TOP L1 Trigger Algorithm

Nisar N.K
Vladimir Savinov

Depatment of Physics & Astronomy
University of Pittsburgh

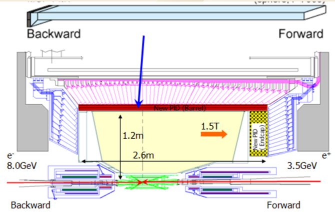
September 5, 2016

Imaging TOP Detector (from Kurtis N.)

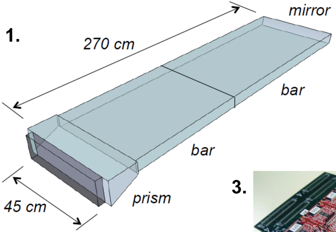
Belle II Imaging TOP Detector

Key elements:

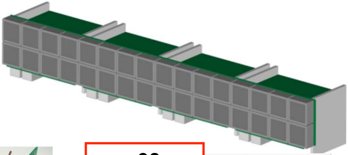
1. Quartz radiator and expansion.
   - Limited imaging relative to *x*,*t* TOP.
2. MCP-PMTs for fast timing.
3. Integrated readout electronics.



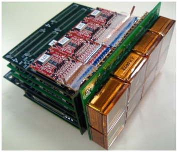
1.



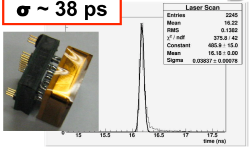
2.



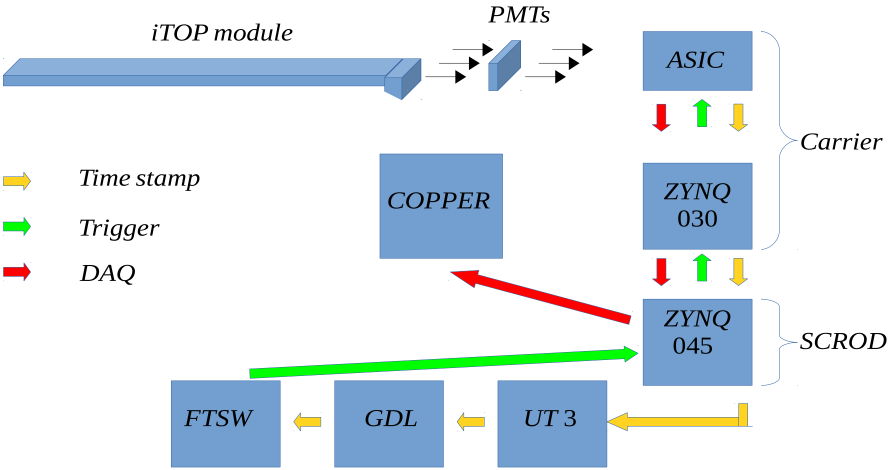
3.



