SVT Hit Fitting Studies

Rory O'Dwyer
Cameron Bravo

Structure of Presentation

Study Objectives

General introduction to pile-up reconstruction (2-6)

Analysis with TH1D Histograms (7-19)

Analysis with TH2D Histograms (20-27)

Cluster Level Studies (27-)

Our findings:

1. We found an (as of yet unexplained) increase in resolution using TH1D and Pulse Shape analysis (19)
2. Insensitivity of the algorithm to chi-sqr threshold (9-15)
3. Misidentification of pulse number in a small number of unlikely events. (24-26)

Study Objectives

In the track time distribution, Alic observed a large shift in the MC t0 distribution mean wrt data. This, as well as the observed difference in resolution, could be explained with incorrect choice in pileUp pulse times.

Our objectives in this study are to:

1. Assess how well pileUp fits reconstruct the T0 and amplitude distribution of hits
2. Determine what changes to the Hit Fitting may be performed to reduced misidentified hits.
3. Ultimately we hope to find an improvement in hit time and time resolution.

Preview: We find evidence for misidentified pileUp pulses in this study :)

How pileUp pulse fitting occurs in Reconstruction

RawHitFitterDriver is given 1 of four fit algorithms: (analytic, linear, pileUp only, and pileUp)

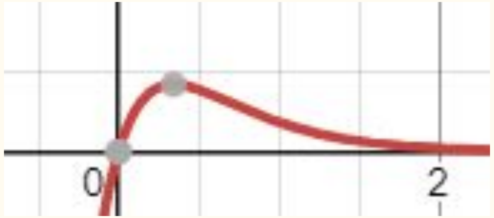
- Analytic performs a fit using RCCR and the amplitude and offset is fitted for. For RCCR there are analytic expressions for best fit amplitude and t0 parameters.
- The other three perform fitting to a four pole fit function, where FitMinuit is used to determine amplitude and t0
- In both fits **τ1** and **τ2** (in the equations to left) are properties of a channel and in a database.

Ultimately what we hope to extract from the pulse profile is its time t0 and V_{i} measurements (which are the only features not characteristic properties of a channel)

Our default algorithm is pileUp

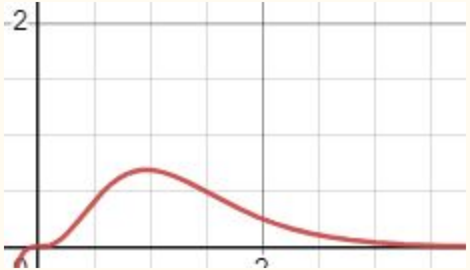
CRRC

$$V_0 = \frac{V_i \tau_1}{\tau_1 - \tau_2}\left(e^{-\frac{t-t_0}{\tau_1}} - e^{-\frac{t-t_0}{\tau_2}}\right)$$



Four Pole

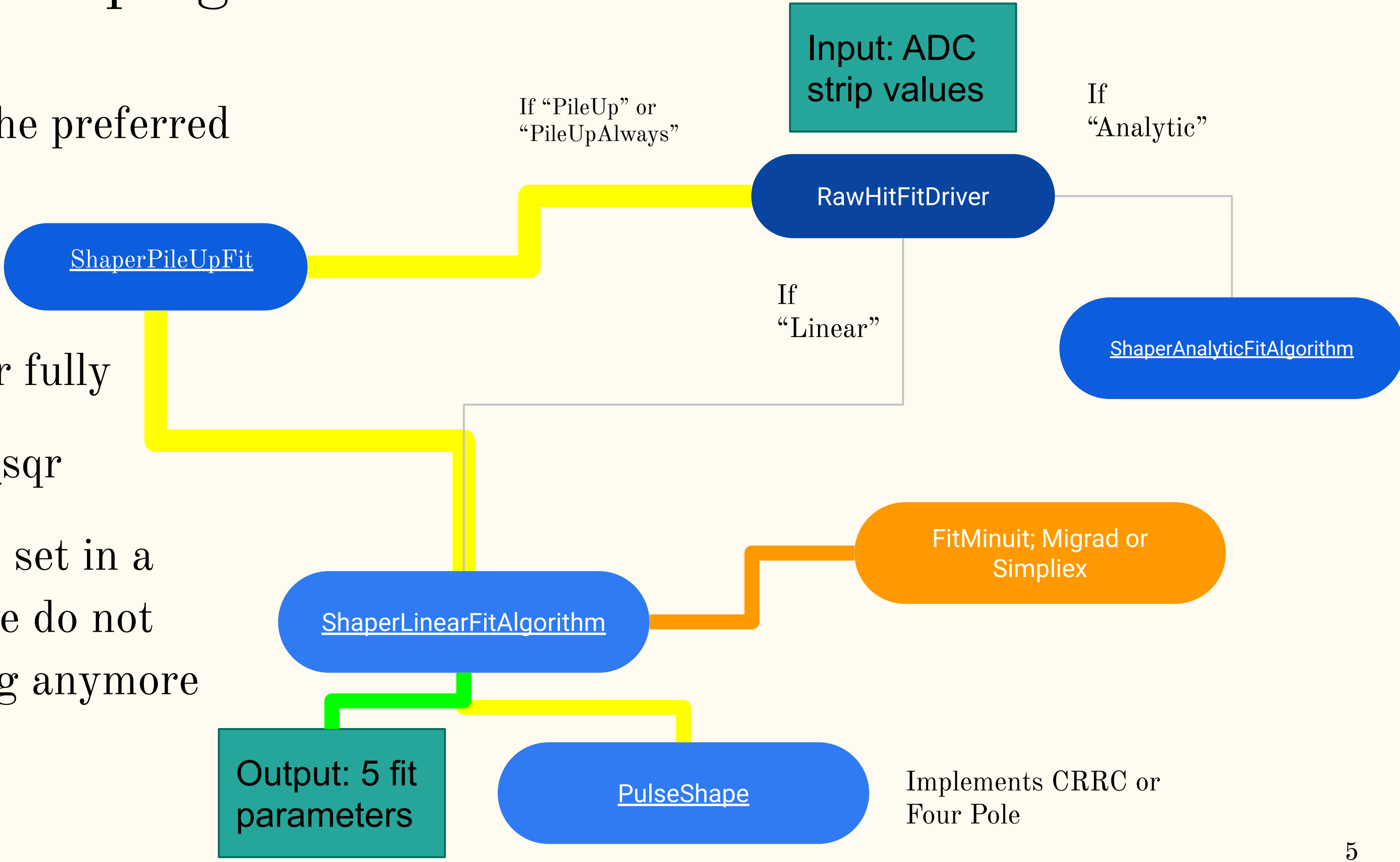
$$V_0 = V_i \frac{\tau_1^2}{(\tau_1 - \tau_2)^3}\left(e^{-\frac{t-t_0}{\tau_1}} - \left(1 + (t-t_0)\left(\frac{\tau_1-\tau_2}{\tau_1\tau_2}\right) + .5(t-t_0)^2\left(\frac{\tau_1-\tau_2}{\tau_1\tau_2}\right)^2\right)e^{-\frac{t-t_0}{\tau_2}}\right)$$



How pileUp Shaping Occurs in Reconstruction

Yellow denotes the preferred path

PileUp allows for fully configurable chi_sqr threshold (this is set in a steering file so we do not require rebuilding anymore between scans).

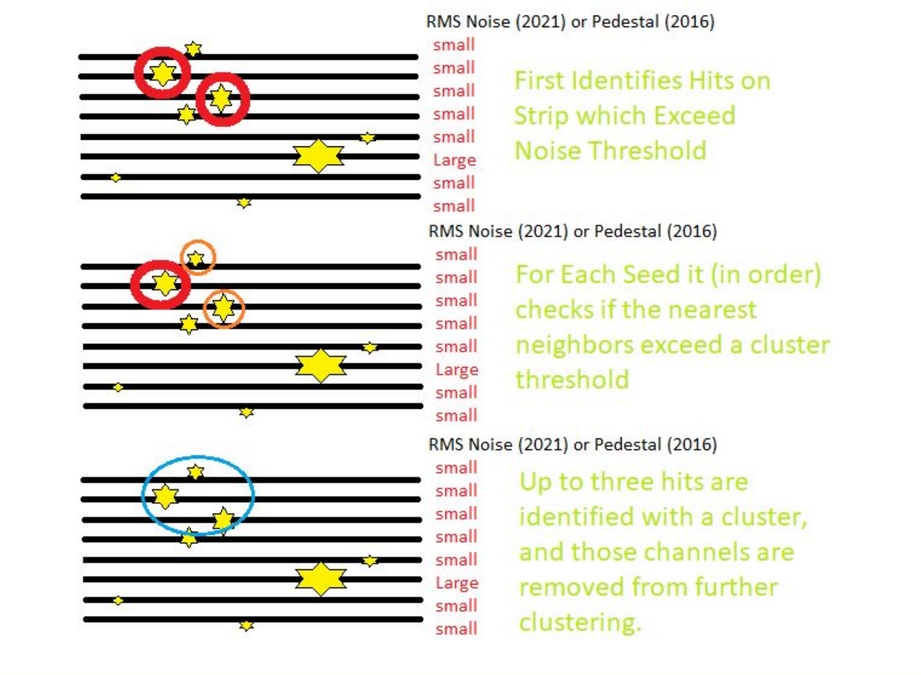


How is pileUp propagated into later Reconstruction

NextNearestNeighbor.java creates seeds for hit clusters and associates hits to a cluster depending on their signal to noise ratio.

