Online Trigger Primitive Generation (TPG) Algorithms

For collection and induction channels

Klaudia Wawrowska
K.Wawrowska@sussex.ac.uk

University of Sussex/ Rutherford Appleton Laboratory

12/01/2023

Trigger Primitives (TPs) at DUNE

TPs are the most fundamental part of the online data selection chain.

What are they?

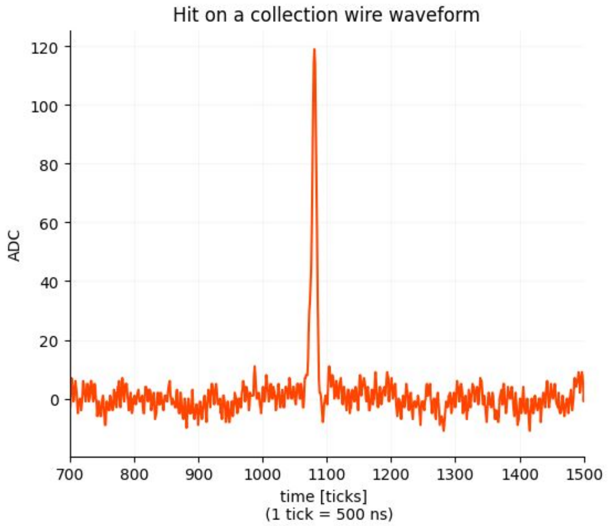
- Data structures containing information about any wire activity ("hits") which appears inconsistent with noise.

How are they generated?

- We apply a series of simple algorithms which process the incoming ADCs, search for signal spikes in the "sea of noise", and output the corresponding TP parameters.

What do we need TPs for?

- TPs are used to highlight regions of activity and filter out noise in the large volumes of the incoming raw data.
- The downsized data is fed downstream to the trigger, for more sophisticated processing (cluster & event building).

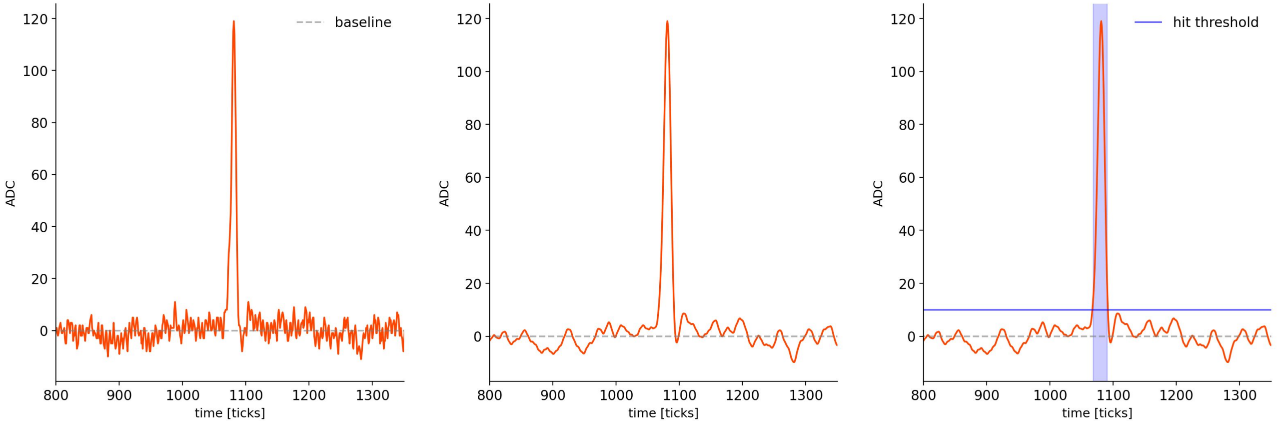



```
struct TpData
{
  tp_word_t m_end_time : 16, m_start_time : 16;
  tp_word_t m_peak_time : 16, m_peak_adc : 16;
  tp_word_t m_hit_continue : 1, m_tp_flags : 15, m_sum_adc : 16;
```

Basic TPG Algorithm Chain

A TPG algorithm chain <u>for collection wires</u> was already successfully demonstrated at ProtoDUNE [P. Rodrigues talk].

