An Electronics Simulation and Improved Noise Model for Milagro

Andrew Smith

Introduction

The Milagro electronics has never been part of the Milagro detector simulation. The GEANT simulation currently only computes the arrival times of photons at the PMT. The time of arrival of the first hit is smeared slightly and declared to be the hit time. Likewise, the number of PEs is simply the sum of the number of simulated photons that reach each PMT with a little bit of smearing. Noise is added by the application of random hits in the routine CalibrateMC(). If a noise hit is added to a channel that is already hit by a primary, the noise hit cancels out the air shower hit. This approach has the following deficiencies,

1) TOT effects are not simulated. It is a common problem that late hits will can lengthen pulses. In the data, 2 widely separated PEs can reconstruct as 4-5 PEs because the time separation lengthens time over threshold (TOT) measurement. This makes our charge resolution function quite complex. It is not simply related to nPEs or sqrt(nPEs) as one might expect, but it depends on the temporal dynamics of the event.
2) The noise model used is highly unrealistic. Real noise is almost entirely due to cosmic ray secondaries unrelated to the simulated cosmic ray shower. The noise should be constructed from correlated shower fragments, not uncorrelated random noise.
3) Noise hits that occur just prior to, during or just after the a hit can combine with the hit to form a single modified hit. The dynamics of this process can only be properly simulated by combining the wave forms from the multiple PE sources.
4) Reliability of the simulation depends on it's accuracy as a representation of the detector. The effect of an unsimulated detector element can never be really known. The electronics are often blamed for inconsistencies between the Monte Carlo simulation and data. Without simulating the electronics, the arguments that they are the culprit can never be ruled in or out.

In this memo I describe a simulation of the electronics contained in function CalibrateMC2() which can be directly substituted for CalibrateMC(). CalibrateMC2() utilizes the raw PMT hit times from the GEANT simulation, combines them with simulated “noise” events and “dark” random noise events, computes wave forms channel by channel, simulates the multi-threshold front-end boards, the edge finder and the calibration.

Noise Model

“Noise” events are generated using the standard GEANT Monte Carlo simulation. Proton primaries are thrown over a 1 km squared area with isotropic sky coverage out to theta = 70 deg. The simulated proton events have a spectral index of -2.75 (measured by some balloon) and are thrown from 5GeV to 100TeV. From this sample, events with at least 1 hit in any PMT, but no more than 10 air-shower layer PMT hits are stripped. About 20,000 of these “noise” events were generated, and are stored in a file in the CONFIG_MILAGRO directory.

The noise hits are distributed randomly in a 5 microsecond window encompassing the simulated event. Random hits from the noise sample are added at a rate that will produce a 20kHz/PMT average noise hit rate for the air-shower layer. 2kHz of random “dark” uncorrelated 1 PE noise is also added.

After the primary event hits and noise hits are combined, the photons arrival times are sorted and used to construct wave forms for each PMT. The wave forms are scanned to identify the times of the crossings of the low and high threshold discriminators. Currently, the low and high thresholds are set at 0.25PEs and 4.0PEs respectively. The process of scanning the waveforms in general is to slow to be practically applicable to the Milagro data. For this reason, the single PE waveform is approximated as having an instant rise and an exponential fall,

Amplitude = A*exp(-t/tau).

This simple shape allows for the addition of photon amplitudes, saving computation time.

Trigger Simulation

The list of sorted PE arrival times are used by a slightly modified version of the VME trigger simulation called SimulateVMETrigger2(). This function uses the PE arrival times, and not the hit times as in the old trigger simulation, to compute the trigger mask and the time of the event trigger. This is a more accurate representation of the VME trigger, because it allows for more than one hit in each channel, but it still has some short comings which I will describe later.

TDC Simulation, Edge Finder and Calibration

The times of level crossings are offset by the trigger time and an additional delay is added to approximate the delay prior to the common stop. The times are digitized in 0.5ns bins (TDC channels) and subtracted from 6000 to emulate the reverse time lists generated by the real TDCs. The mock TDC data is loaded into the raw data structures for easy access by users. Below are plots of AS layer first edge times for this simulation and for real data. Note the lip on the right edge. Do you know what causes it?

