Calorimeter Activities in Mainz

H. Ahmadi, S. Ahmed, A. Aycock, L. Capozza, A. Dbeyssi,
P. Grasemann, S. Haasler, D. Izard, J. Jorge Rico, J. Köhler, F. Maas,
O. Noll, D. Rodríguez, S. Wolff

PANDA LIX Collaboration Meeting

6. December 2016

1 News

2 Feature Extraction

3 Baseline Adjustment

News

News

Uppsala ADC

- Functioning with TRB (Beamtime 5.16) ✓
- Got it running with Bonn firmware (Talk: J. Müllers) ✓

Gießen HV-Distributor

- PCB is working (readout with PI) ✓
- Waiting for MCX-connectors (We have MMCX)

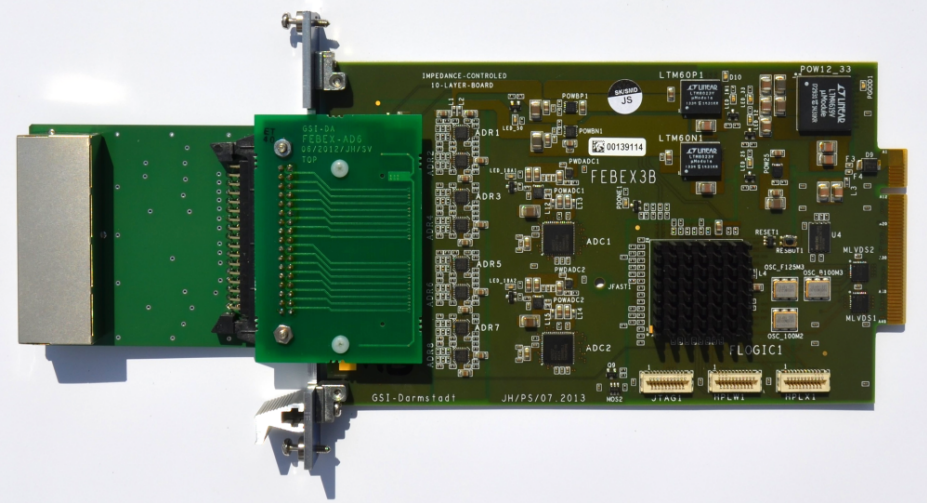
Taking Responsibility

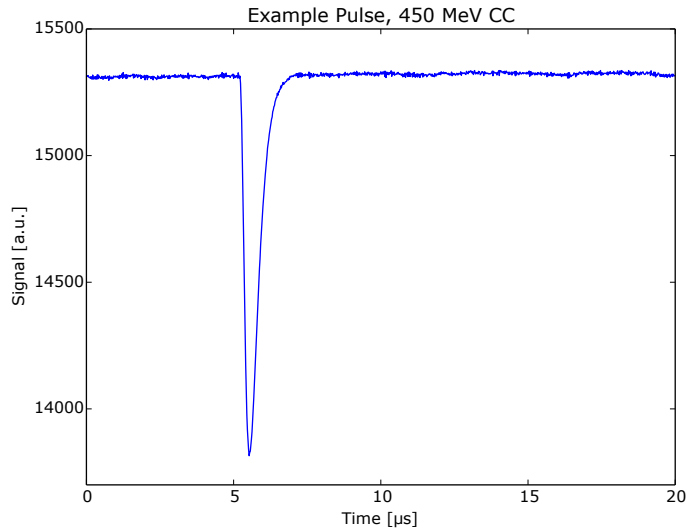
- Mainz will develop the feature extraction for ASIC pulses
- Mainz will develop the baseline adjustment for the ASIC

