray Signal



Graph 4. Example of a Noise Signal

The data was collected using ScopeOut, a program developed in the Miskimen lab. The detector reads data from a photomultiplier tube, or PMT, a high-sensitivity device for the detection of electromagnetic waves. The graphs above depict examples of waveforms collected in ScopeOut; the first is an actual cosmic ray signal, while the second is noise produced by the PMT. Our program, written in Python, imports all the data collected in ScopeOut as a matrix, separates the data into separate matrices, then integrates the data. From there, it will take the ratio of the maximum height of the waveform to the integral, and use this ratio to determine the validity of the signal.

```
for j in range(0, nWaves):
    beginWave = waveStartIndex[0][2*j]+2
    endWave = waveStartIndex[0][2*j+1]
    l = j + 1
    Wave = str("Wave" + "%s" % l)
    setattr(A, Wave, A[beginWave:endWave, 0:2])
    f = A[beginWave:endWave, 0:2].astype(float)
    A.Wave = f[beginWave:endWave, 0:2]
    np.array(A.Wave)
```

Figure 8. An excerpt of the code to chop up the data

## What's Next?

There are several improvements that can be made to the design of our MWPC. First of all, some of the traces on our PCB board leave room for charge buildup, and the PCB's connector is very fragile. Our Python program still needs cleaning up and finishing before it can be used in the lab

## Contact

Sahar Khashayar<sup>1</sup>

Ike Qian<sup>2</sup>

<sup>1</sup>Email: saharastro@gmail.com

<sup>2</sup>Email: qqian1@student.gn.k12.ny.us

## Acknowledgements

1. Rory Miskimen – professor in charge of the project
2. Andrew Schick, Alexander Moscella – others working in the lab
3. Erin O'Rourke – organized the research program
4. Sahar's parents – Mojgan and Amir Khashayar
5. Ike's parents - Henry Qian and Yun Yao