Neutrino Detector R&D

Adam D. Schneider

January 29, 2004

Contents

1 Introduction

1.1 Neutrino History

In 1930, Wolfgang Pauli first hypothesized the existance of a tiny neutral particle as a means of explaining the apparent non-conservation of energy and momentum in nuclear β-decay that had recently been encountered. This hypothesized particle was considered to be massless (or of mass so small that it was by far undetectable) and was named the 'neutrino' by Enrico Fermi, meaning 'little neutral one'. The first evidence for the existence of of a neutrino did not come for another 26 years. This had a tremendous impact on particle physics and the development of the Standard Model. By 1989 it had been confirmed that there are in fact three types (or flavours) of neutrinos: electron, muon and tau neutrinos.

As the neutrino began filling in the gaps of the latest theories, it was calculated that the sun was in fact producing phenomenal amounts of neutrinos every second. More specifically it is electron neutrinos that are produced by the sun, and conveniently these were the easiest type of neutrino to identify. It was in 1964 in the chlorine detector of Ray Davis, located in the Homestake Mine, that the first solar neutrinos were detected, but there was a problem. The number of electron neutrinos observed was drastically less than that predicted. This 'Solar Neutrino Problem' prompted the idea that the neutrinos may be changing type on route to the earth (flavour oscillations). By 2001, a comparison of data from the Sudbury Neutrino Observatory (SNO) and Super-Kamiokande (Kamioka Nucleon Decay Experiment), had strong evidence that solar neutrinos do in fact oscillate and in turn must have some finite mass. There was also evidence of neutrino oscillations coming from the nuclear reactor neutrinos in the KAMLAND experiment.

By 1999 the worlds first long-baseline neutrino experiment had begun in Japan, in hopes of more accurately defining the characteristics of the neutrinos and their behaviour. K2K (KEK to Kamioka) used the KEK synchrotron to generate a neutrino beam that would travel hundreds of kilometers to the Super-K detector in Kamioka. The K2K project was logging enough events to confirm the neutrino oscillations, but to get more information about the neutrino a more powerful synshrotron will be needed. This synchrotron is now being replaced by one producing 50GeV protons and will be constructed at Tokai, giving way to one of the newest and largest scale neutrino projects in history-T2K.

1.2 The T2K Project

The T2K project begins with the new proton synchrotron in Tokai and ends at the Super-Kamiokande 50kTon water Cerenkov detector in Kamioka. The proton beam is fired at a target which spews out pions along with a bunch of other debris. Magnetic focusing lenses called 'focusing horns' are used to focus the pions into a beam, leaving the rest behind. The pions then move down a decay tunnel where they decay into muons and their counterparts, muon neutrinos. There is then a stopping block which allows only neutrinos to pass through, giving us our muon neutrino beam. The beam imediately

passes through a near detector giving us initial readings from the beam, before travelling some 295km to the Super-K far detector.

The Super-K detector is a gigantic cylindrical tank, walls lined with 11,200 photomultiplier tubes, and holding 50 kilotons of water. The near detector consists of a smaller version of the far water Cerenkov detector (using the same type of detection reduces systematic errors), as well as a more fine grained component to gain more detailed information of the particles' behaviour. The research and development of this fine grained detector has been assigned to the Canadian Neutrino Group at TRIUMF (located in Vancouver BC). Designs for this detector involve layers of water and scintillator, and areas of research include the submersion of scintillator in water for extended periods of time, and the relative light output gain from the use of different wavelength shifters.

1.2.1 Scintillator

Scintillators can be made with various organic and inorganic materials, taking on either liquid or solid form. The feature of a scintillator is that as radiation passes through it, its molecules get excited to higher energy states and then release radiation of their own as they return to ground states. This radiation is in the form of light, ususally near the visible spectrum. One important characteristic of a scintillator is that it is transparent to its own radiation.

1.2.2 Wavelength Shifters

A wavelength shifter is a chemical whose molecules absorb light of one wavelength (and in turn one colour) and emitt that light at another wavelength. This can be very usefull in rendering invisible light (relative to the observer) visible.

1.2.3 Photomultiplier Tubes

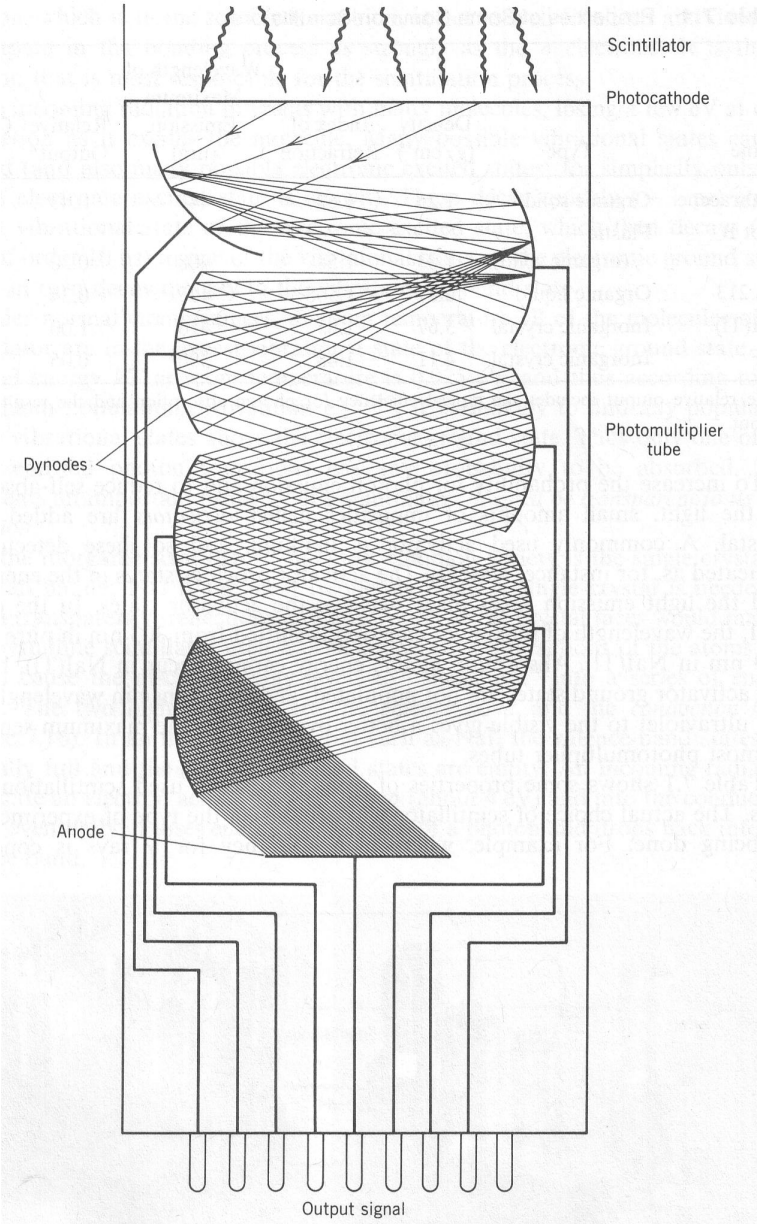

Figure 1: Photomultiplier Tube (PMT) Schematic from Krane, Introductory Nuclear Physics

The photon signal incident to the PMT (figure 1) strikes the photocathode, releasing photoelectrons (fewer than the incident photons). These electrons are then focused down a series of electrodes called dynodes. These dynodes are powered by an external voltage strong enough that each electron that hits will cause the release of several more. Thus the signal is multiplied as it travels down the chain of dynodes. One key feature of a PM device is that the gain is directly proportional to the voltage applied, and that this gain is constant as long as the voltage is stable.

1.2.4 Cerenkov Radiation

When a particle enters a medium at speeds greater than the local speed of light in that medium, something like a sonic boom occurs, and radiation is released in the shape of a cone. The majority of this light is in the ultraviolet spectrum, but some is in the neighbouring part of the visible spectrum.

2 Experiment 1: Scintillators and Wave Length Shifters in Water

2.1 Objective

By monitoring the light output from a system of wavelength shifter (WLS) fiber and scintillator, we can observe the relative effects of adding water to the system. More importantly, by automating the data retrieval, we can see the long term effects of having scintillator submerged in water.

2.2 Apparatus/Setup

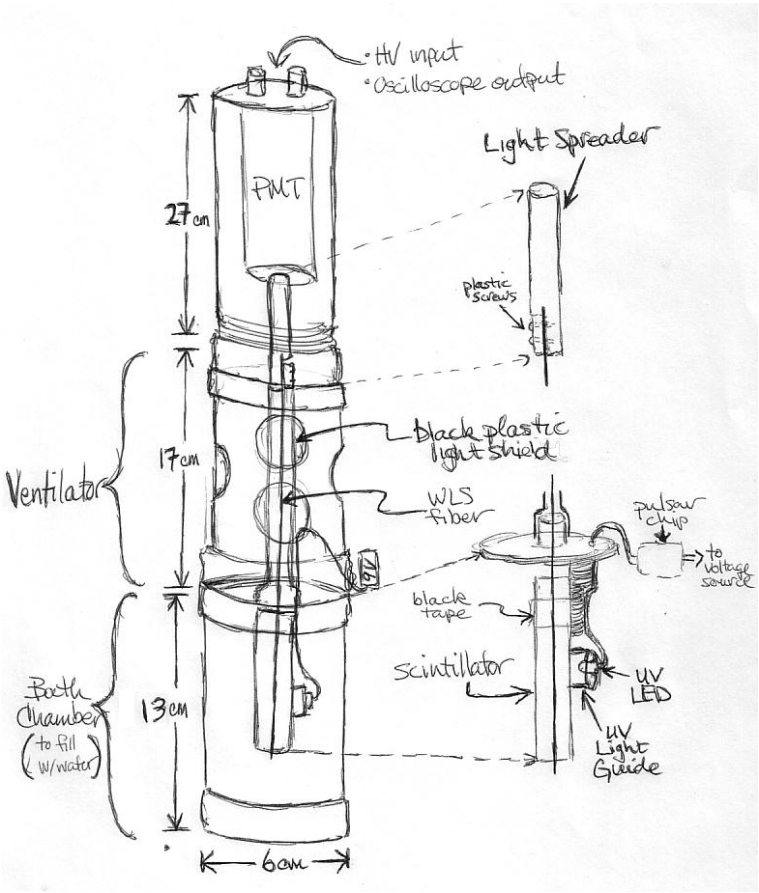

Figure 2: The water test apparatus.

The apparatus consists of a rectangular piece of scintillator with a small (nearly cylindrical) hole (1.5-2.0mm) running lengthwise through it and a UV LED embedded in its side. The sides of the scintillator are coated with a white reflective material, and in the second run of this test the ends were covered in black tape. On one side, a circle of the white coating was cut away where a cylindrical light guide was fixed. The outside of this light guide was covered in black tape and a hole in the end was hollowed out for the LED to rest. The hole was filled with an optical grease so that the LED would fit snugly and not move around throughout the course of the testing. The scintillator is fixed to and

housed inside a cylindrical black plexi-glass bath chamber. A wavelength shifter fiber runs through the scintillator, out of the bath chamber and is attached to a cylindrical light spreader which is mounted flush with a photomultiplier tube (PMT). No optical grease was used in this interface to ensure consistency and repeatability in the event of reassembly. The LED is powered by a 9V voltage source which is run through a pulser chip, sending pulses of UV light into the scintillator. The hole in the scintillator is slightly larger than the fiber, allowing an air gap in the interface. Therefore, when water is added to the bath chamber, the gap fills with water creating a scintillator-water-fiber interface. The signal then runs up the fiber to the PMT which is powered by a variable high voltage (HV) source and monitored by an oscilloscope. From here we can calculate the pulse's area and amplitude.

2.3 General Overview of Procedures

After the initial setup of the apparatus was complete and a steady signal was established, we began testing the reproducibility of the signal. This was initially tested simply by switching the HV off and then on again. Further testing was done by powering off, disassembling and reassembling the apparatus and then powering back up. The level of reproducibility observed here prompted the rotation tests in hopes of narrowing down the cause of our signal's variations. Tests were also conducted exploring the signals dependancies on LED voltage and room temperature. After narrowing down and stabilizing all suspected factors affecting our signal, we were ready to add water to the bath chamber. In the first run of this experiment all data was recorded in a logbook by hand, but in the second run an automated data acquisition system was setup so that the long term effects of having scintillator submerged in water could be monitored.

2.4 Setup

First of all we prepared a length of fiber. The fiber was cut from a large supply stored on a wheel. The ends of the fiber were then sanded by clamping the fiber into a block, with the end just coming out of the block. Using a #300 or #400 grade piece of sandpaper, the end is sanded to be flush with the block. A technique of scrubbing the fiber in all directions as evenly and randomly as possible was used to ensure a consistent surface and optimal light diffusion pattern. The fiber should be scrubbed lightly to avoid chipping the outer coating (extreme bending of the fiber should also be avoided).

