Nikhil Jain **September 21, 2007**

Software Guide

I have listed here the purpose of the various pieces of software I have written while completing my NSERC USRA at TRIUMF under the direction of Dr. Scott Oser, in addition to an explanation of their usage and their location.

The directory ~/, whenever here used, implies my home directory located at: neut02.triumf.ca:/home/njain/.

Not all pieces of software in my directory are useful or functional; some of them were designed as experimental pieces of code for me to test new functions or pieces of code in C++ err applying them to a useful program. Only the functional pieces of code that were applied in some actual manner during my tenure at TRIUMF are explained here.

Weekly Reports

~/Work/Weekly_Reports contains the weekly FGD meeting reports which I gave regarding my progress over the course of the summer. The various files are included in both PDF and OpenOffice Impress file formats.

Student Symposium

~/Work/Student_Symposium contains a presentation of the T2K project and my own work that I gave at the annual TRIUMF summer student exposition.

Annual T2K Meeting

~/Work/T2K_Meeting contains the project report and presentation that I gave at the annual T2K meeting in August.

Recovery Analysis (Data taken under direction of Dr. Retiere)

~/Work/TimeChange contains two subdirectories /110pf_Cap and /510Resistor. The names of the two directories represent a component that was switched on the external MPPC circuit in order to alter the long recovery constant (either a 110pf capacitor or a 510Ω resistor). The /110pf_Cap has a file named Graph_time.C which is a ROOT script file, which one runs by using the command ".x Graph_time.C" once in ROOT. This program reads in the data contained in TimeChangeData.txt and makes a graph of the Recovery of the MPPC, by showing the ratios of the charge output from double laser pulses separated by various time delays. In the /510Resistor directory, the ROOT script Graph_time510.C produces a similar graph as did the script in the /110pf_Cap, although exhibiting markedly different recovery behavior.

Recovery Analysis (Data taken under direction of Dr. Retiere)

~/Work/PlotAnalyzer contains a ROOT script PlotRecData_v2.C which produces a graph that depicts the recovery behavior of the MPPC after pulses of different sizes.

Examination of the Non-Uniformity of Light Emitted by the WLS Fiber

~/Martyn_data contains the data collected by physics masters student Martyn Bryant regarding the nature of the distribution of the light emitted by the wavelength shifting fiber. The ROOT script RootScript.C takes his data, stored in Data.txt, graphs it, and then fits a simple Gaussian curve to the data, in order to extract the standard deviation of this distribution. The extracted value was approximately 0.47mm, which was later included in the MPPC simulation.

MPPC Simulation Programs

From this point onwards =/ refers to the following directory:

~/Software/nd280rep/elecSimTest/v0r2/

All the programs in the =/app/ directory are compiled by going to =/cmt and using the following two commands: “source setup.csh” & “make”. All the programs are used by going to =/Linux -i686 and typing ./[PROGRAM_NAME].

Parameters for this simulation are altered by modifying the following file:

~/Software/nd280rep/oaCalibrationDatabase/v0r0/parameters/elecSim.parameters.dat

For all of the pieces of Software included in the =/app directory, one can modify two pieces of code depending on one's desires. Firstly, the loop “ for(int i = 0; i < 30; i++)” can be told to run more times for greater accuracy, but also increased run time. Secondly, the lines “int xInc = 500”; “for (int x = 0; x< 15001; x=x+xInc){” can be modified in two manners. By modifying the xInc variable, one changes the steps at which light levels are increased, by modifying the second line, one can change the range over which light levels are to be sampled.

Reproducing Dr. Retiere's Recovery Graph using Dark Noise

The application “doublePeakFinder.cxx” in =/app aims to reproduce the dark noise data collected by Dr. Retiere and presented at the June 13, 2007 FGD meeting, which gives a good measure of the MPPC recovery and crosstalk parameters. When using this program, one should modify the parameters file such that elecSim.PPD2D.PDESlope = 0.0 and elecSim.PPD2D.DCRSlope.fgd = 20,000,000. The change of these two parameters will ensure that there exists sufficient noise in order to extract the recovery behavior of the MPPC. The data produced by this simulation will be stored in =/Data/FabriceGraph/Peaks_and_Times.txt. One can read in this data and produce graphs by using the ROOT script FabriceGraph.C, which will offer two options, either to view a 1d histogram of the relative probabilities of one, two and three pixels firing, or a 2d histogram of number of photoelectrons detected vs. time in which to examine the recovery properties of the MPPC.