Raw ADC input → Pedestal subtraction → FIR filtering → Hit Finding → TP output



Centres the baseline of the incoming waveform at 0; ensuring that hit finding is consistent across channels.

Filters high frequency noise.

Finds ADC regions above threshold (signal), and calculates the corresponding hit parameters.

Can we improve things?

Induction TPs aren't currently used in online DAQ to form trigger decisions.
But, there's a lot to gain if we can make induction hit finding efficient enough to utilize information from all three wire planes:

1) More resilience against noise & greater signal significance.

2) TP stream [J. Klein talk]:

   - TPs could be stored offline for longer periods of time for the purpose of detector monitoring, and physics searches without the trigger bias.

3) Low energy physics gain:

   - Improved clustering efficiencies through view-matching events with low hit multiplicities

   - Region of interest (ROI) triggering [J. Klein talk]:
     - Possible to use the 3-view TP information to only read out some section of the APA
     - We could **lower triggering thresholds** without exceeding the 30 PB/yr storage limit.

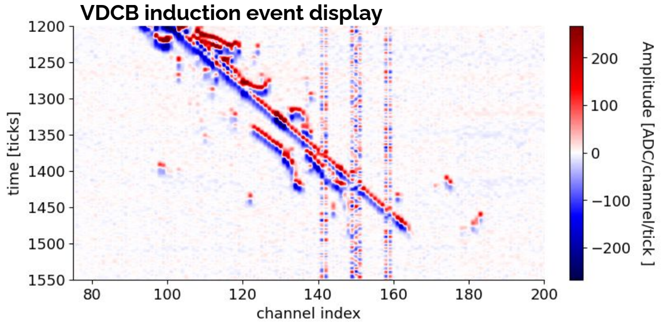
What are the challenges?

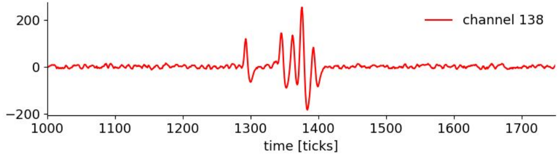
Induction signal processing is more challenging:

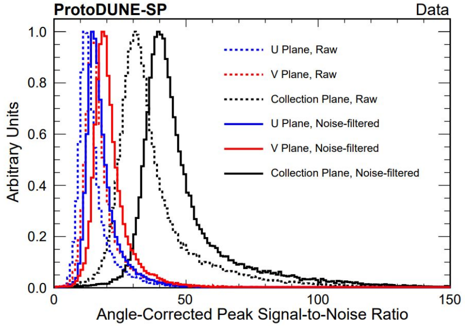
- Higher noise rms
- Lower S/N leads to reduced hit finding efficiency.
- Pulse bipolarity complicates the mapping of hits between planes & leads to ‘disappearing’ induction signals for certain track orientations.

We’re also limited with what we can do online:

- Keeping up with the 2 MHz throughput requires the use of ***simple, single-pass algorithms***.
- Any viable TPG algorithm must be ***universal*** as it’s impractical to distinguish between wire types at the TPG stage.





