

TPG Commissioning for ProtoDUNE-HD in May 2024 at NP04

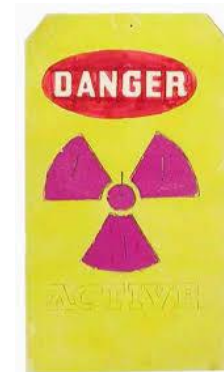
Ivana Hristova, Artur Sztuc, Michal Rigan, Alex Oranday

On behalf of the DUNE TPG/Trigger Group

03 June 2024

TPG Commissioning Goals

- Choose baseline TPG configuration for stable running with cosmics and beam
 - Available TPG algorithms
 - 1. SimpleThreshold (ST)) - no modification of the raw ADC waveform
 - 2. Standard Running Sum (StandardRS, SRS) - running sum performed on the raw ADC values
 - 3. Absolute Running Sum (AbsRS, ARS) - running sum performed on the absolute value of the raw ADC values
 - TPG algorithm configurable parameters
 - Memory factor (R)
 - Scale factor (s) - used only for AbsRS
- Look at TPG algorithm configuration
- TP rates
- TPG algorithm comparison - in progress



TPG Algorithm Configuration

- Algorithm configuration parameters are set from daqconf through the DAQ readout module to the final algorithm
- Knowledge of algorithm implementation allows us to create a comprehensive set of parameters
 - Taking into account of parameter type and resolution (due to AVX2 arithmetics used in TPG)
 - Suggest to prepare list of useful parameters based on emulation and compare to real data

AbsoluteRS vs StandardRS AVX2 Implementation

- Absolute Running Sum Algorithm

- Standard Running Sum Algorithm

**RS_adcs* used for hit finding, *raw_adcs* used for TP parameter calculation

AbsRS_adcs[]

RS = 0

R = 0.8

s = 2

for adc in pedsub_adcs:

RS = R * RS + abs(adc) / s

AbsRS_adcs.append(RS)

StdRS_adcs[]

RS = 0

R = 0.8

s = 1

for adc in pedsub_adcs:

RS = R * RS + adc

StdRS_adcs.append(RS)

algorithm configurable parameters

Both algorithms implement second pedestal subtraction on **RS_adcs*

TPG Running Sum (RS) Algorithm Configuration

- Absolute Running Sum Algorithm
- Standard Running Sum Algorithm

`readoutlibs/schema/readoutlibs/readoutconfig.jsonnet`

```
s.field("tpg_rs_memory_factor_default", self.pct, 0.8,
doc="Default memory factor (R) for
the TPG running sum algorithms."),
pct : s.number("Percent", "f4",
doc="Testing float number"),
```

1. $R = 0.8, s = 2$

```
s.field("tpg_rs_scale_factor_default", self.count, 2,
doc="Default scale factor for the TPG
running sum algorithms."),
count : s.number("Count", "i4",
doc="A count of not too many
things"),
```

TPG RS Config Propagation - Module Level

- Configuration per plane

```
fdreadoutlibs/src/wibeth/WIBEthFrameProcessor.cpp
```

$X = 0, 1, 2$ (induction, induction, collection)

1. $R = 0.8, s = 2$
 2. $R = 0.8 * 10 = 8$
 $s = 10 / 2 = 5$

```
m_tpg_rs_memory_factor_planeX = config.tpg_rs_memory_factor_planeX != -1 ?
```

```
10*config.tpg_rs_memory_factor_planeX : 10*config.tpg_rs_memory_factor_default;
```

```
m_tpg_rs_scale_factor_planeX = config.tpg_rs_scale_factor_planeX ?
```

```
10/config.tpg_rs_scale_factor_planeX : 10/config.tpg_rs_scale_factor_default;
```

TPG RS Config Propagation - Algorithm Level

- ProcessAbsRSAX2.hpp

- ProcessStandardRSAX2.hpp

`fdreadoutlibs/include/fdreadoutlibs/wibeth/tpg/`

$$RS = RS * R / 10 + \text{abs}(\text{adc}) * s / 10$$

1. $A = RS * R$
2. $B = \text{abs}(\text{adc}) * s$
3. $RS = (A + B) / 10$

INT16 arithmetics
R, s: limited precision

$$RS = RS * R / 10 + \text{adc}$$

1. $A = RS * R / 10$
2. $RS = A + \text{adc}$

1. $R = 0.8, s = 2$
2. $R = 0.8 * 10 = 8$
 $s = 10 / 2 = 5$
3. $R = 8 / 10 = 0.8$
 $s = 5 / 10 = 0.5$

supertuxkart/stk-code

#960 Impossible to
configure controls
(wired Xbox360)