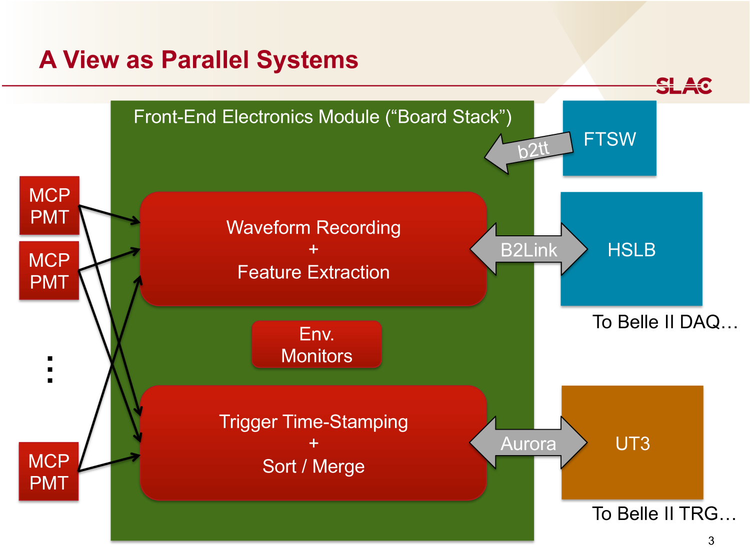
σ ~ 38 ps



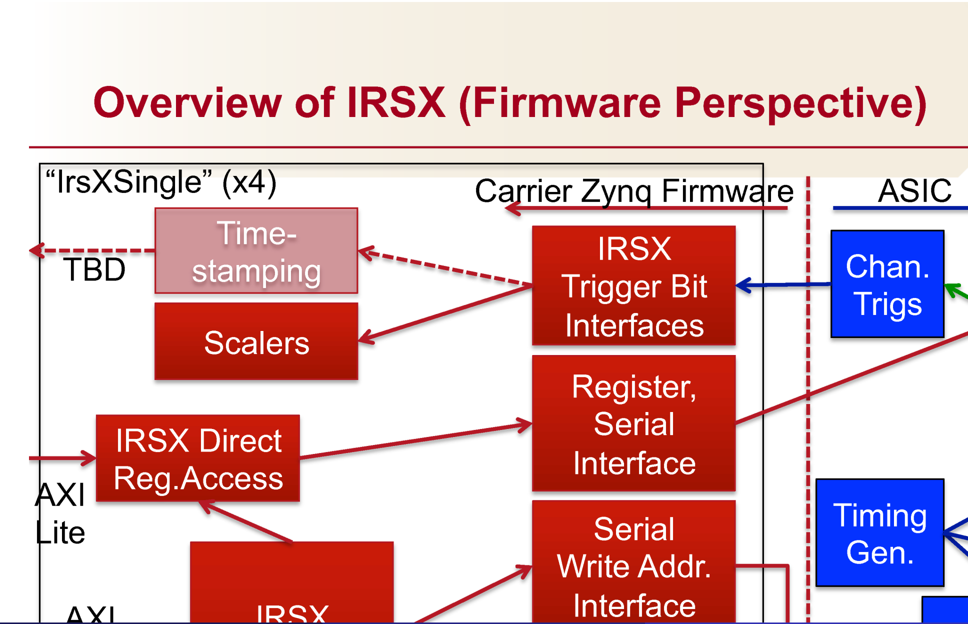
Data and Trigger Control Flow Diagram



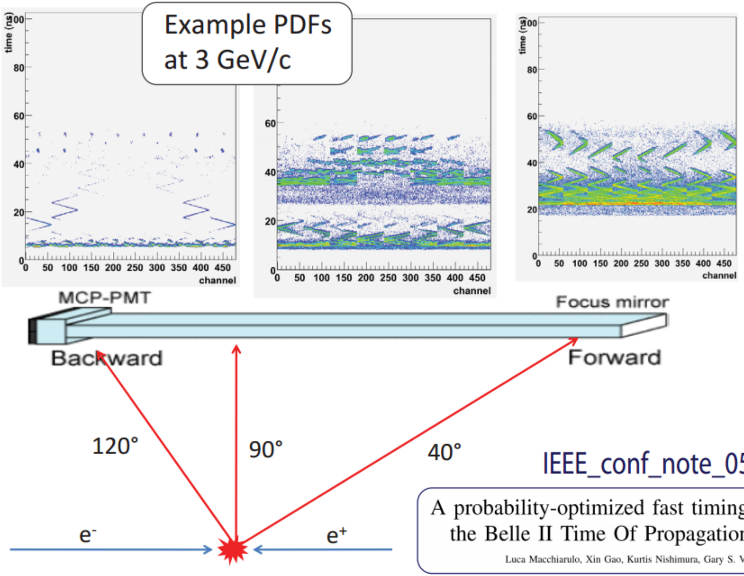
iTOP Data and Trigger Paths (from Kurtis N.)



Overview of IRSX (relevant excerpt, from Kurtis N.)