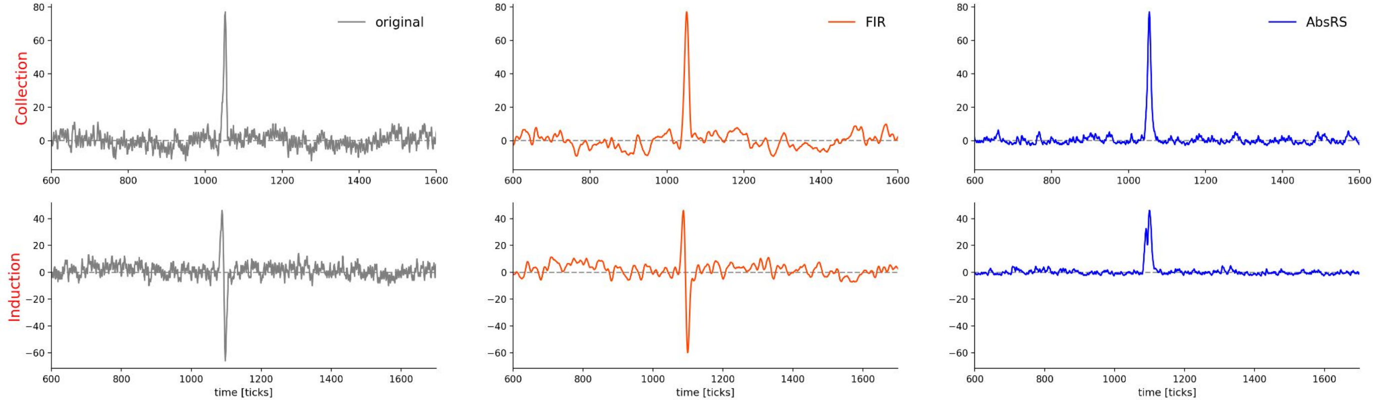


Changes to the TPG chain

Introducing changes is a balancing act between improving things and keeping the algorithms as lightweight and simple as possible.

The proposed approach is to remove the FIR and replace is with a new filter, AbsRS (or [Matched Filter]), which will allow us to process induction and collection signals with equal efficiency:

Raw ADC input → Pedestal subtraction → AbsRS filter + pedsub → Hit Finding → TP output



*filtered waveforms scaled for direct comparison

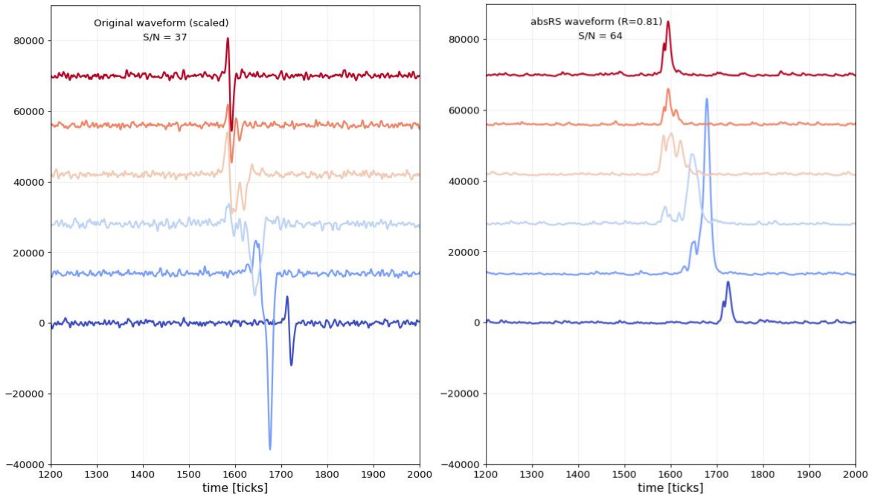
Absolute Running Sum (AbsRS) Algorithm

AbsRS is an absolute-value, weighted accumulator, which effectively integrates the incoming ADC signal in real time (DUNE-doc-27148) :

$$y_n = |x_n| + R \cdot y_{n-1}$$

$0<R<1$ is a free parameter.

- All output is strictly unipolar after taking the absolute value of the incoming ADC sample, x_n.
  - That way induction TPs begin to resemble their collection counterparts, ensuring the entire region of activity is flagged at the hit finding stage.
- Contributions from previous samples are de-weigted by R.
- The overall effect is to smooth-out noise and amplify signal, making online hit finding easier.

