Progress Report #2

Daniel Gawerc
Mentor: Maria Spiropulu
Co-Mentor: Cristian Peña

California Institute of Technology

29 July 2016

1 Background

At the Large Hadron Collider (LHC), the High Luminosity upgrades plan to achieve luminosities of greater than 10^{34} cm^{-2}s^{-1}, with 140 – 200 pileup events per bunch crossing every 25 ns [1]. Because of the physical spread of each bunch, the collisions occur over a large region rather than at an isolated point, which leads to a spread in time ($\sim$ 200 ps) [1]. Thus, detectors in the calorimeter must have a good (low) time resolution in order to resolve individual collisions. In order to reconstruct the vertex of a collision with an increased pileup, the electromagnetic calorimeter (ECAL) detectors should have a time resolution of about 10 ps [1].

2 Experimental Setup

This summer, the Caltech Precision Timing group took data at the Fermilab Test Beam Facility (FBTF). The experiment involves placing different calorimeter detectors in the beam line, in order to observe how the addition of detectors in the perpendicular transverse plane and/or parallel longitudinal direction along the beam line will improve the overall detector time resolution. The layers of detectors are meant to mimic the proposed High Granularity Calorimeter (HGC) detector, which shall replace the CMS ECAL endcap, and is described in detail below [2]. The results of this research may be useful in preparing the LHC High Luminosity upgrades.

In the FTBF experiment, different types of detectors were utilized. In the front of the setup, a scintillator crystal and PMT are used as a trigger. Depending on the configuration, there are a few radiation lengths (X_0) of either lead or tungsten absorbers behind the PMT in order to spread out and propagate the particle shower and produce more secondary electrons and photons. A depiction of the setup is given in Figure 1.

The High Granularity Calorimeter (or PicoSil when referring to a single pixel's readout that has been optimized for timing) layer has an array of separated silicon pixels, in a

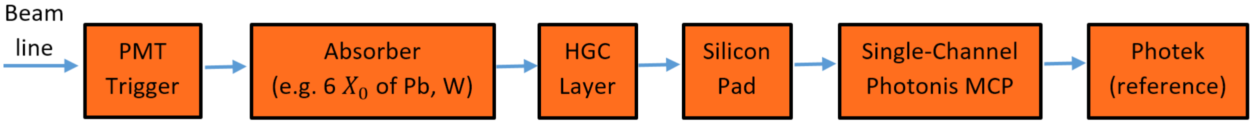

Figure 1: A typical experiment configuration.

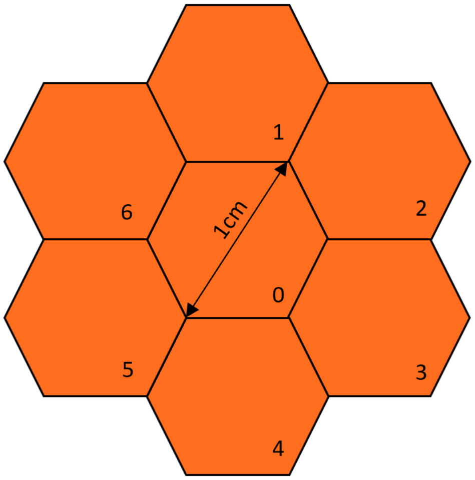

Figure 2: A depiction of the innermost 7 pixels of the HGC layer. The actual board has more than 7 pixels in a single layer.

hexagonal honeycomb geometry [2]. A depiction of the HGC layer is given in Figure 2. Because the beam is focused at the HGC center pixel (0 in Figure 2), its signal is reduced using a 6 dB attenuator due to the high charge detected relative to other pixels.

The Photek 240 MCP is used as a reference detector because its time resolution is good ($\sim$ 10 ps). Because the readout board for all the detectors is divided into 4 groups, which have some time delay between them, the Photek signal is split into 4 cables and plugged into each group as the reference signal from which the TOFs (time of flight, or Δt) are calculated. Specifically, the time taken for the particle to travel from a device to the reference Photek - the time of flight – is represented by Δt. In the data analysis, the Δt values over all events in the run (with certain *bad* events excluded) are combined in a histogram, forming a Gaussian peak. The peak σ parameter is denoted the time resolution.

