EMTiming Simulation

Eunsin Lee, M. Goncharov,
D. Toback, *P. Wagner, and **V. Krutelyov

Texas A&M University
*University of Pennsylvania
**U.C. at Santa Barbara

Outline

- A New Webpage for Full Description of EMTiming Simulation Instruction:

  http://hepr8.physics.tamu.edu/elee/EmtimeSimul.html

  http://www-cdf.fnal.gov/internal/physics/photon/emtiming.html
- More detailed technical information:

  CDF NOTE 7928

  by M. Goncharov, David Toback, P. Wagner, V. Krutelyov
- Introduction
- Monte Carlo(MC) Generation and Reconstruction
- Coding Details: MC Framwork and Modules
- Example Results for $W \rightarrow e\nu$ Samples

Introduction

- EMTiming System is Simulated using MC that is run, independently, after event generation (PYTHIA) and detector simulation (cdfSim)
- Goal is to reproduce the arrival time of single particles and handle with all MC particle types as in real data
- Takes into account physics effects like vertex position and event time
- Correct for energy slewing effects as well as channel-by-channel energy threshold
- Can be used in the search for long-lived particles decaying to photons as well as the study of prompt decays (W→eν)

MC Generation and Reconstruction

- 7 steps of the process of turning MC events into TDC readout

- 1. Calculate the true arrival time and correct for the time of flight(TOF) and vertex time

$$t_{arrival} = t_{vert} + \frac{|\vec{x}_f - \vec{x}_{vert}|}{|\vec{v}_{part}|}$$

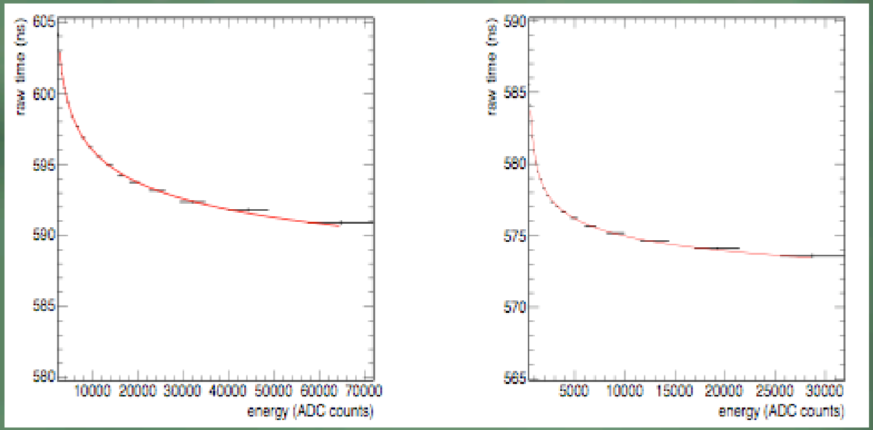
where $\vec{\mathrm{v}}_{\mathrm{part}}$ is the velocity of the final state particle

$\vec{\mathrm{x}}_{\mathrm{vert}}$ and t_{vert} are the position and time of its parent - particle decays

$\vec{\mathrm{x}}_{\mathrm{f}}$ is the position where the final - state particle interacts with the detector

For example, for neutralino decaying into a photon and a gravitino, tvert is the time the neutralino decays

- To get TDC, run the slewing calibration in reverse.



The Functional Form of the slewing correction from calibration table as a function of energy sum of the PMTs of an example tower in the CEM and PEM with the data

- These calibrations assume that particles come from the center of the detector, $t_{arrival}$ is corrected for the TOF of the particles:

$$t^{corr}_{arrival} = t_{arrival} - \frac{|\vec{x}_f|}{c}$$

- This is the corrected arrival time used to get the TDC time after smearing for the various effects

- 2. Check to see if it hits an EMTiming-instrumented part of the detector
- 3. Smear the corrected arrival time for the intrinsic resolution
- 4. Check to see if it has energy to create a hit
- 5. Convert the generated arrival time to a raw time using the ASD(Amplifier Shaper Discriminator) slewing curve
- 6. Truncate the raw time to an integer to simulate the TDC
- 7. Put the raw time into the *EMTD_StorableBank*

Coding details :

