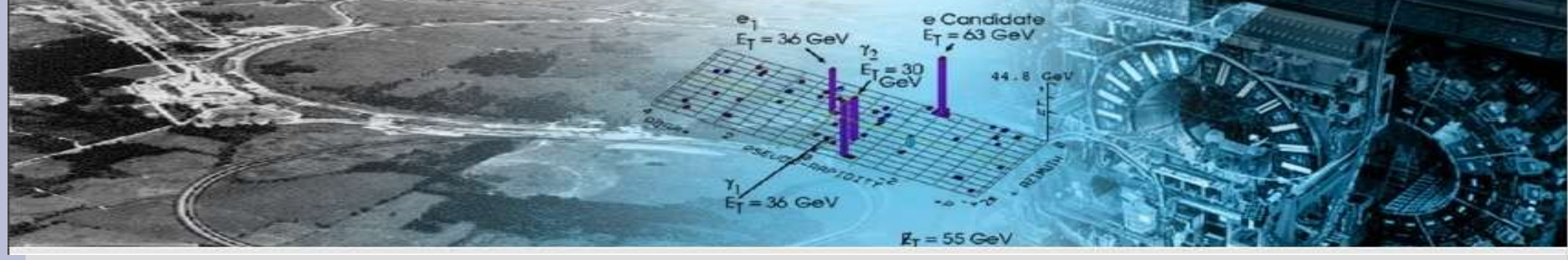


Monte Carlo Simulation for the EMTiming system

Peter Wagner
Texas A&M University

Outline

- Overview
- Goal
- Plots/Results
- Conclusion

Overview

- Monte Carlo is generated with cdfSim and then run through Production
- After that, run a module to simulate EMTiming (see next slide)
- Advantages:
  - No conflicts with calorimetry simulation
  - Can run EMTiming simulation on Monte Carlo files after Production (independent; takes less time)
- The idea is to simulate the correct physics, not to simulate/debug hardware issues ⇒ check the MC results by comparing to Z and W samples from real data.

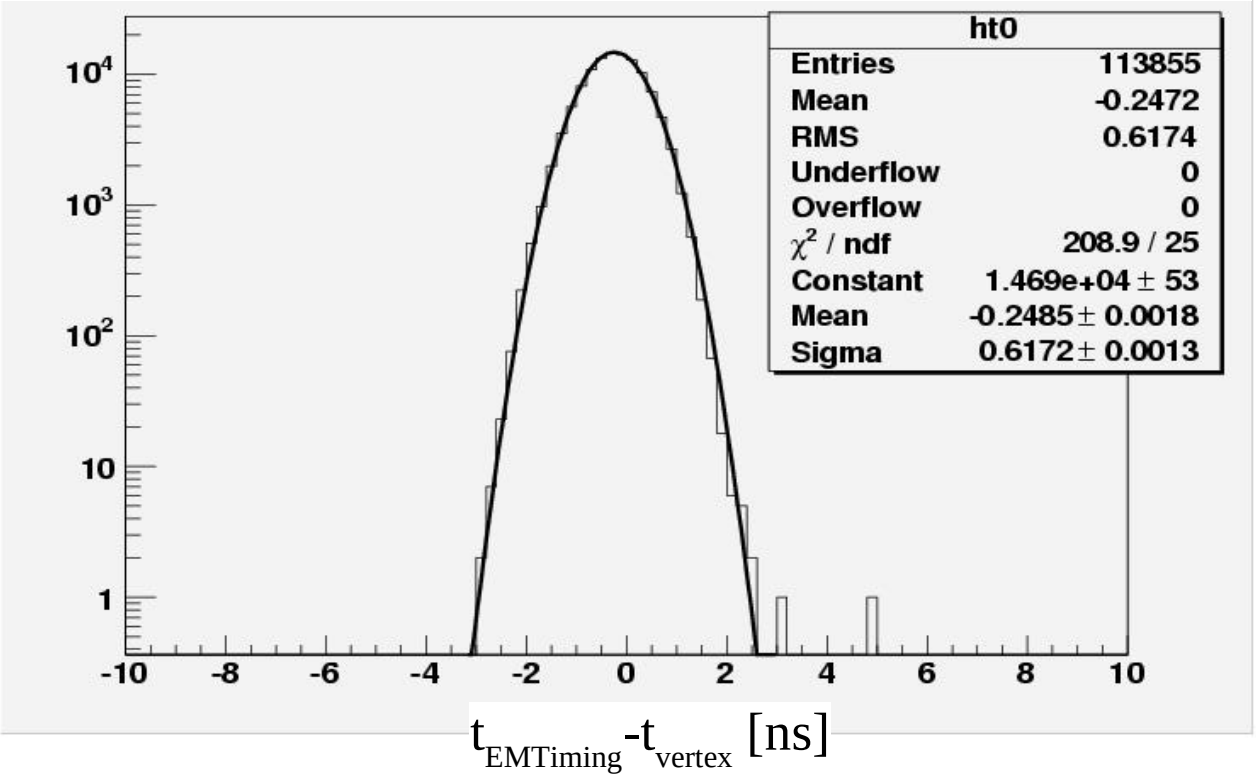
Overview II

Will walk through the steps of Monte Carlo generation:

- Generated time for one EM-tower / for multiple EM-towers
- Conversion to "raw time" which is filled into the timing StorableBanks using calibration tables from data:
  - Use threshold calibration tables to simulate the energy turn-on for each tower (negligible run dependence)
  - Use slewing calibration tables to convert the Monte Carlo arrival times to a time format that resembles the one in the TDCs from data. This is not a simulation of the TDCs but it only alleviates further data processing.
- Reconstructed time from a $Z \rightarrow ee$ Monte Carlo sample
- Treatment of real effects (z-position of vertex, event t_0)