There is a 64-channel and a single-channel Photonis MCP. The 64-channel Photonis was not present for most runs, and the single-channel Photonis was instead used. Lastly, a silicon pad (SiPad) detector was present for most runs, however it only detected pulses well in the very early runs, and does not seem to be useful in most of the analysis.

3.1 Data Analysis: Overview

The Caltech Precision Timing group already had developed some analysis code for previous test beam experiments. The program *dat2rootCP* analyzes the .dat files (with low-level information) from the runs and turn them into .root files containing a tree with plots and fit results (higher-level information). Since some of the data contained ringing noise, which often resulted in the wrong peak being identified as the pulse, *dat2rootCP* was updated to correct for these events. Additionally, new macros were written in order to automate batch file analyses and merging the results into a single .root file (for when the setup configuration and beam type were the same over many runs). These programs utilize the ROOT programming language, which is based largely on C++. It is the main programming language used by scientists at CERN and in the field of HEP (see: root.cern.ch). The .root file extensions are associated with the ROOT language.

Because the HGC layer has multiple pixels, each pixel detects different signals and has a different time resolution. The *MultiChannelStudy* code was created (here channel and pixel are used interchangeably), which calculates and combines the TOF (Δt) histograms for different pixels in order to return a better overall time resolution. The Δt values are calculated for every event in a run by subtracting the pulse time in one device from the pulse time in the Photek (making sure that it is from the same cabling group 1-4 as the device). A 1-D histogram is then filled event-by-event with the Δt values. In an ideal world with perfect detectors and no variation in each event, the histogram would be a Dirac delta function with a time resolution of zero. However, due to detector jitter, the signal-to-noise ratio and the particle shower development in the material will vary event-by-event, resulting in Gaussian histograms [2]. After fitting the histogram to a Gaussian, the σ parameter gives the time resolution of the detector/pixel. If weighted correctly, combining the HGC pixels' Δt histograms will result in a combined histogram with better time resolution (smaller σ). This improvement is due to the combination of non-overlapping information about the shower size and development in each of the detectors.

Because there are also multiple devices along the beam line as well as multiple HGC pixels, the analysis of the coplanar HGC pixels has been called *transverse analysis*, and the analysis of multiple devices *longitudinal analysis*. This report shall deal mainly with the longitudinal analysis: analyzing the improvement in the time resolution when taking the HGC layer, single-channel Photonis MCP, and SiPad into account. However since it is best to use the time resolution from the entire HGC layer, much of the transverse analysis will be included as well. Due to re-cabling the wires of the detectors into different groups between runs 52 and 53, multiple longitudinal analysis codes were written for each setup. Additionally, the silicon pad has noise dominating for most of the runs, and substantially deteriorates the combined detector time resolution. For example, Figures 3 and 4 give the time of flight (Δt) histograms for runs 65-83 (same configuration) for the HGC center pixel and the Photonis MCP, along with their Gaussian fit σ parameter. The Gaussian fit used the log-likelihood maximization method. Figure 5 gives the TOF histogram for the silicon pad. Figure 6 is one possible combination histogram (more details below on how the individual device Δt's can be combined). Clearly the combined histogram improves the silicon pad resolution by about 4 times, but it is 4-5 times *worse* than the individual resolutions of the HGC and MCP. Additionally, the silicon pad histogram is peaked, but certainly not Gaussian. Thus, because it is usually more favorable (in most - but not all - runs) to simply exclude the silicon pad, code that does not account for the silicon pad was also created.

Therefore, because of the change in cabling and the noise in the silicon pad, the following longitudinal analysis codes were used: *MultiDeviceStudy*, *MultiDeviceStudy_InitialWiring*,