9 comments

supertuxkart-sourceforge-migration opened on February 24, 2014

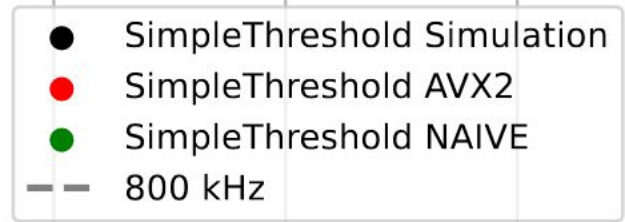
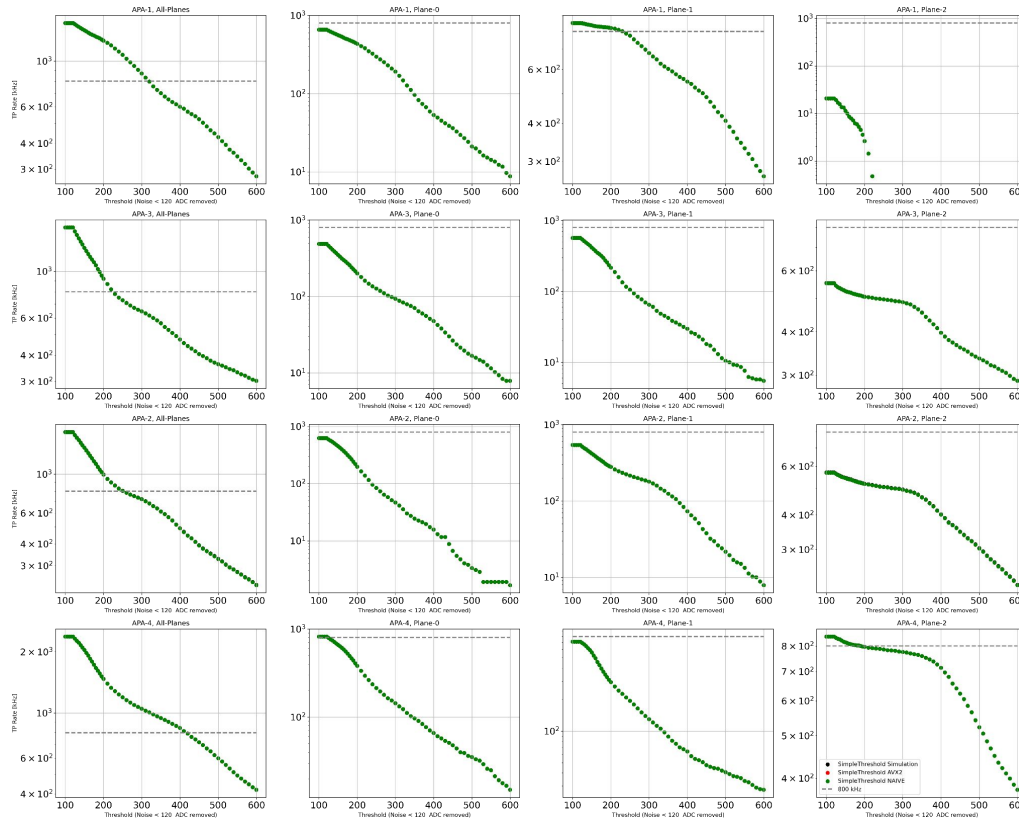


1. $R = 0.8$
2. $R = 0.8 * 10 = 8$
3. $R = 8 / 10 = 0.8$

TPG Rates

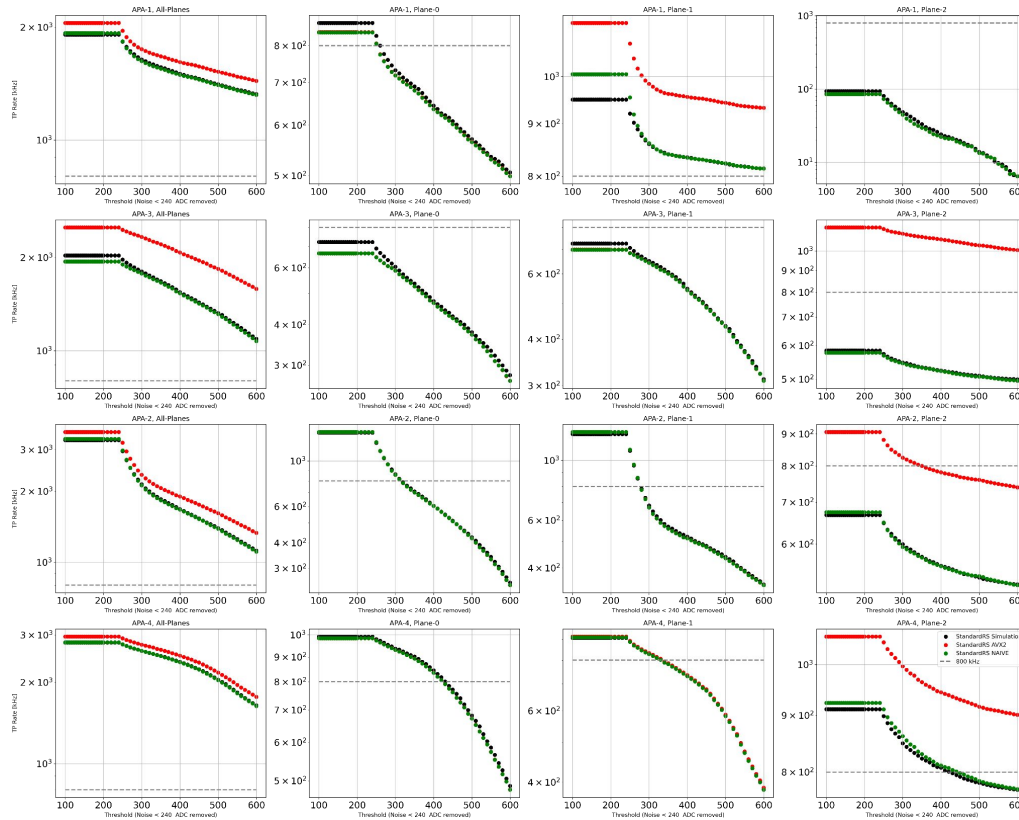
- Unexpected high TP rates of 800 kHz per plane
- Here ADC thresholds are NOT calibrated to energy
 - ADC to MeV calibration needed at some point, e.g. using Bismuth radiation source
- Next plots
 - Running Sum configurations: $R = 0.8$, $s = 1$