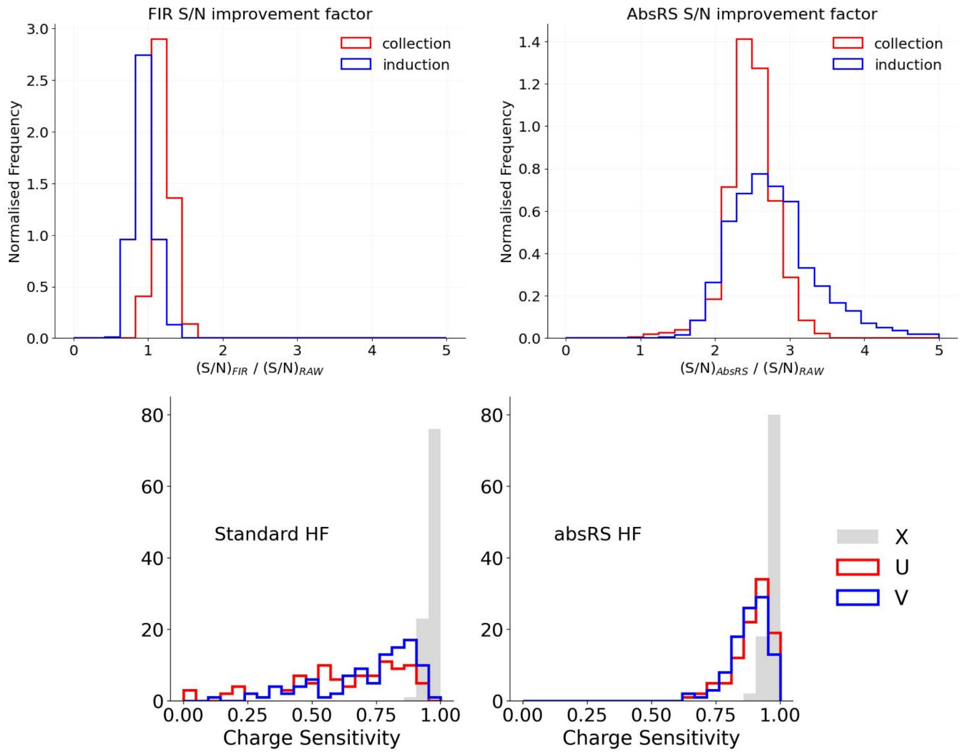


Waveforms from different induction channels before and after AbsRS.

AbsRS performance studies

Testing the new TPG algorithm with LArSoft-simulated data gives consistently promising results.

- **S/N**: observe and improvement by a factor of ~2.5 relative to raw & FIR-filtered waveforms [talk1, talk2].
- **Angular Dependence**: Improvement in S/N regardless of track orientation w.r.t. wires [talk].
- **Hit Finding**: Improvement in sensitivity at low energies without the loss of precision [talk].
- **Noise Resilience**: Better handle on harmonic noise [talk]



UNIVERSITY OF SUSSEX Particle Physics STFC Rutherford Appleton Laboratory

AbsRS vs FIR

Exchanging the FIR filter for AbsRS also makes sense from a practical point of view.

	Computational Resources	Universality	HF performance
FIR	**Gain depends on N-taps** which have to be stored in memory and iterated through for every ADC sample.	The values of the **taps need to be tuned for each plane** in order to get the optimum S/N [F. Lopez talk].	**Marginal improvement** compared to unfiltered waveforms [talk].
AbsRS	Single pass & depends on **single parameter**, R.	**R can assume the same value for all planes**; can use the exact same TPG algorithm chain for collection & induction signals.	Better noise/signal distinguishability & **improved HF capabilities** at low energies.

Both, the FIR & AbsRS, are low pass filters. Why the improvement in performance?