- In both the 2016 and 2021 we use noiseRMS; the noise recorded from each detector component w.r.t a hit.
- This is highly dependent on Luminosity; we use the same seeding and clustering thresholds however for dramatically different luminosities.

$$t_{cluster} = \frac{\sum_{i \text{ a neighbor of seed}} t_0^i * PE^i}{\sum_{i \text{ a neighbor of seed}} PE^i}$$



The cluster averaging between incorrectly chosen time pulses may cause both the observed shift in mean and lowered resolution. In general there are problems with using PE for weighting

Analysis from our TH1D Histograms

In the next section we use histograms of t0 and amplitude across a scan of pileUp thresholds. We will do the following

1. Demonstrate the usefulness of of pileUp in T0 resolution, but its insensitivity to chi_sqr threshold values
2. Plot Pulse profiles to demonstrate visually how well our Fits work in the 2021 reconstruction pipeline.
3. motivate possible improvements to the pile-Up pulse algorithm, particularly how often our algorithm misdiagnoses pulse number
4. Lead into TH2D Histogram discussions by finding signs of mislabelled single and pileUp pulse events.

Data for plots

We use /sdf/group/hps/data/physrun2021/hps_014552/ as our source for high lumi runs;

There are 127 M events at 20 nAmps, but EB does crash at the end (1 hr long).

We use /sdf/group/hps/data/physrun2021/hps_014166/ as our source for low lumi runs

There are 10 M events at 8 mA,ps, lasts 36 minutes; it was used for the SVT commission test for SVT alignment.

.evio files are processing using the PhysicsRun2021_pass0 steering file, and reconstructed using kalSimTuple_cfg.py

The flat nTupple data and the cuts performed upon them were obtained using the SvtRawDataAnaProcessor Cam and I are writing.

T0 for Single Pulses (no PileUp) changing χ2

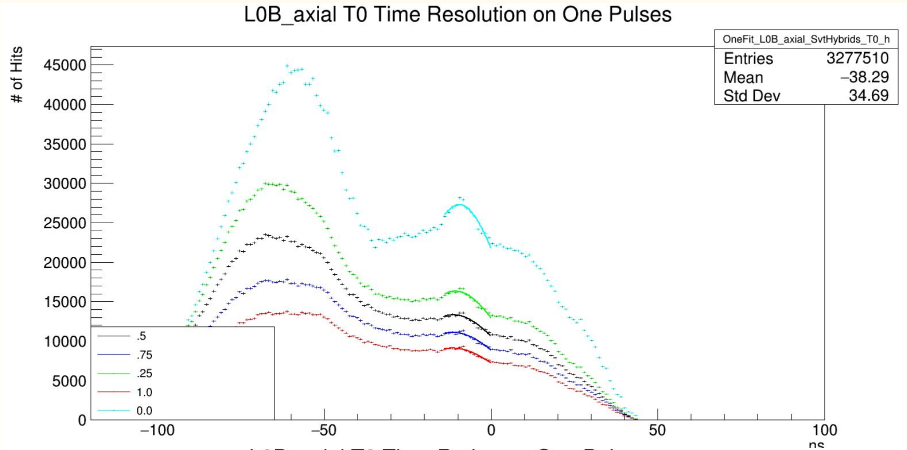
The profiles of the ratios after chi_sqr nonzero all have a valley at -60

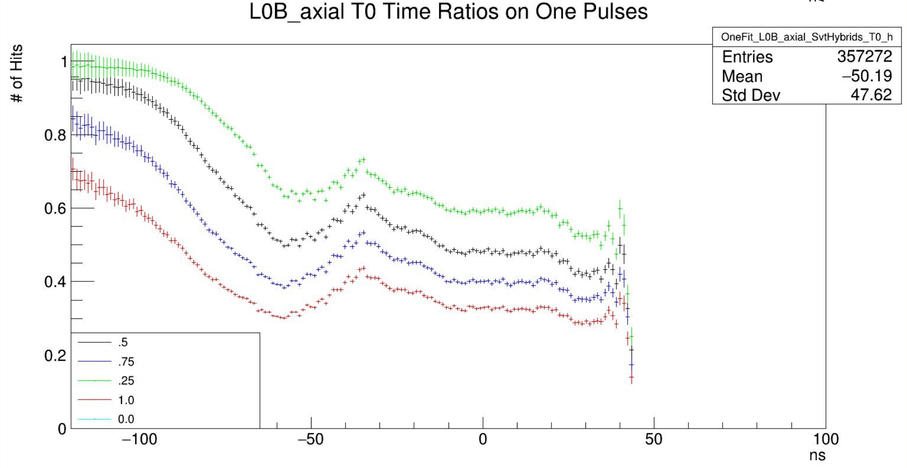
Afterwards the shape does not change to much, but the overall occurrence of one tracks decreases.

The overall reduction in OneFit is now proportional to the likelihood a genuine OneFit fails the chi_sqr criterion.

This tells us that while PileUp is important for identifying true PileUp events, it doesn't show too much sensitivity to chi_sqr once it is allowed.

The actual sigma of our real peak doesn't change to much (imaged besides the histogram)





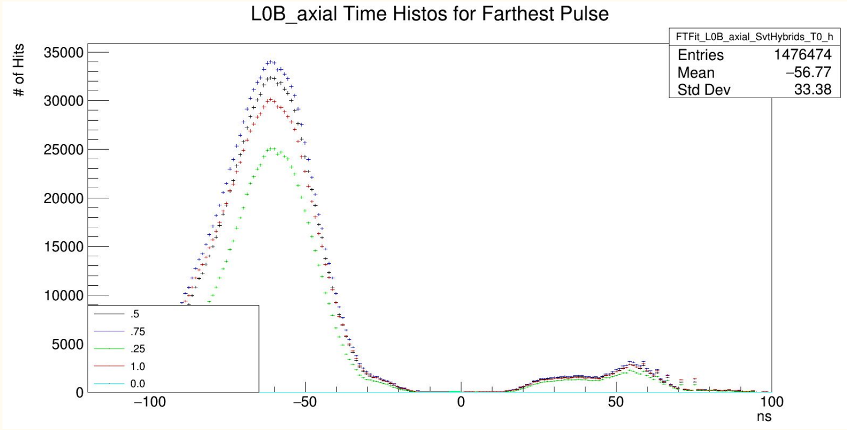
T0 for the Farthest Fit T0 in PileUp

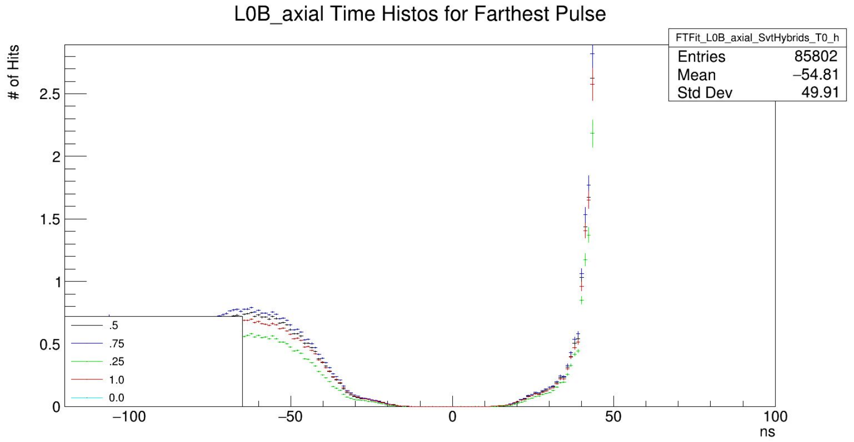
The first fit in the pileUp sees a preferential peak at -60 over ratio plots, and a decrease around -40.

