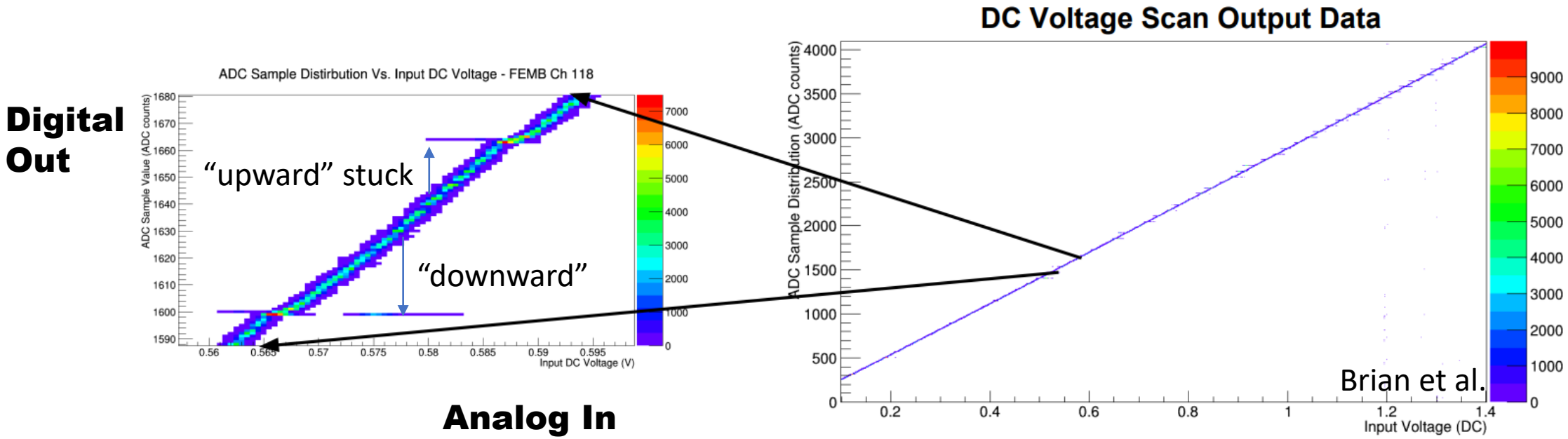
Sticky Code Mitigation for protoDUNE

Wenqiang Gu

Brookhaven National Laboratory

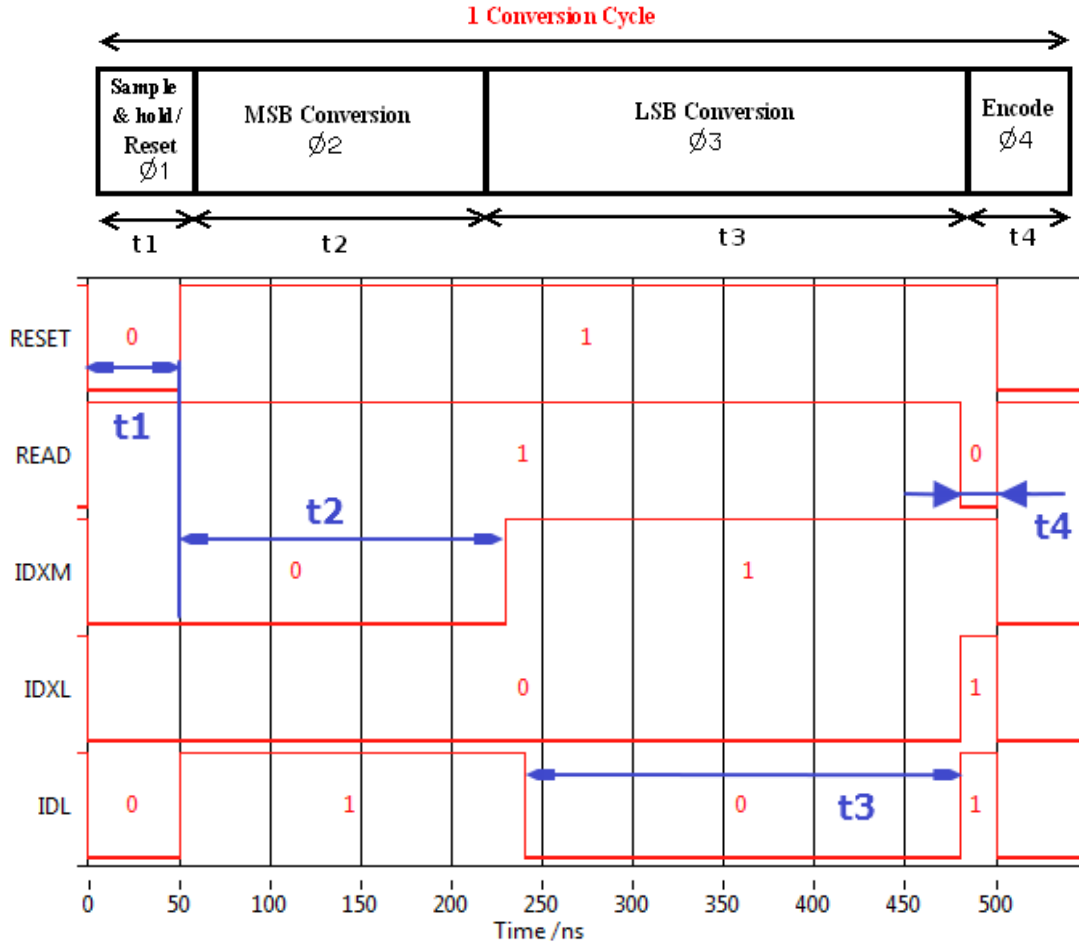
Sticky Code

- The 6 LSBs in ADC ASIC was found to be “sticky” around 000000 (0x00) or 111111 (0x3F)
- So called sticky code, or stuck bit



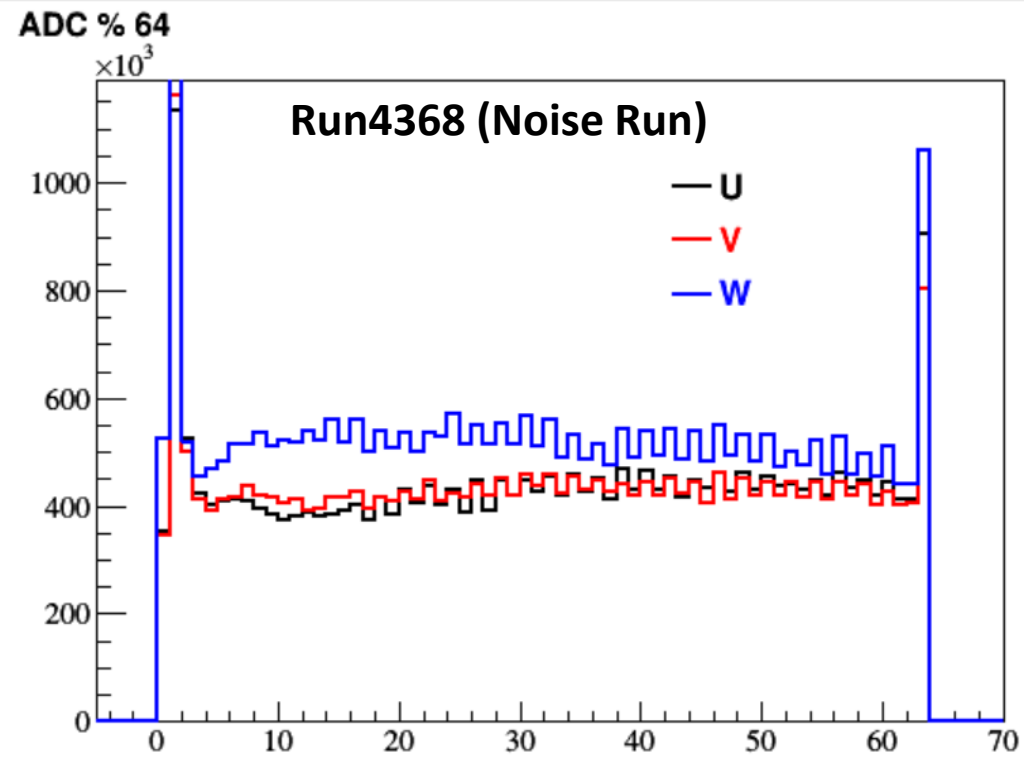
ADC Digitization and Sticky Code

- Two stages of a 12-bit digitization
  - 6 MSBs (most significant bits)
  - 6 LSBs (least significant bits
- Analog input compared with MSB first
- Sticky code issue happens between the conversion of MSBs and LSBs
- Sticky code represents a loss of information