Goal

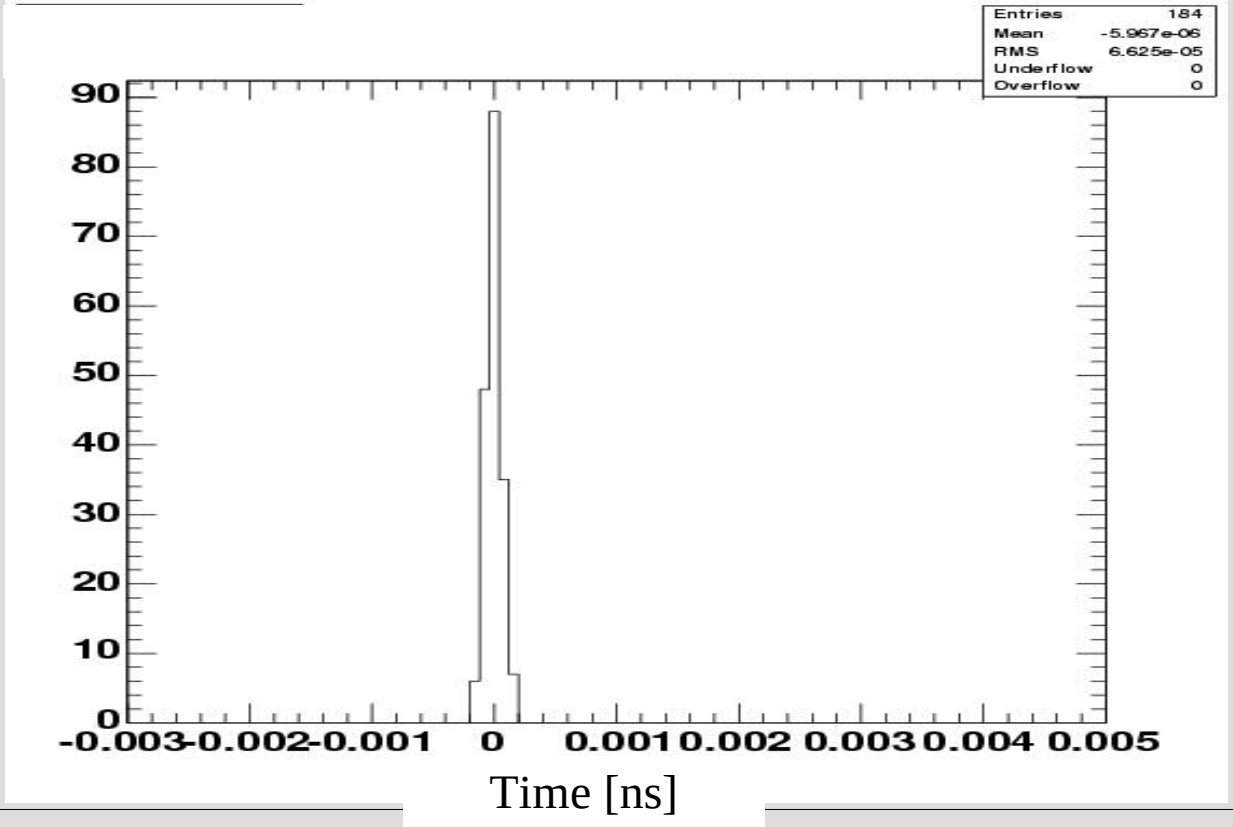
- At the end the simulation should look like data: Vertex-corrected arrival time of e's from a $W \rightarrow e\nu$ sample. $\sigma = 0.61$ns



Generated Time

Time at generator level for one tower with collision point/vertex fixed in position and time (Single Particle Gun) for an energy range of 4GeV-80GeV, corrected for the time of flight:

Centered at 0ns, Negligible RMS

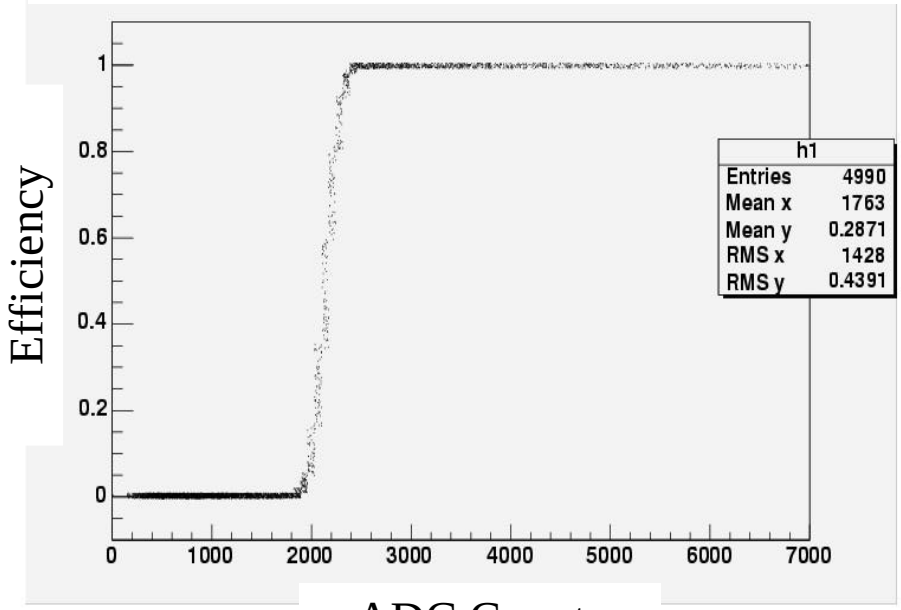


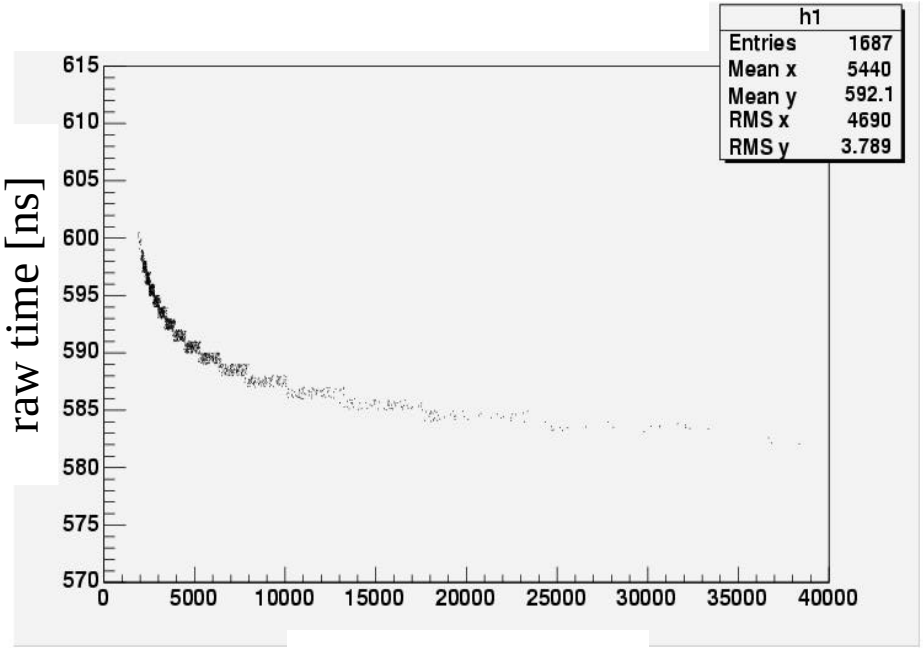
Calibration Information

Example of a Threshold curve and Slewing curve for one tower.