Pulses occur at -9 or -22 depending on the wiring of the ADC's; our timing cut may get rid of this farthest pulse but the closer pulse places a large shoulder on the Final T0 distributions (as we shall see)

These are events that are ejected! Therefore a significant number of our events occur in time with the trigger. I.e. we are seeing reduced hit efficiency.

Determining precisely how much would require a decent understanding of the background rejected; we don't have a clear picture of that just from this plot.



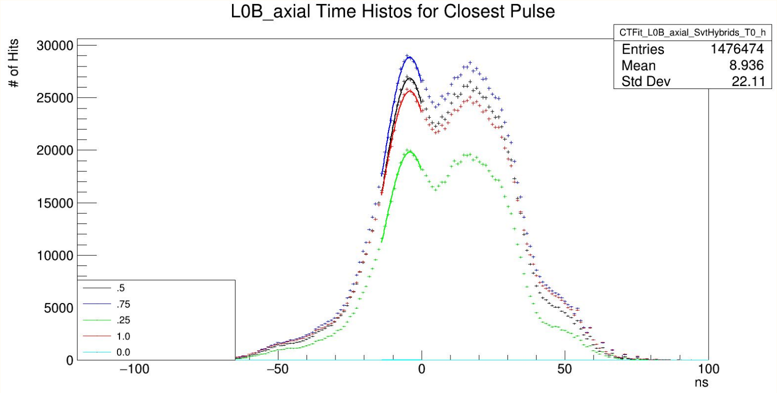


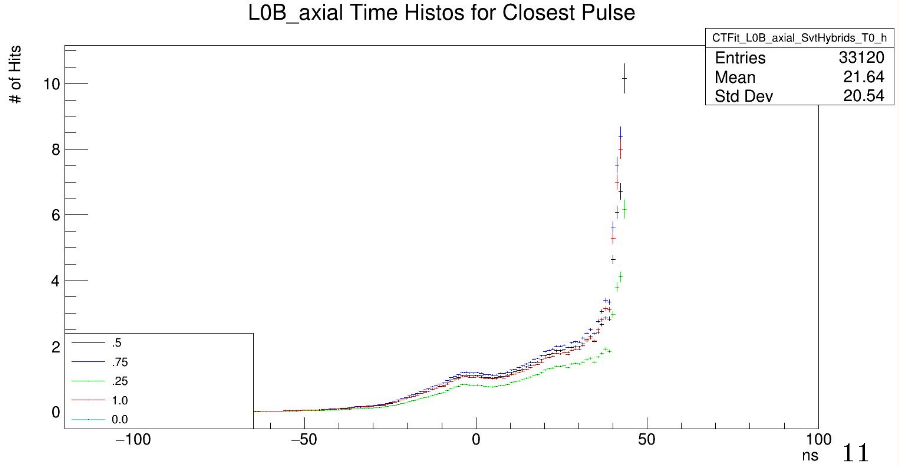
T0 for the Closest Fit to T0=0 in PileUp

The closest pulse distribution is bimodal; we see authentic hits in time with trigger but secondary hit peak far displaced from this.

The Time Window of -48 ns used in NearestNeighborClustering does not eliminate contributions from this peak; it warrants further analysis.

Again the ratio plots demonstrate this distribution is largely invariant w.r.t just turning on the pileUp.



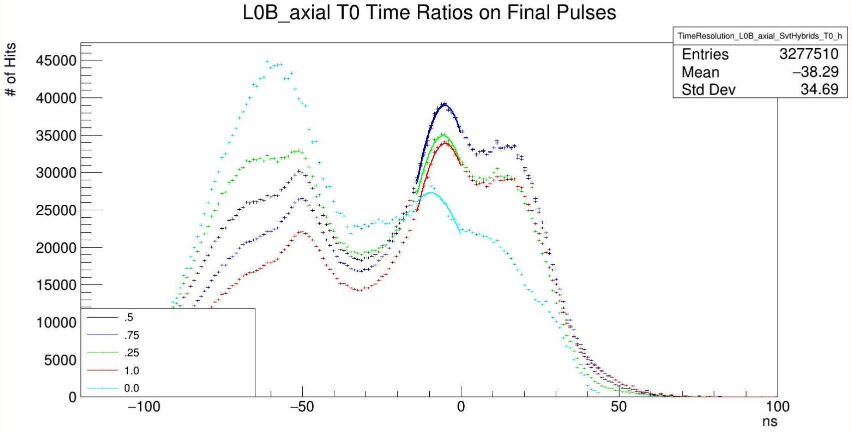


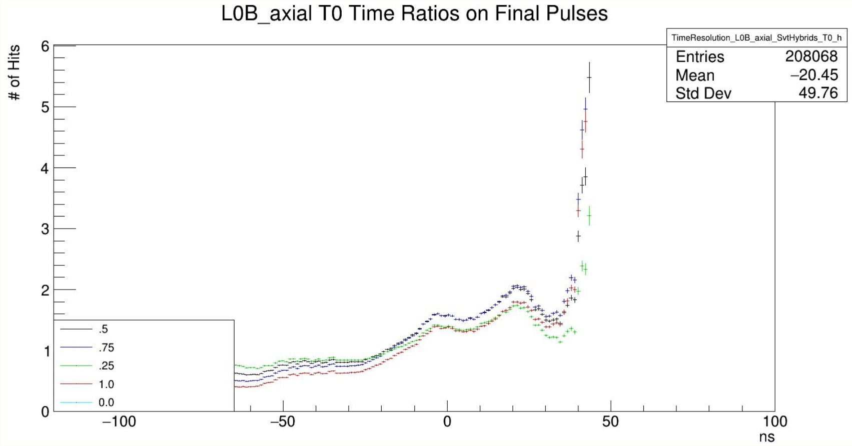
T0 for the Final Fit T0 in PileUp

This shows what T0 distribution we would see in the Final Fit for L0_axial sensor.

There is a large should of time due to the second peak being displaced from our normal time of -9 ns for layers 0 and 1.

It demonstrates that pileUp is insensitive to chi_sqr in time resolution, but when turned on preferentially boosts the contribution of the second hit in pileUp



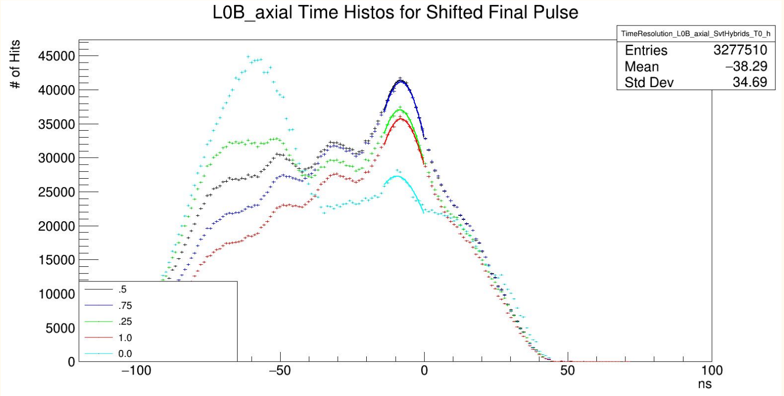


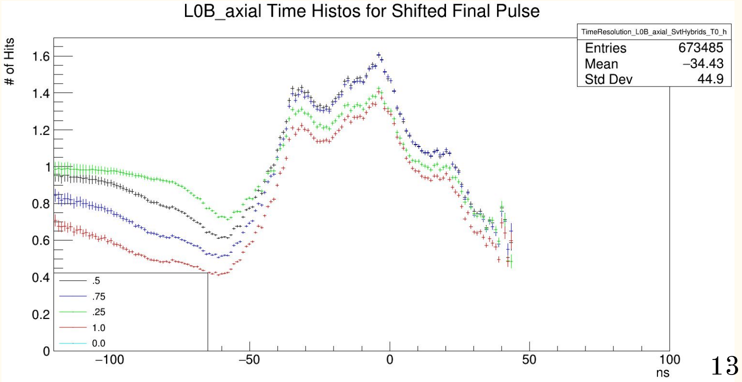
Possible Decrease in T0 First Layer Shoulders via calibrating Second Pulses to appropriate place

Before a magic fix, whenever we have an incidence of pileUp we translate the time back -27 ns.

This is the difference between the prominence at 21 ns from two pulses and the standard time of pulses in the first layer.