TPG Rates - SimpleThreshold



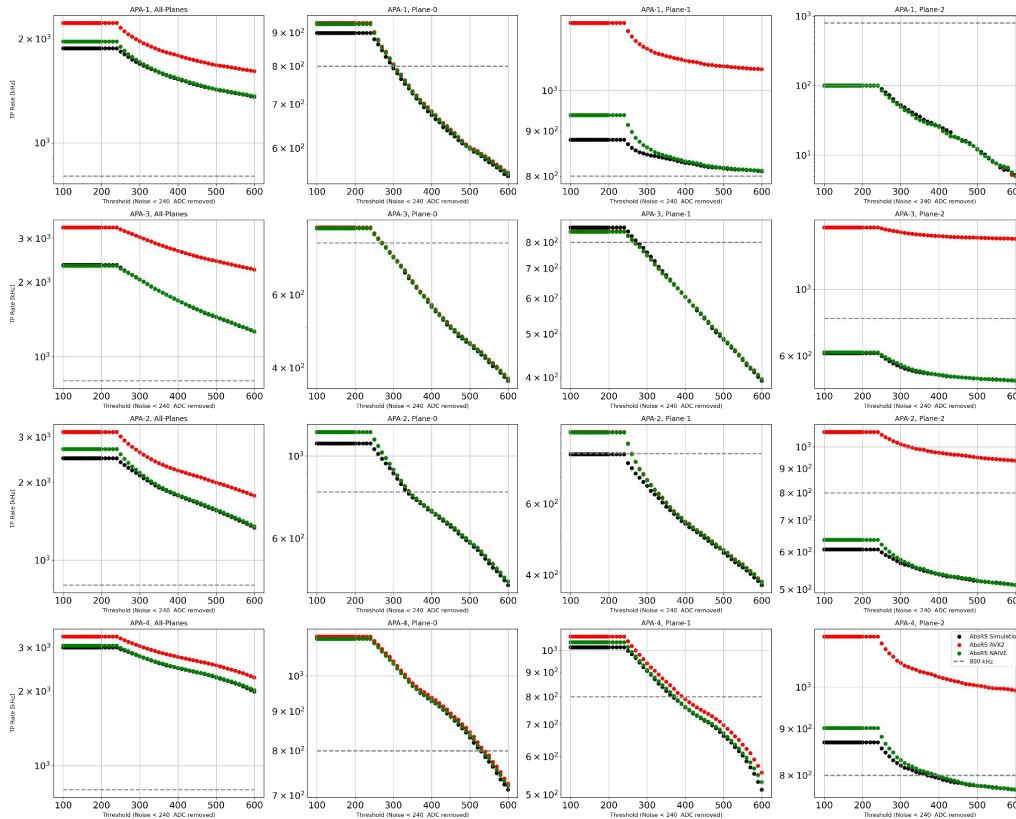
- Total, per APA, per plane
- Validation of TPG algorithm implementation
 - AVX2 = running online
 - NAIVE - offline emulation
 - Simulation
- Perfect agreement - any discrepancy due to RS algorithms

TPG Rates - StandardRS



- Good agreement between AVX2, NAIVE and Simulation for induction planes
- Discrepancy for collection plane between AVX2 and NAIVE/Simulation - not understood now