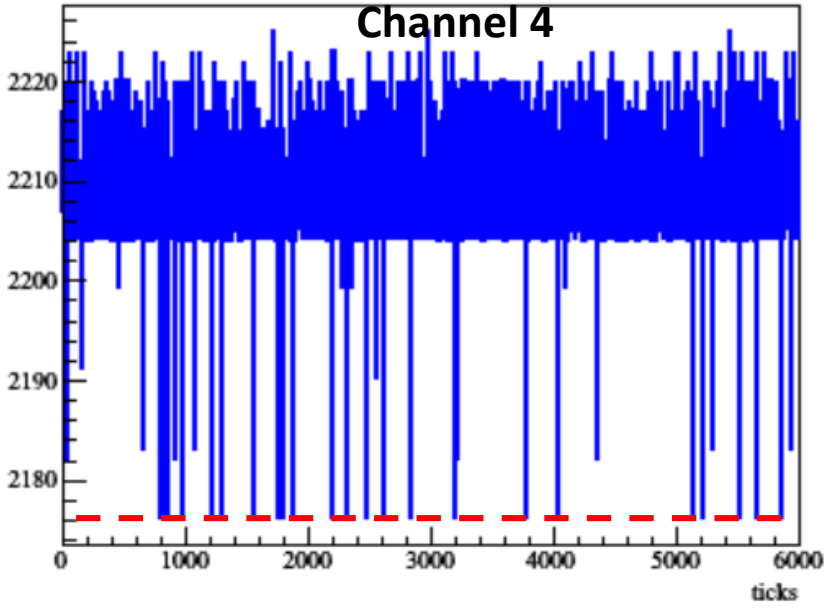
ADC % 64 (mod64)

- Sticky at bit 0, 1, 63



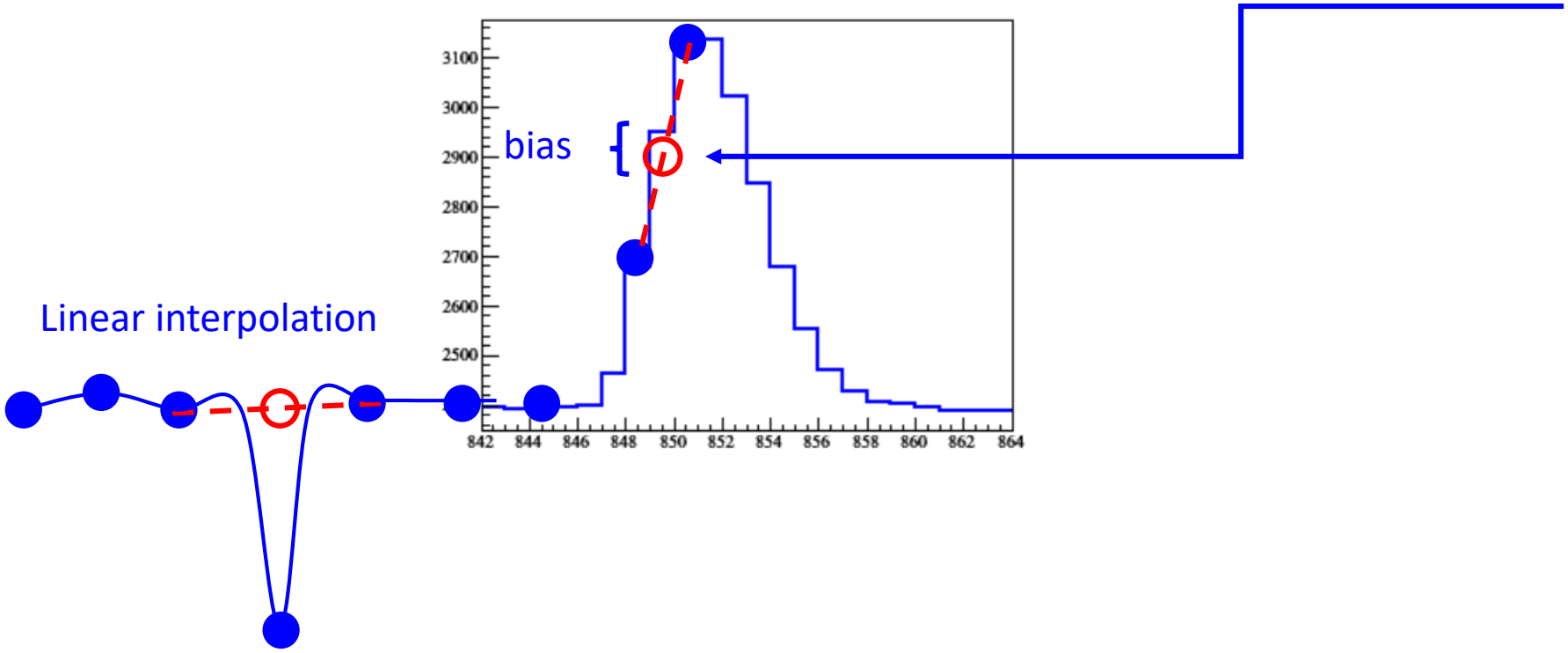
- Example waveform of sticky at bit 0

$2176 = (100010{,}000000)_2$



Sticky Code Mitigation

- Linear interpolation between “un-sticky” codes is a good first step
- However, linear interpolation may not be sufficient for signal region



https://en.wikipedia.org/wiki/Fourier_transform

Interpolation via Fourier Transform (FT)

- Shift in time domain

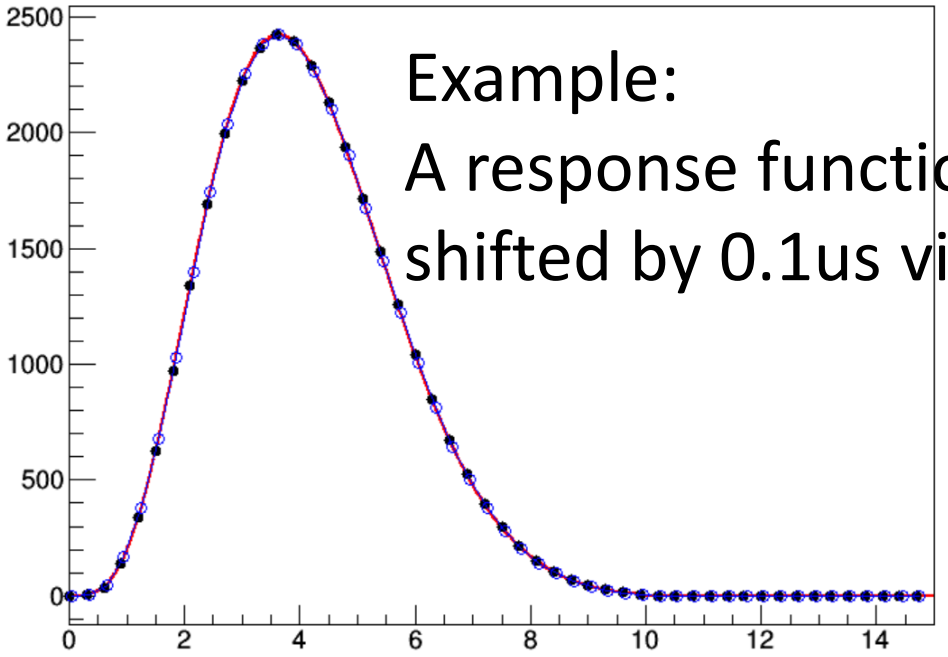
  ⇔ Phase shift in frequency domain

$$f(x-a) \Leftrightarrow e^{-2\pi i a \epsilon}\, \hat{f}(\epsilon)$$

- Advantages of FT
  - Only phase changed. No change of magnitude in frequency domain