Both taken from the calibration database ⇒ threshold simulation and slewing energy correction work well

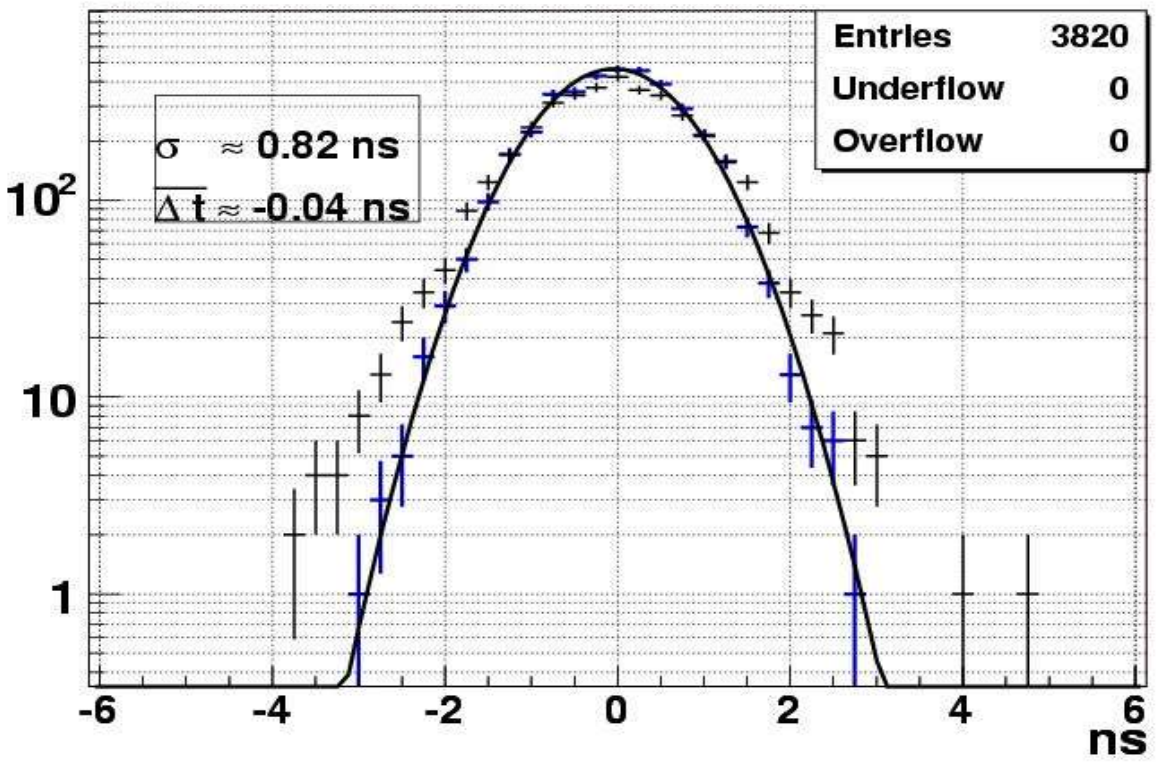
Raw time out of the TDC in ADC Counts, 1000 ADC Counts ≈ 3 GeV





Resolution from Data

- EMTiming resolution taken from data: time difference between the electrons of $Z \rightarrow ee$
  $\sigma(t(e_1)-t(e_2)) = 0.81\text{ns} \Rightarrow \sigma_{EMTiming} = 0.58\text{ns}$ (Checked with W's)



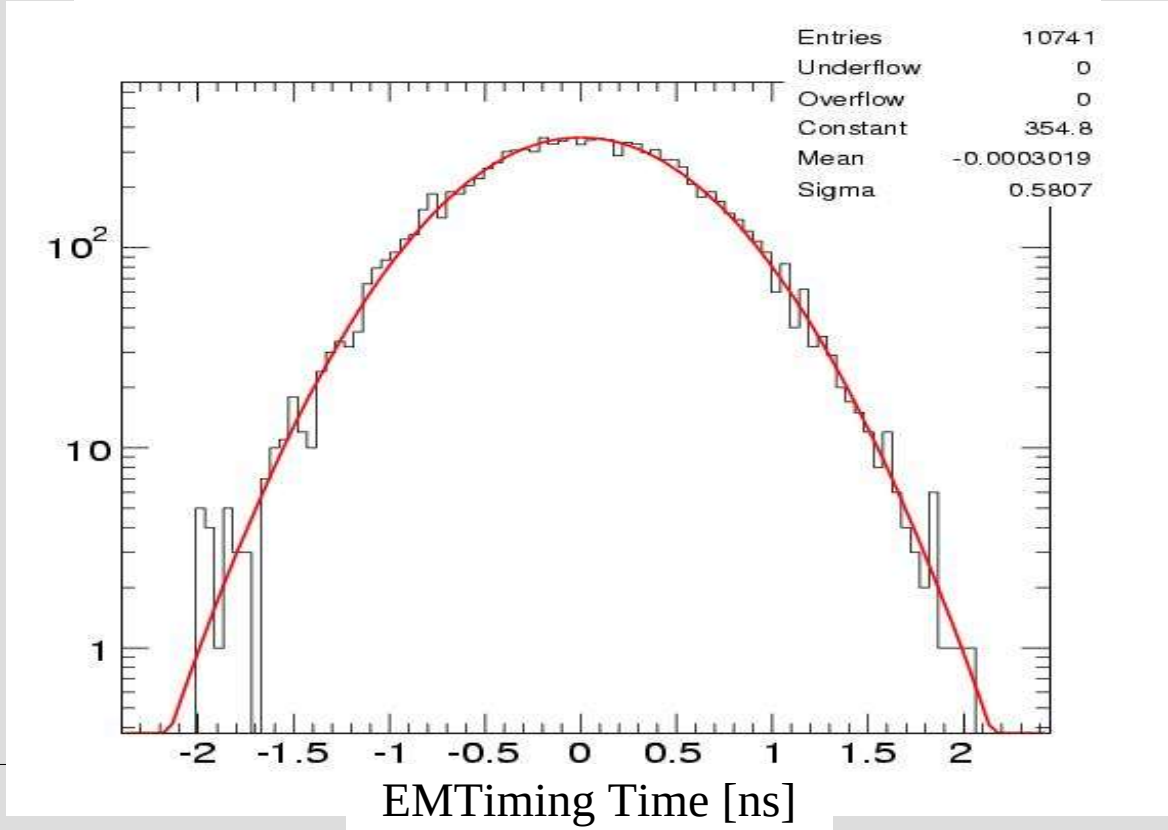
EMTiming Time [ns]