Next, we ensured that the apparatus was assembled correctly. The plastic tubing should all fit together snugly. The WLS fiber should be all the way inside the light spreader when clamped down, and the light spreader should fit flush with the PMT. The PMT is spring loaded and must be compressed against the light spreader for the tubing to screw together. The PMT HV source was set to -1200V. This causes our signal pulses to be negative as well (most references to increase or decrease in signal refer to its magnitude). The LED voltage source was 9 volts, originally powered by a single 9v cell. The actual output voltage of the cell was too inconsistent, which lead to its replacment by a variable (but

consistent) voltage source whose output could be constantly monitored.

The HV source must always be switched off when the apparatus is disassembled to ensure that the PMT is not damaged by high currents caused by exposure to ambient room light. Consequently, when powered on, the PMT requires approximately 30 minutes to warm up and settle down to a stable signal. The signal can then be read off an oscilloscope set to display the signal's integrated area and peak amplitude. In the first run of this experiment, these readings were taken manually from a scope and recorded by hand in a logbook. Readings were taken approximately every half an hour along with a measurement of the LED voltage and the room temperature and humidity. In the second run of this experiment, an online data acquisition system was set up which takes a signal reading five times every half an hour(throughout the evening), along with two temperature readings.

2.5 Reproducibility Tests

Once it was clear that turning off and on the HV source did not change the stabilized signal, it was suggested that the apparatus be completely disassembled and reassembled. It was then disassembled and reassembled exactly as it would be when the water was to be added. Seeing how much this changed the signal would allow us to have a better idea of how much of an effect the water itself had. In this procedure the bath chamber was removed from the ventilator and opened, but the fiber was not disconnected from the light spreader and the spreader was not disconnected from the PMT.

Repeating this several times, it was found that although the signal stabilized in 20-30 minutes each time, the area and amplitude of the signals varied over a range of about 10%. After much consideration, these results prompted the rotation tests.

2.6 Rotation Tests

In this test, the PMT and consequently the fiber itself were rotated with respect to the scintillator/LED fixture. As the fiber had a slight curve in it from being stored on a wheel, its positioning inside the scintillator changed throughout this rotation. The results of these tests (one trial shown in figure 3) clearly indicate a signal dependency on the rotational position of the PMT (fiber position) with respect to the scintillator and LED.

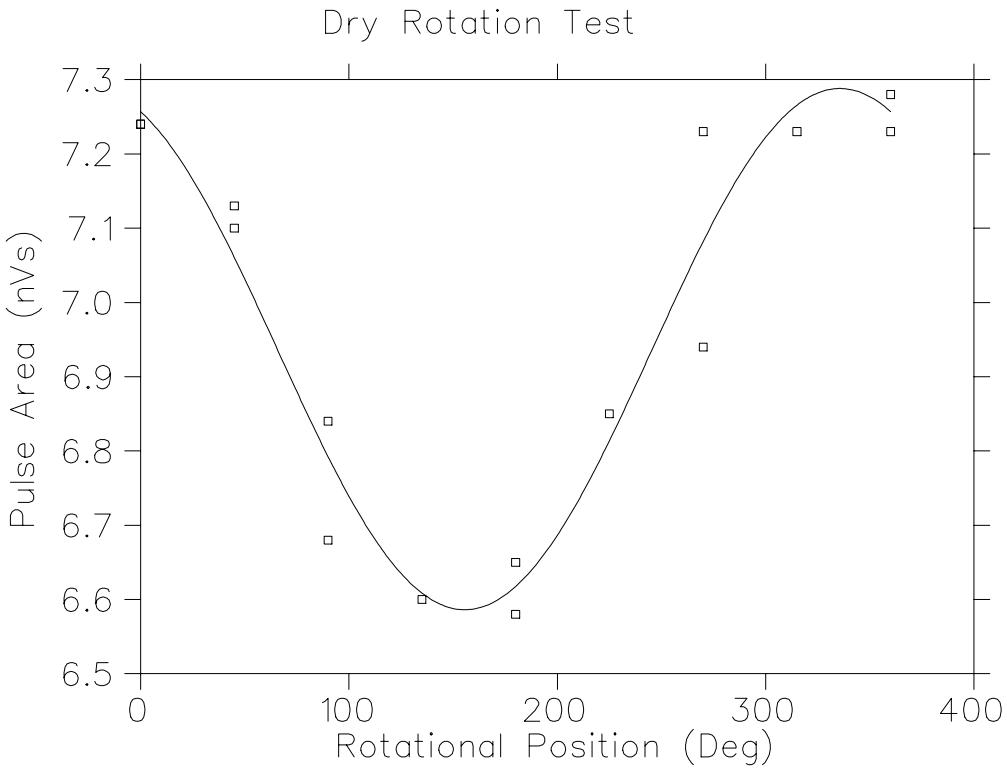

Figure 3: Signal Area vs Rotational Position

This test was repeated several times (all with no water) and each time similar behavior was observed. In some of these cases however, after a rotation of 360 degrees, the signal would not return to exactly where it had originally been. After a further rotation of 360 degrees, the signal would sometimes stray further from its original position. It seemed that after continued rotations this deviation would stop, suggesting that maybe the fiber was binding with the sides of the hole causing it to twist, taking a few rotations to reach an equilibrium. Although this behaviour was never adequately explained, keeping the rotational position constant in reassembling the apparatus reduced the signal variation to approximately 2%. This was agreed to be sufficient to move on to other factors.

2.7 LED Voltage Calibration

It was noted early on that the voltage output of a 9v cell was in fact not constant. Being in use for extended perdiods of time would cause its output voltage to drop, and unplugging it overnight would let it charge back up a bit. The instability of this voltage source motivated the use of a constant (but variable) voltage source. After replacing the 9v cell, some calibration data (figure 4) was taken in hopes of correcting our old data to 9 volts. This data was generated by varying the voltage from 9.0 volts to 8.4 volts and back up to 9.0 volts again. There seems to be a clear linear relationship in this range.

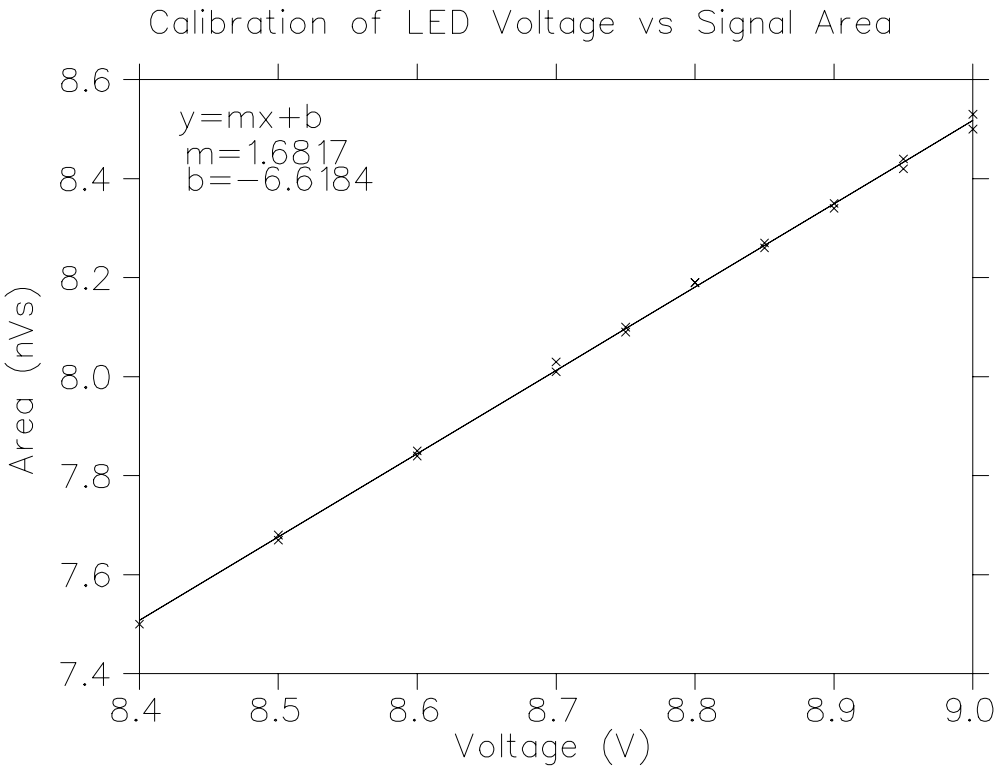

Figure 4: Signal vs LED Voltage.

2.8 Temperature Dependancy Tests

During the first run of this test we had no climate control and the temperature of the room would vary by about 5 degrees over the course of the day. This seemed to have a significant effect on the signal, and prompted us to move to a climatized room for the second run. From figure 5, we can see how the temperature fluctuated over time (since the automated temperature measurment was implemented). The spike at day 318 was due to turning the air conditioner off to test the signals dependancy on room temperature. Before this test, the only data in the signal versus temperature plot (figure 5) was the cloud of data points around 19-20 degrees. This seemed to suggest that there was no dependancy, but our previous experience with the first run suggested that there was. This is why the air conditioning was shut off. From the data at higher temperatures (20-22.5 degrees) it would seem that there is a dependancy, but this only makes it more difficult to understand the cloud at the lower temperatures. Although, with the temperature remaining between 19-20 degrees (with the AC on) the variation in signal is small enough that it is not a major concern.

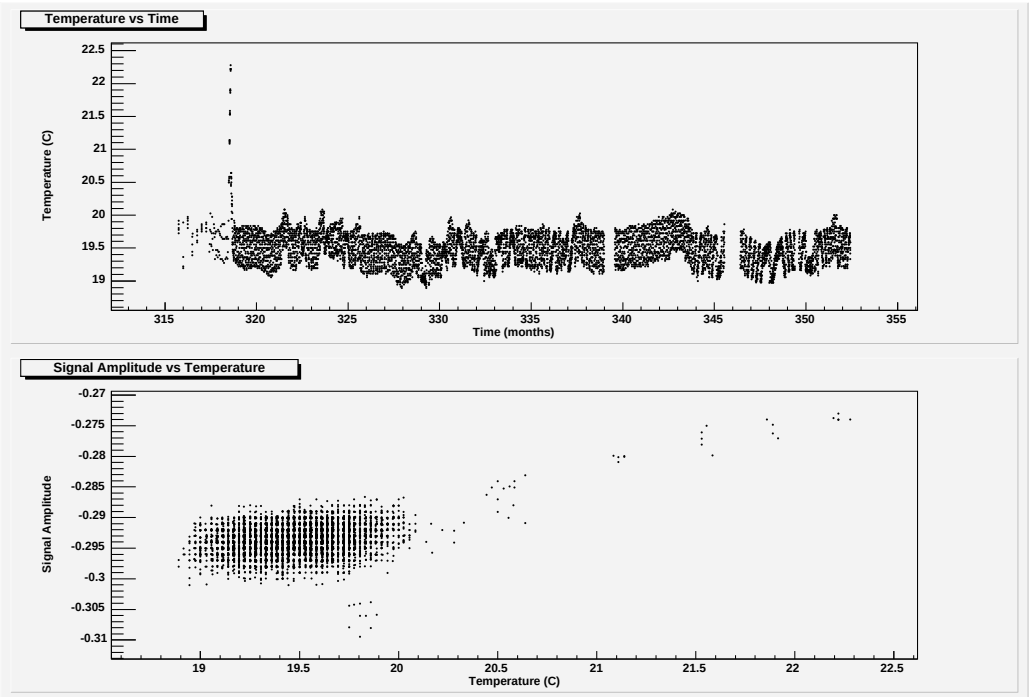

Figure 5: Temperature data

2.9 Water Addition and Long Term Results

After achieving several satisfying trial reassemblies with reasonably low signal variation (2-4%), we were ready to add water. The first time this was done we saw a signal gain of about 20%. Unfortunately, during the dissassembly procedures one of the pulser chip connections broke and needed to be resoldered on the fly. This did not likely have much of an effect on the signal, but was troubling none the less. In the second run, the only difference to the setup (other than the room change) was the addition of black tape over the ends of the scintillator. In this run a signal gain of at least 35% was seen. This is the discontinuity observed at day 303 in figure 6. This is almost double the signal gain originally seen! The results were actually rather surprising, as since the tape is black, one had expected any light originally lost would only be absorbed by the tape and not reflected back into the fiber. It is not clear that the black tape is the cause of this change.