Performance of the Original L1 Trigger Algorithm



IEEE_conf_note_05873835

A probability-optimized fast timing trigger for the Belle II Time Of Propagation detector

Luca Macchiarulo, Xin Gao, Kurtis Nishimura, Gary S. Varner

Simulation: "Hardware", "Software" and basf2-based

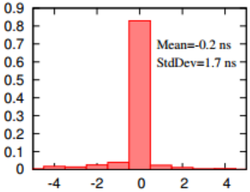
1. Hardware (functional and behavioral) simulation
   - Implemented in Xilinx IDE (ISE)
   - Assessment if FPGA (Virtex 6) on UT3 can execute firmware
   - Operates with continuous stream of timestamps
2. Software simulation
   - Implemented as a standalone C program
   - Validation of the algorithm implemented in firmware
   - Further development of iTOP-based L1 trigger algorithm
   - Operates with timeframes of hits, where, currently, each timeframe is 819.2ns (digitization frame in MC, $2^{14} \times 50ps$) long
3. Full simulation (of the detector)
   - Written in C++, implemented in basf2, creates timestamps from reconstructed hits
   - Provides tools to simulate iTOP-based trigger algorithm (but not the trigger hardware), to prepare PDFs and to study performance of the trigger algorithm
   - Currently, it estimates t_0 according to maximum likelihood within ±10ns of its true value. This needs to be changed.

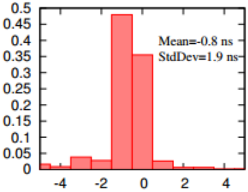
Xin's and Luca's Original t_0 Algorithm

- t_0 is estimated by maximizing the likelihood for a set of hits by matching them with PDFs that correspond to 10 logical segments of a quartz bar. As we do not know the arrival time of the first signal hit, we also scan PDFs (in time, over 100ns) to allow for an ambiguity in time position of the first signal hit.
- Here is how the algorithm actually works
  1. Take the fist hit in the frame as t_{initial}
  2. Estimate $\chi^2 = \sum_i \ln \mathrm{L}_i$ for a set of hits using PDFs
     - i For all 10 PDFs
     - ii For each PDF, shift hits in time (in 1ns increment, 100 times)

     to get the max value of χ^2.

     This allows us to estimate t_{PDF}, so we can estimate t_0:
  3. $t_0 = t_{\text{initial}} - t_{\text{PDF}}$



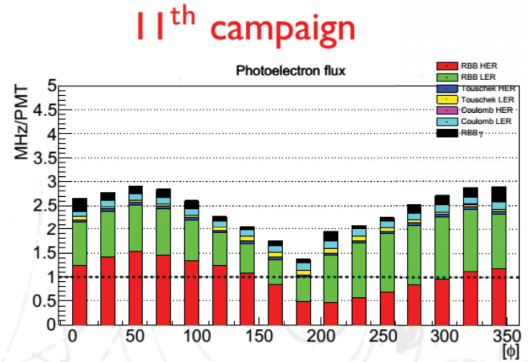


Trigger timing error for 1ns and 2ns hit time quantization (L. Macchiarulo, K. Nishimura, G. S. Varner and X. Gao, doi:10.1109/NSSMIC.2010.5873835)

Beam Background Rates

- Estimated photon detection rate on PMTs

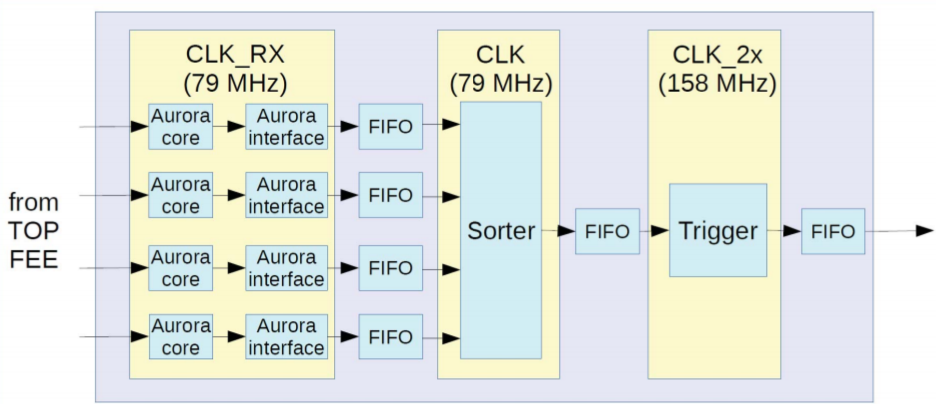
- 1.5-3 MHz/PMT
- 3-6 MHz/Carrier
- 12-24 MHz/SCROD
- 48-96 MHz/TOP bar
- Implications: a background hit every 10ns



from T.Nanut, 20th B2GM (Feb 2, 2015)