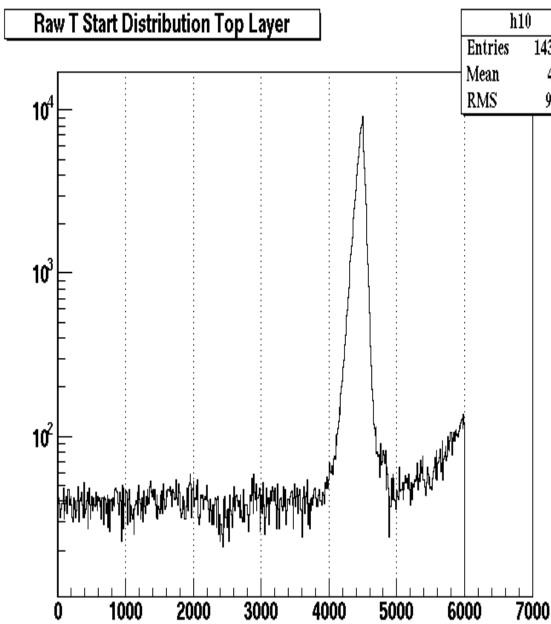

Figure 1 First edge distribution for MC data.

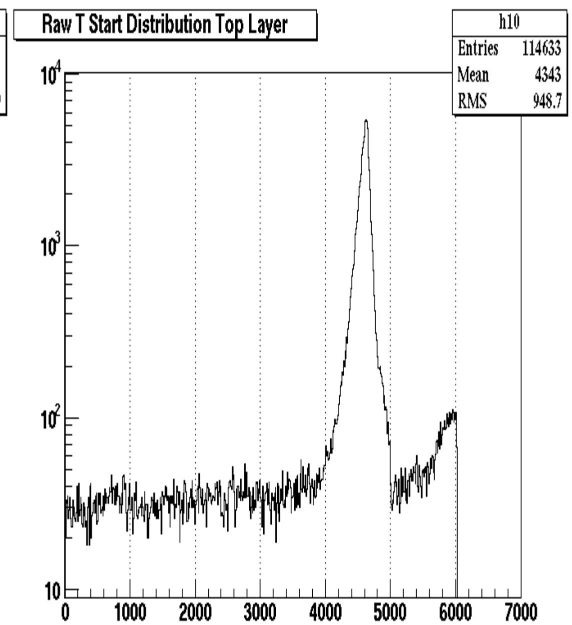

Figure 2 First edge distribution for real data.

The lists of TDC edges are then examined, and a slimmed down version of the edge finder is used to identify good 2 and 4 edge hits. The time over threshold is used to compute the number of PEs, assuming all the PEs were liberated at a single time as is done in the calibration. Also, 2 edge hits that have reconstructed PE levels higher than the high TOT threshold are assigned the PE value equal to the high threshold. This is also done in the real data calibrations.

Conclusion

Code to simulate the electronics has been checked into the Milagro CVS archive, but is not the default. One can utilize this simulation by changing the call to “CalibrateMC()” in function “CalibrateRaw()” to

“CalibrateMC2()”. Although it is functional, the code is still in its testing phase. This simulation provides a nice tool, but may not be properly optimized to best represent the Milagro data. A number of details were left out due to lack of knowledge of the electronics that may impact the accuracy of the results. Listed below are a number of what I still consider to be outstanding issues with the electronics simulation.

1) VME Trigger: The VME trigger simulation does not properly account for multiple hits in a single PMT during a trigger window. If a PMT is hit, a trigger pulse is generated with a width of 190ns. The old trigger simulation assumes that each channel is hit only once and in a 2 microsecond window, and the time is the time of the first hit. In this improved version of the trigger simulation, if two hits are separated by more than 190ns, they are both included in the simulated trigger waveform. However, this is not a complete description either, because a pair of hits in the same PMT separated by 50ns will generate a trigger pulse of length 240ns. Neither the new or old trigger simulations properly account for this lengthening of trigger pulses.
2) Edge Finder: The edge finder, and the front end board edge generators, have a number of idiosyncrasies that are not simulated. The leading high threshold edges are delayed by ~50 counts compared to leading low threshold edges, likewise the trailing high and low edges are forced to be separated in time. These effects are not simulated.
3) TDC jitter: There is no accounting for TDC measurement error. I would assume that the accuracy of the edge times and their digitization by the TDCs introduces an error. None is simulated.
4) Waveforms: The shape of the wave forms is simplified, and super position is assumed to work. In reality, the pulses have a different shape, and super position breaks down for hits containing lots of PEs, as is illustrated by the observation of after pulsing and non-linearity. The PMTs are also AC coupled, so the pulse amplitudes can actually be negative.
5) Many Constants: The values for the thresholds, the single pulse amplitude variation, the waveform exponential fall rate and many other constants are utilized in the this simulation. The number were chosen to be more or less sensible, but are probably not optimal.

Despite several short comings, this new electronics simulation is a substantial improvement on the current non-simulation of the electronics.