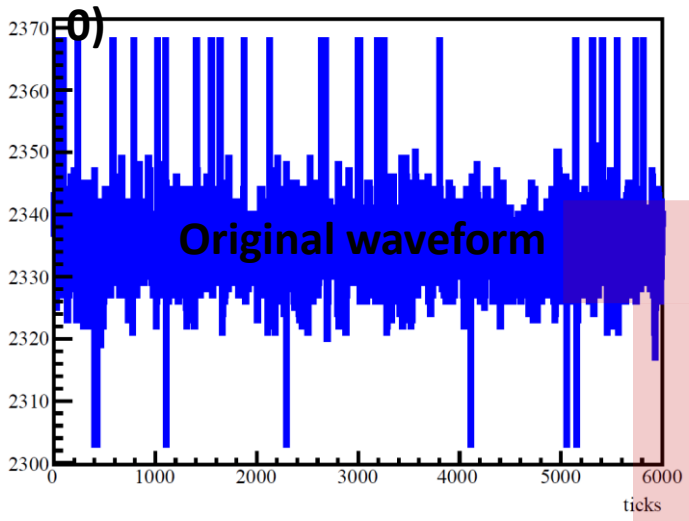
    ➔Respect the shaping of electronics response function
  - Sometime good codes tagged as “sticky”, FT interpolation presumably minimize the biases

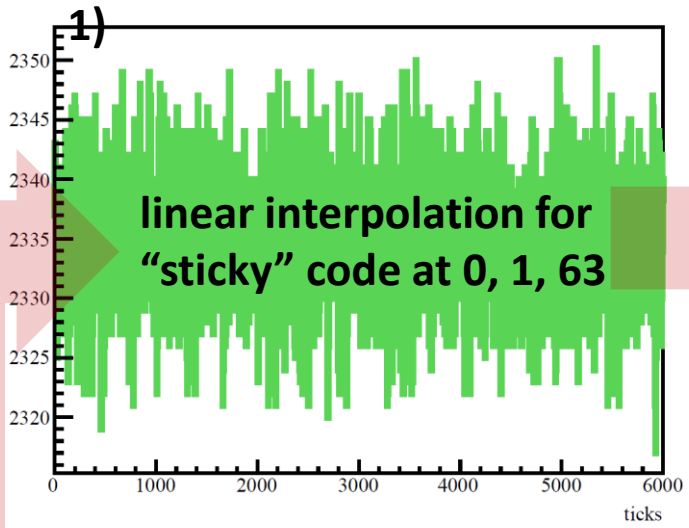
    ➔ Balance of efficiency and accuracy for sticky code tagging

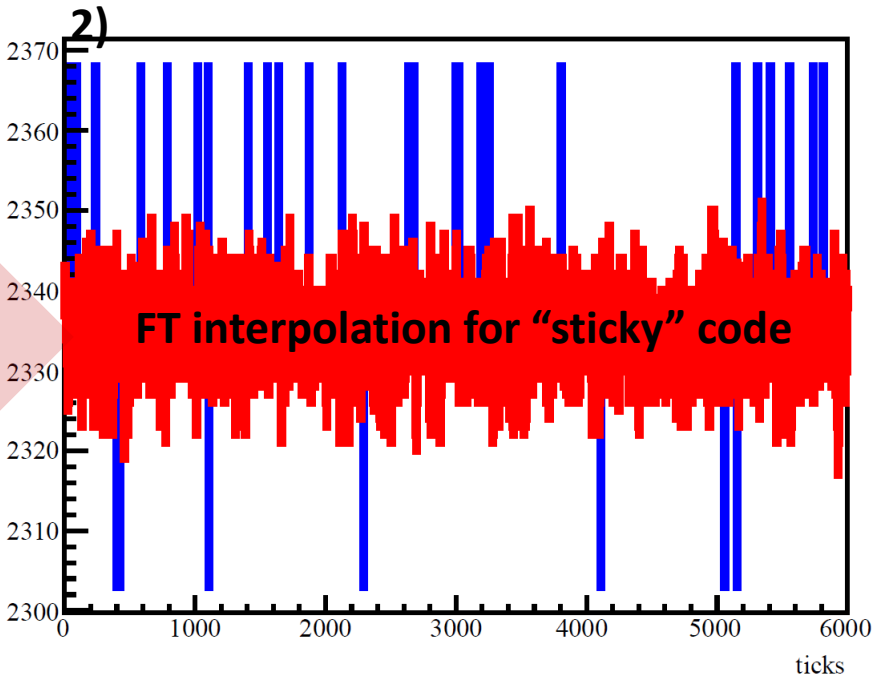


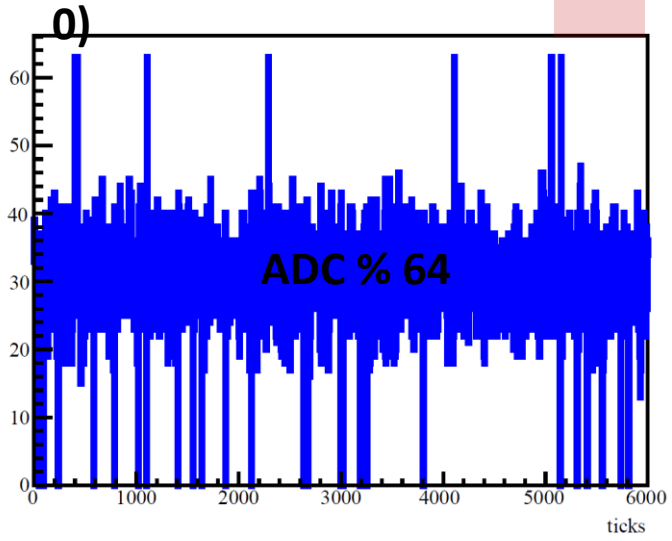
Example:
A response function shifted by 0.1us via FT

Mitigation Procedure







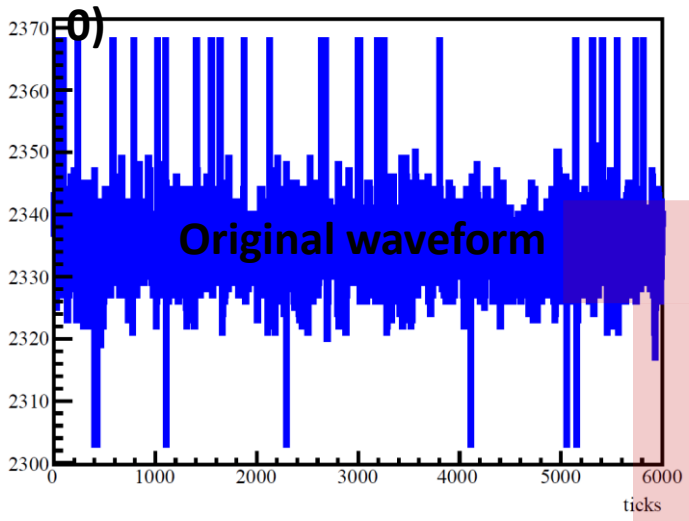


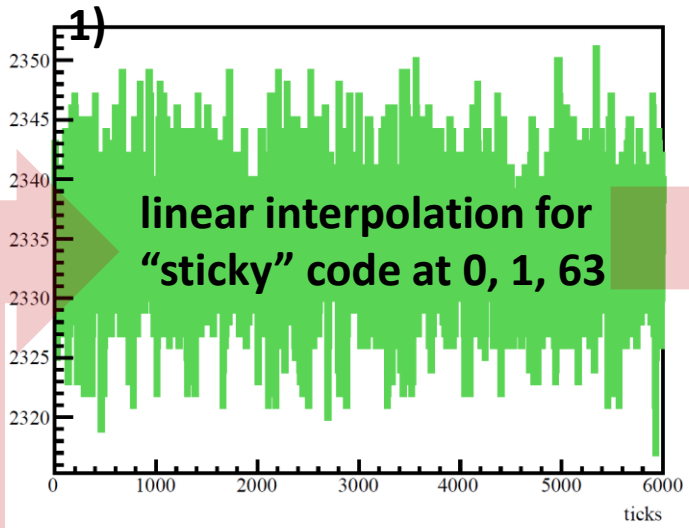
0) Identify sticky codes by bit 0, 1, 63