You do see a small increase in the left shoulder, from originally 'good' pulses in the first T0





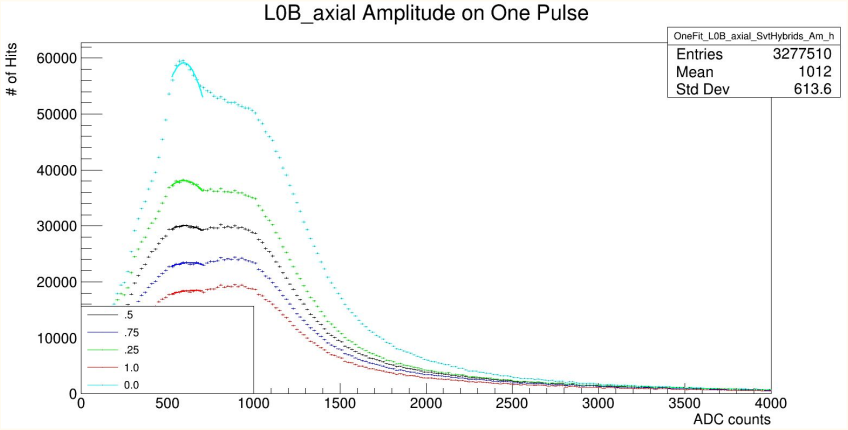
Amplitude distributions across Threshold Scan

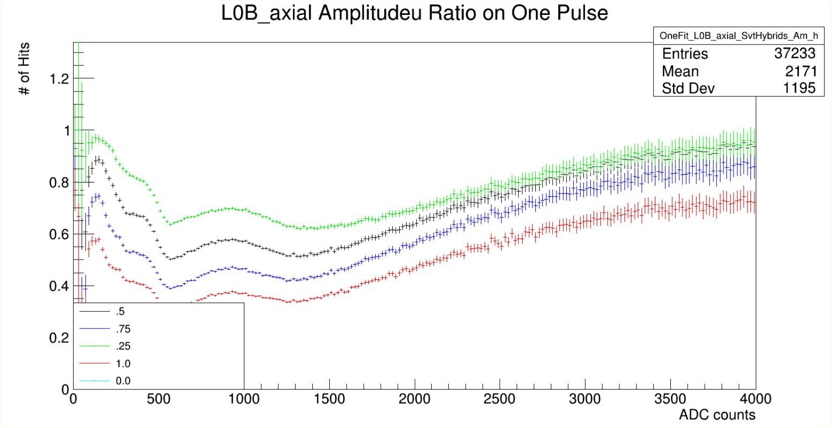
One Pulse monotonically decreases as you go from threshold 0 to 1.

From the ratio plots, we again see that the initial occurrence of pileup preferentially removes peaks at 500 ADC counts.

It increases pulses at 1000 ADC counts, so its achieves more realistic pulses.

It is, however, not too sensitive to $\chi 2$ threshold (after $\chi 2$=fifty, it just removes events).





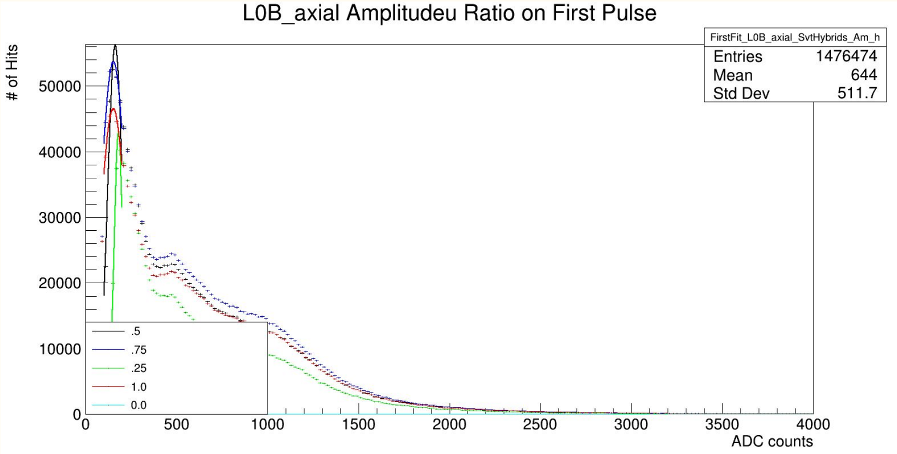
Amplitude distributions across Threshold Scan

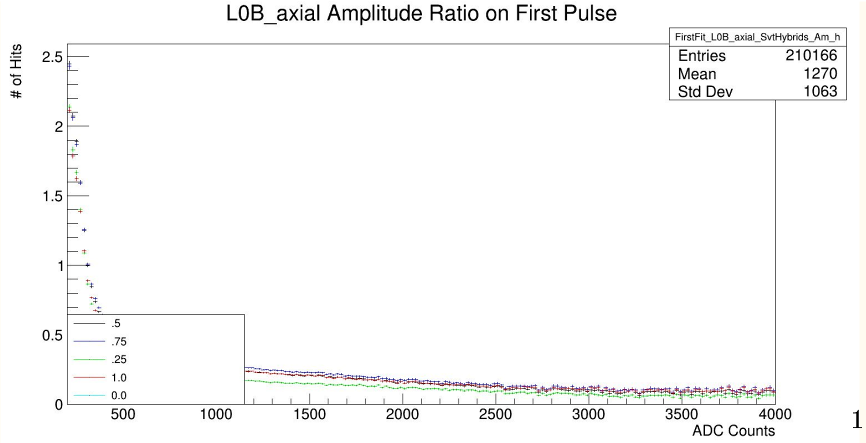
The time of clusters are PE weighted times; from these plots we see that pileUp is naturally suppressed.

It could be more beneficial to use time Error weighted averages

the lower PE of pileUp is just an artefact of the fit splitting amplitude; seems more appropriate to use time-error

In the ratio plot we see a large preferential gain directly at adc=0,





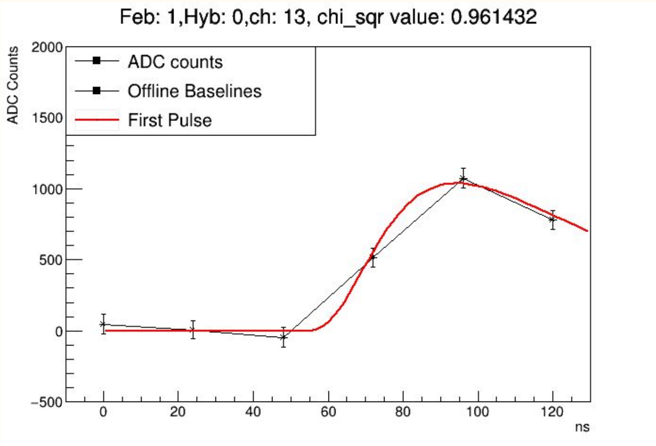
Svt Pulse Profiles

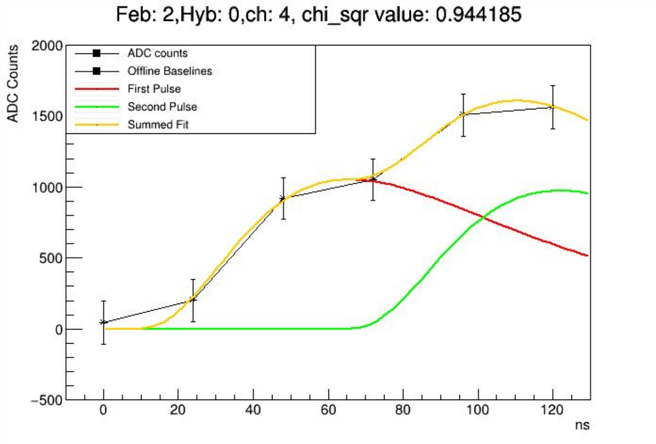
Another powerful way to evaluate our algorithm's effectiveness are Svt Pulse profiles.

For run 14552, we read in the offline baselines and pulse shape information from .dat files in our database.

For each event, we subtract our baselines. We then (using T0 and Amp) plot the reconstructed profiles upon the pulses.