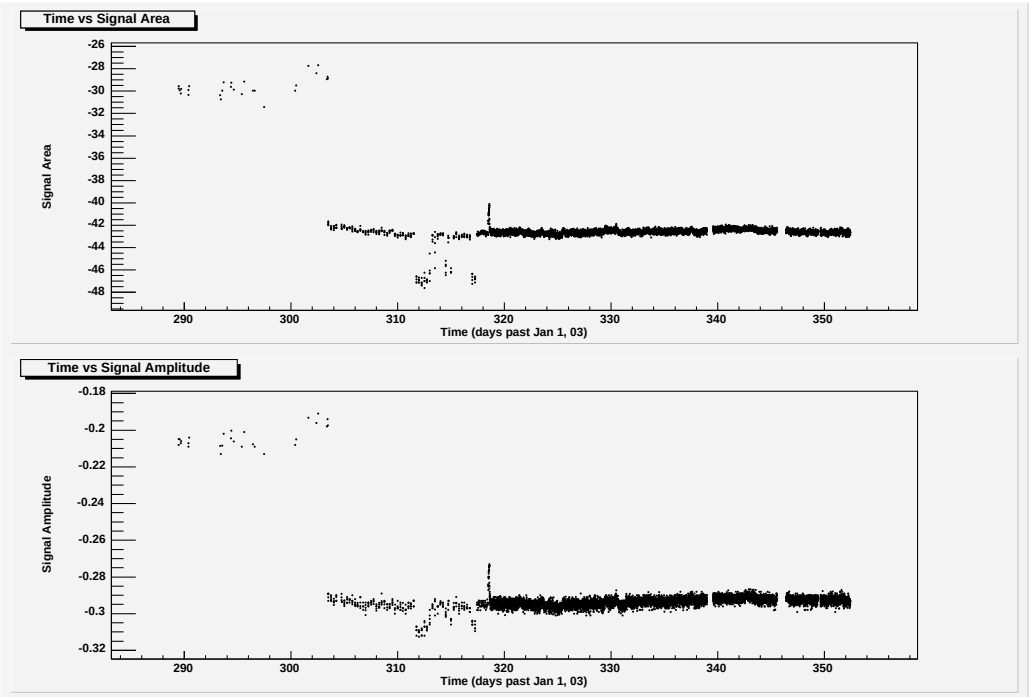

Figure 6: Water test plots of Signal vs Time.

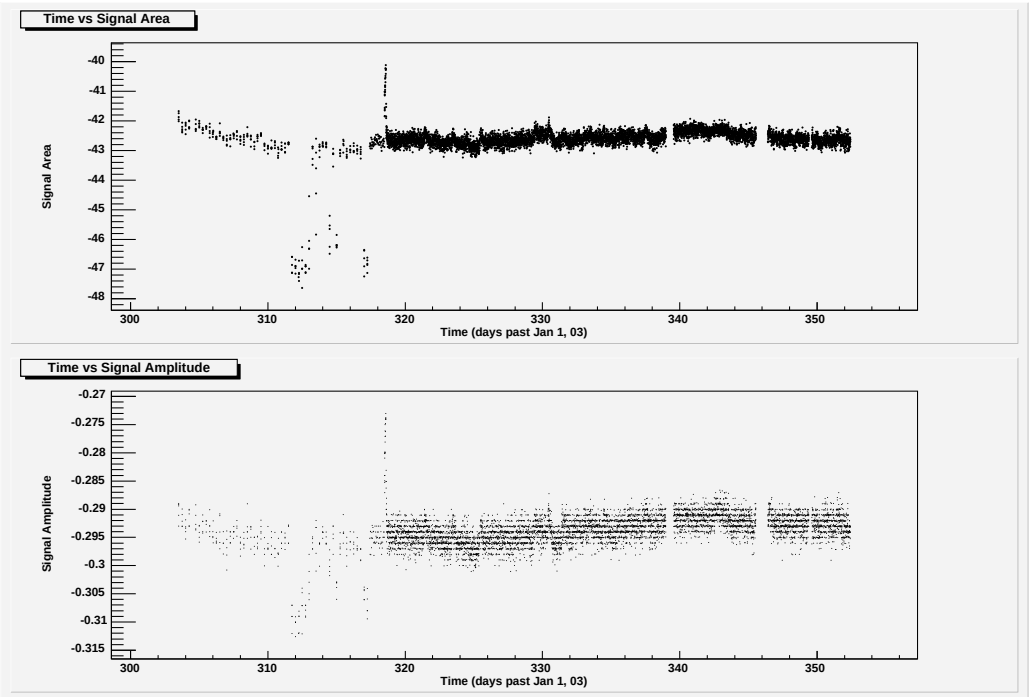

Figure 7: Water test plots of Signal vs Time.

After adding the water for the second time, the signal was continually monitored and recorded by an automated data aquisition system. Although for the first week, the signal seemed to be slightly increasing, after a month, the signal seemed relatively stable. This initial increase could have been due to air bubbles left in the scintillator-fiber interface, or the scintillator absorbing water. The signal will continue to be monitored for a few more months to see if any new trends appear.

There is one small anomaly that can be seen in figure 7. Between days 310-315, the signal to jumps up by about 10% and then jumps right back. This happened a couple times over a weekend when nobody was around to see what may have happened. We know that this was not due to temperature as there was a thermometer that remembered the most extreme values reached. We also checked to see if the cyclotron magnets had been switched off or on those days, but they had not. To this day we have no satisfying explanations or even suggestions as to the cause of this change. It has not occured since.

2.10 Automated Data Acquisition System

The addition of an automated data acquisition system has allowed us to continuously monitor the behaviour of this setup for an extended period of time. By having the oscilloscope connected to the TRIUMF network (detscope01.triumf.ca) we are able to collect a data file consisting of points tracing out the signal. We also have two temperature probes connected through a serial port, we were able to create scripts that collect this data. The computer was then programed to run these scripts every half an hour all day every day. The data could then be processed by another series of scripts run with ROOT analysis software.

2.11 Conclusions

From the results of the two runs of this experiment, it is clear that we get a stronger signal with water between the scintillator and fiber than with air. Increases from 20-35% have been observed, but what the factors are that might affect this value are difficult to determine.

After having the fiber and scintillator submerged in water for almost two months, there seems to be no evidence of signal deterioration. But, we would like to be convinced that this will remain true over the course of years. Further tests will eventually be done, where by heating the water we will be able to speed up the chemical reaction rate and see where things might be headed.

3 Experiment 2: Water Cerenkov Detector and Liquid Wavelength Shifter

3.1 Source Tests

3.1.1 Objective

By monitoring the light output of a small water Cerenkov counter irradiated by a source of 0-2.27 MeV beta particles (from a Sr-90 source), we can measure the relative light output gain from different concentrations of liquid wavelength shifter solutions.

3.1.2 Apparatus/Setup

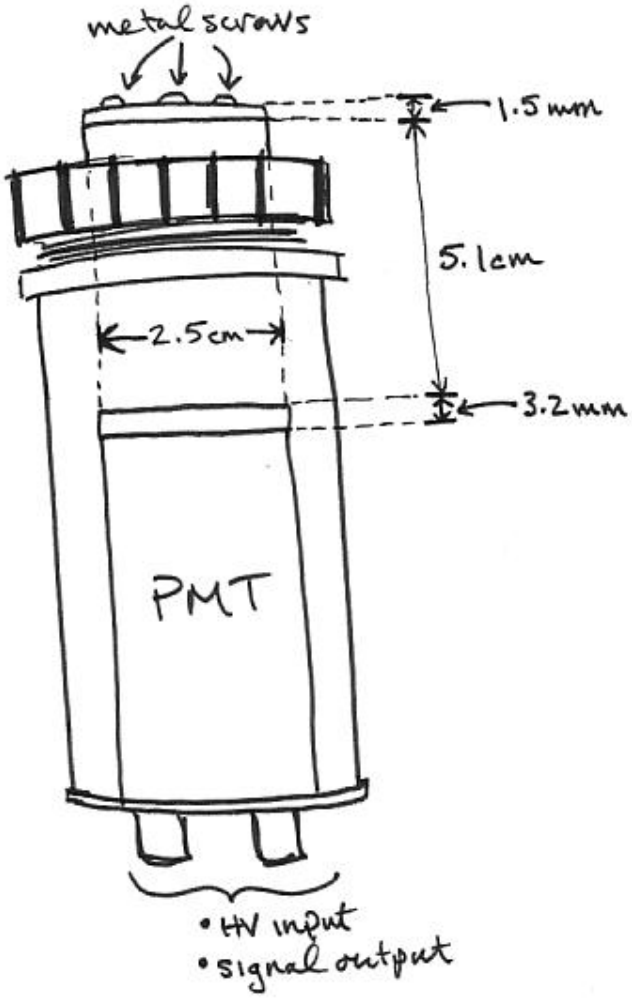

Figure 8: Water Cerenkov Detector.

Our water Cerenkov counter consists of a PMT with a small bath chamber made of black plexiglass mounted to the end, with an ultra-violet transparent

(UVT) plate interfacing the two. The chamber is light and water tight, and has a volume of 25 ml. A thin plate (thin enough for electrons to penetrate) made of black plexi-glass is attached on the top by metal screws, and can be removed for the solution to be changed. A beta source rests on top, directed downward into the bath chamber. The signal from this detector was monitored using an oscilloscope as well as a Camac and NIM logic system. A schematic of the data acquisition electronics can bee seen in figure 9.

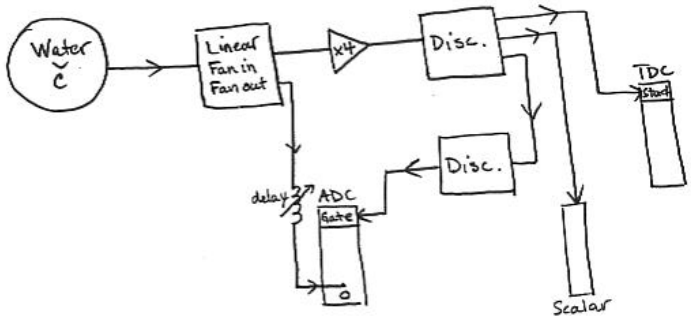

Figure 9: Camac Schematic for Source Tests.

The signal from the detector was split, running one signal into an ADC (analog to digital converter) and the other into a discriminator. The discriminated signal tells the TDC when each count started (alerting the computer software) and the scalar counts the number of hits. The discriminated signal was also run through another discriminator to stabilize the width of the gate that triggers the ADC. A computer running MIDAS data acquisition software was used to process the signals and record the data to disk.

3.1.3 General Overview of Procedures

After the initial setup was complete, the electronics calibrated and the data acquisition software configured, it was suggested that we use a reflector against the cylindrical walls of the chamber to increase the light output of our Cerenkov counter. Aluminum and Tyvek reflectors were both tested. Once an adequate signal was finally obtained, a series of 5 minute runs were made for each concentration of the Carbostyril 124 solution and each beta source. The concentrations 0, 1, 2, 5, 7, 10, 20, and 100ppm (by weight) were all tested with three different sources: a strong Sr-90, a weak Sr-90, and a Ru-106.

3.1.4 Calibration of electronics

For these tests the PMT was powered by -1900 volts and as before it required about 30 minutes to warm up after being opened. We began using the wavelength shifter Carbostyril 124 at a concentration of 100ppm (to ensure a visible signal). We then had to calibrate the electronics. An analog scope was used to see the signal well enough to set the threshold of the discriminator and set the delay so that the signal was within the ADC gate. The 4x multiplier was

used so that the threshold could be set just lower than the single photo-electron peak (which could best be seen using the analog scope).

3.1.5 Reflectors: Tyvek vs Aluminum

From comparing figures 14 and 15 with figures 11 and 16 respectively, we can see the increase in signal from adding aluminum and tyvek reflectors. It does not appear that Tyvek is significantly better than Aluminum for these purposes, but it is in general suposed to be a better UV reflector. The Tyvek was also easier to handle, and was used in the further concentration tests.

3.1.6 Concentration of Wavelength Shifter

From figures 10-12 we can see the relative increase in signal from the lower to higher concentrations of wavelength shifter. Although the increase is not huge, the signal does appear to be slightly larger using the higher concentrations. From figure 13, we can also see that with no source there is almost no signal, confirming that we are in fact 'seeing' beta particles.

The more interesting observations can be made from figure 16 which shows the signal using pure water with no wavelength shifter at all. Here we seem to have a signal whose integrated area is almost as large as with the wavelength shifter present. This would suggest that the frequency ranges of the Cerenkov radiation and the PMT sensitivity overlap more than had been anticipated.