Resolution in MC

$\sigma_{EMTiming} = 0.58ns$

- 0.28ns of which are from rounding to integers in the TDC
- The rest of the resolution is not understood ⇒ implemented 0.50ns gaussian smearing in quadrature:

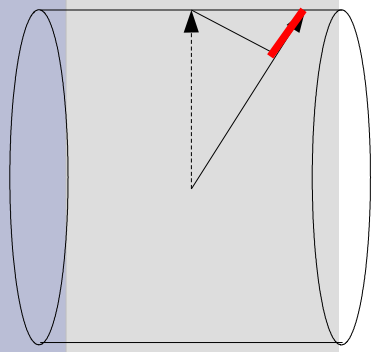
Single particle gun, one tower, vertex fixed

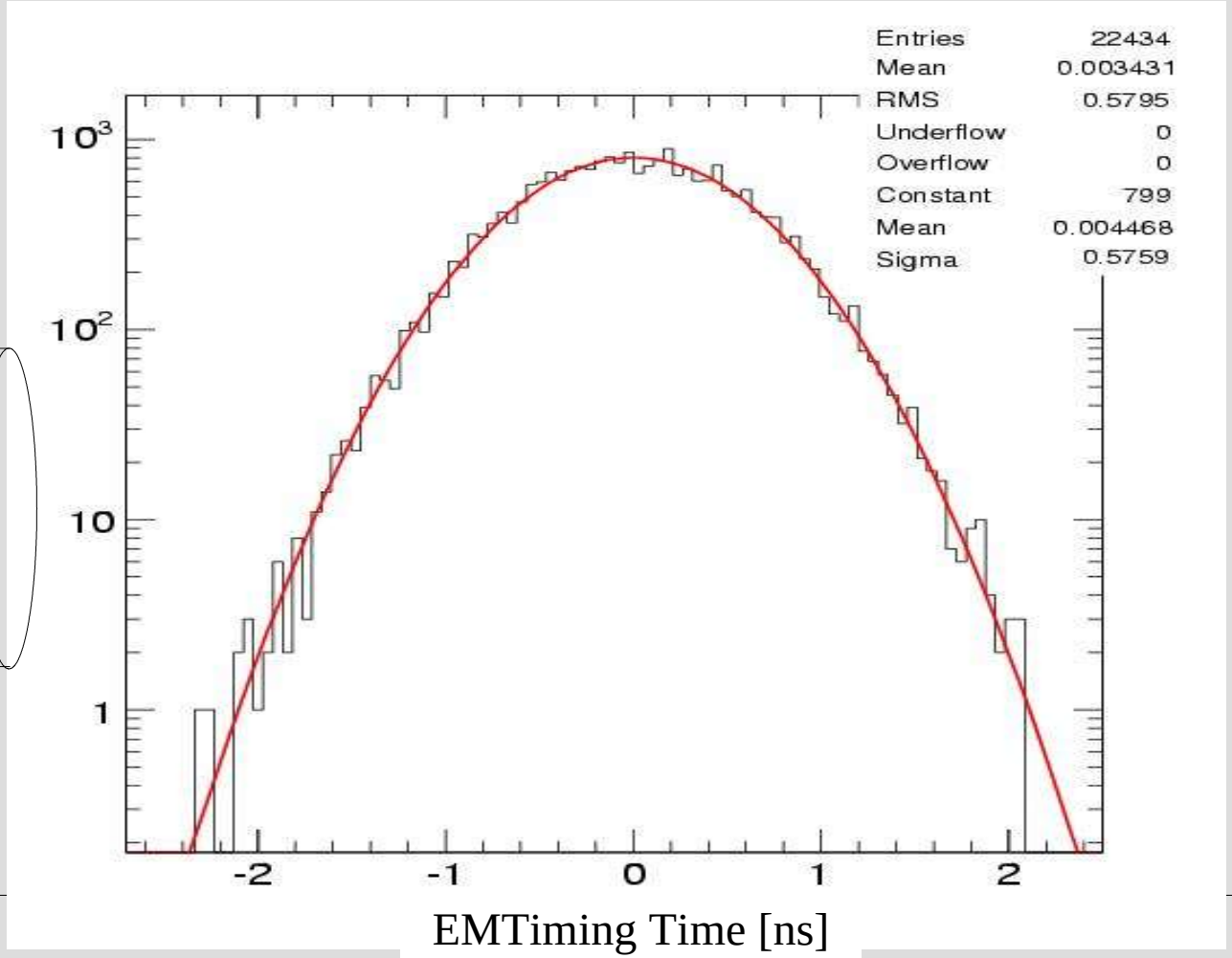


Results for Z → ee

Now the same plot for all towers. One has to compensate for the distance of the towers as a function of η

⇒ Same result (from a Z → ee MC sample, vertex fixed):