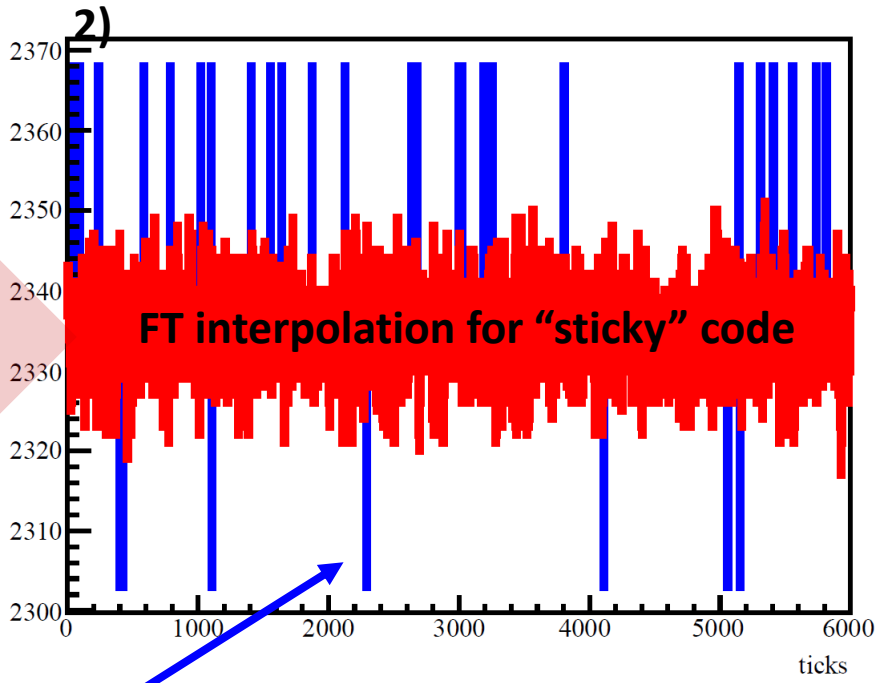
1) Linear interpolation for sticky codes

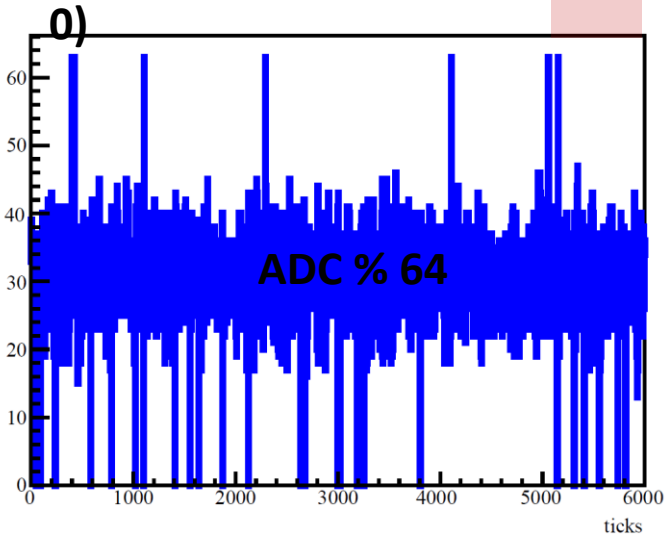
2) Apply FT interpolation on the linearly interpolated waveform

Mitigation Procedure (Cont')









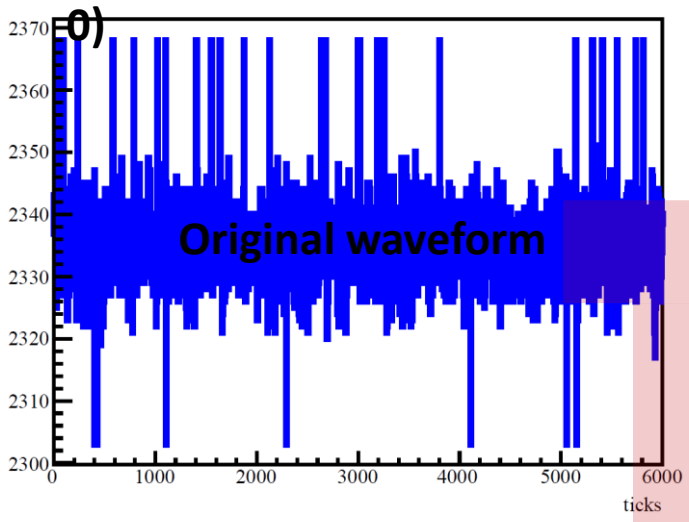
- For a single sticky code,

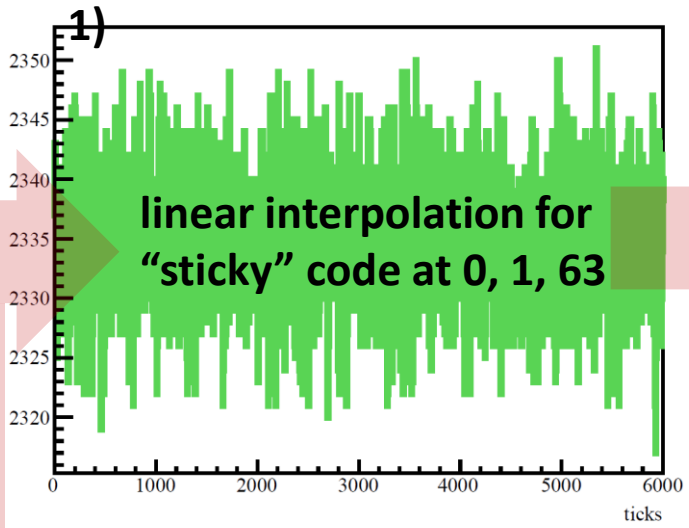
- If the ticks number is even, interpolate this tick with odd-numbered waveforms, and vice versa.