- Hits and PDFs are generated using the same detector geometry
- We use data from beam-related background campaign circa 2012, background hits from beam-related particles are distributed uniformly in time
- Signal hits could happen anywhere, so we simulate them with a predefined shift w.r.t. the start of the frame with the hits

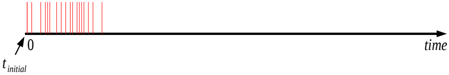
Trigger Algorithm



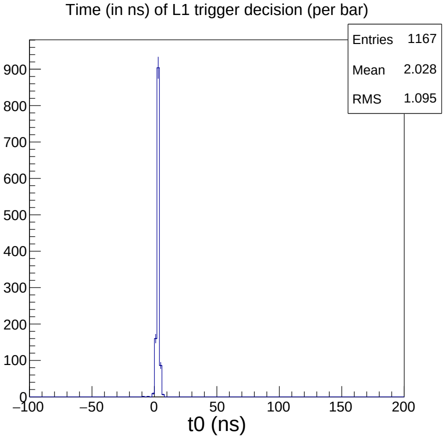
- Currently, TOP trigger algorithm runs (on UT3) at 127 MHz, so our bandwidth is barely sufficient to handle background alone. At projected rate of background there will be no empty clock cycles, so there will be no gaps that could be otherwise used to identify where signal hits possibly start (as was implemented in the original algorithm)
- Recently we came up with an algorithm to identify the location of signal hits in the frame (the sequence of hits), so we solved the "how to find the first signal hit" problem

Xin's algorithm (software simulation, no background), signal hits start right away (true $t_0 = 0$ns)

$p(\pi) = 2.5\text{GeV}, \theta = 90^\circ, \phi = 90^\circ, t_{\text{shift}} = 0\text{ns}$

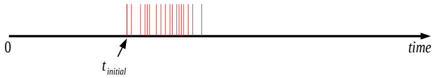


- Sample contains signal only
- Signal hits start at the beginning of the frame
- All good, as expected

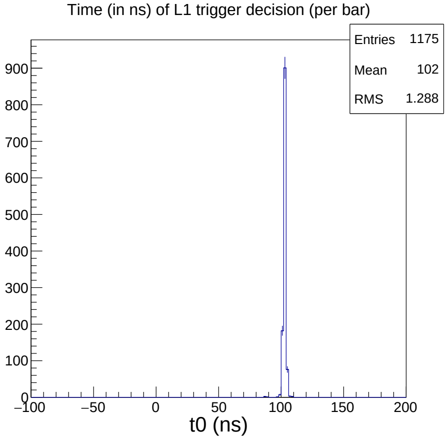


Xin's algorithm (software simulation, no background), signal hits are shifted by 100ns

$p(\pi) = 2.5\text{GeV}, \theta = 90^\circ, \phi = 90^\circ, t_{\text{shift}} = 100\text{ns}$

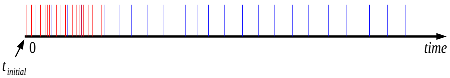


- Sample contains signal only
- Signal hits start after 100ns
- All good, as expected

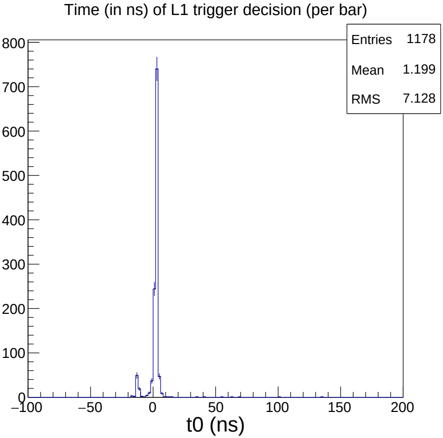


Xin's algorithm (software simulation in presence of background), signal hits start right away (true $t_0 = 0$ns)

$p(\pi) = 2.5\text{GeV}, \theta = 90^\circ, \phi = 90^\circ, t_{\text{shift}} = 0\text{ns}$