Feature Extraction

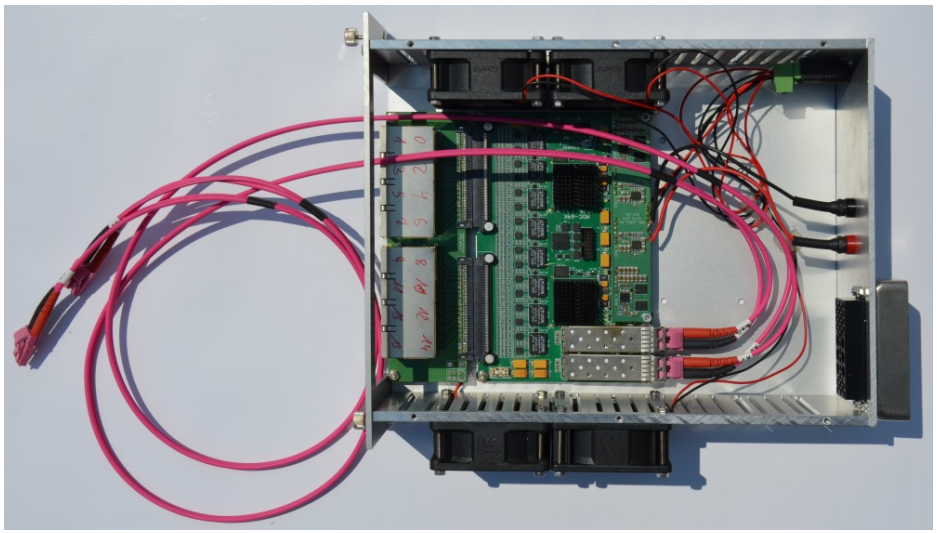
Mainz ADC Collection

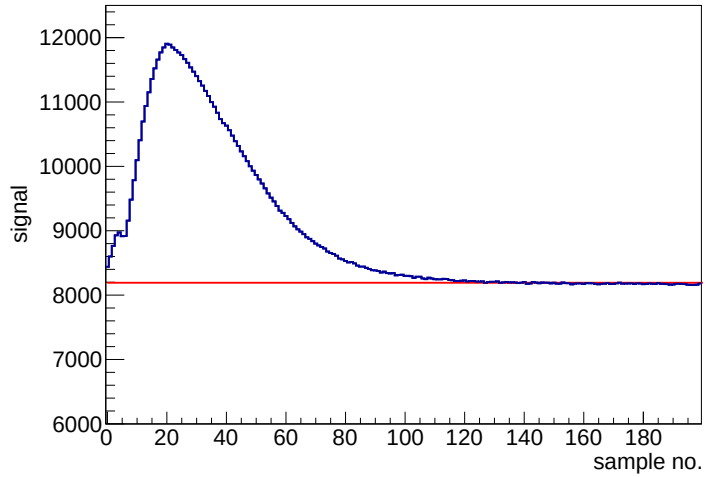
FEBEX 14 bit, 50 MSa/s, 16 diff. input channels