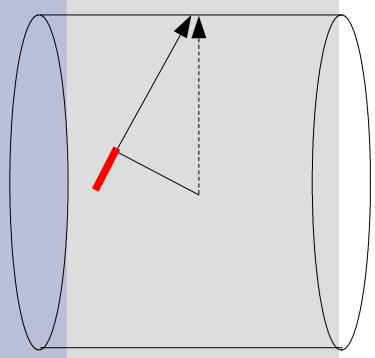


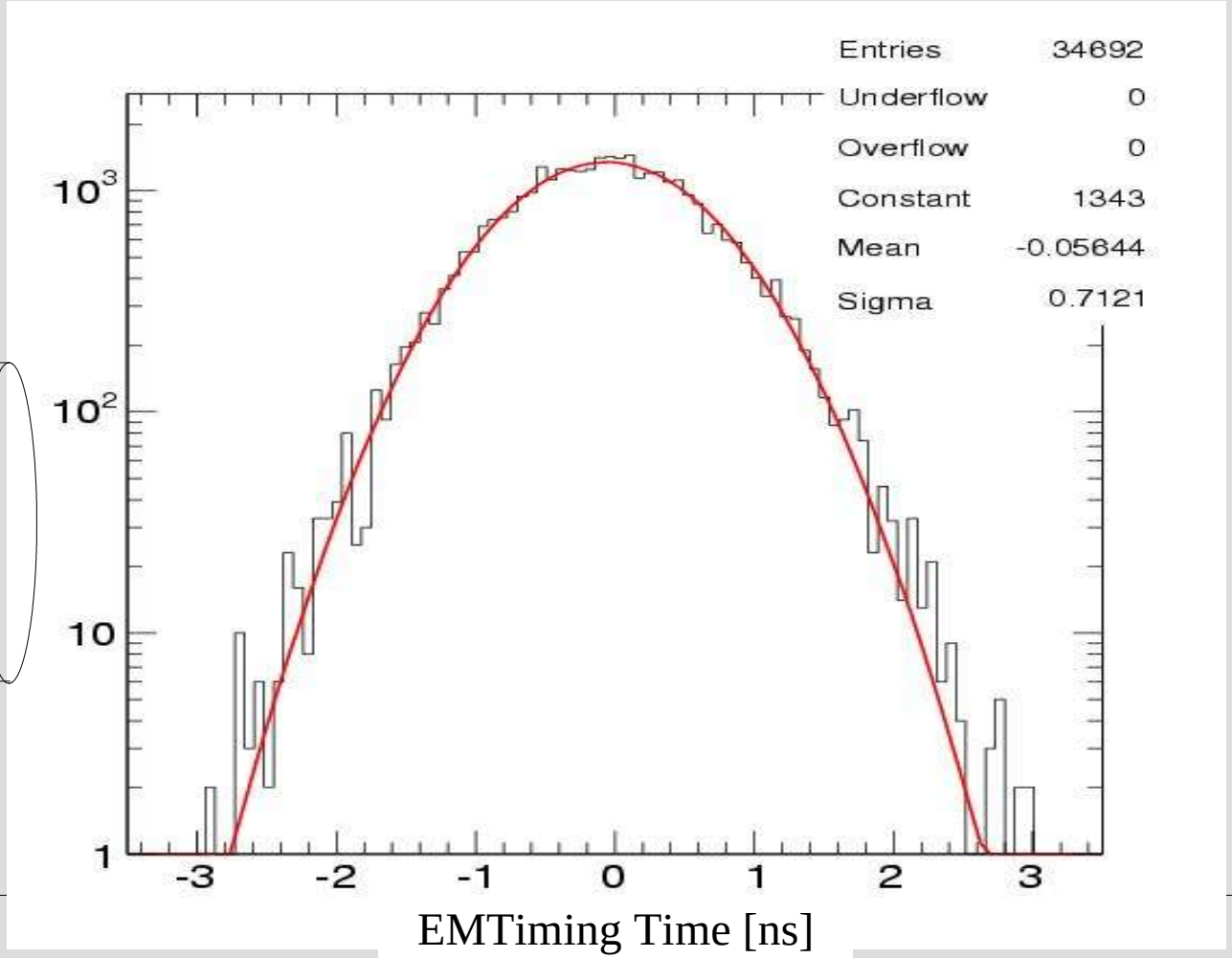


Results for $Z \rightarrow ee$

Now the same plot for all towers, vertex t_0 = 0ns, but allowing the vertex position to vary ($\sigma_{\text{vertex pos.}}$ = 28cm). $\Rightarrow$ uncorrected σ = 0.71ns

Need to compensate for the shifted z-position of the vertex...

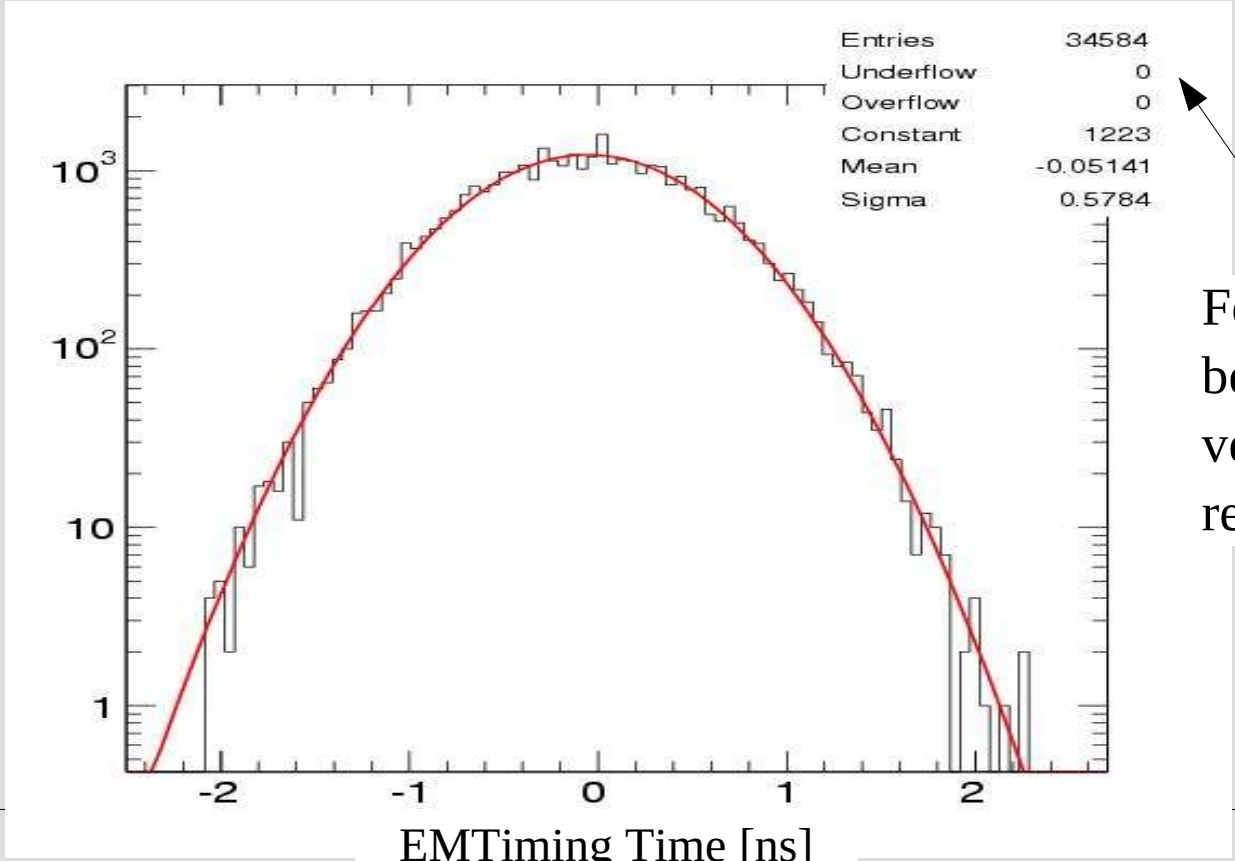




Results for $Z \rightarrow ee$

- Vertex information recontructed with Max's modified Zvertex collection.
- After compensation: $\sigma = 0.58$ns