This allows us to evaluate precisely how well our FitMinuit is performing; we have diagnose a couple errors in my implementation of reconstruction using this tool



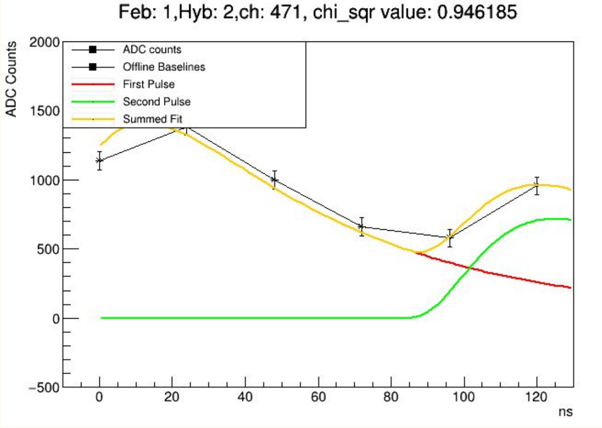


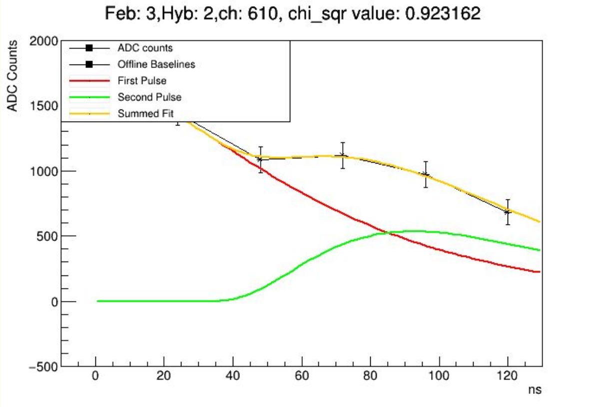
What clean pileUp pulses normally look like

If we explore the region between -40 and -80 for one of the amplitudes, we find pulses that characterize an authentic pileUp event

For the vast majority of these pulses, what we see is a quick increase in the last two entries, corresponding to a rising edge.

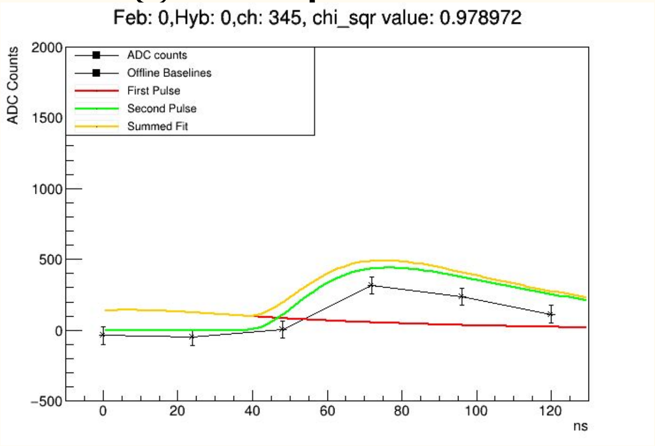
Whenever the later amplitude is near to zero, we instead see a quite large amplitude for the first pulse





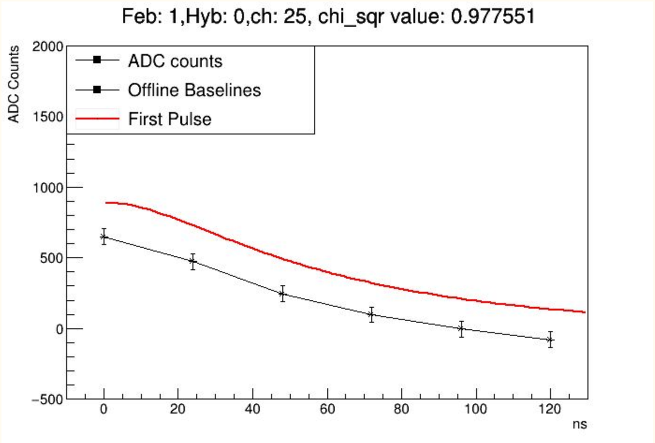
A Quick Story: diagnosing a steering file problem

While calibrating these pulses, we came across a couple problems: we were seeing pulse shapes like that on the right:



This is because the time correction algorithm in the PhysicsRun2021 was either incorrect or not reverse engineered properly (timeless run was perfect)

After many manipulations, we corrected the time shift. It also corrected the secondary peak; implying that we have made some progress!



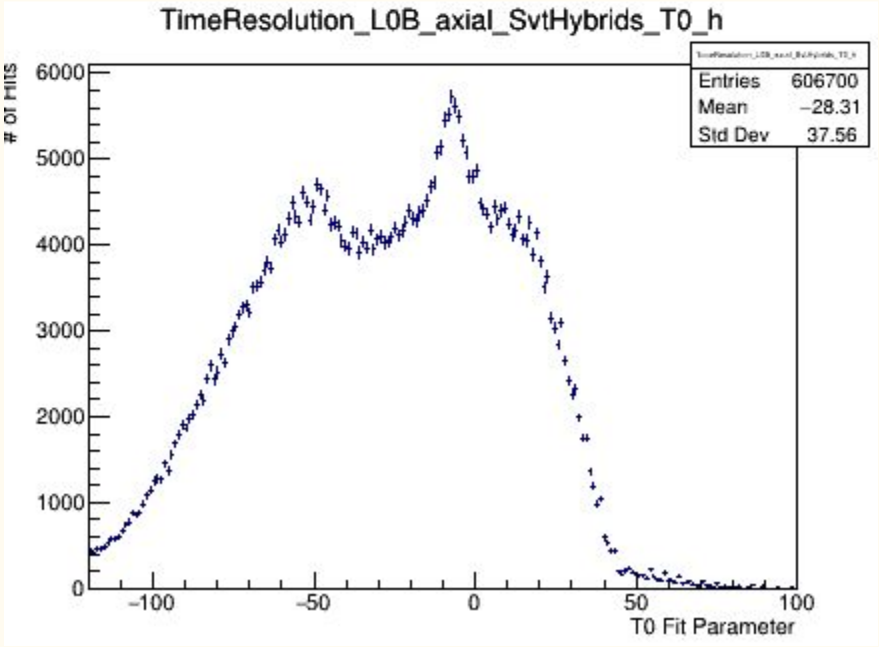
Main issue: cannot retrace what was changed to lead to increase in resolution.

Time Profiles After Fix

You may immediately notice an increase in signal events and removal of shoulder (without weird effects).

We are still diagnosing the fix. It was not a case of misused steering file; a genuine fix was made in the Pulse Shape fitting algorithm

This change was made in the very recent past; the most immediate steps I will take is finding why this occurred. I didn't have enough time to do a threshold scan.



TH2D plot analysis

The peak at ADC=0 for FirstFit pileUp events suggest that the majority of the charge from an event is placed in a single charge; this is not reasonable for authentic pileUp.

In this section we

1. Devise a normalized ADC curve plot to visualize what effects cuts have on the shapes of our curves
2. Use timeDiff of two pulses along with timeError and AmpError to diagnose misidentified hits
3. Evaluate the possible improvements to be made on pileUp and their effect on Hit efficiency and time resolution of pulses.

ADC Time and Baseline Shifted Curves

One of the more important metrics we use in this work is the ADC Time/Baseline shifted curves.

From Alic's scan of the APV24 asic, we obtain the profile of a pulse through our silicon strips.

For a given hit, we can translate the 6 adc counts (24 ns separated pulses) in time by the pulse T0 and down in amplitude by a baseline, then divide by the pulse amplitude, to obtain the shape of a pulse.

We plot all of these curves for every event that passes our cut, and this can determine when a one pulse event is mislabelled.

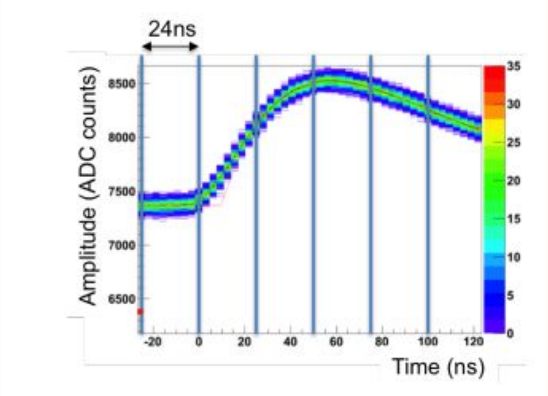
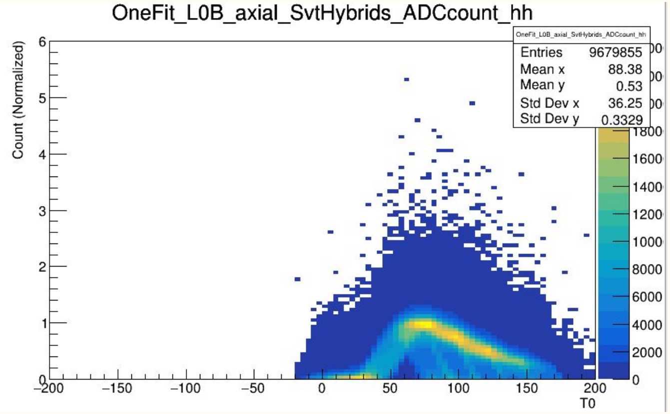
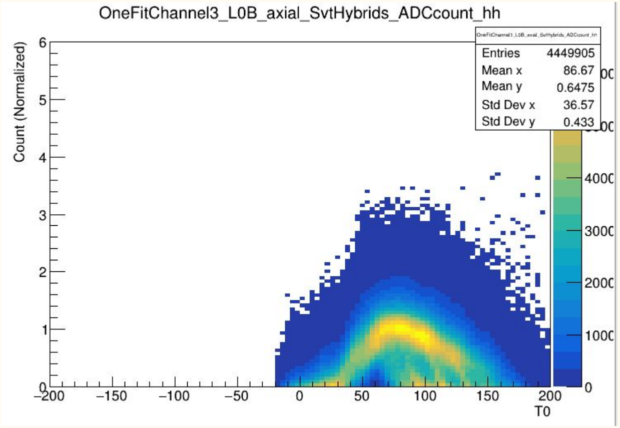


Fig. 4. Pulse shape of the APV25 ASIC.