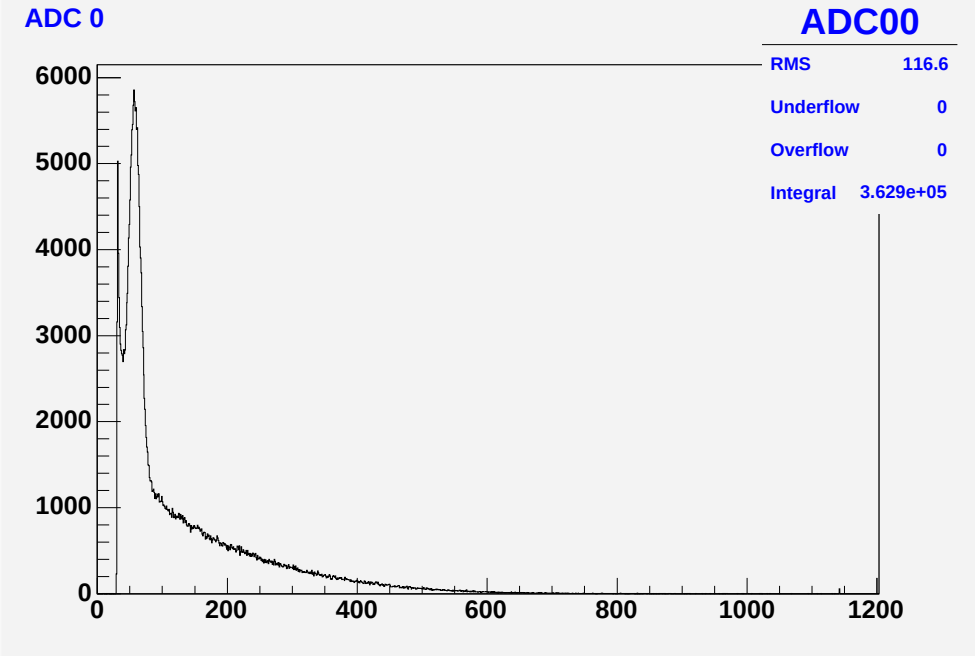

Figure 10: 100ppm Carbostyril 124 with weak Sr-90 (w/Tyvek)

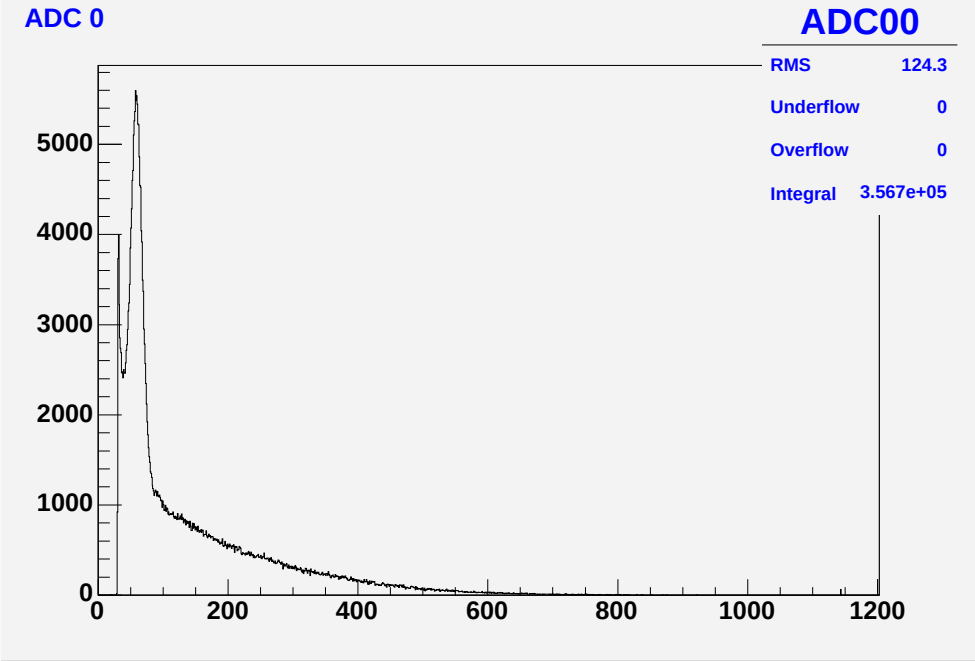

Figure 11: 20ppm Carbostyril 124 with weak Sr-90 (w/Tyvek)

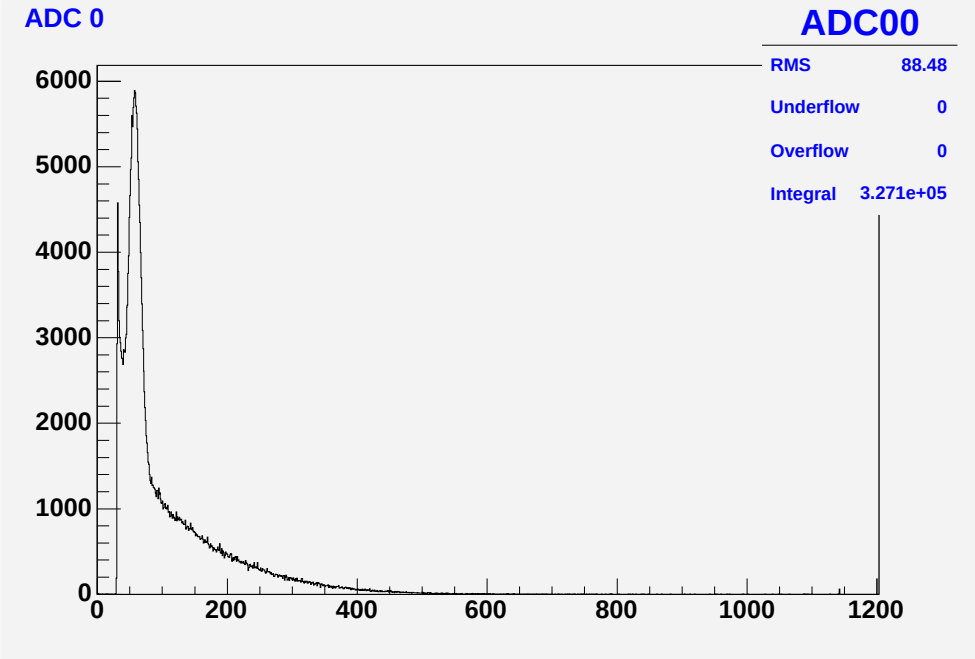

Figure 12: 1ppm Carbostyril 124 with weak Sr-90 (w/Tyvek)

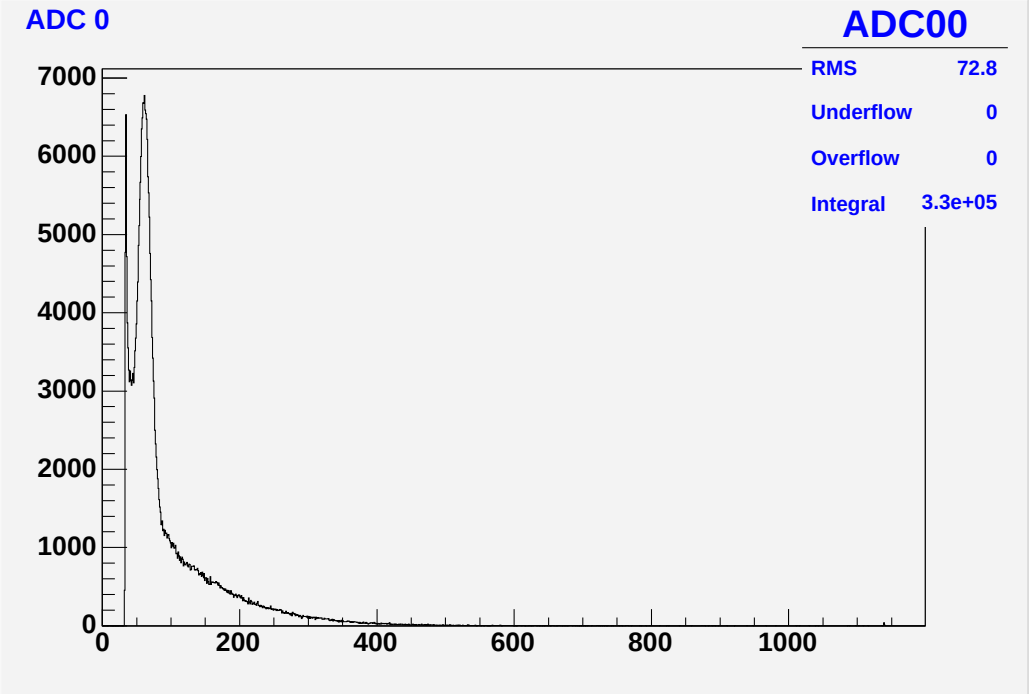

Figure 13: Pure water with weak Sr-90 (w/Tyvek)

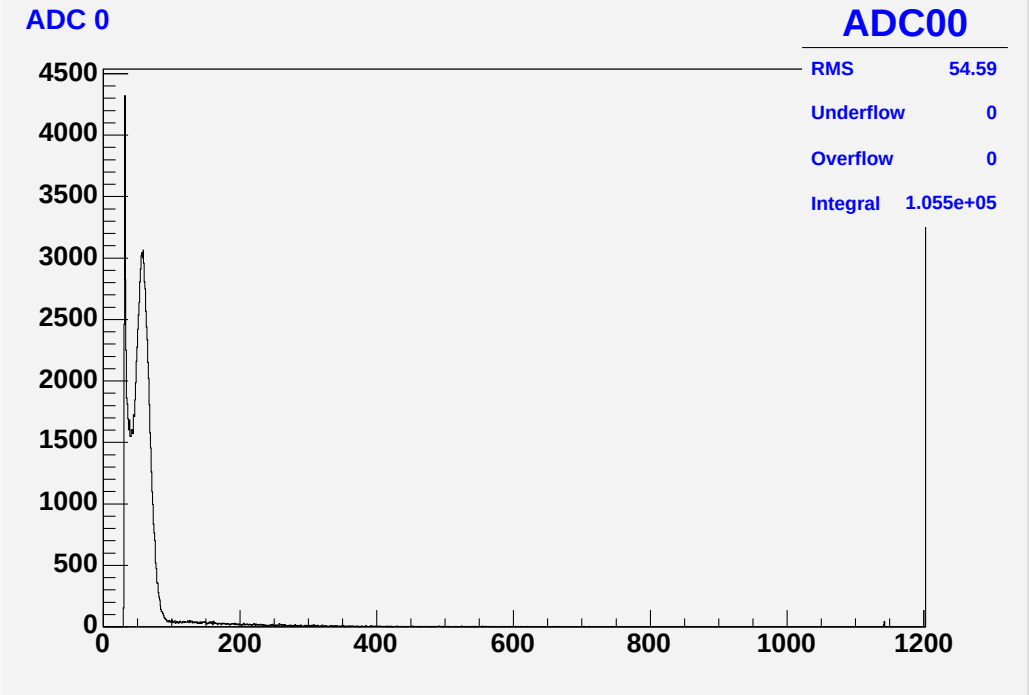

Figure 14: No source (w/Tyvek)

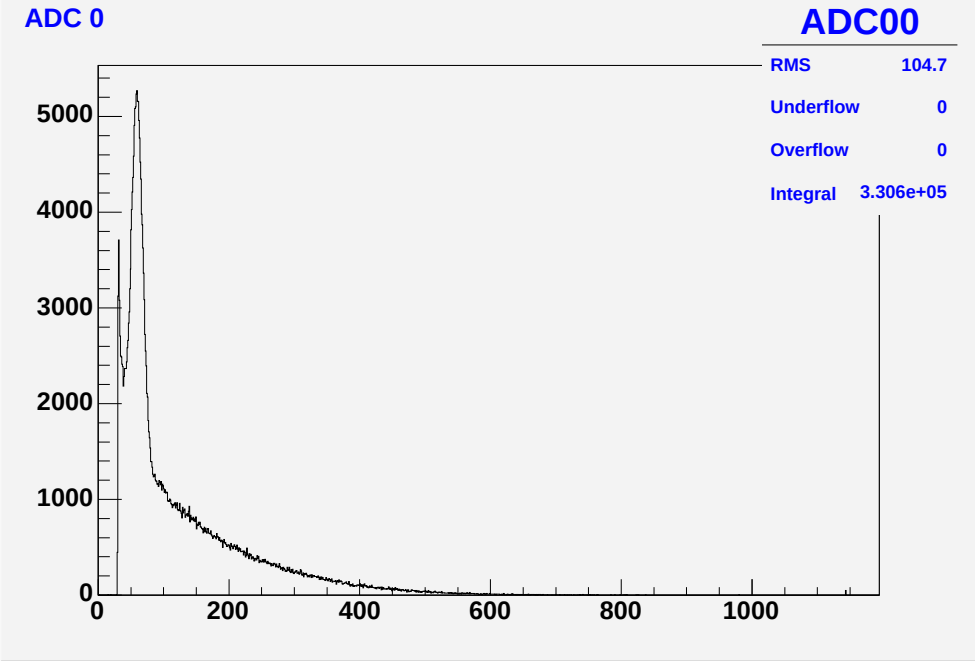

Figure 15: 20ppm Carbostyril 124 with weak Sr-90 (w/Aluminum)

