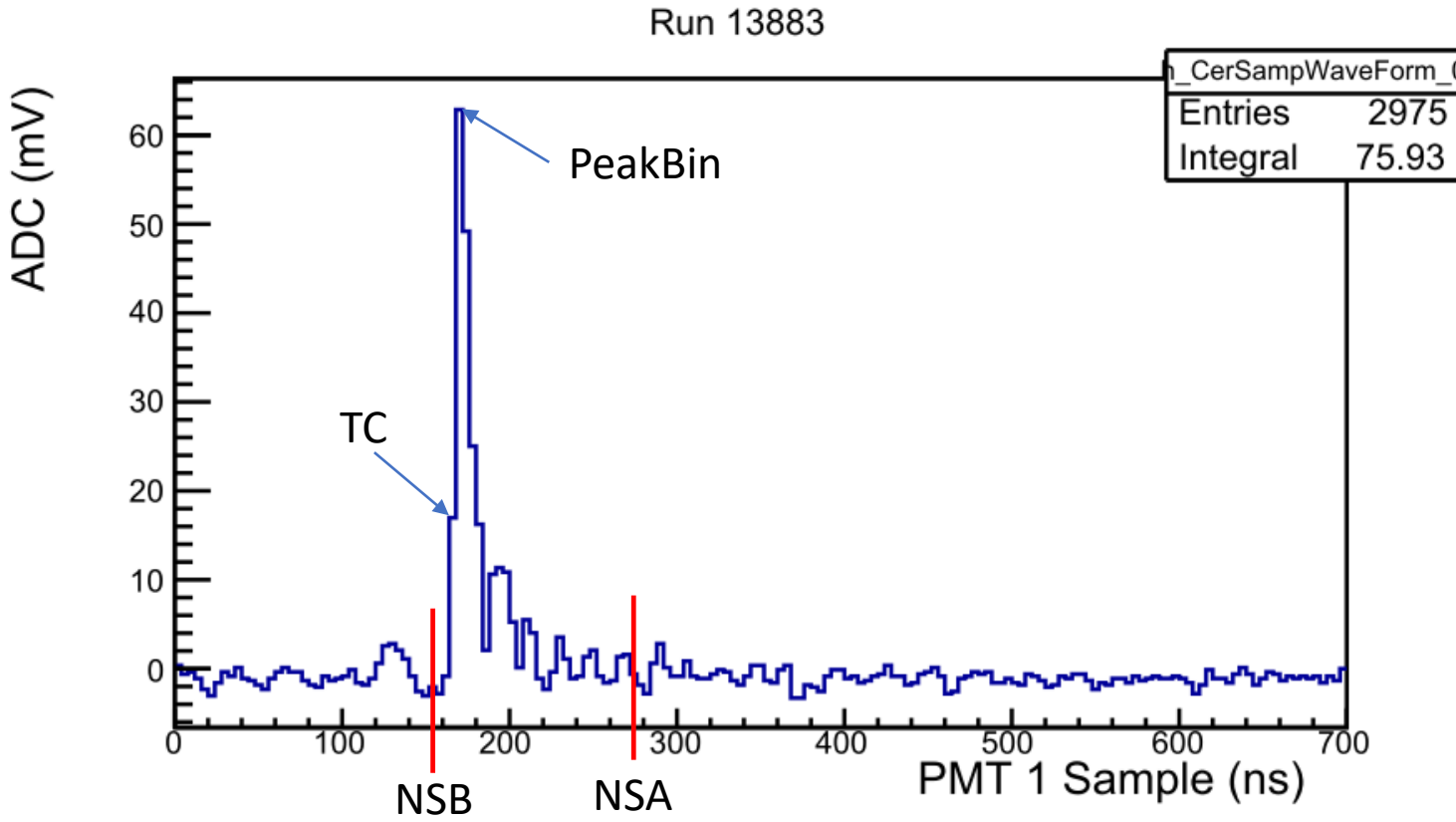
HCANA update for new FADC Firmware update and analyzing waveform

- Older firmware:
  - If FADC could not determine the pedestal , because signal in first 4 samples above the threshold, the code would return pulse information: pedestal, integral, amplitude = 0 and time = threshold crossing bin. This was given an error_flag ==1 in HCANA.
  - The HCANA fix to this problem was if the amplitude = 0 then use a default pedestal in parameter file to subtracted from the integral.
  - Another case when FADC would return amplitude =0 but only for the effected pulse. When the FADC could not determine the amplitude for the pulse because it was at the far edge of the time window which cut-off the peak.
- Asked to update firmware so that it would return the waveform in addition to the pulse info
  - Unfortunately, the firmware was updated to just return the waveform and not return any pulse information
- Decided to readout the Cerenkov and aerogel in FADC mode 10
  - This gives the option of using the Waveform.
  - Now when Waveform is present and no pulse information , HCANA analyzes the Waveform to get the pedestal (from the last 4 samples of the waveform), integral, amplitude and time. The Waveform data is placed in the variables for the usual pulse data and the analysis proceeds as usual. This is given an error_flag = 3 to be able to monitor these type of events.
  - There is a parameter so that for every event the Waveform data is placed in the variables for the usual pulse data and the analysis proceeds as usual.

Define terminology

- NSAT = Number of Samples above Threshold
  - For FADC configuration NSAT=1
  - HCANA can set NSAT with default = 2
- Pedestal determined by the average of ADC in first four time bins.
- Threshold is 10mV above pedestal.
- HCANA can set the threshold.
- TC = Threshold Crossing is first bin in the NSAT that is above threshold relative to the pedestal
- NSB = Number of Samples Before TC
- NSA = Number of Samples After TC
- In F250 configuration NSB=3 (12ns) and NSA=26 (104ns).
- HCANA can set NSB and NSA. Default is to use the F250 configuration.