Signal and noise frequency spectrum at DUNE

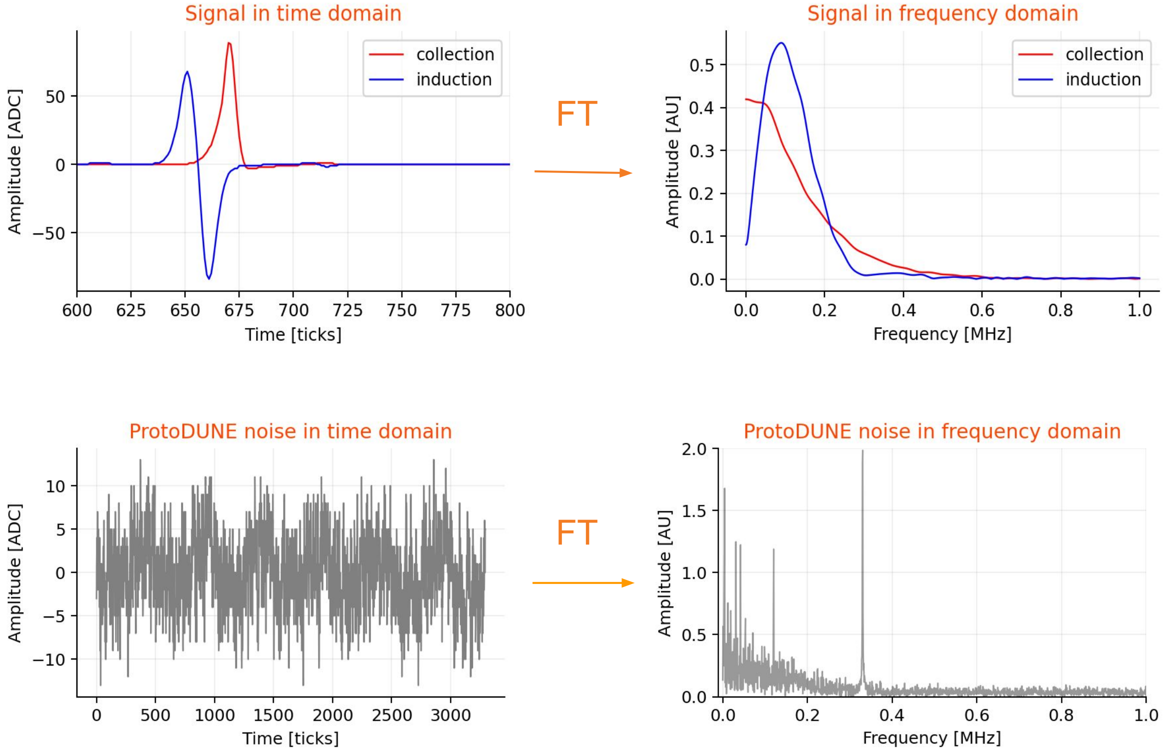
Noise and signal peaks at DUNE overlap in the frequency domain.

Signal frequency depends on electron drift time, typically < 0.3 MHz.

Electronic noise also peaks in this region; Low frequency noise is a problem as it causes baseline variations and can mimic hits.

At DUNE, the use of low pass filters amplifies low freq. noise and signal equally, causing the observed lack of HF improvement.

However, this is not true for AbsRS..



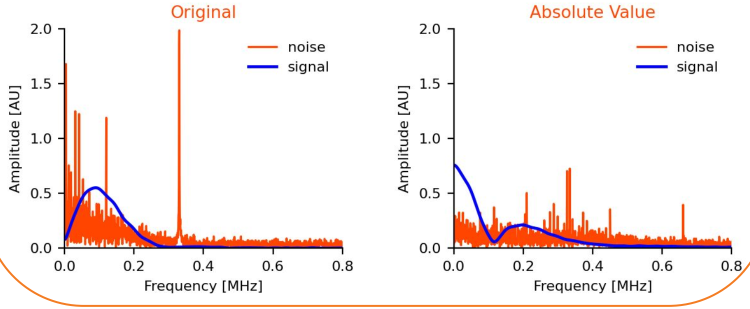
AbsRS Algorithm

Two main parts:

$$y_n = |x_n| + R \cdot y_{n-1}$$

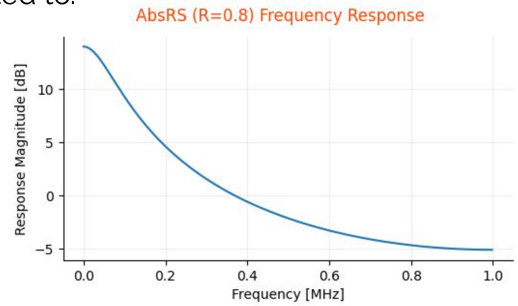
Absolute value

- Non-linear process which alters signal and noise frequency spectra, allowing for improved distinguishability in f-space (work in progress).
- Harmonic noise peaks are subdued and dispersed, while induction signals peak at lower frequencies.