Uppsala ADC 14 bit, 80 MSa/s, 64 diff. input channels





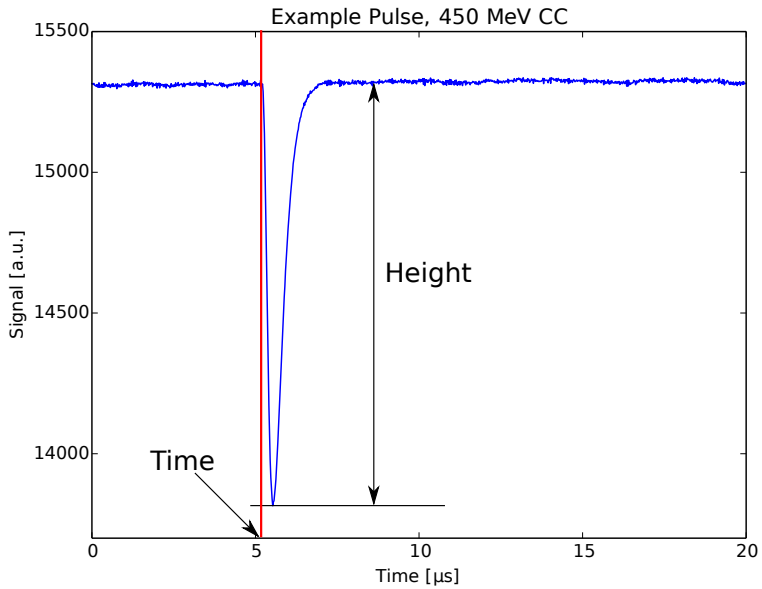
Feature Extraction

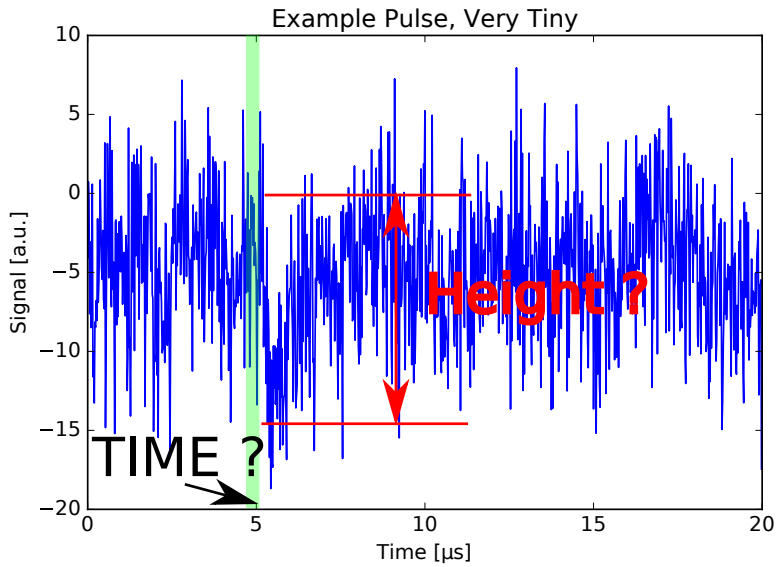
Needed Information

- Time
- Energy (pulse height)

Requirements on FE

- Fast (calculation time)
- Sensitive to ASIC pulse shape
- Linear
- Threshold as low as possible (2-3 MeV)
- Dead time as short as possible





Trapeze Filter

Trapeze Filter

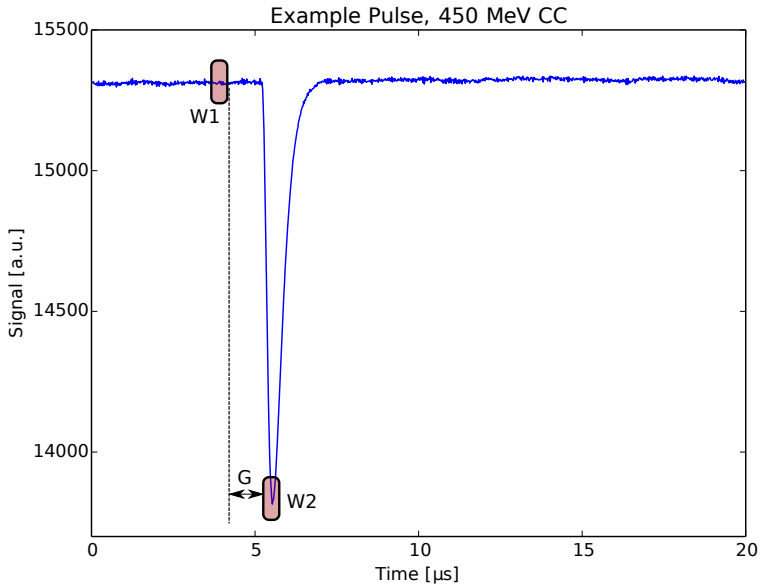
- Robust and efficient filter
- Saving calculation time
- Already implemented on FEBEX FPGA ($W_1 = W_2$)