Using the Integration Gate method

Of the two methods mentioned in my report in converting from a flash ADC waveform to the number of MPPC pixels activated, this piece of software utilizes the integration method, where it measures the total charge under a pulse. The application "IntegGate.cxx" uses the MPPC simulation and finds the charge released by the MPPC for a variety of different light levels. The collected data will be stored in =/Data/IntegGate/IntegGate.txt. To view the resulting data, one can go to =/Data/IntegGate and use the ROOT script Profile_Graph.C, which will allow one to a saturation curve of the generated data. Lines 57 and 58 of the code uses an empirically found conversion factor to convert from charge to pixels; this may need to be modified if one has changed the gain or voltage in the parameters file.

Using the Peak finding Method

The file "PeakLoop.cxx" uses the other method, where it merely finds the height of the peak, and associates that, via some empirically found conversion factor, to the number of pixels fired. The data from this program is stored in =/Data/PeakLoop.cxx. It needs to be moved or copied, however, to the directory =/Data/SatCurves/HighStats/ after the program has fully completed generating the data. One can then examine the results by going to the directory =/Data/SatCurves and using the ROOT script Profile_Graph.C. This script will create a saturation curve, in addition to fitting the curve and printing to the terminal a series of important values: the maximum difference between the number of pixels predicted by the curve and the generated data; and the predicted maximum number of pixels that can be activated.

Waveform Generator

The program " test_li_waveform.cxx" will produce a simulated waveform of the ASIC output.

Varying MPPC Parameters

The program "VarParam.cxx" effectively does the same thing as PeakLoop.cxx, but uses fewer light levels and fewer flashes at each level, so as to run quicker. It was used primarily to simulate the data examined in Section C of my report. The data generated by this program is outputted to =/Data/VarParam/VarParam.txt, where it can be read in by using the ROOT script Profile_Graph.C in the =/Data/VarParam directory, and selecting the option 0.

Accounting for various LED imperfections

In the directory =/Data/SatCurves there are present a variety of programs. The various programs simulate a serious of different imperfections with the LEDs in the light injection system. These were the scripts used to produce the results for section B in my report. For example, NomPhotons.C simulates Poisson fluctuations due to shot noise in the LEDs. The most important of these is AllEffects.C which simulates all of the following effects: shot noise in the LEDs; unknown nonlinearity of light output in the LEDs; Gaussian variation in LED output due to power supply fluctuations; and Gaussian variation in the charge output of the MPPC. This script generates a set of 100 simulated data sets which exhibit these effects. The programs (NomPhotons.C, NonLinPhotons.C, LowStatGen.C, RandomPhotons.C) all similarly generate 100 simulated data sets exhibiting one of the above-mentioned effects each. Any one of these programs can be modified in two ways: firstly, the number of light levels that they are expected to be sampling can be increased or decreased by modifying the step size (yInc); secondly, one can modify the number of samplings at each light level by modifying line 50's variable “SimNumFlashes” to some desired number.

One can examine any one of these particular data sets by using the ROOT script noout_LpReader.C, which will ask for two options: choosing which effect's data set one wishes to examine; and which generated data set of that type one wants to look at. This script produces a saturation curve of this specific data set, and fits it with the linear sum of the three exponentials that I used throughout my saturation curve fitting.

Another important program in this directory is Fit_Comparer.C, a ROOT script that will ask which of the above effects' data sets one wishes to examine for the divergence of it's fit from that of the high statistics set (=/Data/SatCurves/HighStats/PeakLoop.exe). Fit_Comparer.C will go through all hundred of the generated data sets for the effect selected and fit each of them, and output a graph that shows the ratio of each of the sets' fits to that of the high statistics set fit. There will also be two red lines which represent the 5% error tolerance that was desired.

This concludes the my explanation of the pieces of software I used/created during my time at TRIUMF. If anyone has difficulty examining or understanding my results or my programs, please feel free to email me at Nikhil.Jain87@gmail.com and I shall reply within a few days' times.