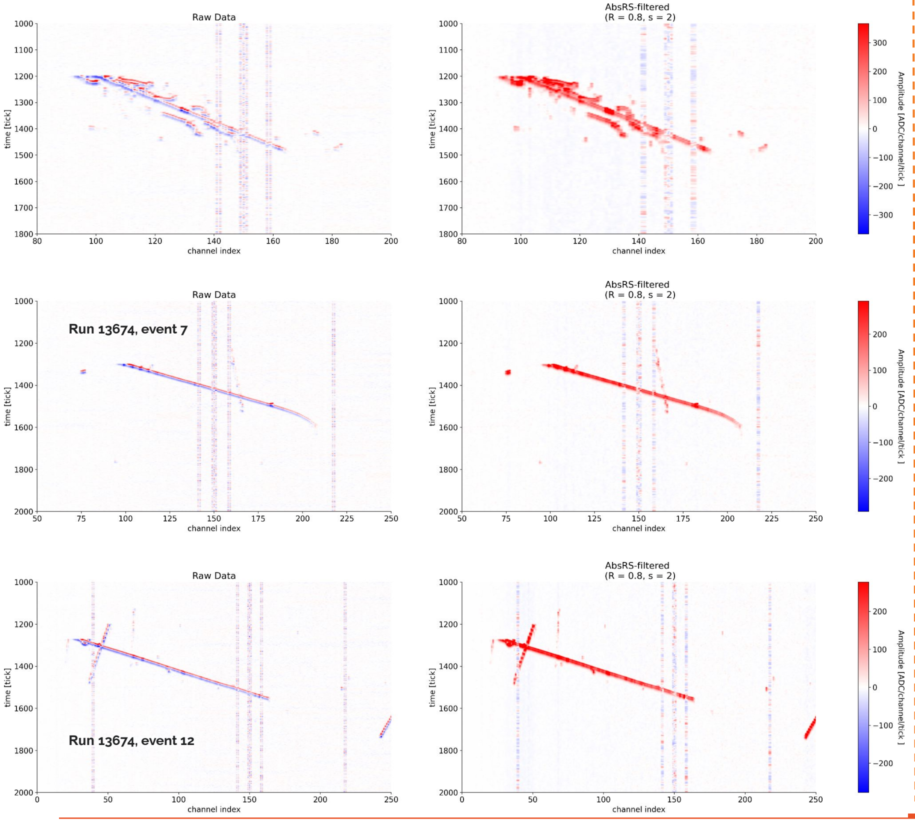
Weighted running sum

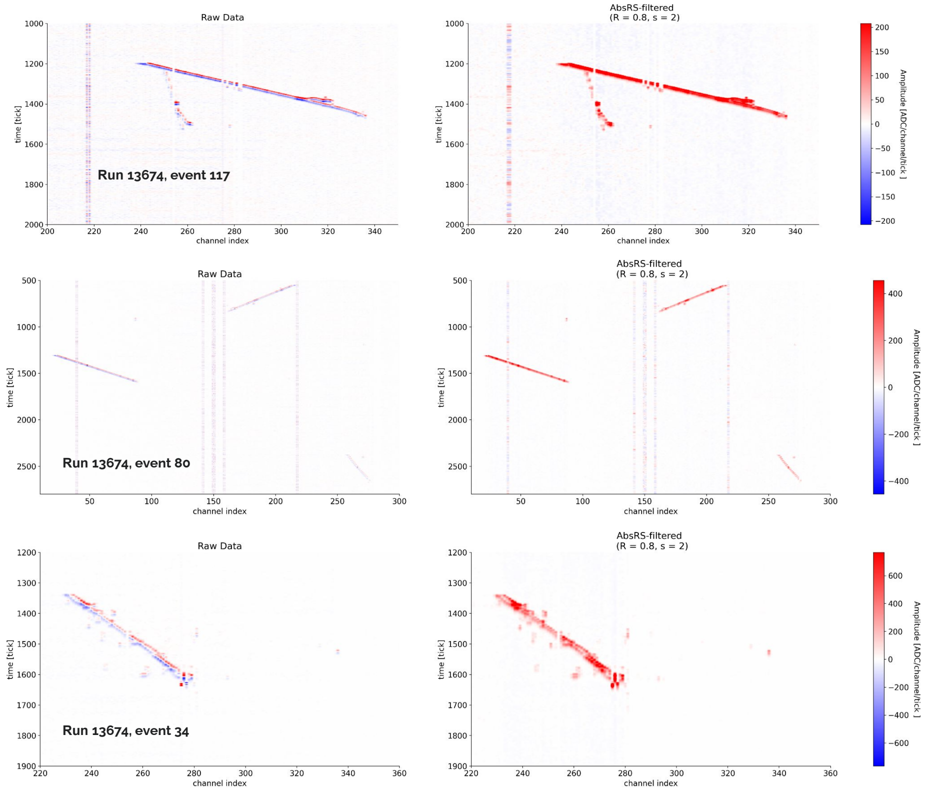
- RS is a low pass filter with steep gain <0.3 MHz.
- Highest gain for frequency space that's occupied by signal after abs. Negative gain above 0.4 MHz where a lot of the harmonic noise f-spikes get shifted to.