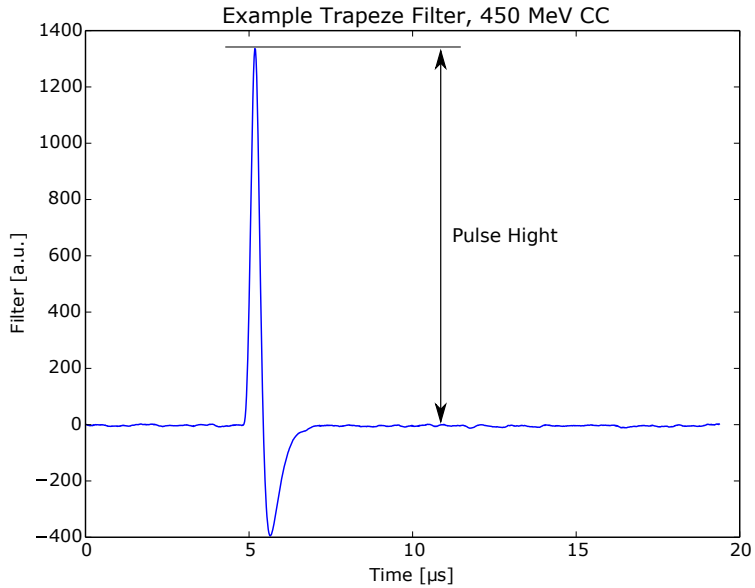
Recursive calculation:

$$S_0 = \frac{1}{W_1}\sum_{i=0}^{W_1} T[i] - \frac{1}{W_2}\sum_{i=W_1+G}^{W_1+W_2+G} T[i]$$

Filter value:

$$F_n = S_n = S_{n-1} + \frac{1}{W_1}\big(T[W_1+n] - T[n-1]\big) - \frac{1}{W_2}\big(T[W_1+G+n-1] - T[W_1+W_2+G+n]\big)$$