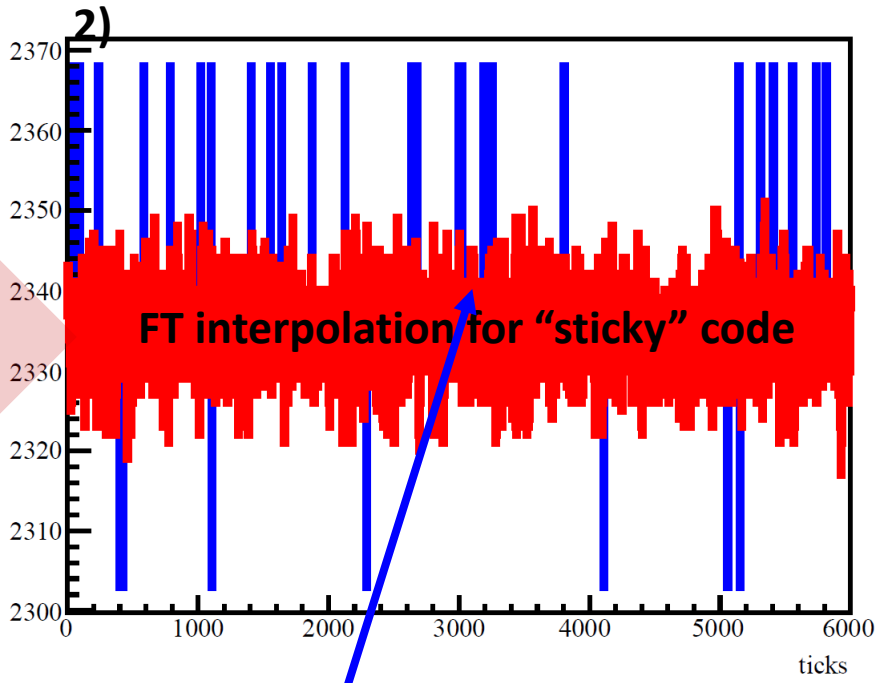
- This basically "reuse" the nearby waveform, while not "create" new waveform

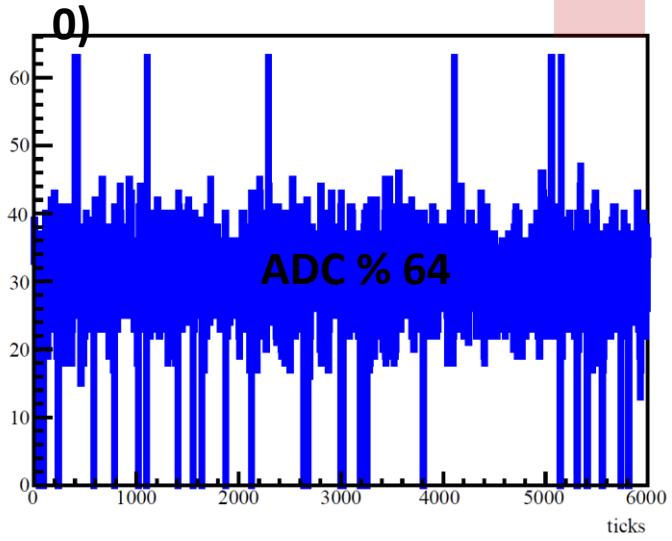
- Thanks to the 2MHz oversampling

Mitigation Procedure (Cont')



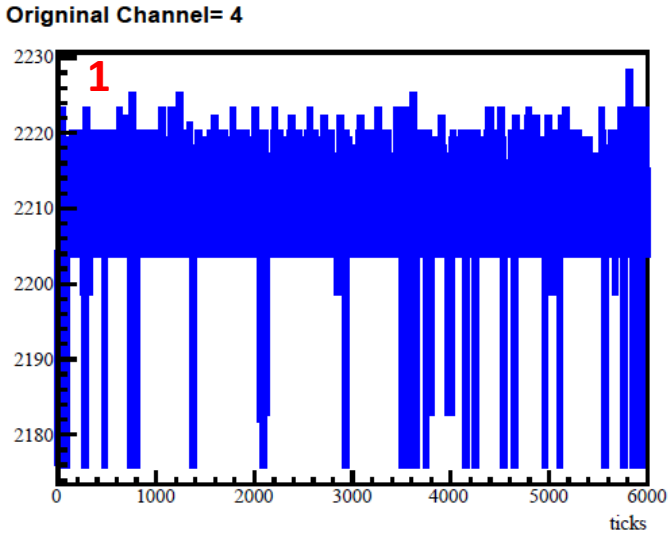


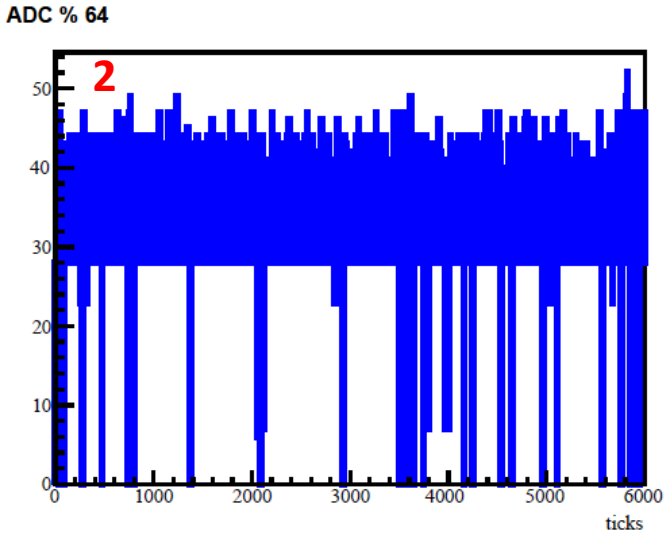




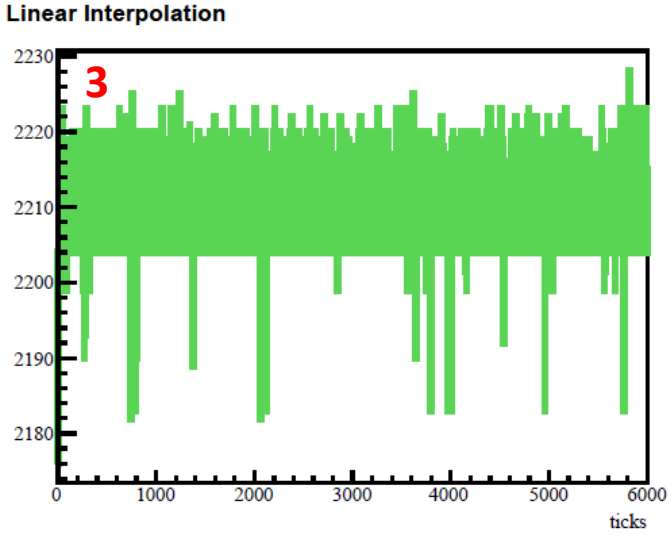
- For a few adjacent sticky codes,
  - FT interpolation based on the linearly interpolated waveform
  - Avoid the biased information from nearby waveform

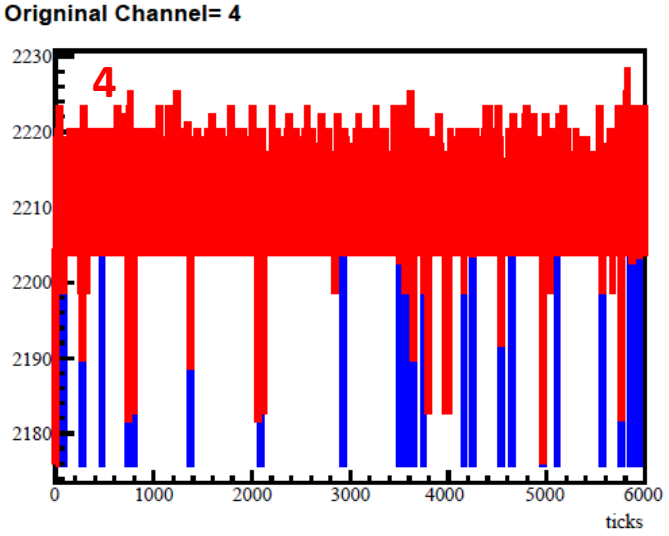
Example (Run 4368, Event 82)