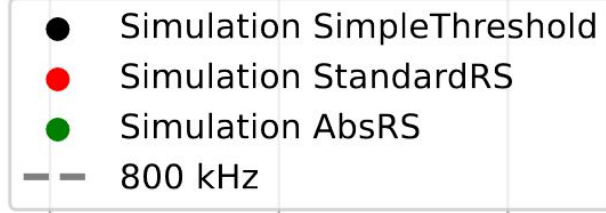
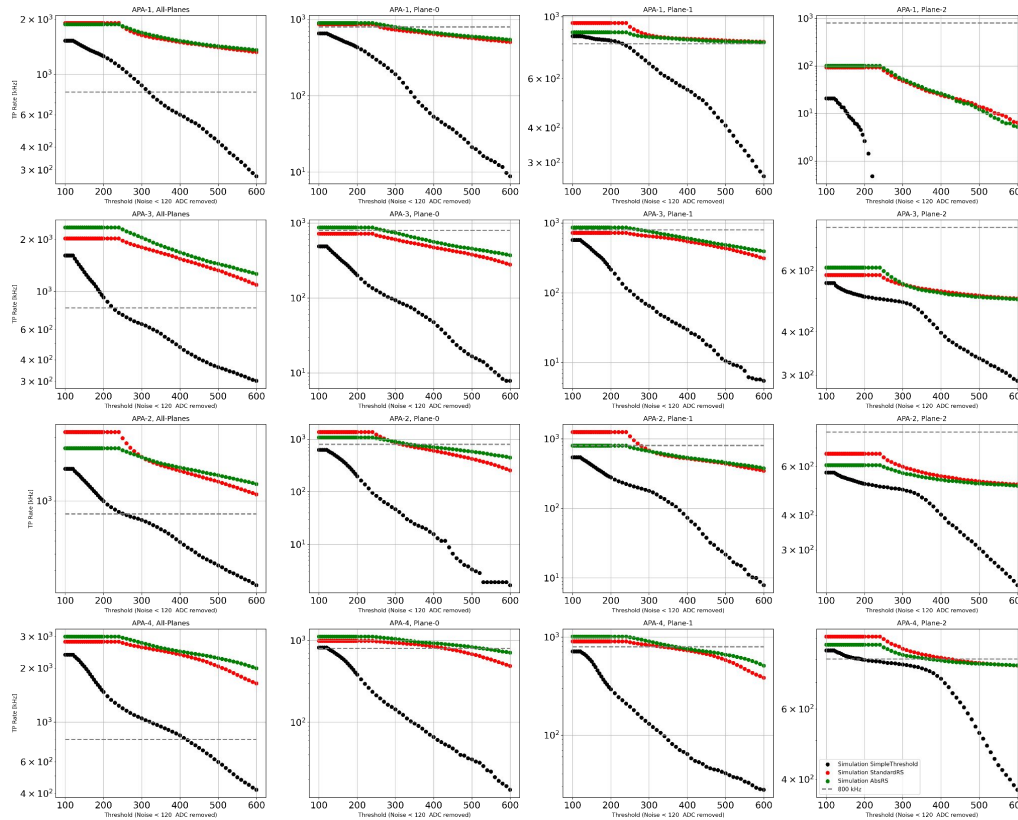
TPG Rates - AbsRS



- AbsRS Simulation
- AbsRS AVX2
- AbsRS NAIVE
- 800 kHz

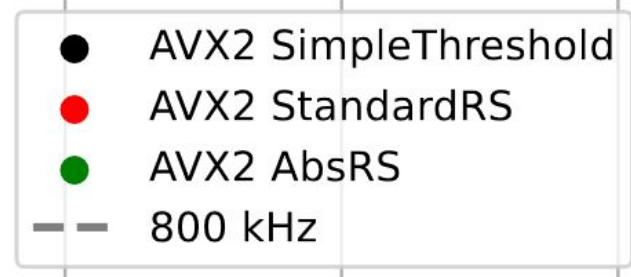
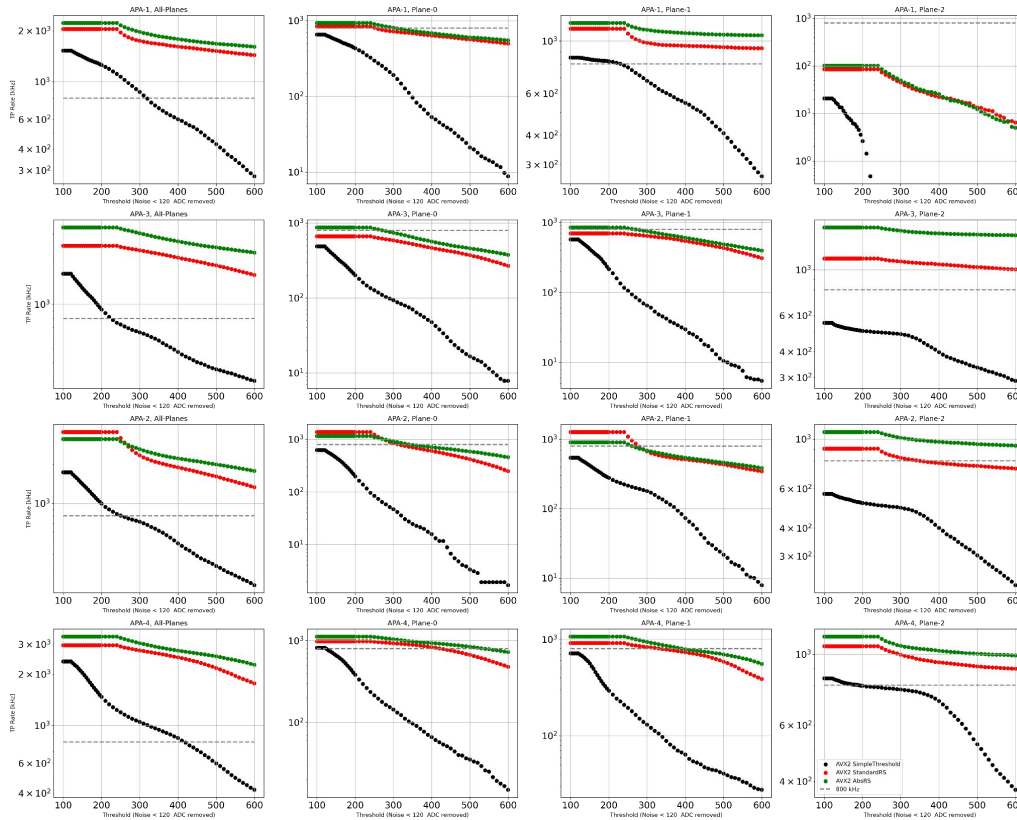
- Good agreement between AVX2, NAIVE and Simulation for induction planes
- Discrepancy for collection between AVX2 and NAIVE/Simulation - not understood now
 - Both StandardRS and AbsRS