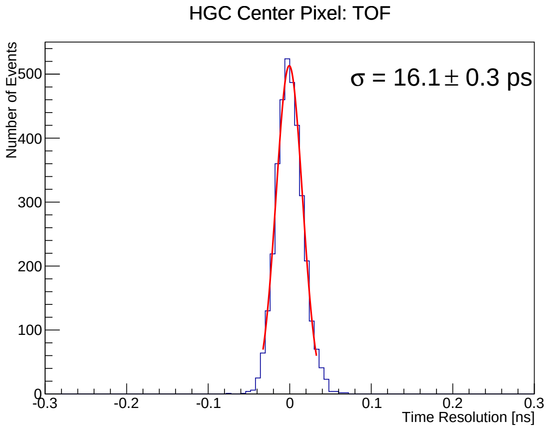

Figure 3: TOF of the HGC Center Pixel

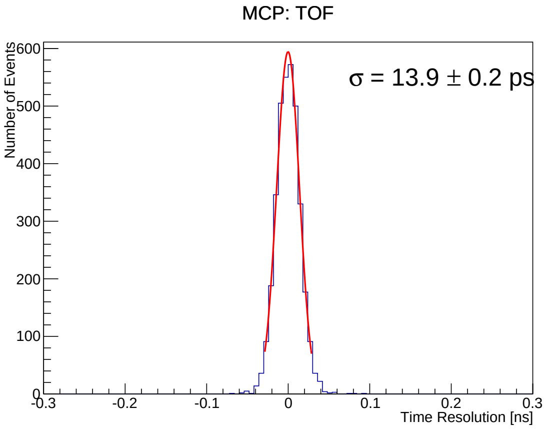

Figure 4: TOF of the Single-Channel Photonis MCP

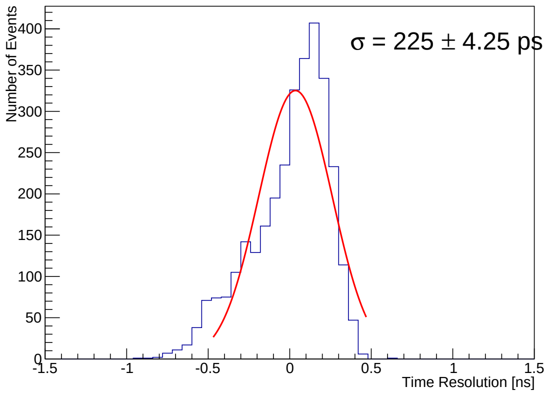

Figure 5: TOF of the Silicon Pad

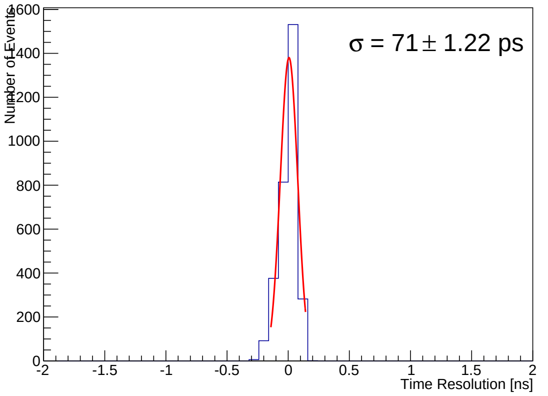

Figure 6: TOF of the HGC Center Pixel, Photonis, and Silicon Pad Combined with Total Device Charge Weighting

MultiDeviceStudy_PicosilMCP, and *MultiDeviceStudy_InitialWiring_PicosilMCP*. Because the most applicable file is *MultiDeviceStudy_PicosilMCP*, it is more extensive than the others, incorporating all of the ring 1 pixels (the 6 pixels surrounding the center pixel; refer to Figure 2) in the analysis. Thus, the code implements transverse analysis of the HGC as well as longitudinal analysis by adding the Photonis MCP.

3.2 Data Analysis: Event Selection

Before discussing the various ways to combine the Δt values from different detectors, the methodology for selecting events that contribute to their individual TOF histograms should be reviewed. Because not every event contains a detected pulse, and some have a lot of ringing noise and a small signal-to-noise ratio, some events need to be cut out of the Δt calculations.

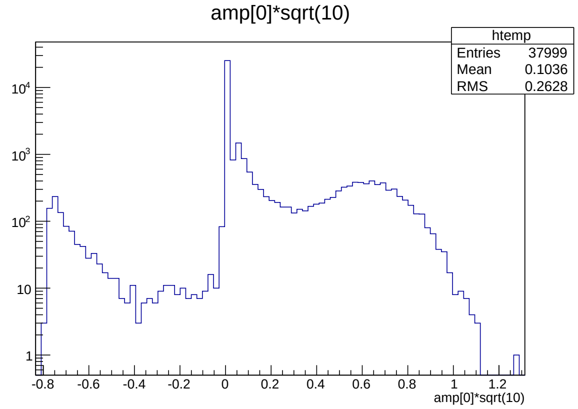

Figure 7: Pulse peak amplitude distribution for the Photek 240 MCP. The $\sqrt{10}$ factor accounts for a 6 dB attenuator.