($Z \rightarrow ee$ MC sample, realistic vertex simulation, event $t_0 = 0$ns)

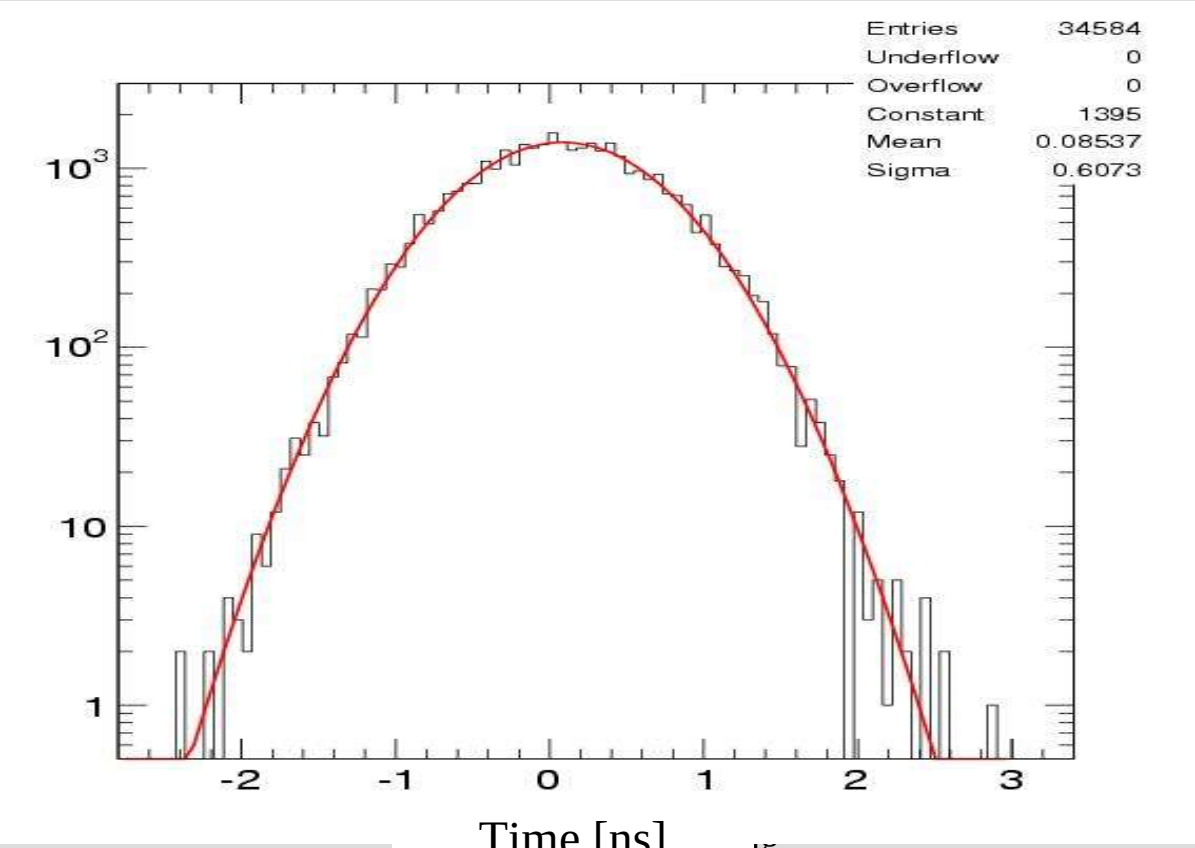


Fewer entries because of vertex finding requirements.

Fully Corrected Time

- Now plot the fully corrected arrival time using the $Z \rightarrow ee$ MC sample. Corrections: (1) vertex z-displacement and (2) vertex/event t_0.
- Both reconstructed using Max's modified Zvertex collection.

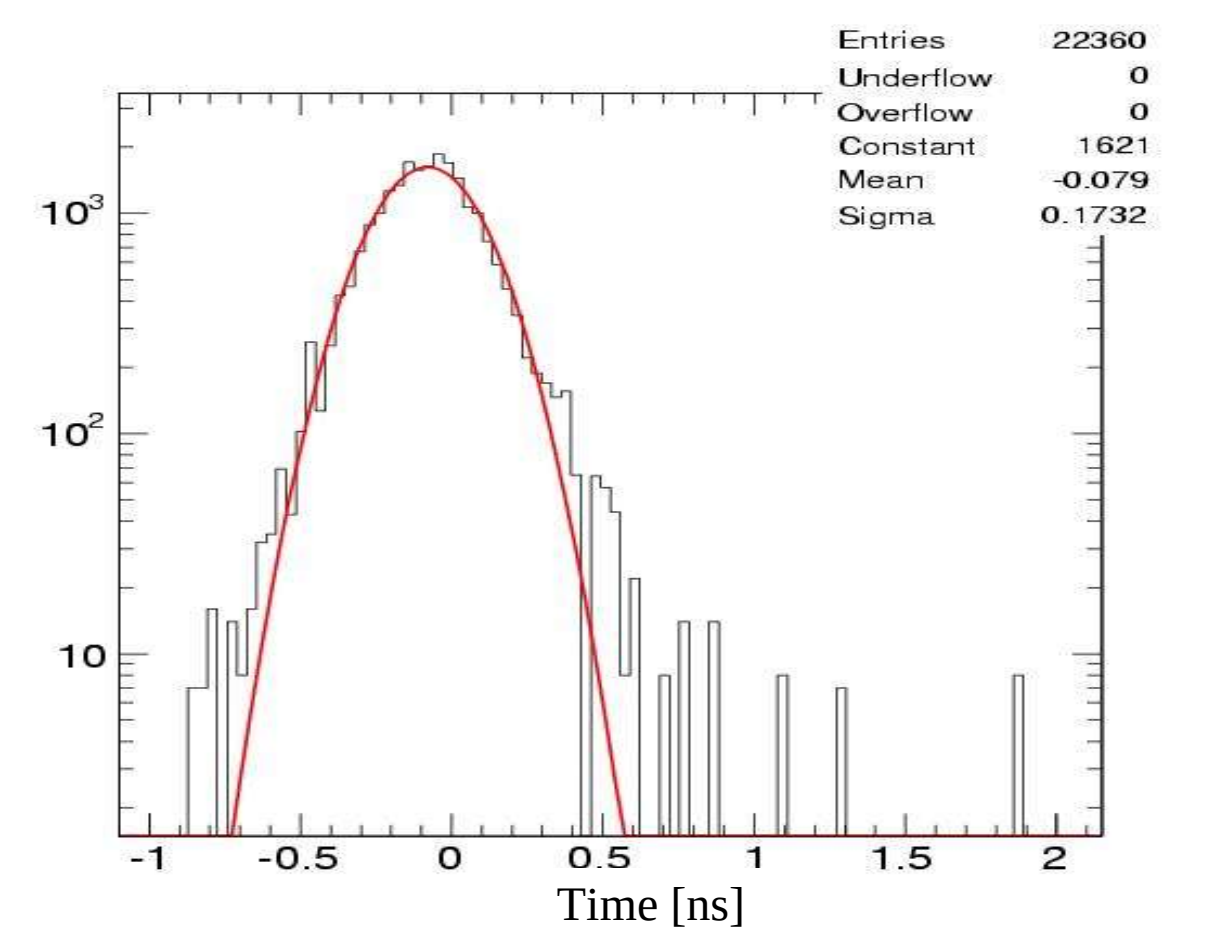
$$\Rightarrow \sigma_{EMTiming} = 0.61\text{ns}, \text{ not } 0.58\text{ns}.$$