TPG Rates - Simulation



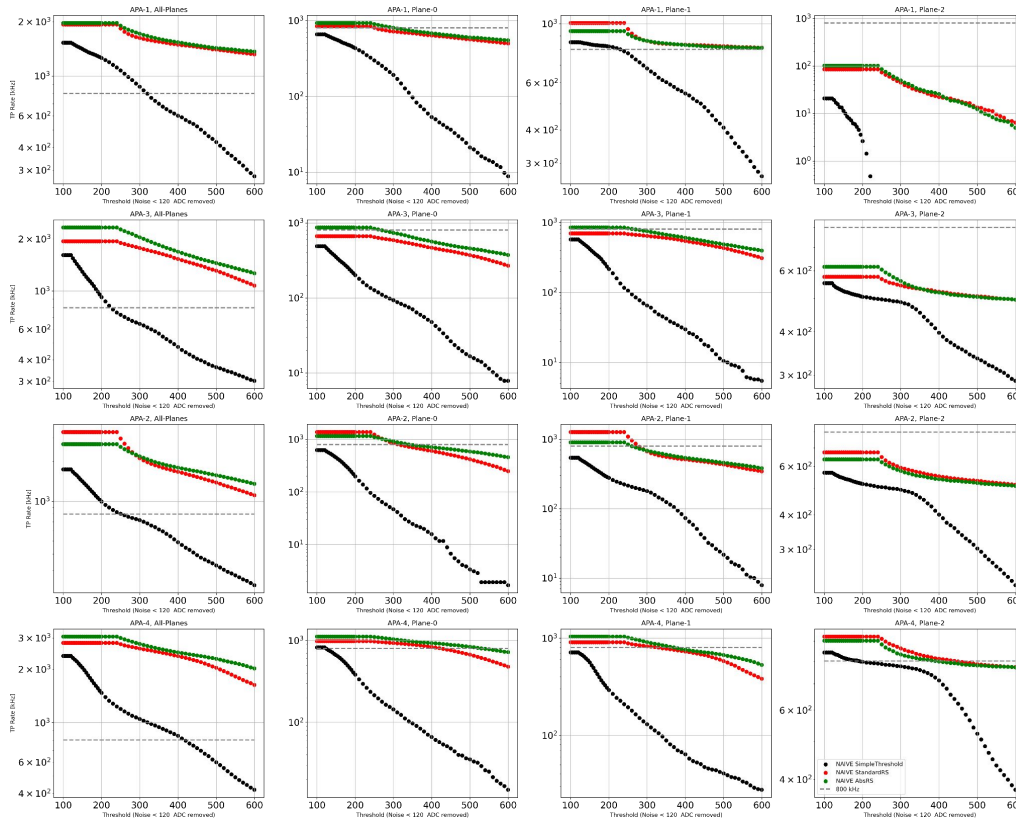
- Good agreement between StandardRS and AbsRS
- There is no evidence that StandardRS and AbsRS would behave significantly different on induction planes

TPG Rates - AVX2



- Discrepancy for collection plane between StandardRS and AbsRS
 - To be investigated

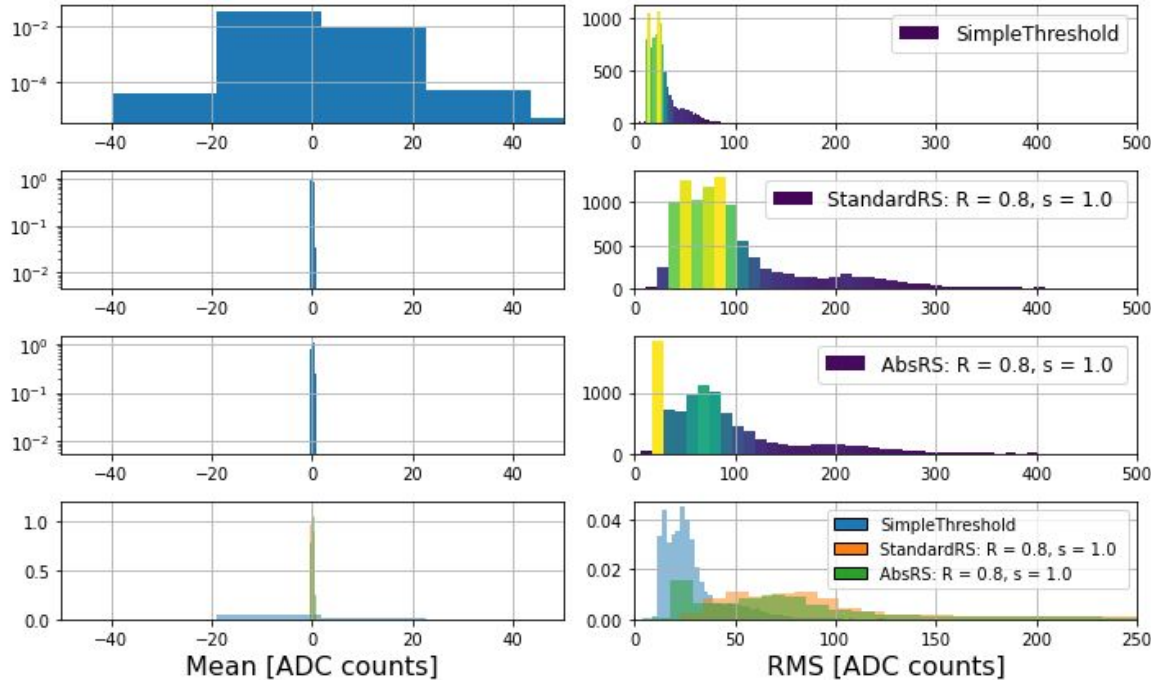
TPG Rates - NAIVE (AVX2 EMU)