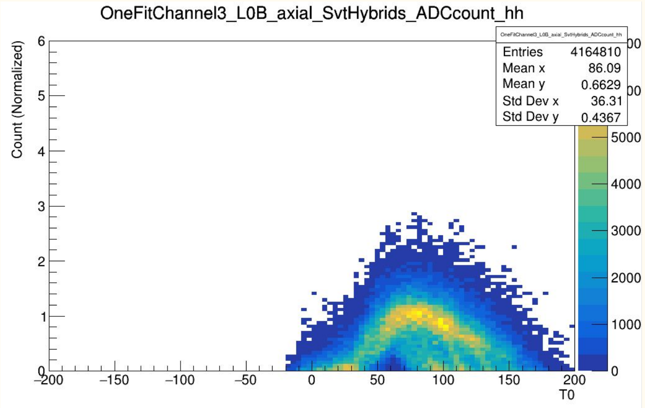


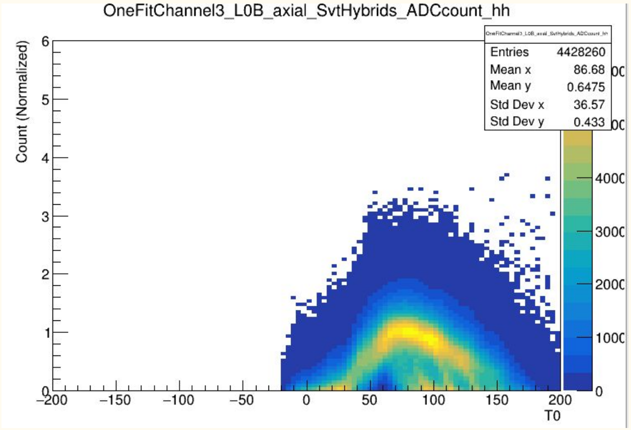
Single Channel Scans One Fit (Channel 3)



Some Single Channel Scan still show substructure. This should occur iff pileUp is not being identified

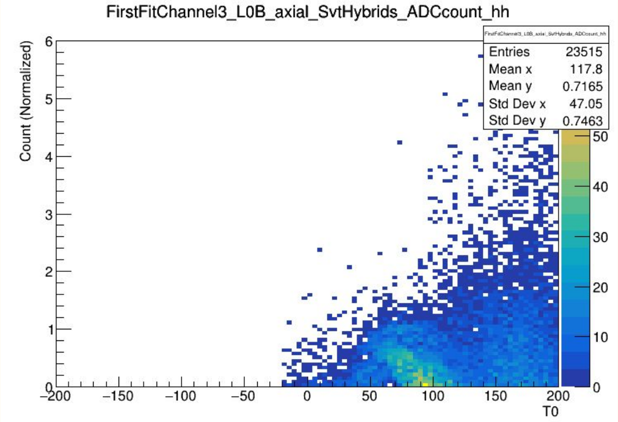
This is consistent with these pictures

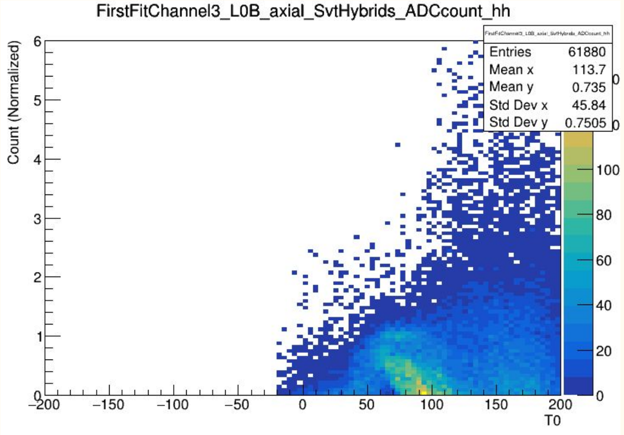




Single Channel Scans First Fit (Channel 3,7)

This may require higher probability, but the channel profile seems to change for the first pulse in a two pulse pileUp, which is weird since the shape ought to be a property of the channel

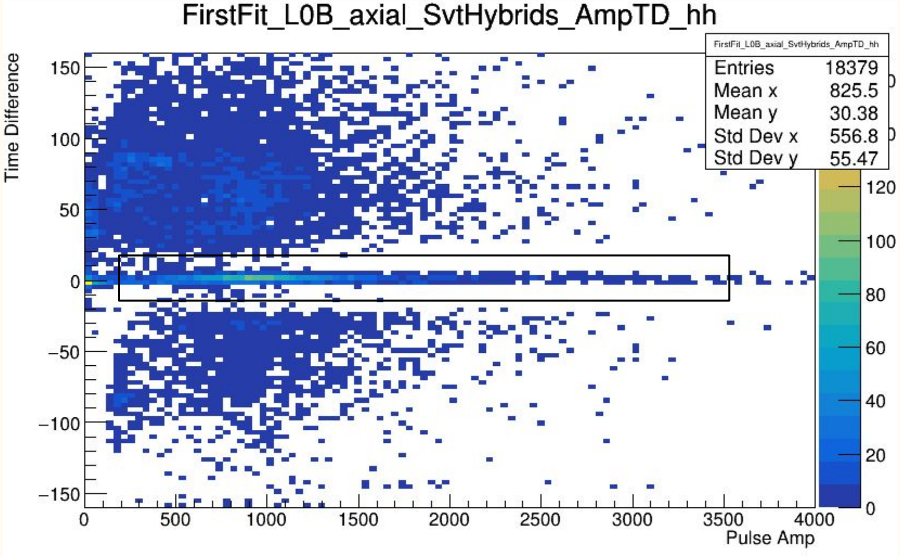




General Properties of the PileUp Algorithm at Low and High Lumi's

At Low Lumi runs (like run 14166) pileUp accounts for 1/40 of the hits with low variation w.r.t. Threshold. At High Lumi (14552) it varies (0,.29,.41,.44,.51)

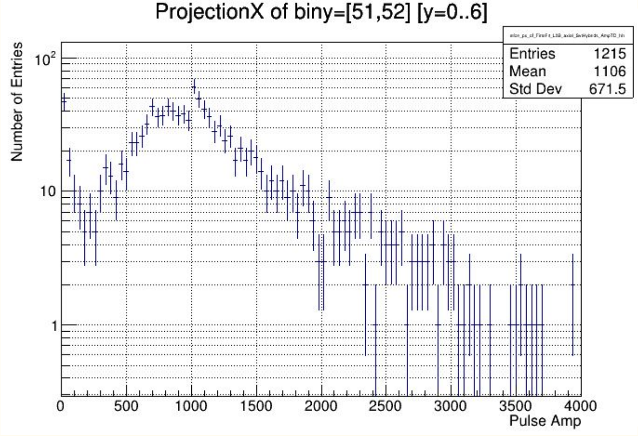
Non-Sensible Features in some T0 and A0 Histograms occur in .001 of the total cases. These we will show correspond to single pulse misidentification