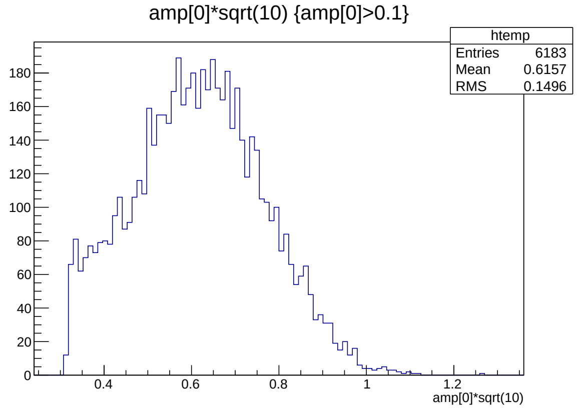

Figure 8: The Photek 240 MCP amplitude distribution with the cut in effect. Non-positive amplitudes represent *"bad"* events.

In order to ensure that most of the *"bad"* events are removed from the histograms, a cut is applied to the calculated peak amplitude (in mA) and to the overall charge (in pC) of the event in that detector, forcing them both to be higher than some threshold. This requires that the peak found is not just noise and that the charge (found by integrating the plot) is significant enough to represent a pulse.

For example, analyzing the data (with *dat2rootCP*) from runs 65-83 returns a maximum amplitude value in the Photek for every event. Filling a histogram with all of these values in the command-line in ROOT gives Figure 7. Clearly the amplitudes that are near zero or negative are events without pulses or very noisy events. These events should be ignored, so a cut is implemented to select for amplitudes above a certain value. In this case the value appears to be $0.1 \times \sqrt{10}$ mA. Figure 8 shows the histogram of the amplitudes with only the events that passed the cut. Figures 9 and 10 show the same methodology being applied to the charge distribution, with a cut at $2 \times \sqrt{10}$ pC. Similar cuts were made in the HGC pixels and Photonis MCP.

3.3 Data Analysis: Device Combination

The *MultiDeviceStudy_PicosilMCP* code generates many variations of TOF combination histograms. The simplest are the HGC center pixel and the single-channel Photonis MCP histograms (Figures 1 and 2). Another histogram (Figure 3) is the average value of the center pixel and the MCP Δt at every event. Figure 4 adds in the other 6 pixels of the inner ring of the HGC layer, and weights everything by taking the arithmetic average of the Δt's for the devices that passed the cuts.

Another histogram (Figure 5) averages all 7 of the HGC pixel Δt's equally, and then averages that value with the MCP:

$$\Delta t_i = \frac{\Delta t_{MCP_i}}{2} + \frac{\sum_{j=1}^{N} \Delta t_{HGC_{ij}}}{2 \times N}, \quad \textit{event } i, \quad N \textit{ HGC pixels passing cuts}$$

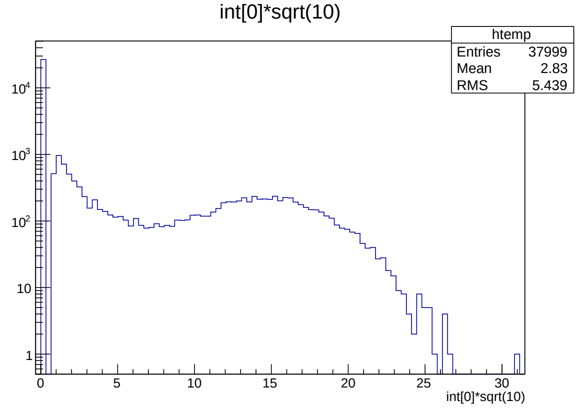

Figure 9: Pulse charge distribution for the Photek 240 MCP. The $\sqrt{10}$ factor accounts for a 6 dB attenuator.

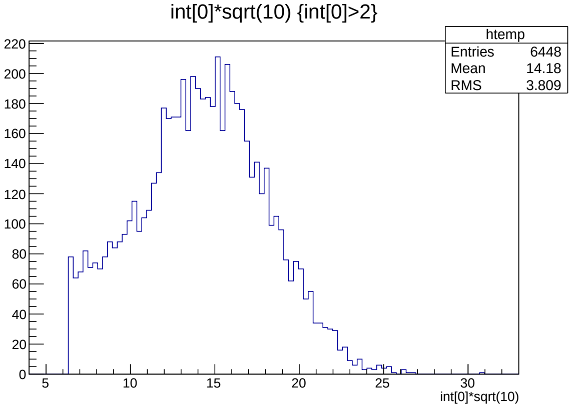

Figure 10: The Photek 240 MCP charge distribution with the cut in effect. Zeroed charges represent events with only noise.