- Better agreement with Simulation than with AVX2
- Some feature of AVX2 algorithm implementation not understood/correctly emulated

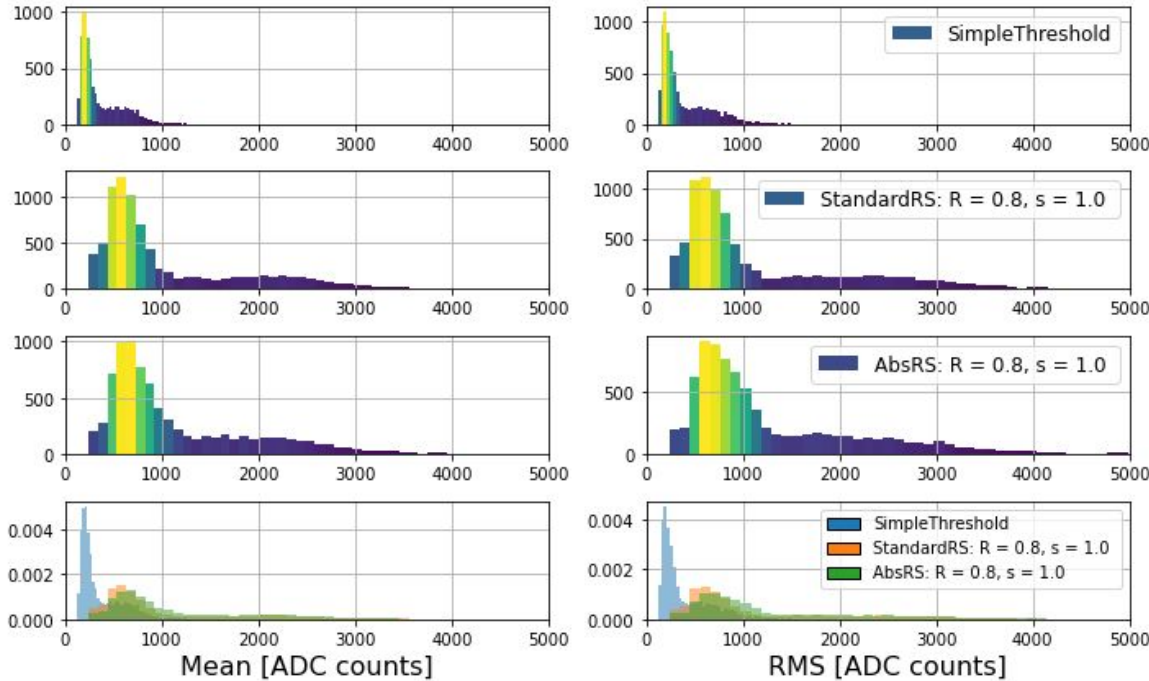
TPG Algorithm Comparison

TPG Comparison - Raw ADC Mean and RMS



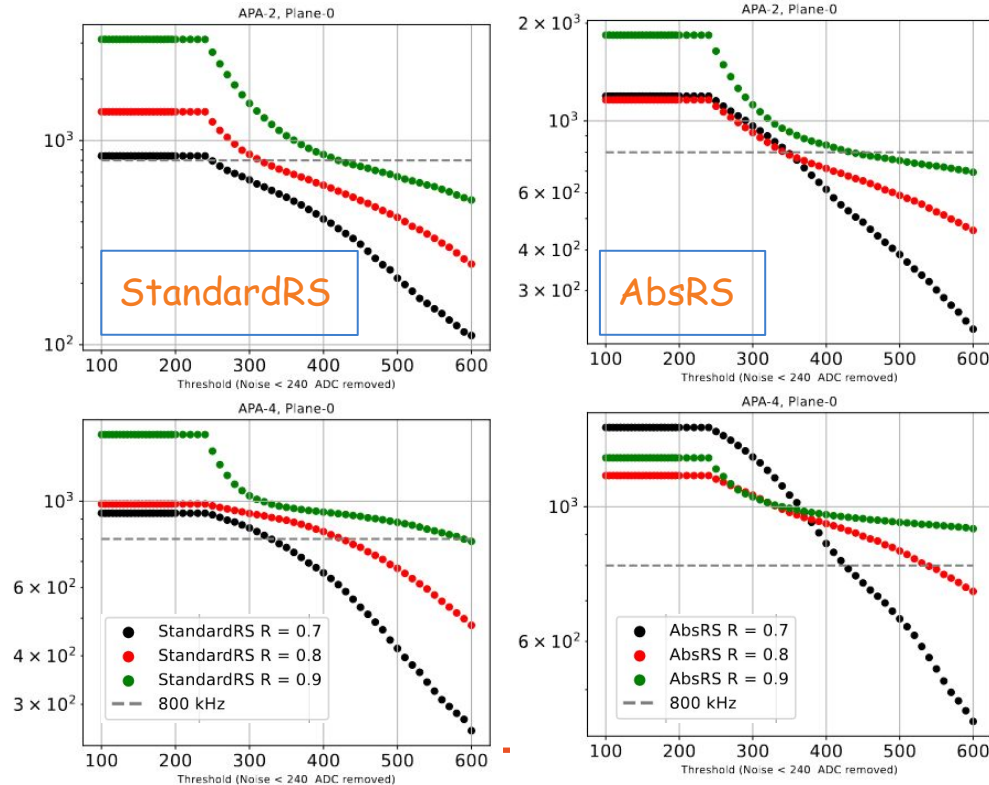
- Compared to SimpleThreshold, any RS amplifies ADC signal
 - RS mean too narrow - to be checked (2-nd pedestal subtraction)
- No major difference between StandardRS and AbsRS
 - AbsRS appears to have narrower RMS
 - Per plane - to check