Remaining RMS difference

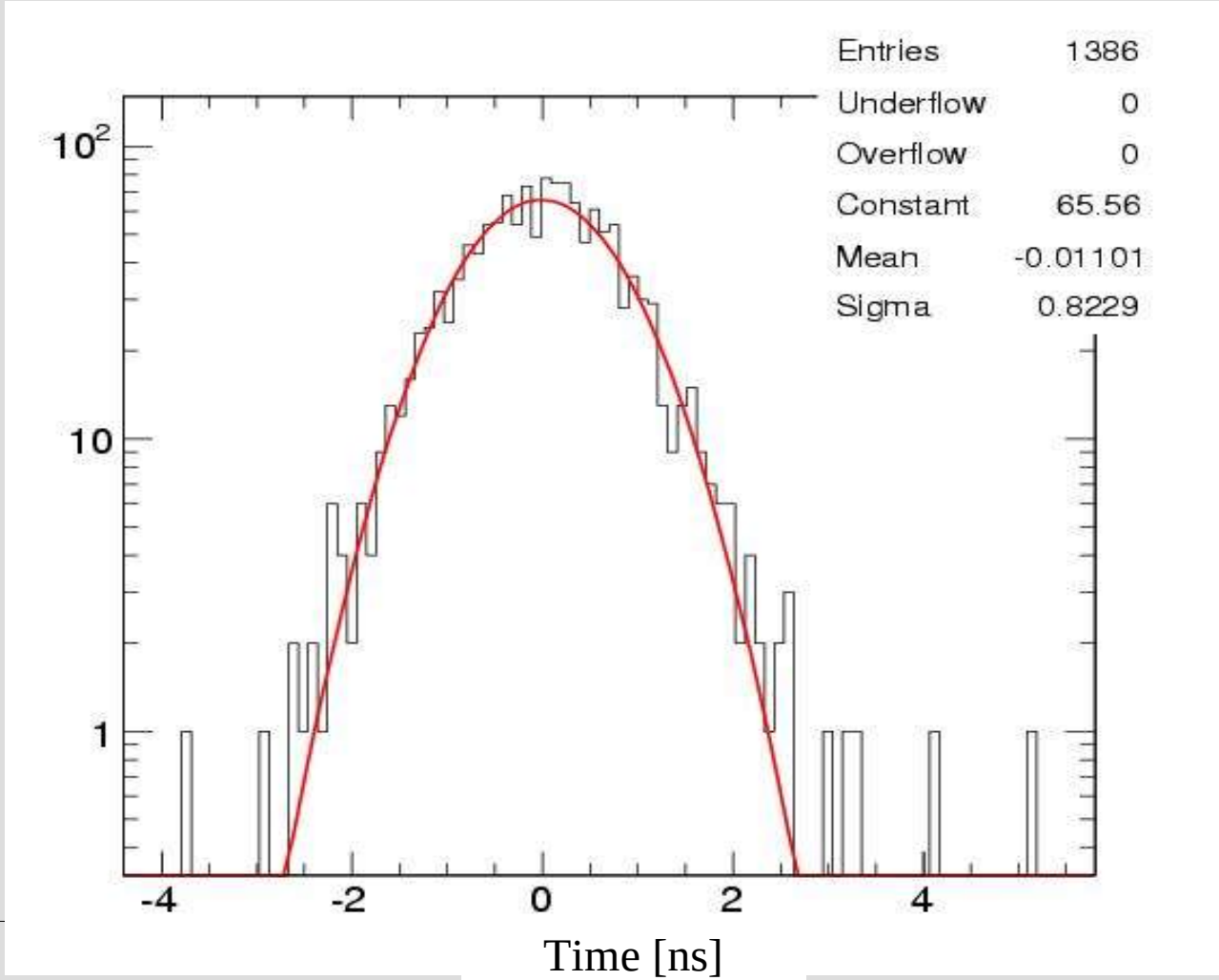
Vertex t_0 measurement resolution (vertex point was simulated at (0,0,0,0): σ_{vertex} = 0.17ns (from the Z $\rightarrow$ ee MC sample with fixed vertex)

$\Rightarrow$ 0.17ns $\oplus$ 0.58ns = 0.604ns. Close enough...



Results

Check: Plot $t(e_1)-t(e_2)$; should be ≈ 0.81ns (after vertex-z correction).



Conclusion & Plan

- It works
- Plan:
  - Commit code to CVS
  - Write up a CDF note
  - Show at lots of meetings for final vetting
  - Start simulating long-lived particles

Backup Slides

Modules

Separate module that is run on on Production output:

Files in:

CalorTimeMods/src/TDCBankSimModule.cc
CalorTimeMods/CalorTimeMods/TDCBankSimModule.hh
CalorTimeMods/test/McTiming.cc
CalorTimeMods/test/run_McTiming.tcl