When used sequentially, the two components of the algorithm act in synergy to amplify signal while subduing noise, improving DUNE's online HF performance!

VDCB data filtered with AbsRS





Conclusions

- Utilising induction and collection TPs in online DAQ could significantly improve DUNE's low energy data selection capabilities.
- There are ways to improve our signal processing without straining the available resources.
- Several promising approaches have been developed (covered only one, today!).

Future work

- The AbsRS algorithm has been extensively tested in LArSoft & on offline data.
- Next step is to test it under realistic conditions with real data.
- Adam Abed Abud has developed an AVX2 software implementation of the AbsRS hit finder, and tests with a live data stream from the VDCB will begin at CERN in the next few weeks.

Thank you!

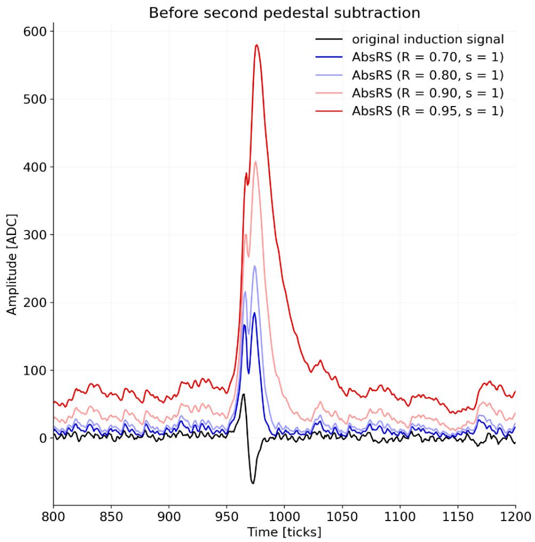
Backup slides

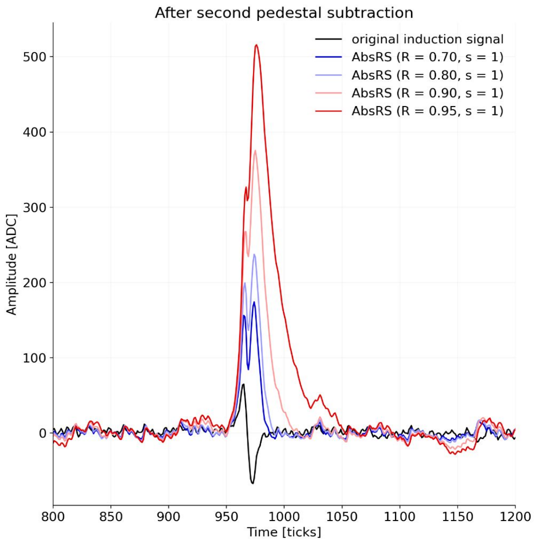
R-value

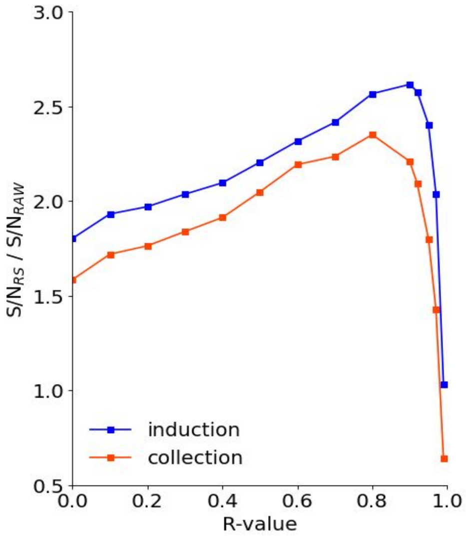
R-value acts as a weighting factor; directly impacting the shape and amplitude of the filtered waveforms, and the S/N.