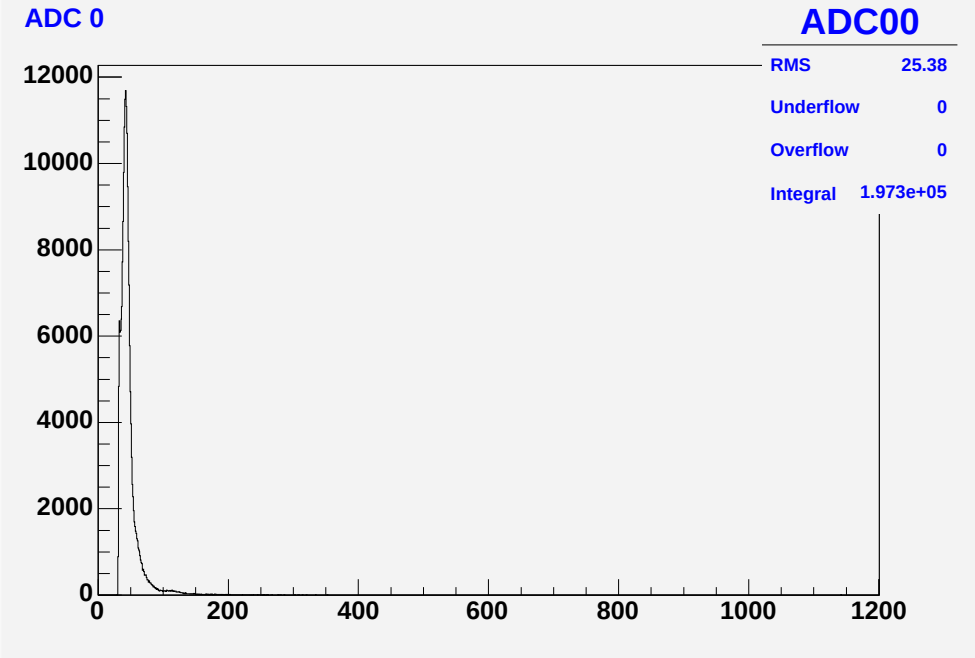

Figure 16: Pure water with weak Sr-90 (no Tyvek)

3.1.7 Conclusions

It is immediately clear that some sort of reflective coating is needed on the inner walls of the bath chamber to ensure that most of the light is shifted in wavelength and reaches the PMT. Adding the WLS solution does increase the light output and it would seem that very litle is in fact needed, as the largest gain is from 0-1ppm. Although, a surprisingly strong signal is recieved with no WLS at all. This suggests that the range of frequencies of Cerenkov radiation and PMT sensitivity overlap significantly.

It is difficult to draw any solid quantitative conclusions from this data, but we may be able to gain some more information by conducting further test with different WLS chemicals.

3.2 Beam Tests

3.2.1 Objectives

By placing a small water Cerenkov detector in a beam consisting of electrons(actually positrons, but as we are not at all concerned with charge we will consider them as electrons), muons and pions of momentum 120 MeV/c, we can test different concentrations of WLS solution with particles of energies closer to that which will be encountered in the full scale T2K project. We will also be able to see the efficiency with which our Cerenkov counter detects the electrons that pass through it.

3.2.2 Apparatus/Setup

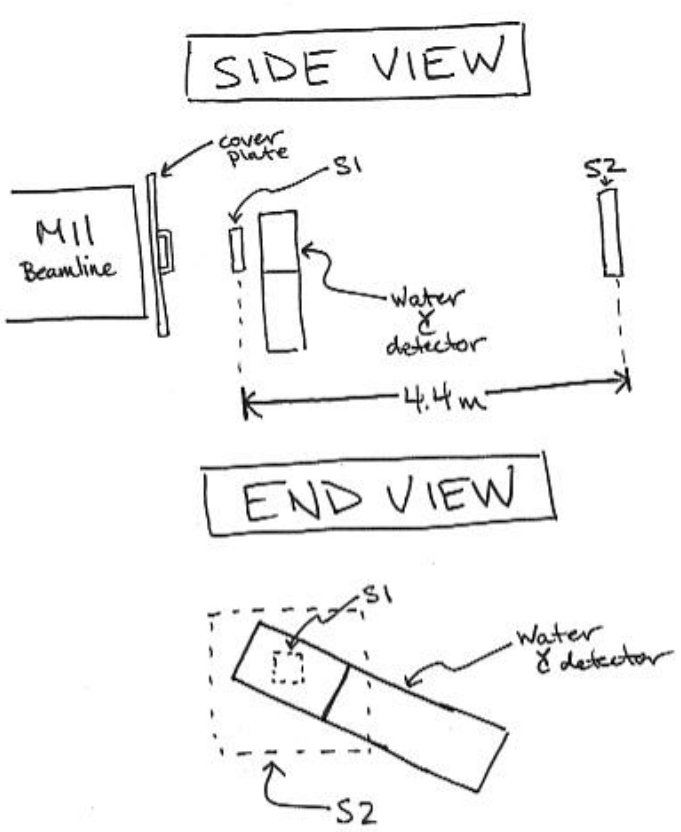

Figure 17: M11 beamline setup.

This setup consisted of our small water Cerenkov detector, as described earlier (see figure 8), placed in the M11 beamline at TRIUMF. The M11 beamline produces a beam of particles consisting of electrons (mostly), muons (fair number) and pions (very few), set at a momentum of 120 MeV/c. Two scintillators, S1 and S2, were used to identify the particles by their times of flight, and to trigger the water Cerenkov counter. S1 was placed immediately upstream from the counter, aligned with the center of the liquid cell and S2 was placed 4.4m

downstream from S1. S1 was initially 1.5cm X 1.5cm, but was later changed to 1.4cm X 4.0cm to cover more of the liquid cell. S2 measures 8.9cm X 8.9cm.

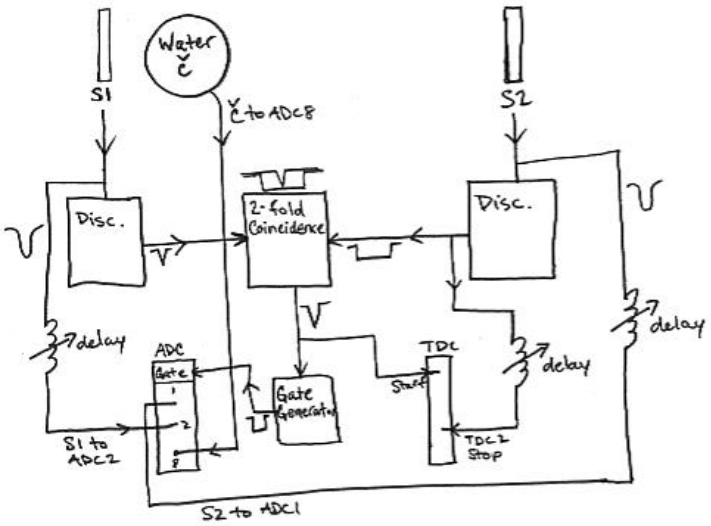

Figure 18: Camac Schematic for beam tests.

The data acquisition system (DAQ) was again set up using camac crates controlled by MIDAS DAQ software. The signals from S1, S2 and the Cerenkov counter were connected to ADC channels 2, 1 and 8 respectively. The ADC is gated by the coincidence unit, which could be set to trigger on S1, S2, or their two-fold coincidence. The discriminated signals from S1 and S2 were sent to a TDC, channel 2 giving the difference in their times as the time of flight of the particles. By later plotting this time of flight spectrum against the S2 signal amplitude, we were able to identify each type of particle.

3.2.3 General Overview of Procedures

After configuring our electronics so that we could see the entire spectrum of Cerenkov pulses coming from our detector, tests of the various concentrations of Carbostyril 124 were conducted as before in the source tests. We then calibrated the timing of our TDC signals so that we would have an idea of the range of our histograms and look at the time of flight spectrum of our particles. At first, there was a significant amount of noise in our timing spectrum which could not be explained, and time was spent diagnosing the problem. After reducing the noise as much as we could, the data had to be replayed with new software cuts in place to focus more colsely on the desired data. From this we were able to calculate the efficiency with which we were detecting beta particles, and come up with some improvements for our detector.

3.2.4 Calibration of Cerenkov Detector Signal

Calibration runs were made over the first few days, adjusting the discriminator thresholds, timing gate widths and timing delays. An analog scope was used to

see weak signals more accurately, but in some cases the signals were too faint and a digital scope set to infinite persistence was needed to get an adequate picture. Triggering on the coincidence of S1 and S2, we would get a count rate of less than 1 Hz, so to test the various concentrations of WLS solution, we set to trigger to S1 alone (as any particle passing through S1 also passes through the Cerenkov counter).

3.2.5 WLS Concentration Tests

With the coincidence unit set to S1 only, we got count rates of several hundred hertz and we were able to test the concentrations 0, 1, 2, 5, 7, 10, 20 and 100ppm (by weight) in two days. As pions and muons of 120MeV/c do not give Cerenkov radiation in water, the Cerenkov pulses in ADC 8 should be due only to our 120 MeV/c electrons.

After sufficiently calibrating our software and hardware settings so that the distribution of Cerenkov pulses fit within the limits of our histogram (see figure 19), we began testing the various concentrations of the WLS solution Carbostyril 124.

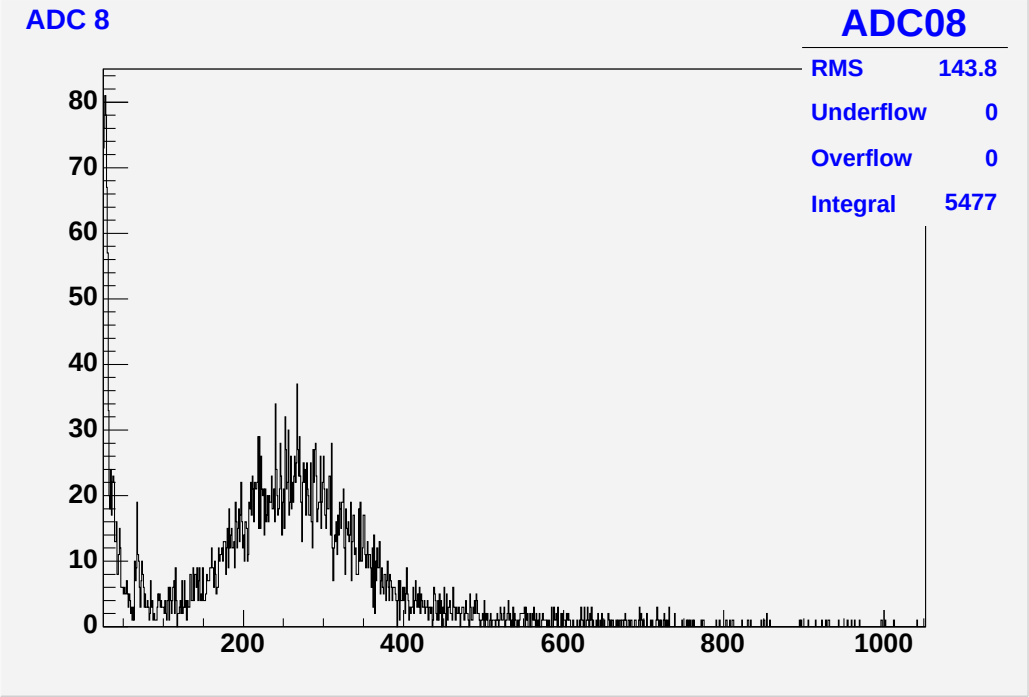

Figure 19: Histogram of Cerenkov Pulse Areas (0ppm WLS)

In this case, we are concerned mainly with the distributions position along the x-axis which is indicative of the number of photoelectronss produced by each beam particle. As the concentration was increased, the distribution shifted to the right, indicating that more light was being shifted by the solution. From a

plot of the pulse position versus the WLS concentration (figure 20), we can see the relative signal increase from each increase in concentration.