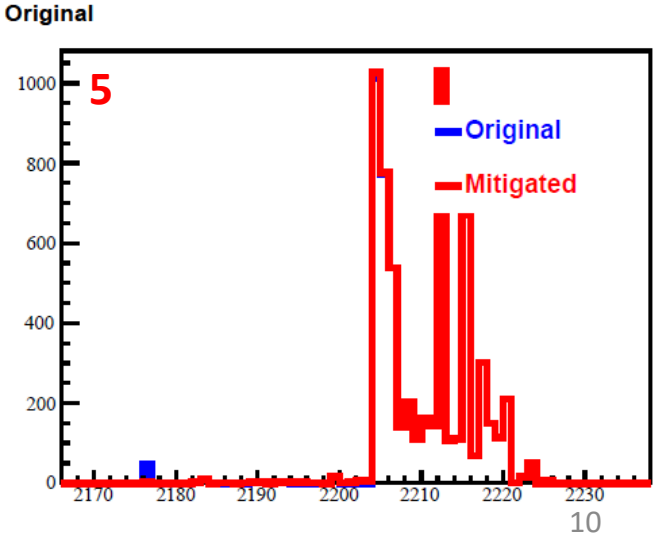




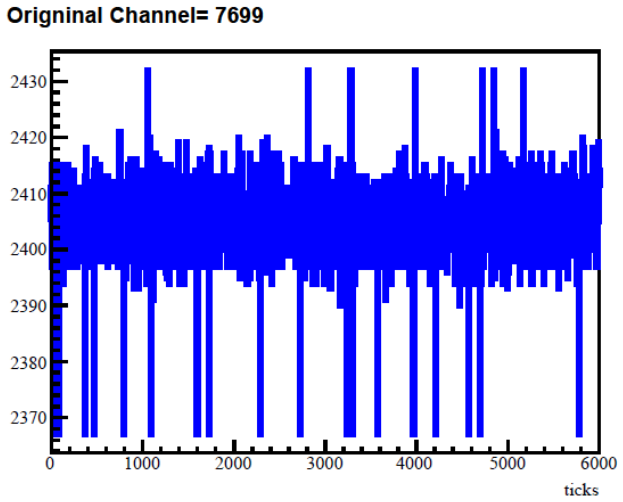
1. Original waveform
2. ADC % 64
3. “Pre-correction”: linear interpolation
4. Original vs. Mitigated
5. Noise level projection of Fig. 4

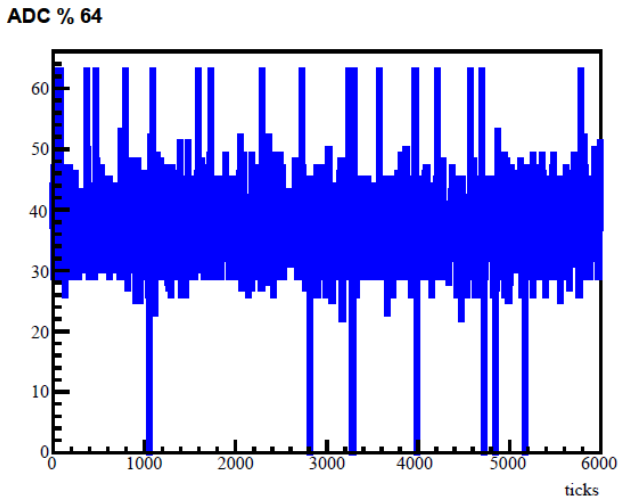


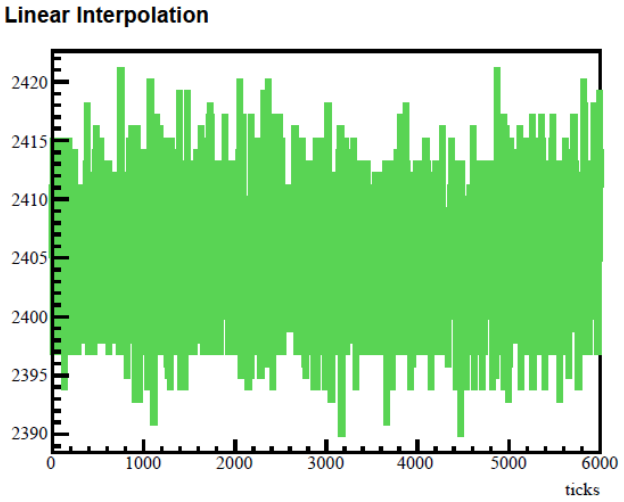


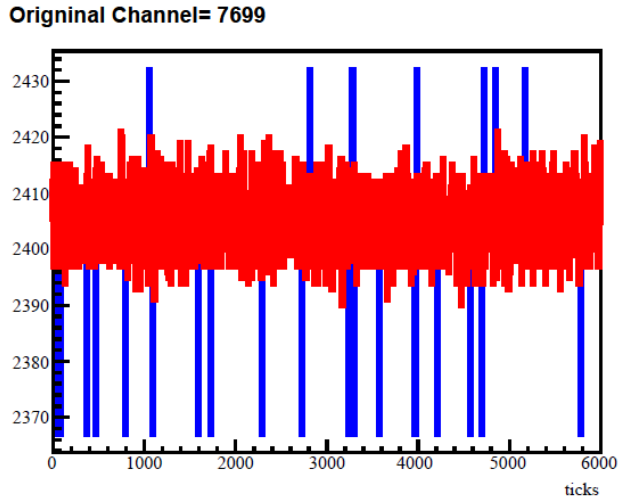


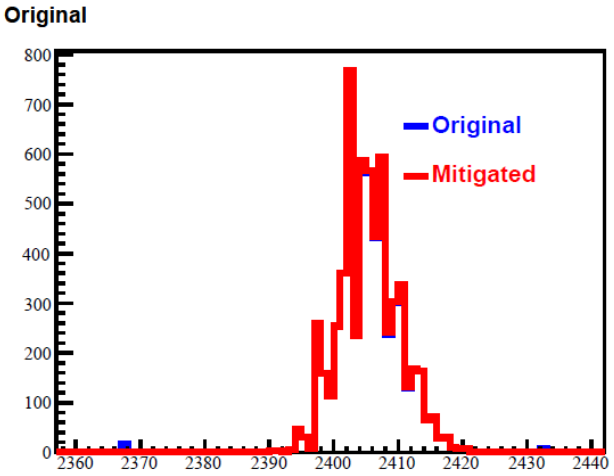
Example (Cont’)





