

Detailed Description of GaussHitFinder

Michael Wang and Giuseppe Cerati

June 20, 2018

Abstract

A detailed description is provided of the algorithm used in the standard implementation of the `GaussHitFinder` module found in the `LArSoft` toolkit.

1 Introduction

The purpose of `GaussHitFinder` is to find peaks associated with wire hits and determine the parameters of these peaks from the deconvoluted waveforms from the wires in each plane of a Liquid Argon TPC. It consists of three major stages which involve hit candidate finding, merging of hits that are close in proximity, and fitting Gaussian functions to the waveforms to determine the peak parameters. We describe each stage in detail below.

2 Hit Candidate Finding: `findHitCandidates`

The first step in `GaussHitFinder` is to identify hit candidates. It does this by looking at all ADC readings from a single wire that lie within a given time window of the waveform which is required to have a minimum of five ticks. Starting with the maximum ADC reading within this window, which it requires to be above a given threshold for that particular wire plane, it scans backwards in an attempt to find either the rising edge of a unipolar pulse or a valley between two such pulses. Prior to performing this backward scan, it verifies first that there are at least three ADC readings upstream of the maximum, otherwise, it moves to the forward scanning stage which we will describe later. Going back to our discussion of the backward scan, it looks for a set of three consecutive ADC readings where the central one is bracketed by a downstream one with a value greater than its own and an upstream one with a value greater than or equal to its own. This search is continued all the way to the very beginning of the waveform until a valley or rising edge is found. When this is satisfied, the code recursively calls itself, but this time using a truncated range of the waveform having the same starting point as the previous definition but which now ends at the third point of the moving window in which the valley or rising edge was detected.

After it returns from this first recursive call, it proceeds to the forward scanning stage in which it searches for a valley or falling edge downstream of the peak. The procedure is similar to the backward scan except in reverse. Prior to the forward scan, it checks to see if there are at least three ADC readings downstream of the maximum. If so, it looks for a triplet of consecutive ADC readings where the central one is bracketed by an upstream one that is greater in value and a downstream one that is greater or equal in value. This search is conducted all the way to the end of the waveform until something is found, in which case, having identified both a leading and trailing edge, a new entry is added to the list of hit candidates. The mid-points of the triplets in which the leading and trailing edges were found are taken as the first and last tick, respectively, of the pulse waveform associated with the hit. The locations of these two ticks and that of the peak, including their associated ADC values, and an estimate of the half-width of the pulse are all stored in the hit candidate list. Next, the code recursively calls itself a second time. In this recursive call, a truncated waveform is passed, which begins at the third point of the moving window in which the valley or edge was detected, and ends at the same point as the previous definition of the waveform.

In this recursive manner, all sub-pulses with peaks above the specified threshold for the wire plane are found, located, and added separately to the list of hit candidates.

3 Merging Candidate Hits: MergeHitCandidates

After all the hit candidates are found, this next step tries to merge consecutive hits that are close enough to each other into a single group before passing it to the next stage where hit parameters are extracted by fitting Gaussian functions to the waveform. This way, parameters for the entire group of hits can be determined with a single fit to a sum of Gaussian functions. Two consecutive hits are considered close enough for merging if the first tick of the downstream hit is less than two ticks away from the last tick of the upstream hit. A whole sequence of hits in which this condition is satisfied between adjacent hits are merged into a single group. As soon as an incoming hit's first tick is more than two ticks away from the previous one, it is used to start a new group of merged hits.

4 Finding Peak Parameters: findPeakParameters

As soon as all hit candidates are merged into groups, these groups are then passed on to the peak parameter finding stage. All this stage does is to fit the merged group of hits simultaneously to a sum of Gaussian functions with three independent parameters per function. One Gaussian function is associated with each hit candidate in the merged group. The initial values of the amplitude, mean, and width (σ) of each Gaussian are estimated from the peak ADC value, its tick position, and the approximate width, respectively, of the hit stored in the hit candidate list. A constrained fit is performed in which the parameters are allowed to vary within a limited range. The amplitude is allowed to vary from its initial value within 0.1 to `AmpRange` (2.0) times its initial value. The mean is allowed to vary from its initial value in either direction within `PeakRange` (2.0) times the estimated hit width, unless the range exceeds the first or last tick of the merged waveform, in which case those limits are imposed. The width is allowed to vary from its initial value from a lower limit of `MinWidth` (0.5 tick) or 10% of the estimated hit width, whichever is larger, up to an upper limit of `MaxWidthMult` (3.0) times the estimated hit width. The goodness of fit parameter, χ^2/DOF is calculated and required to be less than or equal to the maximum value of a `double` type in C++.