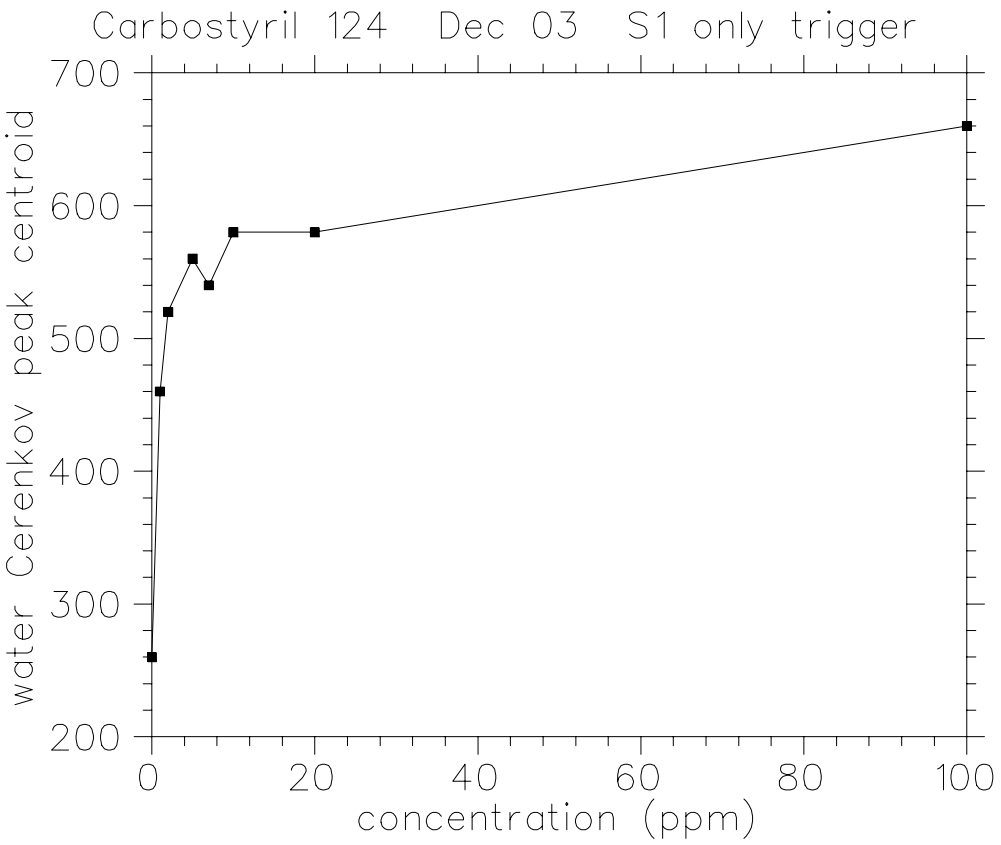

Figure 20: Pulse position vs WLS concentration

We can clearly see that the relative increase decreases significantly by 10ppm, suggesting that the solution is largely saturated at this point, and that at 1ppm the signal has nearly doubled.

3.2.6 Calibration of TDC

After testing the different WLS concentrations, the coincidence unit was then set to S2 only so that we could calibrate the TDC. Thus the TDC was started and stopped by S2. By varying the delay of the S2 stop signal by known amounts, we could watch the shift of the time spike and determine the time/channel in the TDC. Averaging over several delay changes, the time conversion factor was found to be about 0.1ns/channel. Although this doesn't give us the absolute times of the signals, it does give us their differences.

3.2.7 The Time of Flight Spectrum

From our first time of flight diagrams (figure 21), we could certainly see three distinct peaks corresponding to our electrons (channel 390), muons (channel

450) and pions (channel 500), but the there was significant noise around the electron peak (in the form of extra peaks at channels 365 and 410). The cause of this noise was at first unknown.

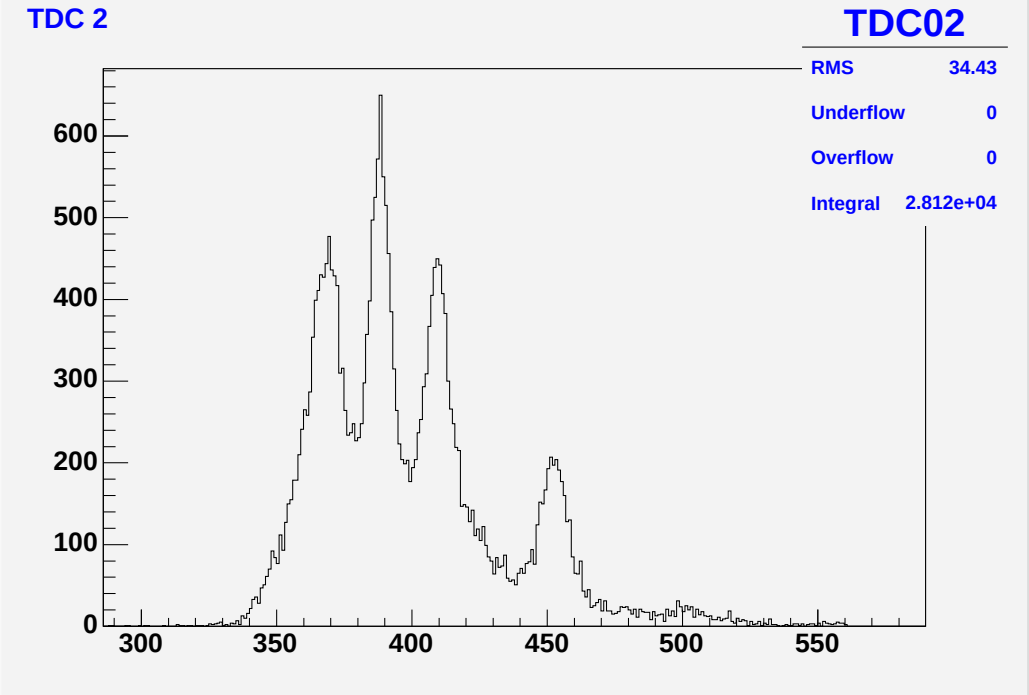

Figure 21: Extremely noisy time of flight spectrum (before software cuts)

By moving S1 along the horizontal axis to either side of the beamline, it was noticed that one direction reduced the noise while the other increased it. From this it was hypothesized that electrons were entering S1 not through the scintillator itself, but through the lucite light guide connecting it to the PMT. To test this, lead bricks were placed in front of the light guide of S1 and it was recentered. The resulting time of flight plot was still quite noisy, but there was a slight drop in the noise on the right hand side of the electron peak. After further speculation, it was realized that electrons could be and most likely were also entering the light guide of S2, which would unfortunately be too difficult to block with lead bricks.

Now since this is a timing plot, the noise to the left of the electron peak would be from something moving from S1 to S2 faster than an electron (not supposed to be possible) and the noise to the right would be from something slower.

Now since the particle beam is not highly collimated, we do not expect all particles to travel through the scintillators of S1 and S2. A few possible paths are shown in figure 22.

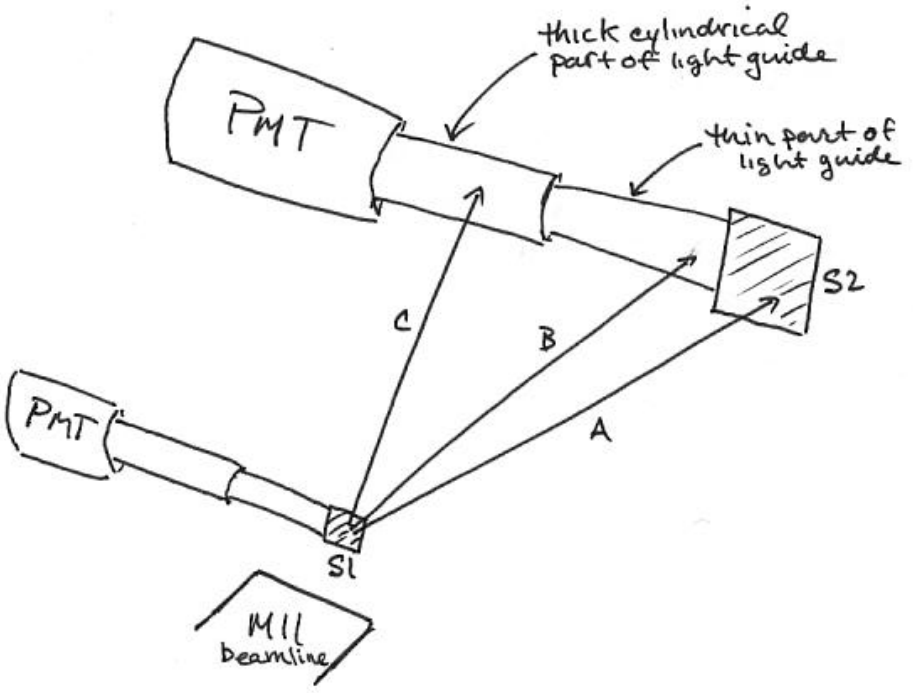

Figure 22: Posible paths of our Positrons.

Path A will give the expected electron flight time, whereas path C will cause S2 to trigger early as the light has significantly less distance to travel to the PMT. This accounts for our superluminal particles. Path B on the other hand will trigger late! Although the light has less distance to travel, this is only very slight. Since the signal pulse is significantly less than that of those electrons that scintillate, it will take slightly longer for the signal to cross the discriminator threshold as shown in figure 23.

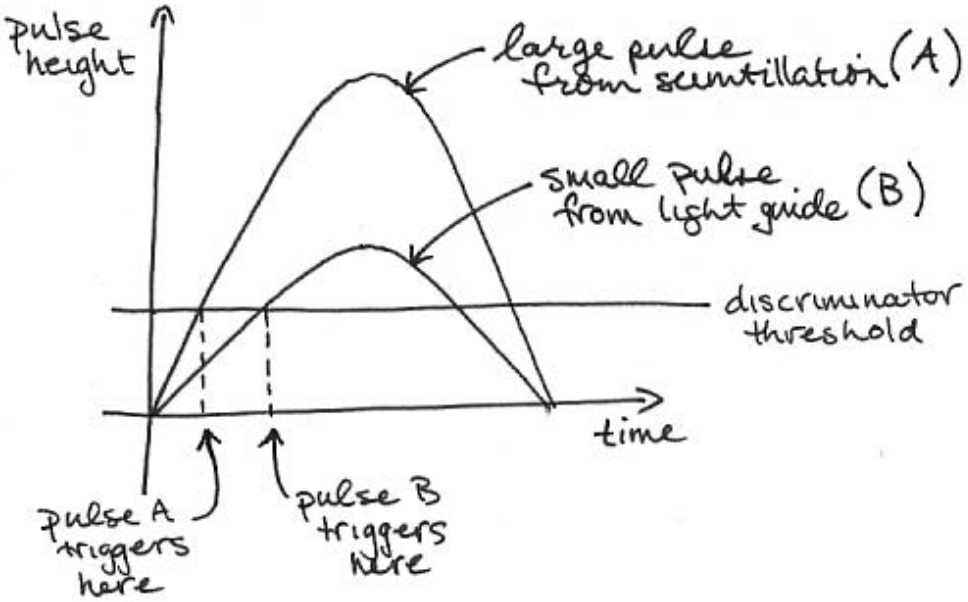

Figure 23: Trigger timing for pulses of different hight.

This would account for our slower particles. This behaviour may also be happening in the S1 scintillator and light guide, as we did see some effect by blocking it with the lead bricks. If this is the sole cause of this noise, then there is an easy way to eliminate it. By replacing each scintillator S1 and S2 with two scintillators each (so that triggering would be on the coincidence of all 4), oriented such that their light guides do not overlap, we will never trigger on a particle passing through a light guide. This approach will be considered for future runs.

At this point, to look at the the data from the electrons passing through S1 and S2 (not the light guides) and consequently the center of the Cerenkov counter, the data had to be replayed with new software cuts in place. As the signal from electrons passing through the light guides are noticeably smaller than those from electrons passing through the scintillators of S1 and S2, we can place cuts requiring the S1 and S2 signals to be above a certain threshold for triggering to occur.

Adding these cuts eliminated virtually all noise around the electron peak as seen in figure 24 (supporting our suspicions of its cause). Further cuts were then placed around the electron peak on the time of flight plot in order to cut out the muons and pions from our Cerenkov trigger. Although muons and pions of this momentum do not produce Cerenkov radiation in water, they may still create some noise in the Cerenkov counter signal. With all of these cuts in place, we were then able to look at the Cerenkov signal from only those electrons passing through the scintillator portions of S1 and S2, and consequently the center of

the Cerenkov counter.