However, the last two combinations (Figures 4/5) are not the best ways to combine the pixels in the HGC. Due to the electromagnetic nature of the particle beam, the shower spreads out as it evolves, so more particles will pass through some of the pixels than others [2]. As it turns out, the center pixel detects significantly more charge than the others, so it should be weighted more. There are two ways to weight the HGC combination by the charge: (1) At every event, compute the relative charge in every pixel and use that as the weighting factor, or (2) loop over all events and find the total charge deposited in a pixel over the entire run, then do this for every pixel and use those values as the weights. The first method will weigh the pixels differently at every event, whereas the second method will weigh the pixels the same way for every event.

Without including the MCP, two other histograms (Figures 6 and 7) plot the event Δt's using the event charge and total run charge weightings:

$$\Delta t_i = \frac{\sum\limits_{all\ pixels} \Delta t_{HGC_i} q_{HGC_i}}{\sum\limits_{all\ pixels} q_{HGC_i}}, \quad event\ i, \quad event\ pixel\ charge\ q_i$$

$$\Delta t_i = \frac{\sum\limits_{all\ pixels} \Delta t_{HGC_i} Q_{HGC}}{\sum\limits_{all\ pixels} Q_{HGC}}, \quad event\ i, \quad total\ run\ pixel\ charge\ Q$$

Utilizing these weighting methods, two other histograms (Figures 8 and 9) weight every pixel in the HGC and the MCP by their event charge and their total charge:

$$\Delta t_i = \frac{\Delta t_{MCP_i} q_{MCP_i} + \sum\limits_{all\ pixels} \Delta t_{HGC_i} q_{HGC_i}}{q_{MCP_i} + \sum\limits_{all\ pixels} q_{HGC_i}}$$

$$\Delta t_i = \frac{\Delta t_{MCP_i} Q_{MCP} + \sum\limits_{all\ pixels} \Delta t_{HGC_i} Q_{HGC}}{Q_{MCP} + \sum\limits_{all\ pixels} Q_{HGC}}$$

These *would* be the best ways to combine everything if we could guarantee that a particle of a specific energy detected in the HGC would give the same charge as a detection in the MCP. However, since the HGC and the MCP have different gains that are not known, a better way to weight the histograms would be to do either an event charge or total run charge weighting for the HGC Δt and then weight that and the MCP Δt each by half (Figures 10 and 11).

$$\Delta t_i = \frac{\left[\sum\limits_{all\ pixels} \Delta t_{HGC_i} q_{HGC_i} \div \sum\limits_{all\ pixels} q_{HGC_i}\right] + \Delta t_{MCP_i}}{2}$$

$$\Delta t_i = \frac{\left[\sum\limits_{all\ pixels} \Delta t_{HGC_i} Q_{HGC} \div \sum\limits_{all\ pixels} Q_{HGC}\right] + \Delta t_{MCP_i}}{2}$$

I have still yet to determine which of the event charge or total run charge weightings is optimal. Note: my code fits every histogram 2×RMS around each peak with a Gaussian.

The following figures were generated through running my analysis code on runs 65-83.

4 Preliminary Results

I am currently using my analysis code for the HGC and MCP to look at how these time resolutions change for different configurations (i.e. absorber type, spacing, and thickness; beam energy; etc). So far, I have only looked at the following configurations:

The last run is a reminder that the cuts should be edited for every run in order to correctly select the proper data. There may also be a beam energy dependence for the value of the cuts. So far, I have determined that the following cuts are the best for the above runs:

As far as the rest of my research goes: I am going back to Caltech at the end of next week to continue my data analysis. I cannot really plan out how my analysis will evolve until I finish looking at the different runs, but then I will most likely take my results to someone in the Caltech Precision Timing group and figure out the next steps of my research.

References

[1] Adolf Bornheim et al. "Calorimeters for Precision Timing Measurements in High Energy Physics". In: *Journal of Physics: Conference Series* 587.1 (2015), p. 012057. URL: `http://stacks.iop.org/1742-6596/587/i=1/a=012057`.

[2] Cristian Pena. For revisions.