The degree of signal amplification/ noise smoothing increases with R. However, as R→1 the peaks become increasingly more deformed which rapidly degrades the S/N.

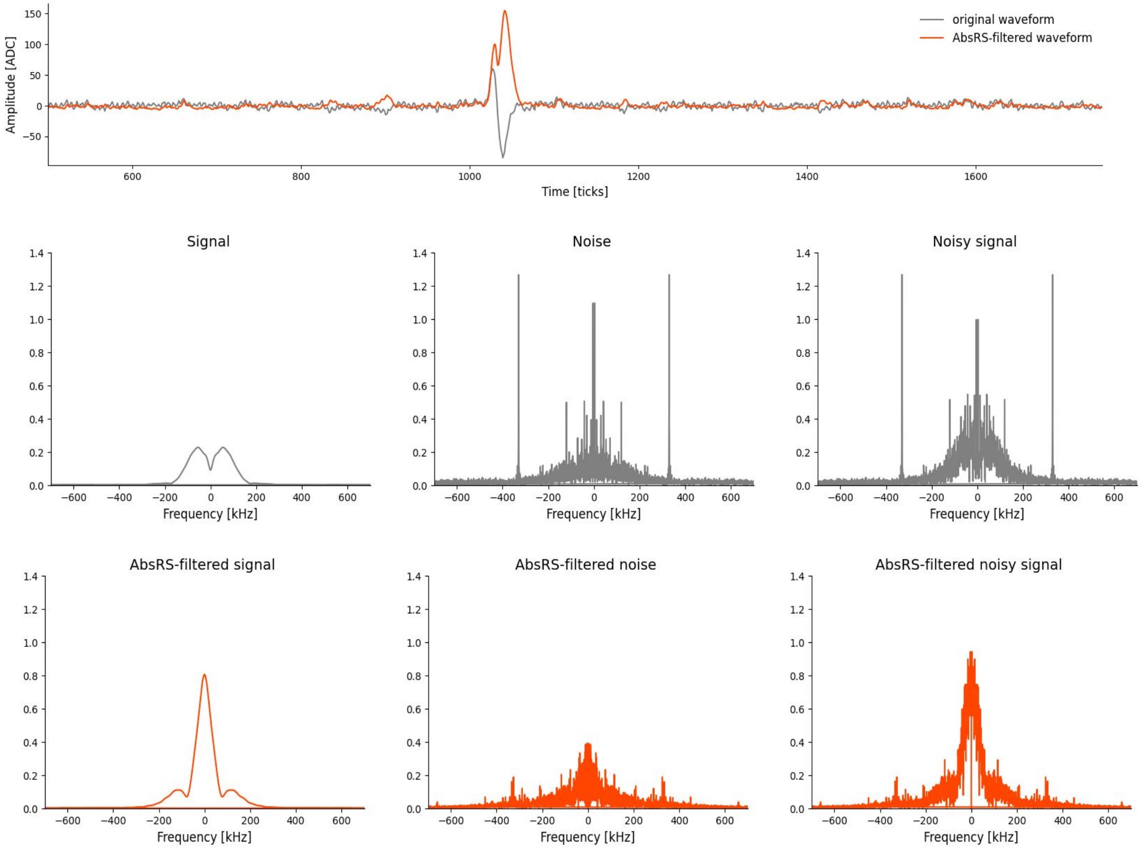
Optimum results are achieved for moderate R-values (0.7 - 0.85) for both wire types.







AbsRS FTs


original waveform
AbsRS-filtered waveform
Amplitude [ADC]
Time [ticks]
Signal
Noise
Noisy signal
AbsRS-filtered signal
AbsRS-filtered noise
AbsRS-filtered noisy signal
Frequency [kHz]