Pedestal subtracted sample waveform
Each time bin is 4ns.



Analyzing FADC Waveform data: Determine the pedestal

- New method SetSampIntTimePedestalPeak in THcRawAdcHit
  - Mimic the algorithm in the FADC firmware
  - Default use NSA and NSB from the event 125 which stores the F250 config info.
  - Default threshold fSampThreshold = 10 mV
  - Default NSAT = 2
  - Use parameters:
    - Each detector can set the threshold by the method : SetSampThreshold(Double_t thres)
    - Each detector can set the NSAT by the method : SetSampNSAT(Int_t nsat)
    - Each detector can set the NSA,NSB by the method :SetF250Params(Int_t NSA,Int_t NSB,Int_t NPED)

- Determine pedestal
  - If pulses with waveform then normal and treat like standard FADC
    - fSampPed = GetIntegral(0, fNPedestalSamples-1)
    - fNPedestalSamples from FADC settings that are saved in CODA file at PreStart and is usually 4.
  - If no pulses and only the waveform then FADC error. Use samples at the end of the waveform
    - fSampPed = GetIntegral(fNSamples-fNPedestalSamples, fNSamples-1)

Analyzing FADC Waveform data: Get pulses integral, amplitude and time

Loop through NS< fNSamples (or stop if fNSampPulses<fMaxNPulses==4)

- Start at sample number NS = fNPedestalSamples-1
- If FADC error then keeping looping until finding sample that is below fSampThreshold.
- Check the next NSAT pulses to see if pulses above fSampThreshold
- If NSAT pulses not above threshold then NS++ and back to start of loop.
- If NSAT pulses above threshold
  - fSampPulseInt[fNSampPulses] = GetIntegral(TMath::Max(NS-fNSB,0),TMath::Min((NS+fNSA-1),int(fNSamples-1)));
  - fNSB (Number of Samples Before) and fNSA (Number of Samples After) from FADC settings that are saved in CODA file at PreStart. Can be set by parameter to supercede FADC settings.
  - Find the PeakBin by looping through the samples NS+1 to NS+fNSA and finding the sample bin which has a smaller ADC value than the previous bin and setting PEAKBIN to previous bin number.
  - fSampPulseAmp[fNSampPulses] = GetSampleRaw(PeakBin);
  - Determine the raw pulse time and mimic the FADC algorithm which stores the raw time in 64 bits
  - Vmid = GetSample(PeakBin)/2. and find which two samples are below (sample number = nt) and above the Vmid.
  - fSampPulseTime[fNSampPulses] = 64*nt + int(64*(VMid-GetSample(nt))/(GetSample(nt+1)-GetSample(nt)));
  - If the code cannot find a PeakBin, then amplitude= GetSampleRaw(NS) and time=NS*64.
  - Set fNSampPulses++ and NS=NS+fNSA and continue the loop

Update to THcCherenkov to analyze FADC mode 10 data

- New parameters (Det == phgcer for example)
  - Det_OutputSampWaveform : =0 no waveform data in tree, = 1 waveform data in tree
  - Det_SampThreshold : Threshold (mV) relative to the pedestal for finding pulses in waveform
  - Det_SampNSA : Number of Samples After Threshold for pulse integral. Integer with sample bin = 4ns.
  - Det_SampNSB : Number of Samples Before Threshold for pulse integral. Integer with sample bin = 4ns.
  - Det_SampNSAT: Number of Samples Above Threshold. Integer
  - Det_UseSampWaveform: =0 do not use waveform data to replace FADC pulse data, = 1 waveform data replaces pulse data and use in the determination of “good” data.
- New tree variables which are the same as the FADC pulse but determined by the FADC waveform.
- The waveform data for all channels in a detector is put in one array that is saved to the tree with the following format:
  - adcSampWaveform[0] == the PMT channel number
  - adcSampWaveform[1] == the number of samples =NS
  - adcSampWaveform[2] == 1st sample Pedestal subtracted ADC (mV) value
  - adcSampWaveform[3] to adcSampWaveform[3+NS-1] == Pedestal subtracted ADC (mV) value
  - adcSampWaveform[NS+3] == the PMT channel number
  - adcSampWaveform[NS+4] == the number of samples =NS
  - adcSampWaveform[NS+5] == 1st sample Pedestal subtracted ADC (mV) value
  - Etc.

Methods that detectors can use to get RawAdcHit information

- New methods in THcRawAdcHit
  - GetNSamples() returns fNSamples, the number of samples in the wavefrom
  - GetSampPedRaw() returns fSampPed, the pedestal integral in ADC channels
  - GetNSampPulses() returns the number of pulse found in the waveform of fNSamples. Max is 4.
  - GetSampPulseIntRaw(iPulse) returns the integral in ADC channels for iPulse
  - GetSampAmpRaw(iPulse) returns the amplitude in ADC channels for iPulse
  - GetSampPulseTimeRaw(iPulse)) returns time in ns for iPulse
  - GetSampPed() returns the average of the fNPedestalSamples in mV
  - GetSample(iSample) returns the pedestal subtracted ADC value in mV for the sample number iSample
  - GetSampPulseInt(iPulse) returns the pedestal subtracted integral in pC for iPulse
  - GetSampAmp(iPulse) returns the pedestal subtracted amplitude in mV for iPulse
  - GetSampPulseTime(iPulse) returns the reference time subtracted Adc time in ns for iPulse