DIR-2-89

The Hawaii DUMAND Monte Carlo

Interim Progress Report

V.J. Stenger

August 23, 1989

[This is an updated version of the interim report issued August 21, 1989. That one had results inadvertently based on a 10 ns time resolution. The following is based on 3 ns and is much better, illustrating how very important PMT time resolution is for the Octagon]

This is an interim report of progress made this summer with DUMC, the Hawaii DUMAND Monte Carlo program, in studying the properties of the Octagon array..

New Spacetime Fit Developed

The previous version used a simple straight line fit through the hit modules to get the first guess to feed to the main fitter. Time information was not used except to establish the sense. I have now adapted an idea I developed while on sabbatical in Italy working on MACRO (see HDC-6-87) that uses the full space and time information to establish a better first-guess.

Basically I have discovered (or re-discovered) that, if you treat all three Cartesian coordinates of the muon's velocity **v** as free parameters (i.e., do not force the speed to be c), write down a χ^2 in terms of these and the other three free parameters, viz., the coordinates of a point r_o on the track, differentiate with respect to the six parameters and set the six derivative to zero, you get six uncoupled equations that can be easily solved for the chisq minimum without the time-consuming iterative search. Each axis has solutions of the form:

$$v_x = \frac{\langle xt\rangle - \langle x\rangle\langle t\rangle}{\langle t^2\rangle - \langle t\rangle^2} \tag{1}$$

$$x_o = \langle x\rangle + v_x\langle t\rangle \tag{2}$$

The simulated pulse widths that measure photoelectrons are used as weights in the computation of the averages.

Note that the first equation says that velocity is the *correlation* between points in space and time.

Of course this fitting procedure is only approximate for DUMAND, since the coordinates the hits are not generally on the track. But the MC indicates that the fitted speed is close to c: 0.28 ± 0.03 m/ns. The median pointing error is 9^{o} for tracks without background but with realistic errors for time and pulse width (time resolution $\sigma_t = 3$ ns used throughout for the results presented in this report). The effect of background and the filtering processes are discussed below; it has been found that, with the background filters developed, the same pointing error distribution for the spacetime fit is achieved.

When the spacetime fit is used as the preliminary guess for the χ^2 search, about 20 % more events fit than previously, raising the effective area correspondingly. However do not put this in the bank just yet. The tighter cuts necessary to reduce the effects of background use up all of that gain.

Four Parameter χ^2 Search

Following up on an idea I got in Moscow this past June, I began to develop a χ^2 search that uses only four parameters instead of five (all χ^2 search algorithms fix the muon speed at c). The idea is to replace one of the spatial parameters with a time, and since time appears explicitly in the χ^2, this parameter can be analytically determined, leaving a four parameter space to be searched for the final solution.

Unfortunately the new algorithm gave about 25% fewer fits, so I postponed further work with it. I have not tried to figure out why it fails so often; perhaps it is simply a program bug. I hope to get back to this later.

Simulation of Random Background

Previously, my version of the Monte Carlo did not mix background hits in with the track hits, although backgrounds were considered in the calculation of false trigger rates and in determining the fraction of false triggers that gave muon track fits.

In the new version, random hits are now mixed in with the simulated track data. For a 1 pe singles rate of 100 KHz and the new single-string trigger scheme proposed by J. Learned: $(t_1 - t_2) - (t_2 - t_3) \leq 15$ ns, where the t's are the times of three adjacent PMT's, about 5% of the triggers are caused by a background hit.

The PMT threshold is set to 1/3 pe. Strategies that involve raising this threshold for all or part of the trigger, or for the Philips tubes in the array, are considered next-resort. Such strategies reduce the effective area and I would like to see how well we can do at minimum threshold first.

In addition to the single track filter, I have required at least 3 strings and 8 total hits. I also require at least two clusters of adjacent 2-fold or greater coincidences. This gives an effective area for the zenith range $-1 < \cos\theta < 0.2$ of 33,000 m^2 for triggers. Of these triggers, 2% are due to random noise at 100 KHz.

With this trigger and noise rate, there are 15 ± 4 background hits and 17 ± 10 track hits in events passing the single-string trigger. These must be filtered before a trying to fit the muon track.

Filters

Simple filters have been developed that reduce the background significantly:

1) All hits are compared with the trigger hit and tossed out of the hit time exceeds the time it would take the Cherenkov wave front to pass between that module and the trigger module.

2) Several iterations (typically three) are made on the spacetime pre-fit. After each, the measured hit times are compared with the expected from the fit and tossed out of the difference, or pull, exceeds one standard deviation from the mean pull.

3) The event is discarded if the muon speed determined by the fit is insufficiently close to c (typically, more than 0.1 m/ns).

4) Two iterations are performed on the χ^2 search, with the same hit toss-out procedure as 2) if it fails the first try.

5) Even if the event passes the χ^2 search fit, it is discarded if the number of strings used in the final fit is less than three and the number of hits used is less than some number, currently 8. Note that this is applied after hits have been discarded in the filtering process. The cut on fit hits was imposed after it was found the the large reconstruction errors were associated with small numbers of hits.

After these filters, 0.5 ± 0.8 (standard deviation) background hits and 17 ± 8 track hits remain. In terms of the distribution of background hits, 56% of the events have zero hits, 24% have one, 10% have two, and 10% have three or more.

I have experimented with other filters, including one along the lines of one used by A. Roberts that basically tosses out hits that are poorly correlated with the others. However, this particular filter is more-or-less equivalent to filter 2) above since, as we see by equation (1), velocity is a measure of this correlation. In any case, I doubt I can get much below an average of one background hit per event.

Effect on Track Reconstruction

As mentioned, the spacetime fit has a median pointing error of 9°, both without background and in the case where the 100 KHz random background is filtered. At the point where the spacetime fit is done, an average of about 2 background hits is present in the data. This gets further reduced in the χ^2 search fit.

The value of χ^2/NDF (number of degrees of freedom) is not a good indicator of poor fits. Very low values often occur in cases of large pointing errors.

Large pointing errors in the χ^2 fit are associated with higher numbers of background hits. Of course, in the actual experiment we cannot tell whether a hit is background or signal, however it is found that when a sufficient total number of hits is used in the fit, the pointing error is minimized. Requiring at least three strings and 8 hits in the final leads to a median pointing error of 1.5^{o}.

With these cuts and filters and a few other details not described here, the Octagon is found to have an effective area averaged over the region $-1 < \cos\theta < 0.2$ of 22,000 m^2 (better than before because of the new spacetime pre-fit).

The pointing error can be further improved to a median of 1^{o}, provided we increase the minimum number of points in the fit from 8 to 12. This reduces the effective area, however, from 22,000 m^2 to 18,000 m^2.

Background filtering seems adequate with current algorithms. I looking into a number of new ideas for fitting algorithms, including neural networks.

I have not yet re-done the optimization, to see if the proposed geometry is still the best in the presence of background. My guess is that is is.