MC Framework and Modules

- The EMTiming MC modules are in *CalorTimeMods* package
- The executable part is in *test/McTiming.cc*
- The library part is *src/EMTDBankSimModule.cc* and *CalorTimeMods/EMTDBankSimModule.hh*
- The binary is run after MC files are generated independently with *cdfSim*
- The EMTiming information filled in the *EMTD_StorableBank*

- Both reconstructed particle information from *CalData* and true information from *OBSV_StorableBank* and *OBSP_StorableBank* are used as input

- More detailed information on *run_McTiming.tcl* and how to run :

http://hepr8.physics.tamu.edu/elee/EmtimeSimul.html

Example Results for 30k W→eν events

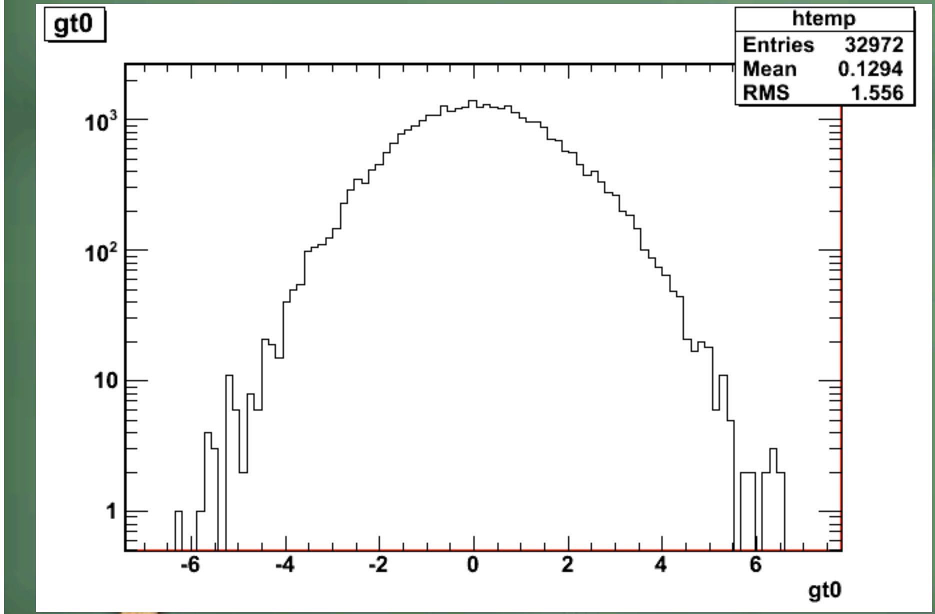
- After running EMTiming simulation we have the event data file as well as TimingNtuple.root file

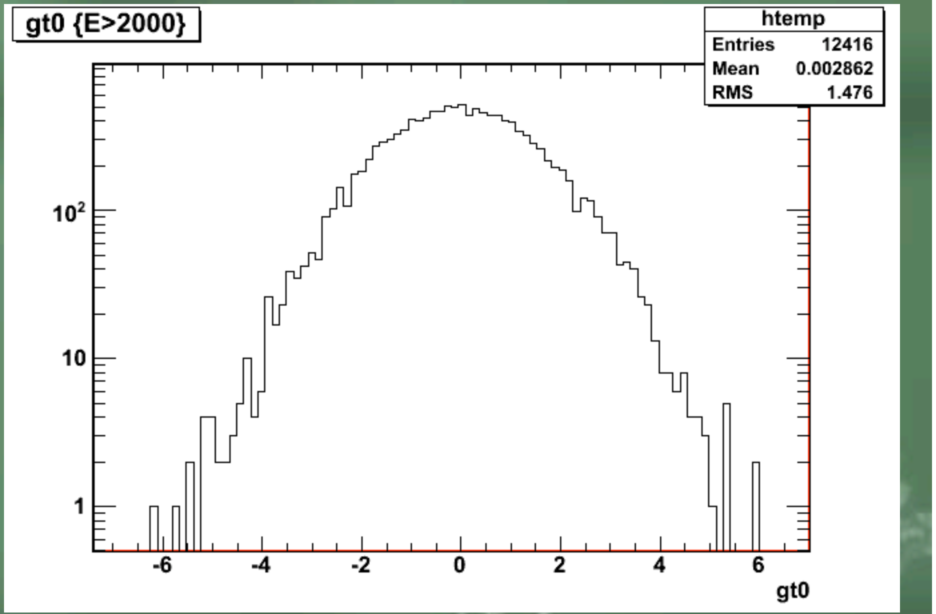
- Inside this root file are two Trees: sim and gen