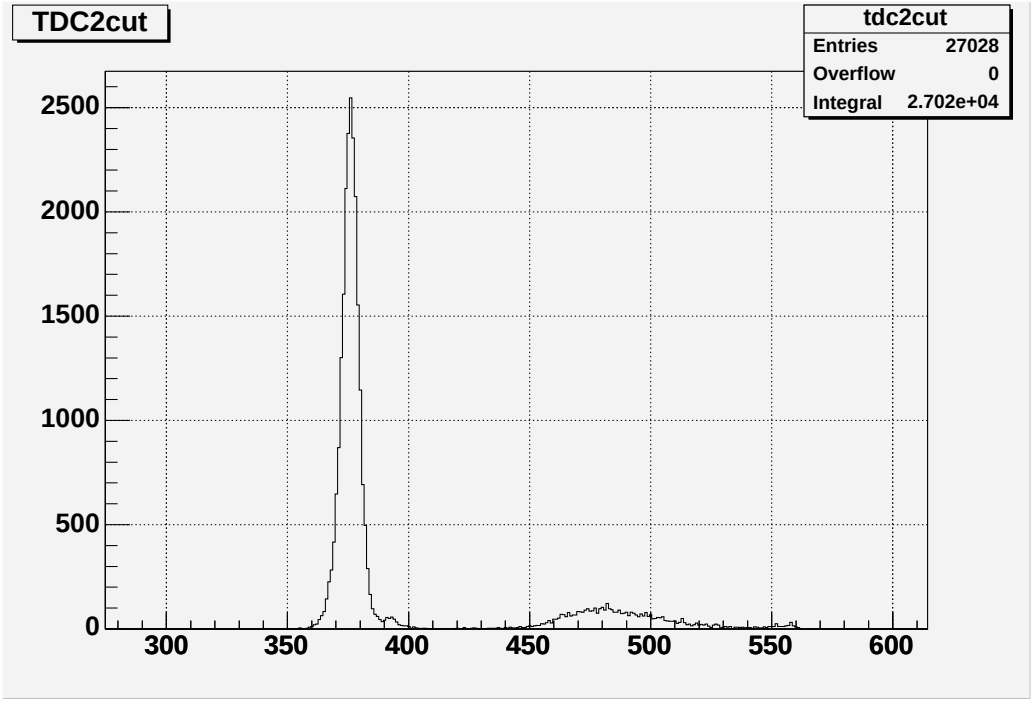

Figure 24: Clean time of flight spectrum (after software cuts)

3.2.8 Cerenkov Detector Efficiency Tests

With a clean time of flight spectrum now defined, we could get a count of how many electrons we had and compare this to the number detected by the Cerenkov counter. This was done with a smaller S1 (1.5cm X 1.5cm) centered at the liquid cell of the Cerenkov counter, and then also with a longer S1 (1.4cm X 4.0cm) positioned along the whole length of the cell.

Now, by comparing the number of hits that the Cerenkov counter picks up with the number of electrons in the newly defined region of the time of flight spectrum, we can calculate the efficiency with which we are detecting elecrons. We then used this method the calculate the efficiency from two runs. In the first run we used the smaller S1 scintillator (1.5cm X 1.5cm) centered in front of the liquid cell of our detector. Here we saw an efficiency of 94.8%. In the second run we used the larger S1 scintillator (1.4cm X 4.0cm) which covered significantly more of the liquid cell. In this run we saw an efficiency of only 85.3%. Now since the larger scintillator covers more of the liquid cell, we would then be triggering on electrons that are passing through closer to the top and sides of the cell. Due to the fact that the reflective Tyvek coating is only on the sides of the cell and not the top, any light hitting the top is not likely to make it into the PMT. Also, light emitted further from the PMT has further to travel and possibly more walls to reflect off of to get to get there. It would

therefore be less likely for light from electrons passing through the top of the detector to reach the PMT and register as an electron count. It is proposed that the top of the cell be coated with the Tyvek reflector in future tests of the water Cerenkov detector.

3.2.9 Conclusions

The concentration tests with the high energy beamline particles has proven to be much more informative than those done with the beta sources. From these tests we can see the signal double after adding only 1ppm of the WLS Carbostyril 124. We can also clearly see that the relative increase diminished drastically at 10ppm suggesting saturation of WLS.

On the other hand, we are still not seeing 100% efficiency, even looking only at betas passing through the very center of our detector. From our previous observations of the significant increase in light output due to the addition of the Tyvek reflector, it was suggested that we also cover the top of our cell with reflector. This should increase the efficiency some more, but we will have to see if we can reach a 100% efficiency.

4 Appendix A: Automated Data Acquisition Setup

The automation of our data acquisition system begins with the crontab. This is a feature built in to the Linux operating system. By creating a file of a specific format (see figure 25) often named 'cronjob' we can schedule commands to be executed with any level of regularity.

```
# min         hour     day month day-of-week command
#-----------------------------------------
 0,1,2,3,4,30,31,32,33,34  * * *       *     ./exp2/AutoWaterData >& AutoWaterData.log
 5,35            *        *    *       *     ./exp2/updateAll >& updateAll.log
```

Figure 25: cronjob

Once this file has been created (using any text editor) it must be put into the systems crontab with the command:

≫ crontab cronjob

If you would like to change the crontab, simply make the changes to your cronjob file (or create a new one) and reapply it with the same command. Our crontab executes two commands, one that runs the data retreival scripts and on that runs the data processing scripts.

AutoWaterData (figure 26) is the first Linux command script that is run by the crontab. This script creates some strings with the current date and passes these strings onto two other scripts which it calls. One called 'scopeget.perl' (figure 27) and one called 'gettemp.perl' (figure 29).

```
#!/bin/tcsh
set x = exp2/water/wtemp/`date +%y_%m_%d_%H:%M`.dat
set y = `date +%y_%m_%d_%H:%M`.dat
set date = `date +%y_%m_%d_%H:%M`
echo $y >> exp2/water/wtemp/datList.dat
echo $y >> exp2/water/wtemp/newDat.dat
./exp2/scopeget.perl > $x
./exp2/water/wtemp/gettemp.perl $date
```

Figure 26: Autowaterdata

AutoWaterData also updates two lists by adding the names of the new data files each time one is created. One is called datList.dat (figure 28) which is the master record of all the data files ever acquired. The other is called newDat.dat which has the same format, but is emtied each time the data is processed. This way we know which data files still need to be processed.

Scopeget.perl is the script that accesses the oscilliscopes page and retreives the current data file. It must be passed one parameter which will be the name of the file to be created.

```
#!/usr/bin/perl -w

my $confHostname = "detscope01.triumf.ca";
#my $confHostname = "send.triumf.ca:8081";
#my $confHostname = "www.triumf.ca";

use LWP::UserAgent;

my $ua = LWP::UserAgent->new;

if (0)
  {
    my $request = HTTP::Request->new(GET => "http://$confHostname/Setting.html");
    my $response = $ua->request($request);

    print "response: ",$response->as_string(),"\n";

    open (OUT,">xxx");
    print OUT $response->as_string();
    close OUT;

    die "Here!";
  }

if (1)
  {
    my $headers = HTTP::Headers->new(Content_Type => 'text/plain');
    my $body = "";
    $body .= "command=select:ch1 on\n";
    $body .= "command=save:waveform:fileformat spreadsheet\n";
    $body .= "wfmsend=Get\n";
    my $request = HTTP::Request->new(POST => "http://$confHostname/getwfm.isf",$headers,$body);
    my $response = $ua->request($request);

    die "Cannot get data from scope: ",$response->status_line(),"\n" if (!$response->is_success(

    #print $response->content();
    my $x = $response->content();
    $x =~ s/\r//mg;

    print $x;

    #open (OUT,">xxx");
    #print OUT $response->as_string();
    #close OUT;

    exit(0);
  }

#end file
```

Figure 27: scopeget.perl

```
03_11_11_18:04.dat
03_11_12_00:00.dat
03_11_12_00:02.dat
03_11_12_00:03.dat
03_11_12_00:04.dat
03_11_12_06:00.dat
03_11_12_06:01.dat
03_11_12_06:02.dat
03_11_12_06:03.dat
03_11_12_06:04.dat
03_11_12_12:00.dat
03_11_12_12:01.dat
03_11_12_12:02.dat
03_11_12_12:03.dat
03_11_12_12:04.dat
03_11_12_18:00.dat
03_11_12_18:01.dat
```

Figure 28: datList.dat

Gettemp.perl (figure 29) is the script that retrieves the temperature data from our probes. It must also be passed a parameter (the current date) which will be stored with the data. This script stores the date and time along with the temperature measurements in a file called 'temp.dat' (figure 30). This file holds all the temperature data ever taken. The new data is also put in a file called 'newTemp.dat' which again has the same format but is cleared each time the data is processed.

```
#!/usr/bin/perl -w

use strict;

my $arg = $ARGV[0];
my $tty = "/dev/ttyS0";

open(IN,"+<$tty") || die "Cannot open $tty: $!";
sleep 2;
my $now = time();
print IN "\n";
#sleep 1;
my $r1 = <IN>; chop $r1;
my $r2 = <IN>; chop $r2;
my $r3 = <IN>; chop $r3;
my $r4 = <IN>; chop $r4;
my $r5 = <IN>; chop $r5;
close IN;

die "Bad data from TempTrax2000: $r5" if $r5 ne "Bat Ok";

my $t1 = ($r1-32)/1.8;
my $t3 = ($r3-32)/1.8;

open (TEMPDAT, ">>exp2/water/wtemp/temp.dat") or die "$! error storing data";
print TEMPDAT sprintf("%s %.2f %.2f\n", $arg,$t1,$t3);

open (NEWTEMPDAT, ">>exp2/water/wtemp/newTemp.dat") or die "$! error storing data";
print NEWTEMPDAT sprintf("%s %.2f %.2f\n", $arg,$t1,$t3);

exit 0;
#end file
```

Figure 29: gettemp.perl

```
03_11_12_00:00 19.89 19.89
03_11_12_00:02 19.56 19.17
03_11_12_00:03 19.44 19.00
03_11_12_00:04 19.39 19.00
03_11_12_06:00 19.72 19.89
03_11_12_06:01 19.78 19.94
03_11_12_06:02 19.83 20.00
03_11_12_06:03 19.83 20.06
03_11_12_06:04 19.89 20.06
03_11_12_12:00 19.56 19.22
03_11_12_12:01 19.56 19.33
03_11_12_12:02 19.61 19.50
03_11_12_12:03 19.67 19.67
```

Figure 30: temp.dat

After AutoWaterData has finished running, the crontab also calls the script 'updateAll'. This is a Linux command script which runs four other scripts with the program 'ROOT'.

```
#!/bin/sh
date
export ROOTSYS=/triumfcs/trshare/olchansk/root/root_v3.05.07.RH8.0
##export DISPLAY=localhost:0
$ROOTSYS/bin/root -b -q exp2/water/wtemp/Proc.C
$ROOTSYS/bin/root -b -q exp2/water/wtemp/plotTemp.C
$ROOTSYS/bin/root -b -q exp2/water/plotWater.C
$ROOTSYS/bin/root -b -q exp2/PlotAll.C
```

Figure 31: updateall

The first scipt that is executed is called 'Proc.C' (figures 32 & 33). This goes through the entries in the new data and temperature files, calculating the peak amplitude of the scope signal and the area of the signal pulse. The algorithms that make these calculations can be seen in the 'Proc.C' code. Once the data has been processed it is all stored in data files. The master data file is called 'allData.dat' (figure 34) and has the shown format. Once the data has been processed some other scripts are run which read in the data and create various plots. An example of a script that will do this is 'PlotAll.C' (figure 35).

With all of this code running automatically, the experiment can be monitored for unusual behaviour by simply checking the desired plots as they are updated.

```
{
gROOT->Reset();
#include "Riostream.h"

ifstream inNew("exp2/water/wtemp/newDat.dat");
ifstream inNewTemp("exp2/water/wtemp/newTemp.dat");
ifstream *in;
ofstream out("exp2/water/wtemp/data.dat",ios::app);
ofstream out2("exp2/allData.dat",ios::app);

Int_t n = 10000;
Int_t t = -1;
Double_t year,month,day,hour,min;
char date[13], currentDate[13];
char file[40];
char xx[10];
char temp;
Int_t nlines;
Float_t time, area, amp, temp1, temp2, avTemp;
Float_t tm[n], ar[n], am[n];

Float_t x[10000], y[10000];
Int_t peakFlag = 0;

 while (inNew->good())
 {
   inNew >> year >> temp >> month >> temp >> day >> temp >> hour >> temp >> min >> temp >>
 temp >> temp;
   inNewTemp >> date >> temp1 >> temp2;

   //if (!strcmp(date,currentDate)) break;

   sprintf(file,"exp2/water/wtemp/%s.dat",date);

   //calc time in days from Jan 1 2003
   {
     time = 0;
     if (year > 3)
       {time = (year - 3)*365;}
     else {time = 0;}

     if (month == 1) {}
     if (month == 2) {time = time +29;}//29
     if (month == 3) {time = time +29+30;}//59
     if (month == 4) {time = time +29+30+31;}//90
     if (month == 5) {time = time +29+30+31+30;}//120
     if (month == 6) {time = time +29+30+31+30+31;}//151
     if (month == 7) {time = time +29+30+31+30+31+30;}//181
     if (month == 8) {time = time +29+30+31+30+31+30+31;}//212
     if (month == 9) {time = time +29+30+31+30+31+30+31+31;}//243
     if (month == 10) {time = time +29+30+31+30+31+30+31+31+30;}//273
     if (month == 11) {time = time +29+30+31+30+31+30+31+31+30+31;}//304
     if (month == 12) {time = time +29+30+31+30+31+30+31+31+30+31+30;}//334
     time = time + day + (hour/24) + (min/(60*24));
   }
```