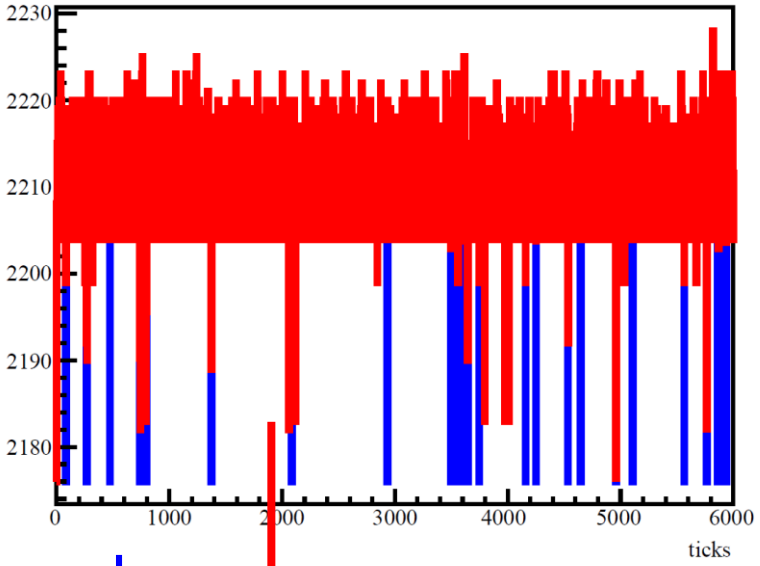


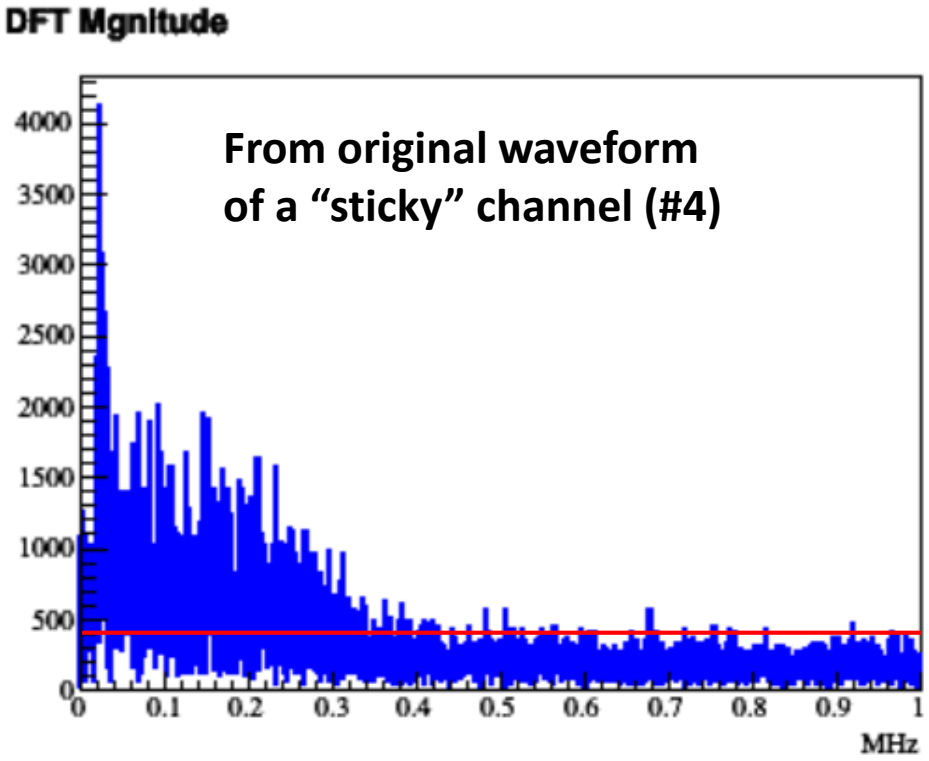


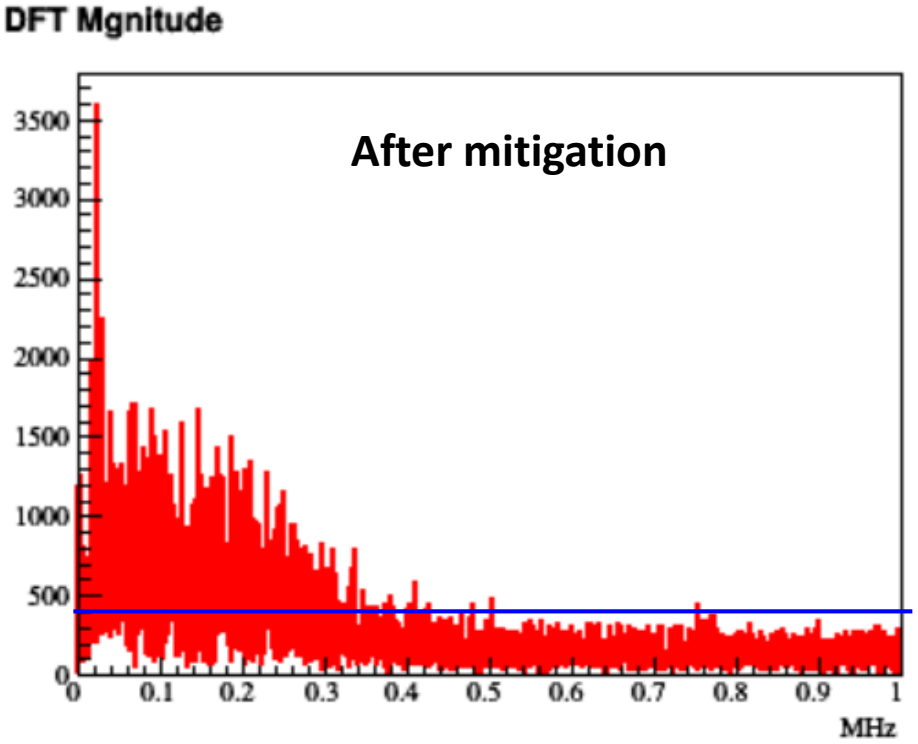


DFT Spectrum

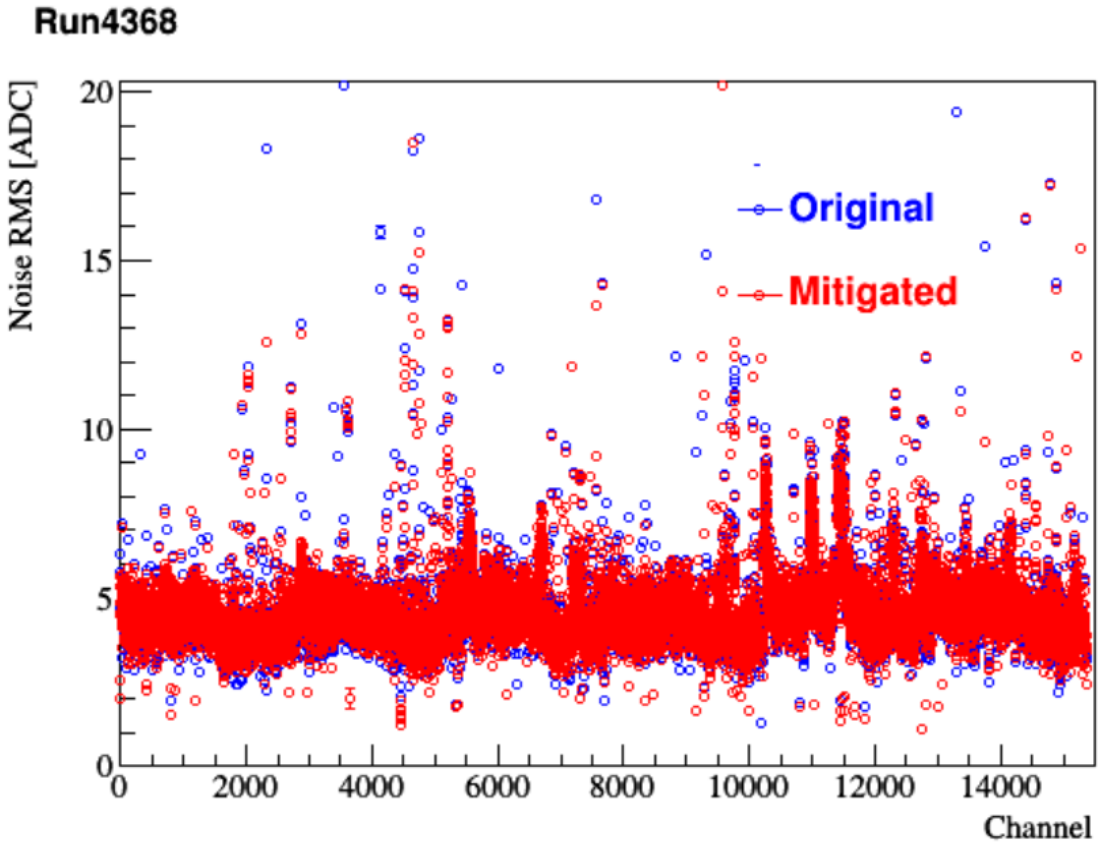
- Amplitude slightly suppressed in DFT spectrum