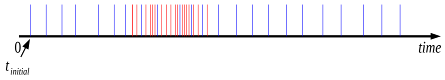


- Sample contains both signal and background
- Signal and background hits start simultaneously at the beginning of the frame
- Appearance of the second peak is not surprising and is expected
- All good, as expected (and is consistent with the results as published)

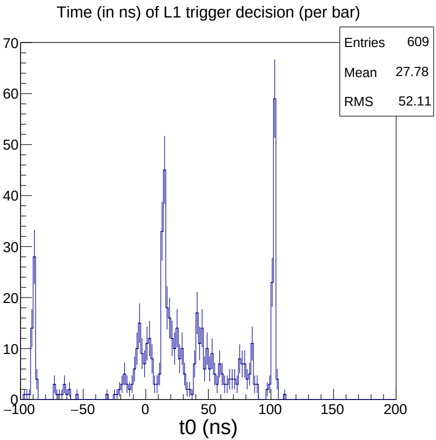


Xin's algorithm (software simulation in presence of background), signal hits are shifted by 100ns

$p(\pi) = 2.5\text{GeV}, \theta = 90^\circ, \phi = 90^\circ, t_{\text{shift}} = 100\text{ns}$

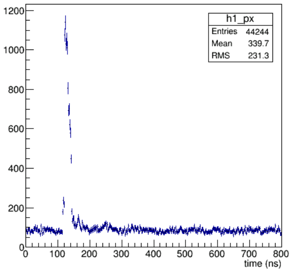


- Sample contains both signal and background
- Background hits start at the beginning of the frame while the signal hits are shifted by 100ns
- As expected, Xin's algorithm does not work when signal hits appear later in the frame (no surprise)
- Xin's algorithm would work perfectly fine if we told it *where* the first signal hit is in the frame
- Identifying where the signal hits are had been a long standing problem that we finally solved

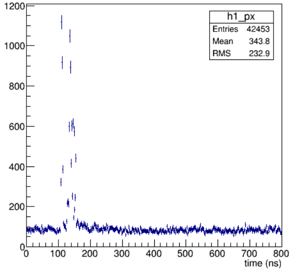


Instantaneous Density of Hits in Presence of Signal

- The hits appear more frequently in a signal region compared to beam background only



$\theta = 45^\circ$



$\theta = 90^\circ$

- We count the number of hits in 8ns windows (actually, we are taking 8 time-sorted hits to estimate instantaneous density of hits USING Δt, this is how it will work in firmware)
- The windows are scanned throughout the range of frame (for the plots shown here, NOT in firmware, where we can't "scan"!)
- We find an enhancement in the number of hits in the signal region
- Average number of signal hits from a high-momentum particle is between 15 and 40, so approx. 8 clock cycles would be sufficient to estimate hit density reliably (using Δt between the first and the last time-sorted hits)

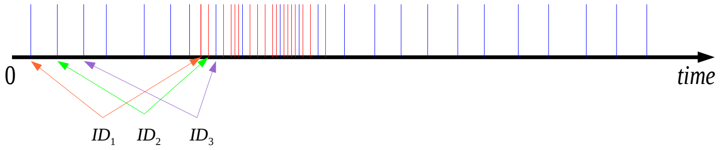
Algorithm Based on Instantaneous Hit Density

- We use real-time hit density estimate to identify the signal
- Consider timestamps in a frame: $t_1, t_2, t_3, ..., t_n$
- Instantaneous Hit Time Density (ihtd):

$$\text{ihtd} = \frac{N}{t_{i+N} - t_i}$$

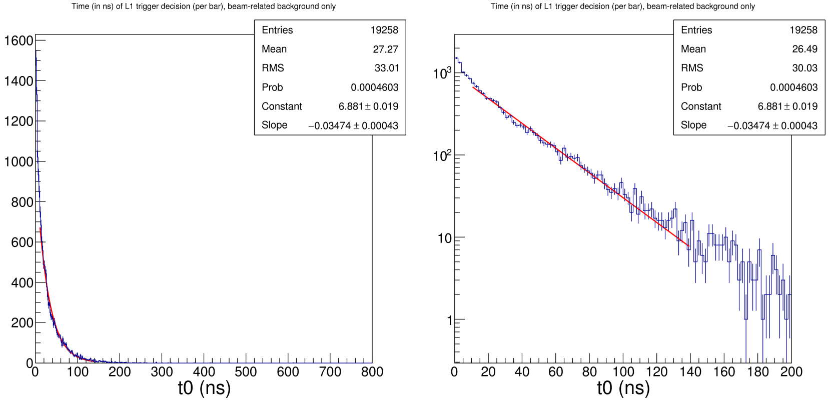
N - the number of hits used to estimate hit time density