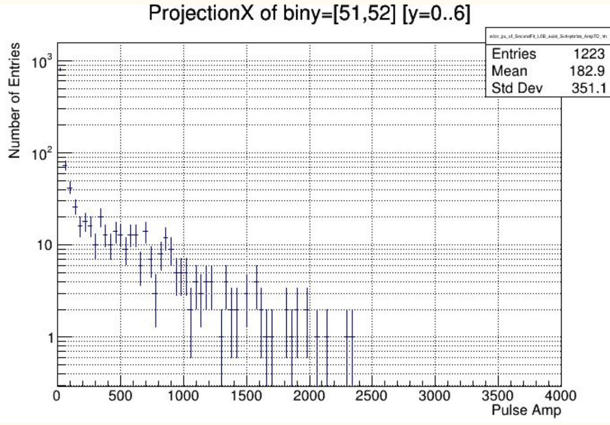


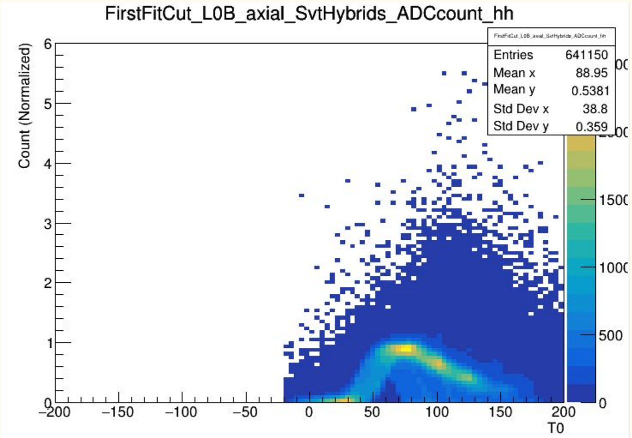
A Look at PileUp Pulses in the Delta T=0 Regime

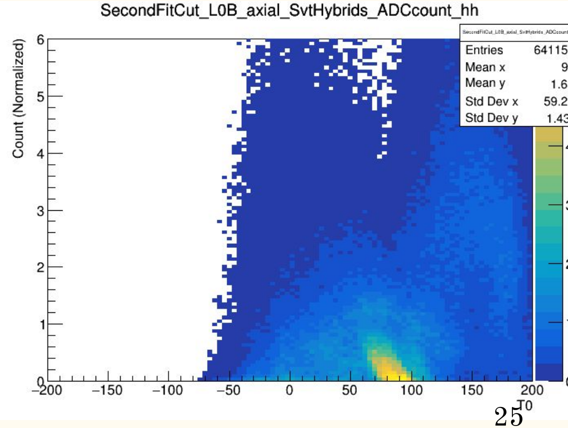
If we project the Amp vs DeltaT0 histogram's three central bins (60 ns total), we obtain the right two graphs. I appears we get a 1000 Amp pulse in First and None in the Other.

To test misidentification, we can make a cut on our pileUp st. the first pulse is 200 Amp away from 1000. We get the following profiles







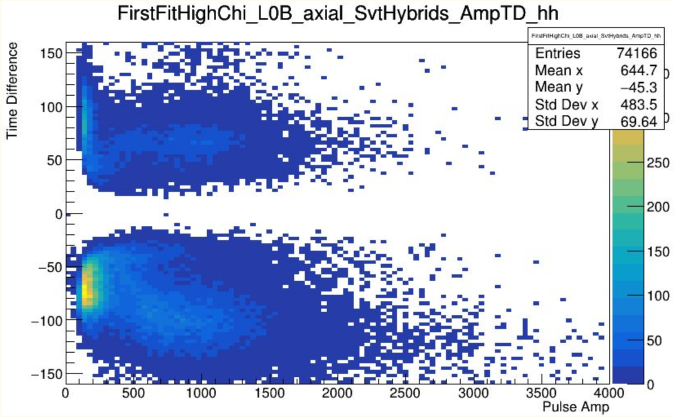


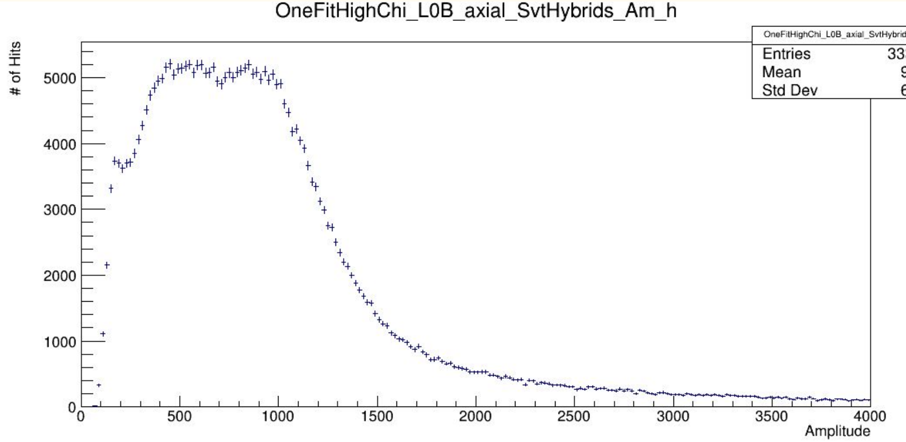
High PileUp Chi Sqr and Eliminating PileUp in Event of Delta T less than 6 nS

High chi-square probability are those events that exceed .9 in chi square probability.

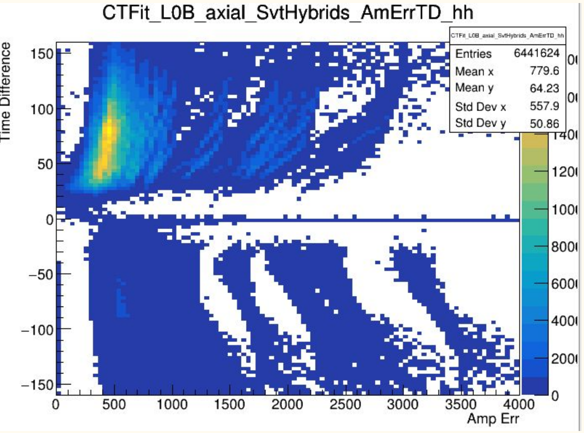
For these events, you do not observe any diff T0=0 in pileUp (so there are no ‘legitimate’ pileUp pulses with low T0 difference).

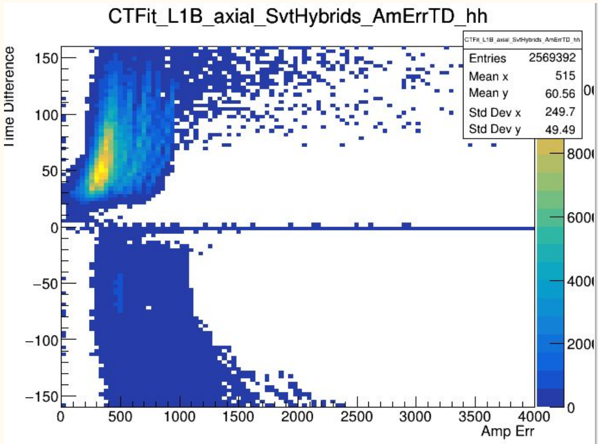
For High Chi square probability, in the one pulse amplitude distribution you do not see a peak at 500 adc counts, indicating that that was likely not a genuine single pulse event.

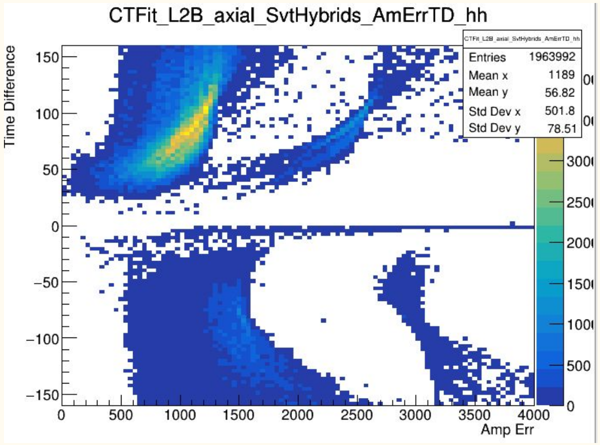


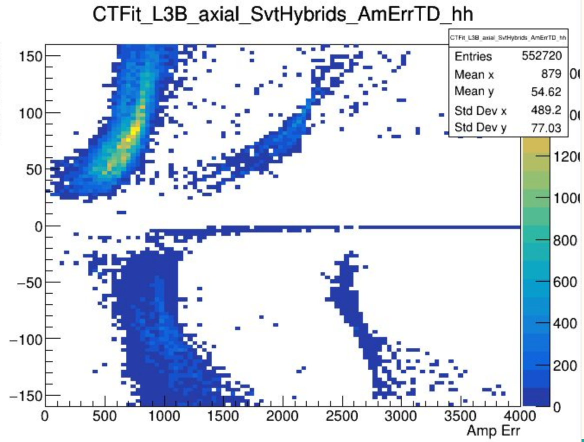


Time Difference (T0 CT) Versus Amplitude Error









For some events off the central bulb, we see positively sloped correlation between Time Difference and Amplitude Error. This indicates that after you obtain some spacing, the farther your pulses are the less likely their fits are reflecting reality. This is with the exception of the large peak at time diff 0, where degeneracy in the fit leads to enormous amplitude errors.