TPG Comparison - TP Peak ADC Mean and RMS



- No major difference between StandardRS and AbsRS
 - Per plane - to check

TPG RS Algorithms Parameters Scan - R



● R = 0.7

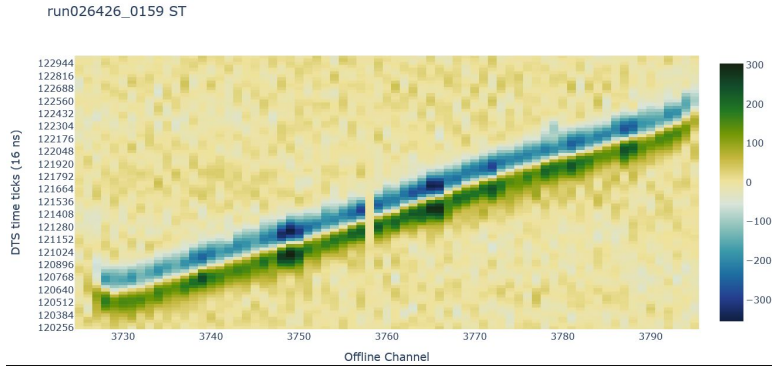
● R = 0.8

● R = 0.9

- RS parameters (R, s) give additional handle on the TP rates
 - In addition to the algorithm choice and ADC threshold
 - Systematic study ongoing
- E.g. for fixed threshold (300 ADCs) StandardRS has somewhat stronger dependence on R than AbsRS

RS Algorithm on Induction Planes (Bipolar Signals)