- We "watch" the timestamps in real time:



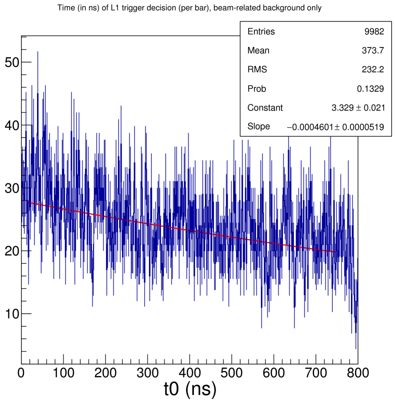
- Whenever the ihtd_i exceeds a (programmable) threshold, we take the t_i as t_{initial}
- The axis on this plot represents the values of timestamps, so many of the timestamps could be the same. However, timestamps arrive at trigger algorithm running on FPGA one at a time. Therefore we can use a small number of clock cycles to estimate time density of hits using the timestamps. In FPGA implementation we will be using Δt between the first and the last time-sorted hits.

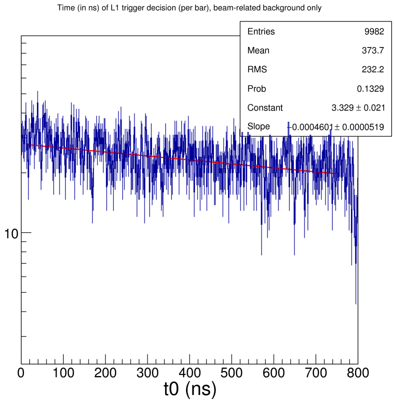
t_0 obtained for background (the original algorithm)



- Time constant from the fit: ns/0.035 = 29ns (the original algorithm)
- Conclusion: beam background-related t_0 decisions are too frequent

t_0 obtained for background (the improved algorithm)

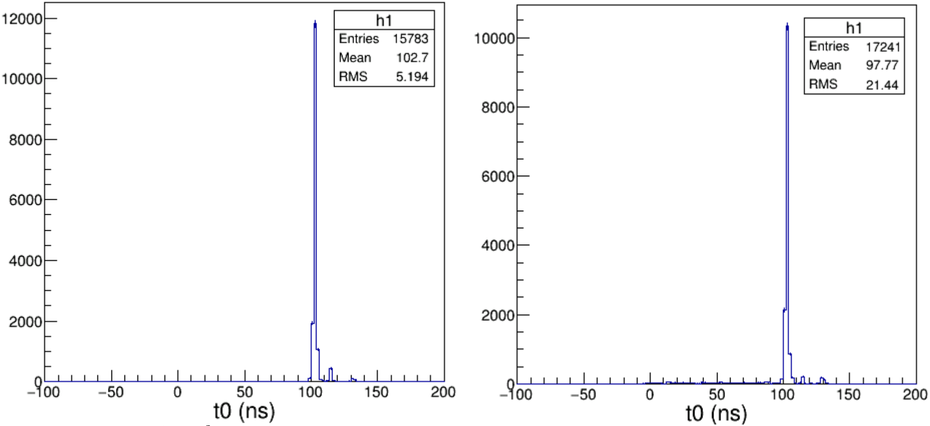




- Time constant from the fit: ns/0.00046 = 2.2μs
- Conclusion: this is acceptable (actually, it's awesome!)