→ The gen Tree is from the *OBSP_StorableBank* and is true information for ALL particles

→ The sim Tree contains the timing information for only those particles that have hits

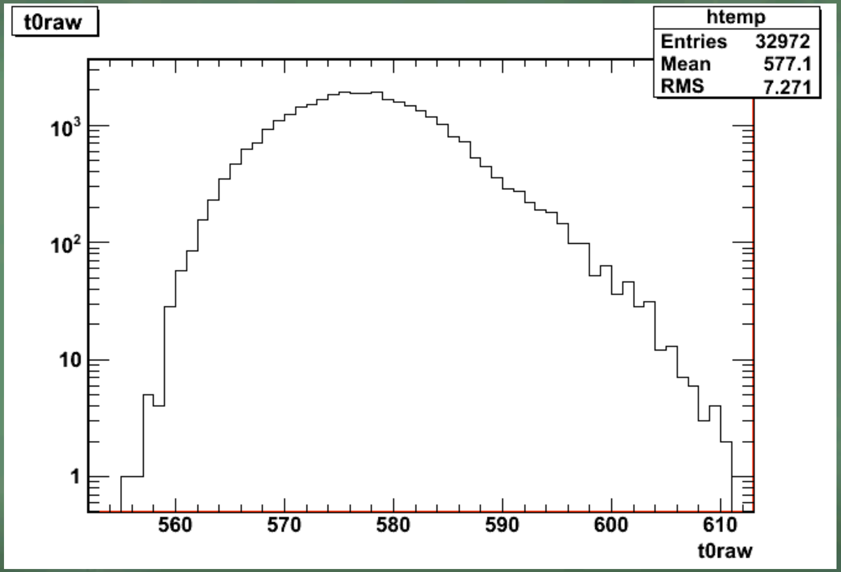
- Generated time plots : The corrected arrival time filled as the gt0 variable in the sim Tree





- The mean shift(~0.1 ns) caused by slow(delayed) heavy particles. For high energy particles(> 2GeV) the mean is centered at zero.
- The RMS(1.47 ns) is from variation of primary vertex t0 = 1.3 ns and TOF variation = 0.6 ns

- Raw time plot : The raw time filled as the t0raw variable in the sim Tree.
- After all seven steps we are left with a set of towers with a “raw time”



- The values range from 560 ns to 590 ns for prompt particles with RMS ~0.8 ns due to energy slewing, the vertex t0 variation, etc.

Summary

- EMTiming simulation works very well and used in Delayed Photon PRL & PRD

- Webpage instruction is available for users to learn how to run the EMTiming simulation

- Full description of EMTiming Simulation Instruction:

  http://hepr8.physics.tamu.edu/elee/EmtimeSimul.html

- Back-Up Slides

Coding details :

a code snipit of *run_McTiming.tcl* file

- include files yourproduction.dst
- module talk EMTDBankSimModule
- genericCast set f : only adds the EMDT bank, but not fill anything
- fillDetector set CEM PEM : add timing info to particles traverse these detectors
- smearTime set t : generated arrival time randomly varied by a Gaussian wit a sigma set by “timeSigma”
- timeSigma set 0.505
- useEffCalibs set t : use calibration tables to estimate the effciency as a function of energy
- dbName set TEXT : use the slewing calibration data of the local calibration tables at the lacation “calibDirName”
- calibDirName set yourcaliblocation
- Algorithm set 2 : slewing calibration algorithm

How to Run

- 1. source ~cdfsoft/cdf2.cshrc
- 2. setup cdfsoft2 6.1.4
- 3. newrel -t 6.1.4 MC_Timing
- 4. cd MC_Timing
- 5. addpkg CalorTimeMods
- 6. gmake CalorTimeMods.lib
- 7. gmake CalorTimeMods.tbin
- 8. bin/Linux2_SL-GCC_3_4/McTiming
     CalorTimeMods/test/run_McTiming.tcl