Cluster Level Steps

We have done a rather in depth analysis of the pileUp pulse algorithm, so we switch gears to see how the next step, Clustering, may benefit in T0 resolution and Hit Efficiency from some changed motivated by previous steps

1. We will show PE count weighted time is not a good metric for determining the time of a cluster; pileUp will give you unfairly decreased dependence
2. We will show that, once done appropriately, pileUp makes a significant difference in Hit Efficiency but that Hit Efficiency (like our previous metrics) is insensitive to chi_sqr.

Changing Cluster Time from PE Weighted to Time Error Weighted

Here are plots of the T0 distribution before and after changing the cluster time def to the following:

$$t_{cluster} = \frac{\sum_{i \text{ a neighbor of seed}} t_0^i * \frac{1}{t_0 Err^i}}{\sum_{i \text{ a neighbor of seed}} \frac{1}{t_0 Err^i}}$$

This is motivated from the fact that our pileUp algorithm unfairly weighs the authentic hits with pileUp with less importance

What Does this Mean For Hit Efficiency

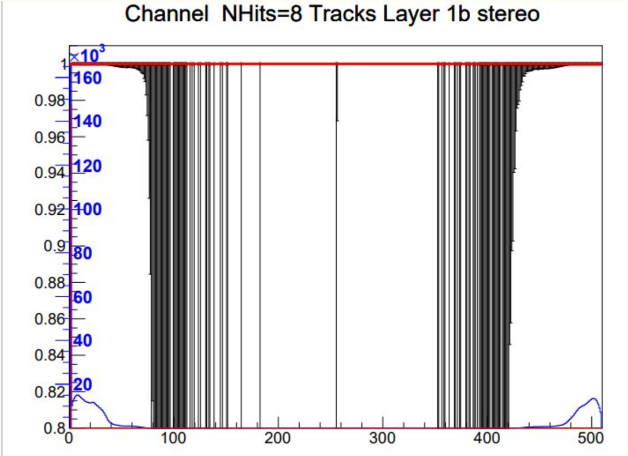
Using the same scan over pileUp chi sqr thresholds, we can also run mgraham's Hit Efficiency

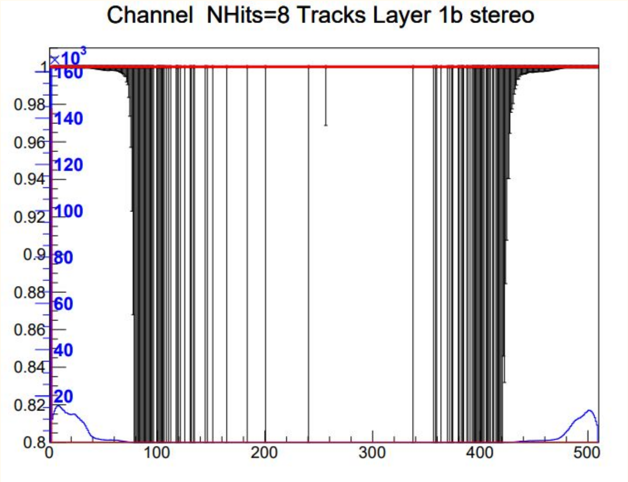
The blue curve represents the number of tracks originally containing some hit, and then the red the number of tracks containing said hit upon reconstructing them from all but that hit.

The ratio is the hit efficiency at that strip.

Notice that for earlier layers middle portions of the efficiency plots are missing simply because the angle is too high to get enough hits to form a track.

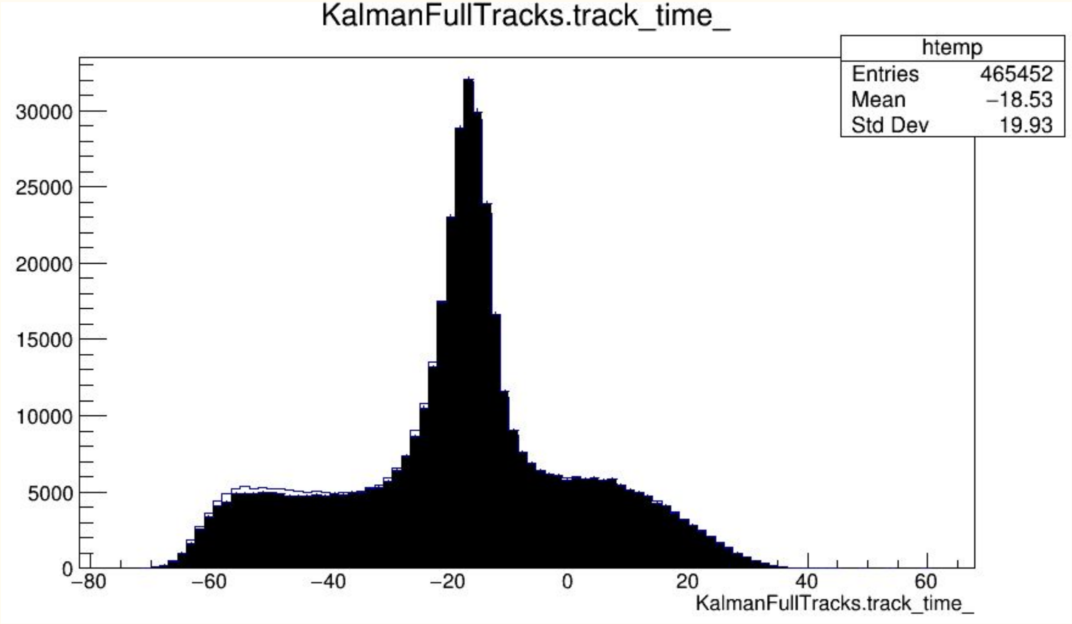
We chose .25 and .5. The point is that while pileUp can dramatically affect time resolution, it seemingly does nothing for hit efficiency.

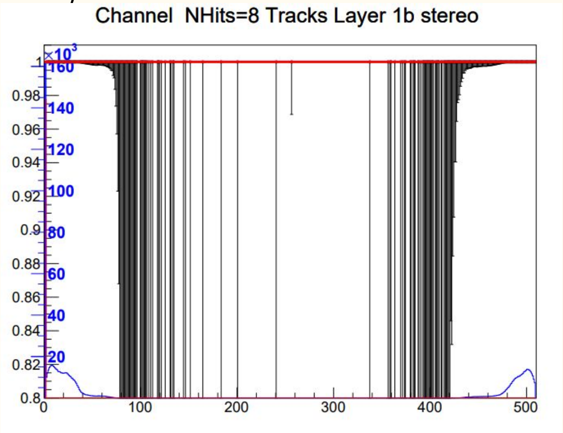


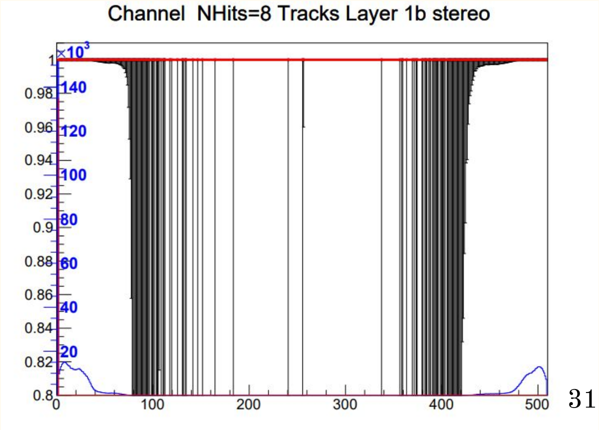


What can we do to increase Hit Efficiency?

Changing the Nearest Neighbor RMS threshold and how we define uncretainty.







Conclusion and Next Steps

For the past 2 months, Cameron and I devised a number of metrics to evaluate the Svt Pulse Shaping algorithm.

Our findings were namely that chi_sqr of around .25 was already sufficient to achieve the desired results of pulse shaping.

Furthermore, beyond finding evidence of misidentified multiple pulses (which has poor chi_sqr), we did not conclude that this algorithm required changes necessarily.

Our Hit efficiency work was largely invariant w.r.t. the changes in chi_sqr threshold and cluster noise thresholds.

We will be exploring why, upon modification of our steering file, we ceased to observe errors in our pulse profiles.

We will also be exploring hand picked profiles with very poor chi_sqr fits to determine if the algorithm can be fine tuned to improve upon these pulses.