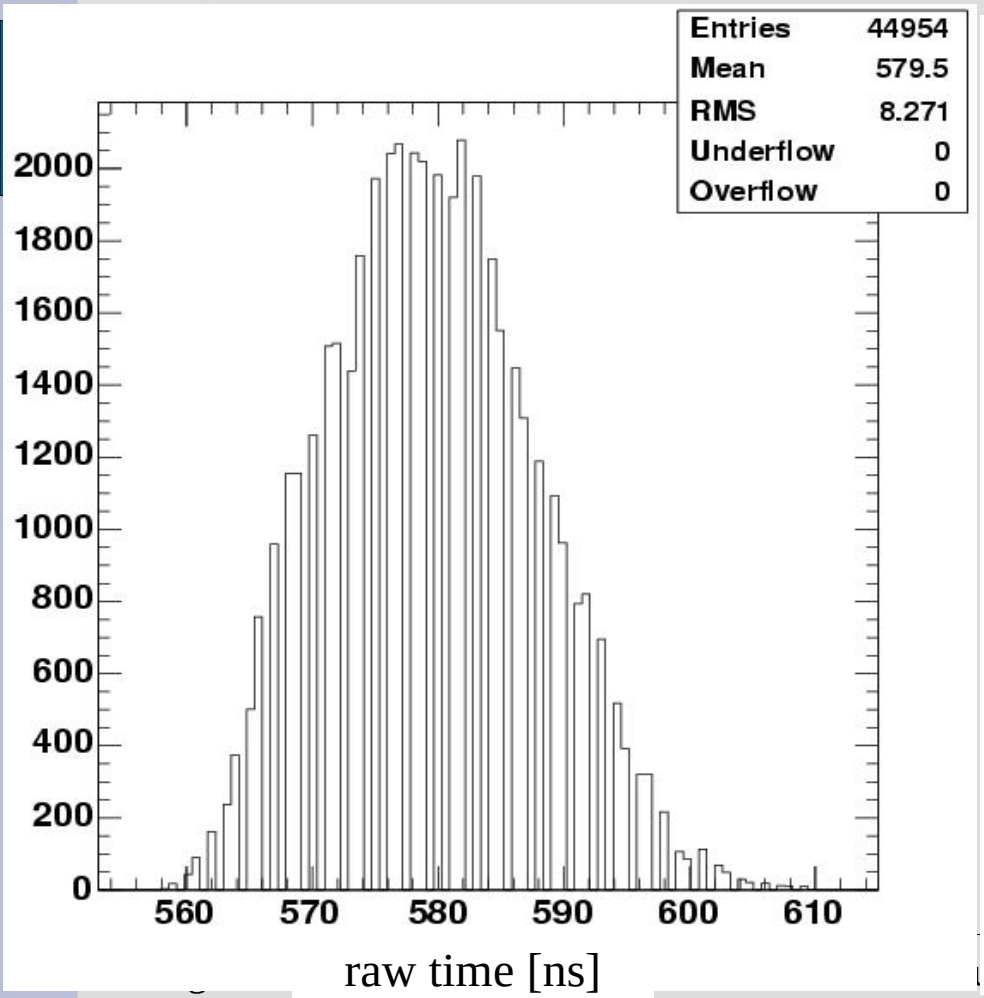
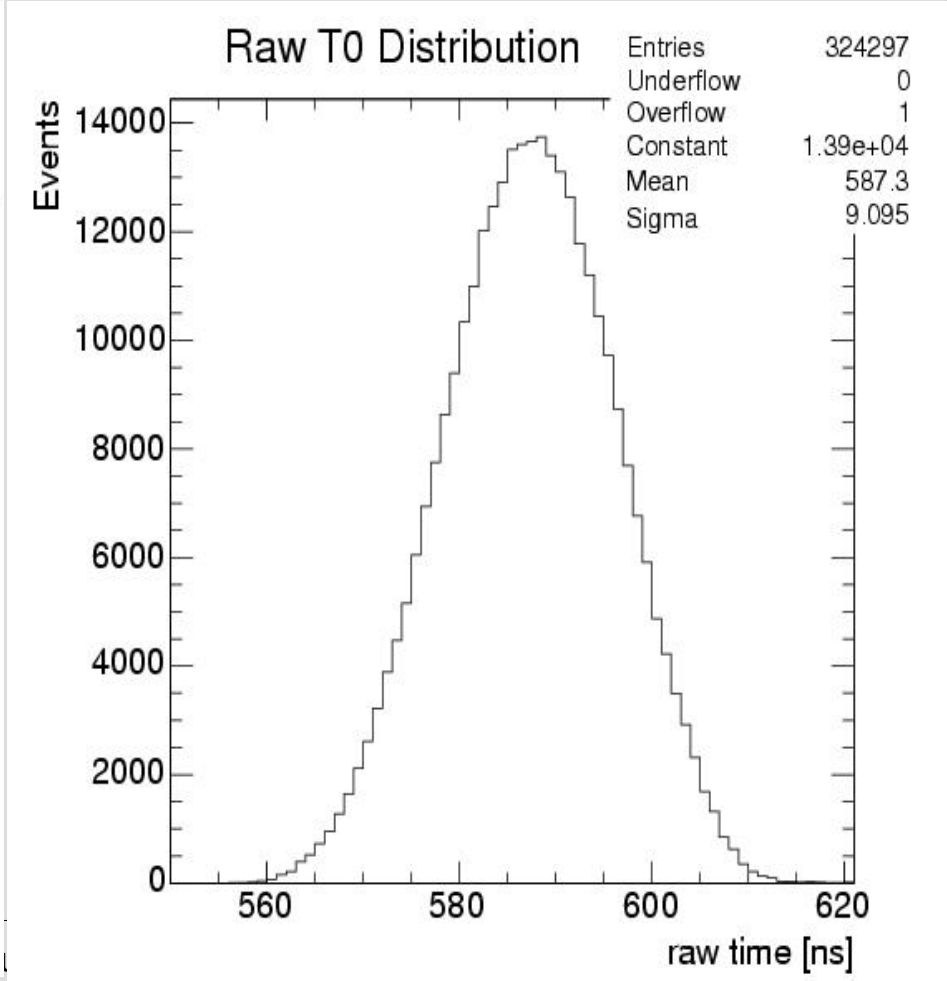
(not yet committed)

Usage:

gmake CalorTimeMods.nobin; gmake CalorTimeMods.tbin
McTiming CalorTimeMods/test/run_McTiming.tcl

Implementation

- Create EMTD_StorableBank and fill it with timing information for each calorimeter tower that has energy deposited above a threshold of ~2 GeV (Plug) and ~3 GeV (Central)
- Use MC-particle information (arrival time) and caldata information (energy deposited; needed for calibrations)
- Drawback: for multiple hits in the same tower cannot say if a particle deposited energy and if yes, how much. Simplification: Take only the first timing hit (e.g. in case of jets). Not an issue for e.g. photons, electrons, muons.
- Use threshold calibration tables to simulate the energy turn-on for each tower (negligible run dependence)
- Use slewing calibration tables to convert the Monte Carlo arrival times to a time format that resembles the one in the TDCs from data. This is not a simulation of the TDCs but it only alleviates further data processing.

Results

Check: Compare the "raw time" from the **Z → ee MC** to the online monitoring results from ObjectMon (uses b-stream: high-Et jets). Close enough...