Figure 32: Proc.C (part1)

```
   in = new ifstream(file);

   strcpy(currentDate, date);

   area = 0;
   amp = 0;
   nlines = 0;
   while (in->good())
   {
     *in >> x[nlines] >> temp >> y[nlines];
     if (y[nlines] < amp)
     {
       amp = y[nlines];
       peakFlag = nlines;
     }
     nlines++;
   }
   if (nlines > 1)
   {
     t++;
     for (pulsePos  = peakFlag - 250; pulsePos < peakFlag + 300; pulsePos++)
     {
       area = area + y[pulsePos];
     }
       avTemp = (temp1 + temp2)/2;

       out << time << " " << area << " " << amp << " " << avTemp << " " << temp1 << " " <<
 temp2 << endl;
       out2 << time << " " << area << " " << amp << " " << avTemp << " " << temp1 << " " <
< temp2 << endl;
       //printf(" Loaded file: %s\n",file);
       //printf(" found %d points\n",nlines);
       //printf(" Area = %5f\n",area);
       //printf(" Amplitude = %5f\n",amp);
       //printf(" Time: %f\n",time);
       //printf(" Temp: %f\n",avTemp);
   }
   in->close();
   delete in;
 }
ofstream outNew("exp2/water/wtemp/newDat.dat");
ofstream outNewTemp("exp2/water/wtemp/newTemp.dat");
}
```

Figure 33: Proc.C (part2)

```
327.565 -42.7967 -0.295898 19.385 19.33 19.44
327.583 -42.7372 -0.293031 19.555 19.5 19.61
327.584 -42.5205 -0.292148 19.585 19.56 19.61
327.585 -42.7674 -0.295219 19.585 19.56 19.61
327.585 -42.3994 -0.294203 19.64 19.61 19.67
327.586 -42.5643 -0.296094 19.5 19.5 19.5
327.604 -42.8783 -0.296039 19.085 19.11 19.06
327.605 -42.7017 -0.295977 19.14 19.11 19.17
327.606 -43.0179 -0.298008 19.195 19.17 19.22
327.606 -42.9122 -0.298195 19.25 19.22 19.28
327.607 -42.8392 -0.296203 19.305 19.28 19.33
327.625 -42.4773 -0.293016 19.47 19.44 19.5
327.626 -42.7003 -0.298008 19.47 19.44 19.5
327.626 -42.6377 -0.294266 19.53 19.5 19.56
327.627 -42.633 -0.293 19.56 19.56 19.56
327.646 -42.7199 -0.297289 19 19.11 18.89
327.647 -42.7711 -0.296367 19 19.06 18.94
327.647 -42.8252 -0.296547 19.085 19.11 19.06
327.648 -42.74 -0.295992 19.11 19.11 19.11
327.649 -42.7153 -0.296938 19.17 19.17 19.17
327.667 -42.635 -0.295 19.385 19.33 19.44
327.667 -42.7593 -0.294 19.445 19.39 19.5
327.668 -42.6926 -0.293133 19.445 19.39 19.5
327.669 -42.7172 -0.296 19.445 19.39 19.5
327.669 -42.5763 -0.294922 19.53 19.5 19.56
327.688 -42.8107 -0.301023 18.945 19.06 18.83
327.688 -42.7661 -0.297055 18.975 19.06 18.89
327.689 -42.6368 -0.299258 19 19.06 18.94
327.69 -42.7841 -0.297023 19.06 19.06 19.06
327.69 -42.7869 -0.296016 19.14 19.11 19.17
327.708 -42.8507 -0.295305 19.385 19.33 19.44
327.709 -42.6522 -0.292141 19.445 19.39 19.5
327.71 -42.8121 -0.291086 19.445 19.39 19.5
327.71 -42.7192 -0.293961 19.53 19.5 19.56
327.711 -42.8062 -0.297039 19.53 19.5 19.56
```

Figure 34: allData.dat

```
{
gROOT->Reset();
#include "Riostream.h"

//data is read from allData.dat
ifstream in("exp2/allData.dat");

Int_t n = 100000,x = 0;
Float_t time[n],area[n],amp[n],temp[n];
Float_t xx;

 while(in->good())
 {
   in >> time[x] >> area[x] >> amp[x] >> temp[x] >> xx >> xx;
   x++;
 }
x--;
TGraph *gr1 = new TGraph(x, time, area);
TGraph *gr2 = new TGraph(x, time, amp);
TCanvas *cv = new TCanvas("cv","Experiment #2 Analysis",1);
cv->Divide(1,2);

cv->cd(1);
gr1->SetTitle("Time vs Signal Area");
gr1->GetXaxis()->SetTitle("Time (days past Jan 1, 03)");
gr1->GetYaxis()->SetTitle("Signal Area");
gr1->GetXaxis()->CenterTitle();
gr1->GetYaxis()->CenterTitle();
gr1->SetMarkerStyle(6);
gr1->Draw("AP");

cv->cd(2);
gr2->SetTitle("Time vs Signal Amplitude");
gr2->GetXaxis()->SetTitle("Time (days past Jan 1, 03)");
gr2->GetYaxis()->SetTitle("Signal Amplitude");
gr2->GetXaxis()->CenterTitle();
gr2->GetYaxis()->CenterTitle();
gr2->SetMarkerStyle(6);
gr2->Draw("AP");

cv.Print("exp2/plot.ps");
}
```

Figure 35: PlotAll.C

5 Appendix B: MIDAS Data Acquisition System

5.1 Detector Facility 'Clean Room'

To access and run the MIDAS data acquisition software from the terminal midtis06 in the detector facility clean room, you must log into the system (or ssh) as:

username: daqt

password: daq_test

From here, enter the directory /home/daqt/midtis06 with:

≫ cd midtis06

Now you can start the software up with:

≫ ./start_daq.sh

You must also open up a web browser to the address:

http://midtis06:8081/
From here you can start and stop runs with the 'Start' and 'Stop' buttons in the top left corner of the window. To shut down the DAQ system, click on the 'Programs' button and stop each application.
Now in order to monitor the data (via histograms of each channel) as it is coming in, start the viewer with:
≫ ./TTriumfFileGUI midtis06
Omitting the midtis06 option in that command will still start the viewer, but you will not be able to view the online data as it is coming in. The data from preious runs can be found in the data/ directory. Here you will generally find a 'his*.root' and a 'run*.root' for each ru that was taken. The * will be the nubmer corresponding to the particular run. The his files contian only the histograms of the data wheras the the run files contain all of the data and can be used to rerun the data with new software cuts in place. As these files are .root files, further analysis and rerunning of data is done using ROOT.

5.2 M11 Counting Room

To access the MIDAS software on the 'daqtest' teminal in the M11 counting room, you must log in (or ssh) as:
username: daqt
password: daq_test
From here, enter the directory:
/home/daqt/neut
Now you can start the software with:
≫ ./start_daq.sh
You must also open up a web browser to the address:
http://daqtest:8081/
From here you can control the DAQ system the same way as described for the midtis06 system.
Now there was an older version of ROOT on this machine in the /usr/local/root directory, but in order to run the histogram viewer you must use a new teminal window to ssh into daqt@midtis06 with the previously mentioned password. Then you can run the viewer from the same folder as before this tim using the command:
≫ ./TTriumfFileGUI daqtest
By adding the daqtest option this time, we will be able to see the live data from the M11 aquisition system.
In order to analize and rerun this data, it is best to send it to the midtis06 machine. It has all been copied to the directory /home/daqt/midtis06/data/daqtest. In this directory you will also be able to find ROOT scripts for rerunning the data.

5.3 Analysis and Rerunning of Data using ROOT

Several versions of root can in the directory
/triumfcs/trshare/olchansk/root

On the midtis06 machine, a path to the newest version (at the time) should be saved in $ROOTSYS.If not, to create one simply enter:

≫ export ROOTSYS=/triumfcs/trshare/olchansk/root/root_v3.10.01.RH8.0

Then to boot up ROOT enter:

≫ $ROOTSYS/bin/root

To run a script with root enter:

≫ .x scriptName.C

To quit enter:

≫ .q

Now a script has been created to rerun any of the previous data files. This file is called run.C. In one of the lines near the top of the code is where you can set which data file you will be reruning. Then there is a large section where every histogram gets defined. Now there is another file which is connected to this run file, called Trigger.C. Each of these histograms must be defined with the same name in both of these files. I would suggest never removing a histogram from the definitions, just create every one that you end up thinking of. You can then, in the run.C file, create canvases with the plots that you would like to display, print or save. For details on the syntax of defining histograms and canvases, consult the ROOT manual and documentation.

```
{
gROOT->Reset();
TFile f("run00088.root");
f.Print();
f.ls();
TTree *t = f.Get("Trigger");
t->Print();
TH2D* adc8vt = new TH2D("adc8vt","ADC8 vs TDC2",50,0,1200,50,0,800);
TH1D* tdc2 = new TH1D("tdc2","TDC2",1000,0,1000-1);
TH1D* adc8 = new TH1D("adc8","ADC8",1200,0,1200-1);
TH2D* s1vt = new TH2D("s1vt","ADC2 vs TDC2",50,0,1000-1,50,0,1500);
TH2D* s2vt = new TH2D("s2vt","ADC1 vs TDC2",50,0,1000-1,50,0,1500);
TH1D* adc8cut = new TH1D("adc8cut","ADC8cut",1200,0,1200-1);
TH2D* adc8cutvt = new TH2D("adc8cutvt","ADC8cut vs TDC2",50,0,1200,50,0,800);
TH1D* tdc2cut = new TH1D("tdc2cut","TDC2cut",1000,0,1000-1);

TSelector *s = TSelector::GetSelector("Trigger.C");
t->Process(s);


TCanvas *c1 = (TCanvas*)gROOT->FindObject("c1");
if(!c1)
  c1 = new TCanvas("c1","c1");
gStyle->SetOptStat(1100011);
gPad->SetGrid(1,1);
tdc2cut->Draw();
}
```

Figure 36: run.C

It is the trigger.C file that actually fills the histograms. It is in the section of this file shown below that we define what the histograms will be filled with. This is where you can define your new software cuts. As I am not sure of what the rest of this trigger.C file actually does, I would recommend changing as little as possible unless you know what it is. Also, if you must, I would recommend

commenting bits out instead of deleting them entirely. There are also a few other .h files that seem to be needed here, but I am not sure of exactly how they are connected.

```
void Trigger::ProcessFill(Int_t entry)
{
  // Function called for selected entries only.
  // Entry is the entry number in the current tree.
  // Read branches not processed in ProcessCut() and fill histograms.
  // To read complete event, call fChain->GetTree()->GetEntry(entry).

  adc8->Fill(ADCS_ADCS[8]);
  tdc2->Fill(TDCS_TDCS[2]);
  s1vt->Fill(TDCS_TDCS[2],ADCS_ADCS[2]);
  s2vt->Fill(TDCS_TDCS[2],ADCS_ADCS[1]);
  adc8vt->Fill(ADCS_ADCS[8],TDCS_TDCS[2]);

  {
    //define cuts
    bool s1good, s2good, tdc2good, good;
    s1good = ADCS_ADCS[2]>350;
    s2good = ADCS_ADCS[1]>350;
    tdc2good = TDCS_TDCS[2]>365 && TDCS_TDCS[2]<385;
    good1 = s1good && s2good && tdc2good;
    good2 = s1good && s2good;
  }
  if (good1)
    {
      adc8cut->Fill(ADCS_ADCS[8]);
      adc8cutvt->Fill(ADCS_ADCS[8],TDCS_TDCS[2]);
    }
  if (good2)
    {
      tdc2cut->Fill(TDCS_TDCS[2]);
    }
}
```

Figure 37: trigger.C

Also, although I would recommend applying strict cuts to the data after it has been recorded, you can also apply cuts to the data as it is coming in. To do this you must edit the modules.xxx file. I have even less understanding of what this file does, but i believe the part where the cuts are defined look like that shown in figure 38.

```
INT event(EVENT_HEADER *pheader, void *pevent)
{
  WORD   *adcdata;
  WORD   *tdcdata;

  /* look for ADCS bank, return if not present */
  if (!bk_locate(pevent, "ADCS", &adcdata))
    return 1;

  /* look for ADCS bank, return if not present */
  if (!bk_locate(pevent, "TDCS", &tdcdata))
    return 1;

  if ((tdcdata[2]>360)&&(tdcdata[2]<380))
    {
      gChist->Fill(adcdata[8]);
    }

  for (int i=0; i<kNumAdc; i++)
    gAdcHists[i]->Fill(adcdata[i]);

  for (int i=0; i<kNumTdc; i++)
    gTdcHists[i]->Fill(tdcdata[i]);

  return SUCCESS;
}
```

Figure 38: modules.xxx