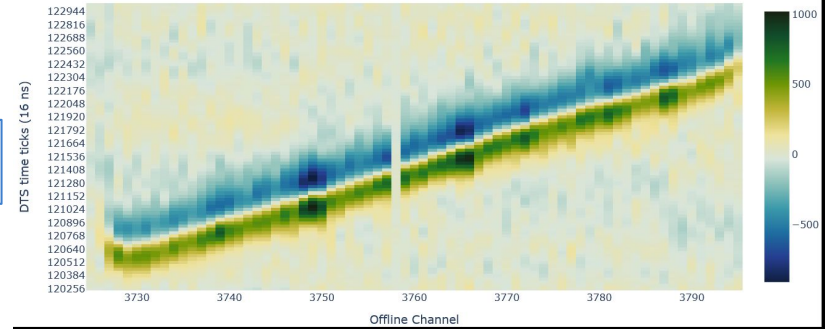
StandardRS



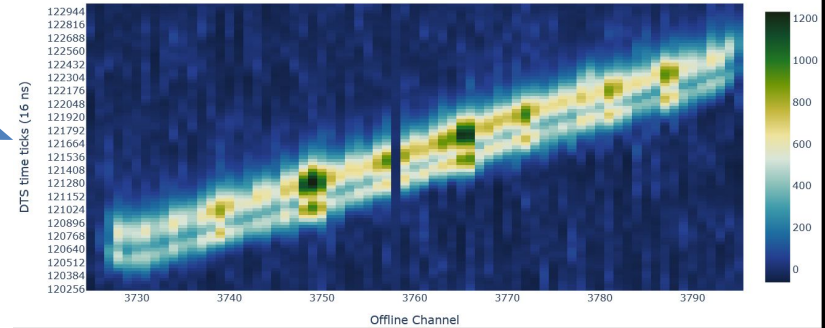
SimpleThreshold

AbsRS

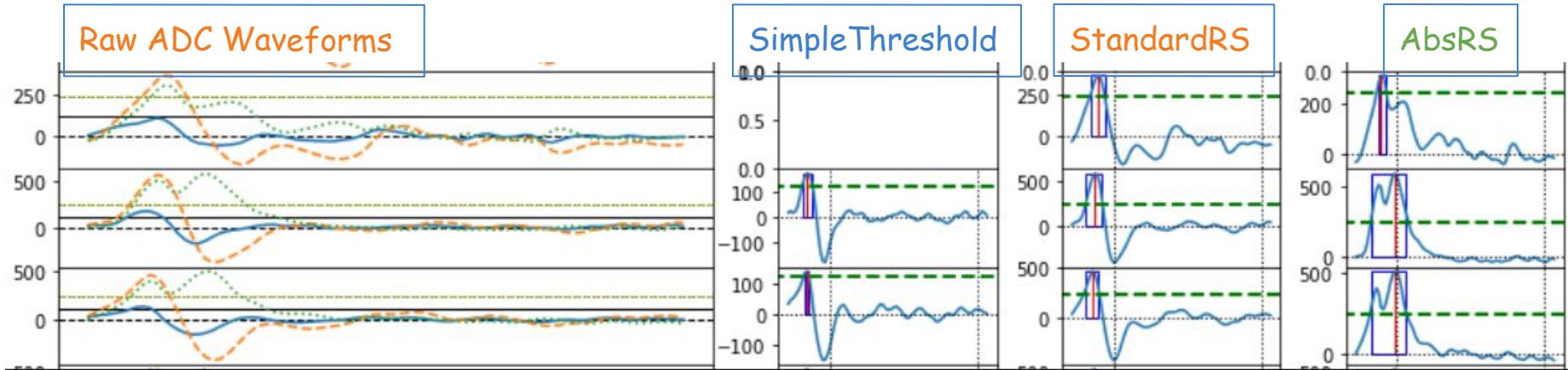
run026426_0159SRS



run026426_0159ARS

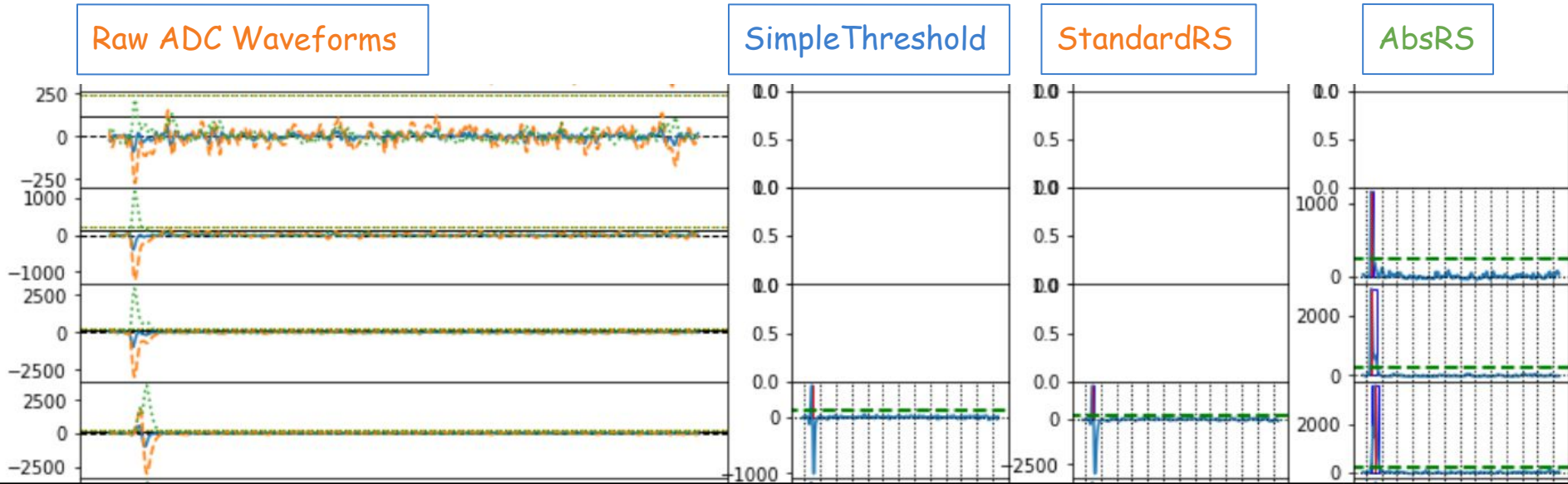


TPG RS Comparison - Waveforms and TPs



- Features of the TPG algorithms
 - Hit finding works correctly across WIB frames - TP parameters match ADC waveform
 - Latest change to TPG RS algorithms extracts TP parameters from RS waveforms
 - RS behaviour seems to be very stable and robust
 - Hit parameter extraction is predictable and under control in most cases
 - RS waferom transformation causes double (multiple) TPs to be created - investigation

TPG RS Comparison - StandardRS vs AbsRS Case



- Special case of asymmetric bipolar pulses on induction wires
 - StandardRS amplifies the negative peak (in the lack of or very small positive peak)
 - Only AbsRS capable of recovering such signals (in case that's desired)

Summary & Conclusion

- TPG commissioning overall successful
- High rate of TPs being found/made along muon tracks and blips from Ar39 and other sources
 - TPG RS algorithms now use the RS (not raw) waveform to compute TP parameters
- Stable baseline configuration found - SimpleThreshold on collection plane, StandardRS on induction planes, with optimal ADC thresholds per plane
- Ongoing systematic checks to better understand behaviour of RS algorithms
 - Improve TPG algorithm emulation/simulation in C++ to match AVX2 arithmetics and the other way around if possible
 -
- Investigation whether the current baseline TPG configuration can be improved, e.g.
 - Run AbsRS on induction planes