Results: t_0 for signal+background, the improved algorithm

$p(\pi) = 2.5\text{GeV}, \theta = 90^\circ, \phi = 90^\circ, t_{\text{shift}} = 100\text{ns}, \text{ihtd} > 2\text{ns}^{-1}$



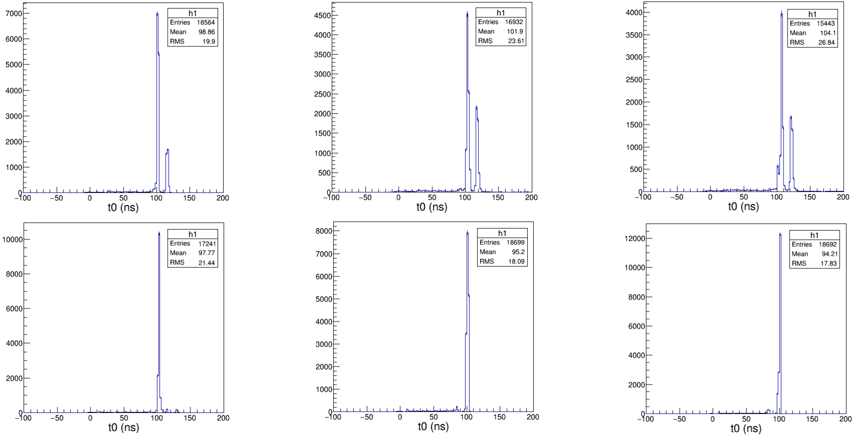
signal　　　　signal+background

- New algorithm works when the signal happens much later than the first hit
- RMS of the peak is below $2ns$, secondary peaks appear where expected

Results: t_0 for signal+background, the improved algorithm

$p(\pi) = 2.5\text{GeV}, \theta = 45/60^\circ, \phi = 90^\circ, t_{\text{shift}} = 100\text{ns}, \text{ihtd} > 2\text{ns}^{-1}$

$\theta = 45^\circ$ $\theta = 60^\circ$ $\theta = 75^\circ$

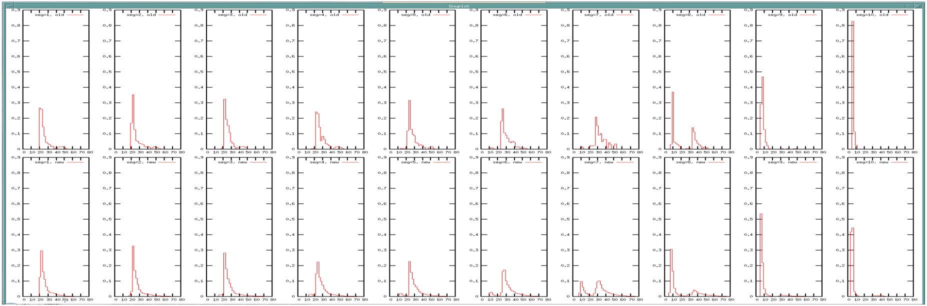


$\theta = 90^\circ$ $\theta = 105^\circ$ $\theta = 120^\circ$

- Double peaks in t_0 distribution are expected for $\theta = 45^\circ,\ 60^\circ$ and 75°
- The second peak in t_0 is from low statistics and PDFs (see the next slide)

PDFs

$\theta = 45^\circ$ $\theta = 90^\circ$ $\theta = 120^\circ$



- Upper row shows Xin's PDFs while the lower ones are PDFs generated by Vladimir using full detector simulation (horizontal axis is in units of ns)
- For small number of signal hits, the first (corresponding to $\theta = 45$) and the last (corresponding to $\theta = 120$) PDFs looks similar to each other
- Matching hits from a wide single-peaked PDF with wrong PDFs gives a wrong t_{PDF} and leads to a wrong t_0, however, as 9th and 10th PDFs are narrow, this does not happen as often for particles hitting the forward region
- Note that we do not need to scan in time through 100ns anymore, as now we know how to locate the signal, so using time windows of ≤ 50ns should be sufficient

Summary

1. The original algorithm could work in real time in low background environment
2. At nominal background we need an estimate of the first signal hit arrival time
3. The instantaneous density of hits ($i.e.$ Δt) can be used to locate the signal
4. The problem of background trigger decisions being too frequent is no more
5. Nisar is currently working on implementing this algorithm in firmware
6. Our next goal is to figure out how to combine t_0's from individual bars on UT3
7. Vladimir now knows what exactly to implement in basf2-based simulation
8. Currently, we are NOT working on timestamping on the boardstacks