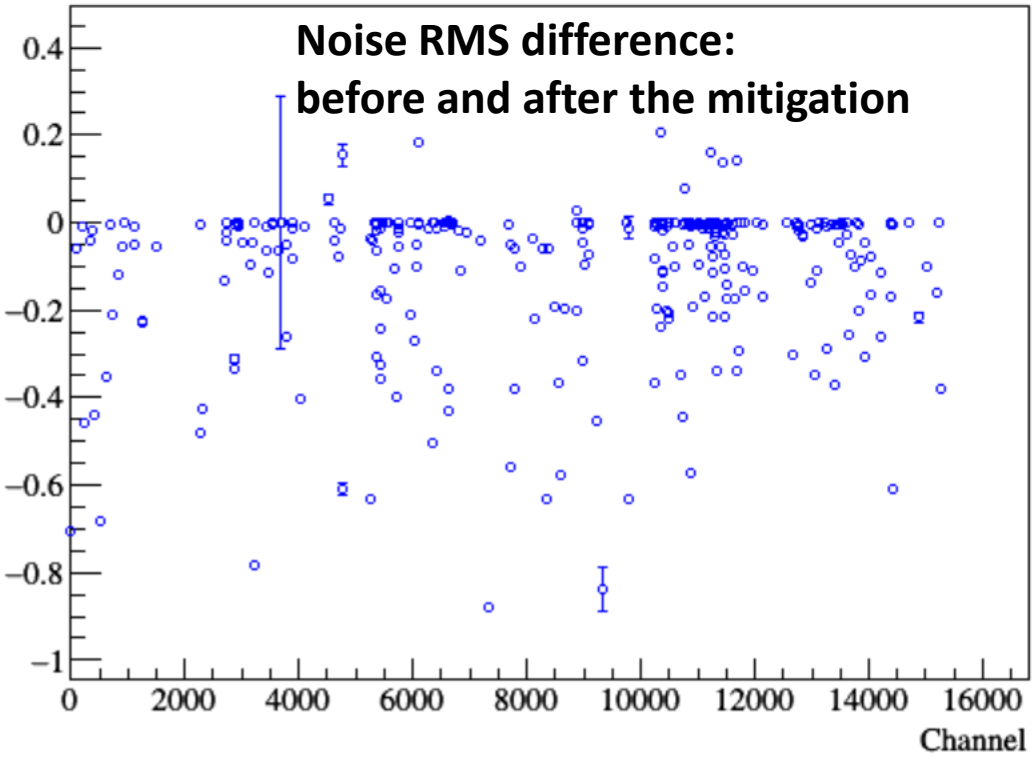




Noise RMS



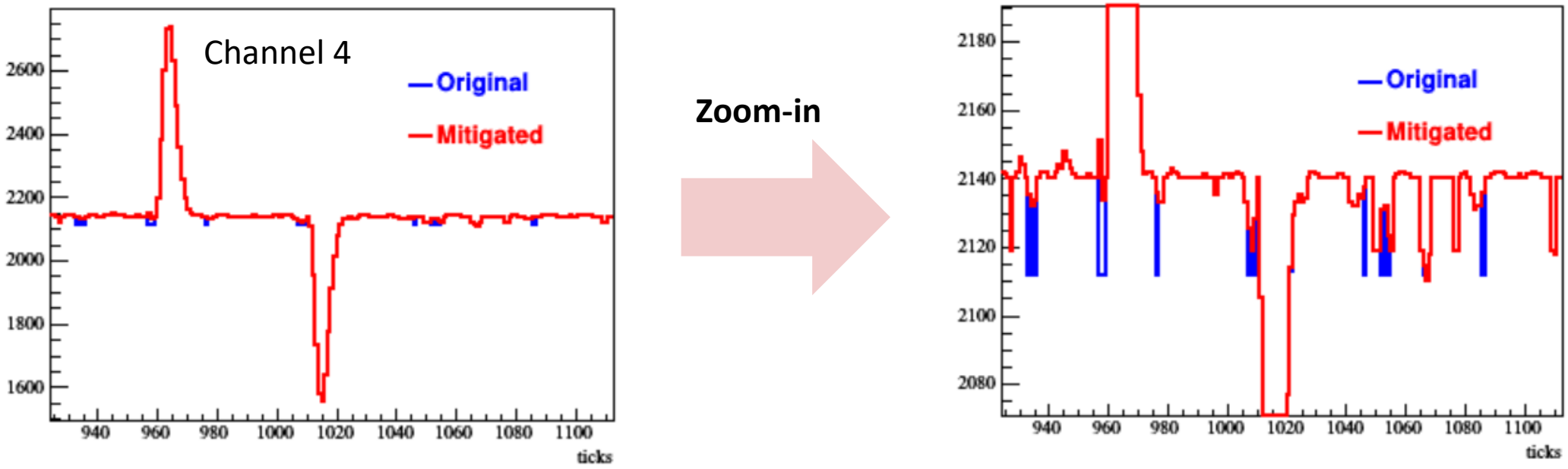
- Noise fluctuation still consistent after sticky-code mitigation
- At least does NOT bias good channels



- For some pre-selected noisy channels, most of them have slightly smaller RMS after mitigation

A Quick Look at Pulser Data

- Run3506, Event42, DAC setting =5 (Aug 21, ADC not “cold” yet)

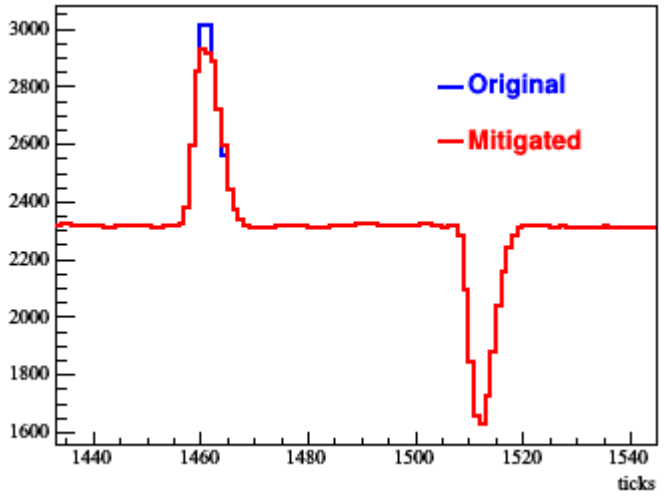


- More calibration data would be helpful since the ASIC changes after immersed in LAr

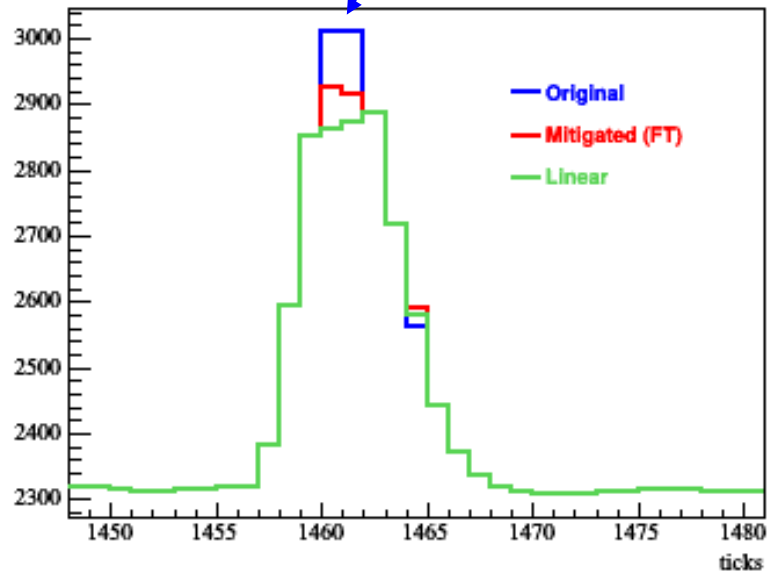
Pulser Data (cont')

Sticky at 3008 = $(101111000000)_2$

Run3506, Event42, Channel 3



Zoom-in



- However, when two adjacent sticky codes happens on the peak region, the mitigation does not work well
- Need to improve this special case
  - Mitigation can be based on original waveform, while not the linear interpolated

Summary

- Sticky code mitigation was studied with protoDUNE noise data
- A linear interpolation and a FT interpolation was applied, some special cases needs to be improved
- Most noisy channels looks better after mitigation
- The mitigation algorithm looks reliable for good channels
- Pulser data was quickly analyzed, looking forward to more “cold” pulser data