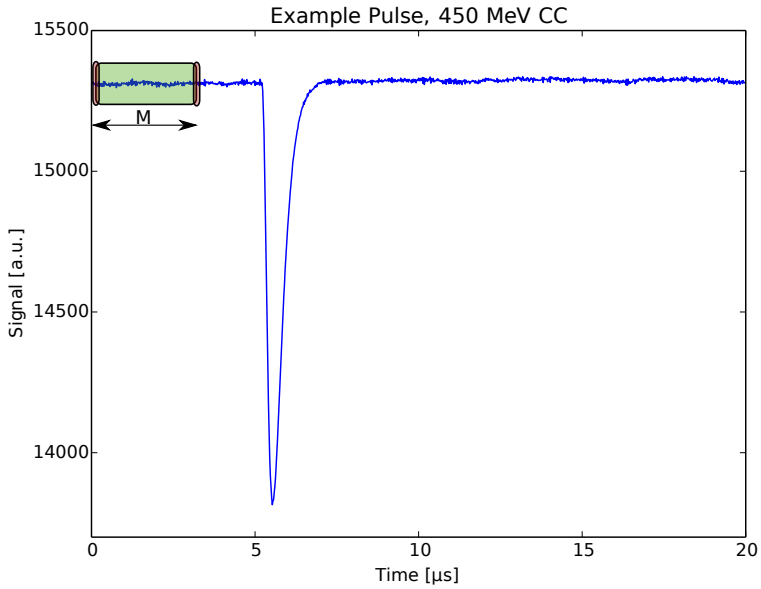
MWD Filter

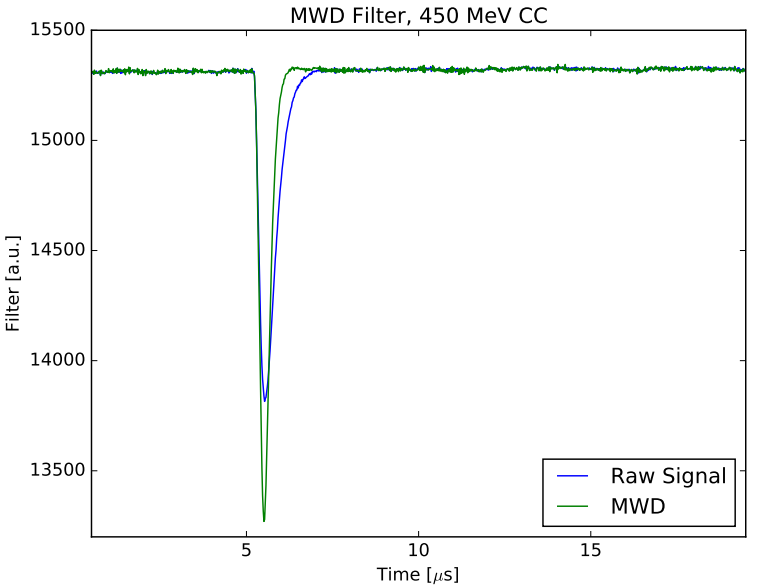
MWD Filter

- Robust and efficient filter
- Shortening of pulse shape
- Smoothing (SMA)
- No overshoot
- Baseline correction needed

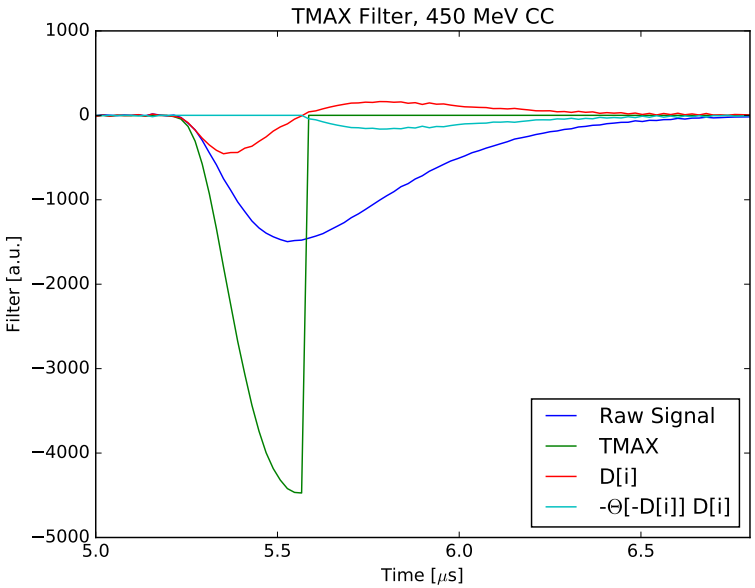
MWD:

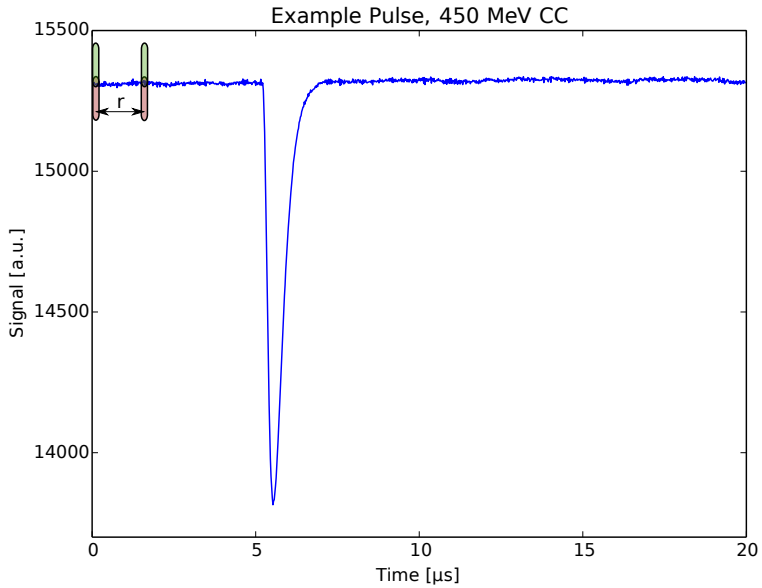
$$F_{MWD}[i] = \underbrace{T[i] - T[i-M]}_{\text{Differentiation}} + \underbrace{\frac{1}{\tau}\sum_{j=i-M}^{i-1} T[j]}_{\text{Moving Average}}$$





TMAX Filter





Derivative:

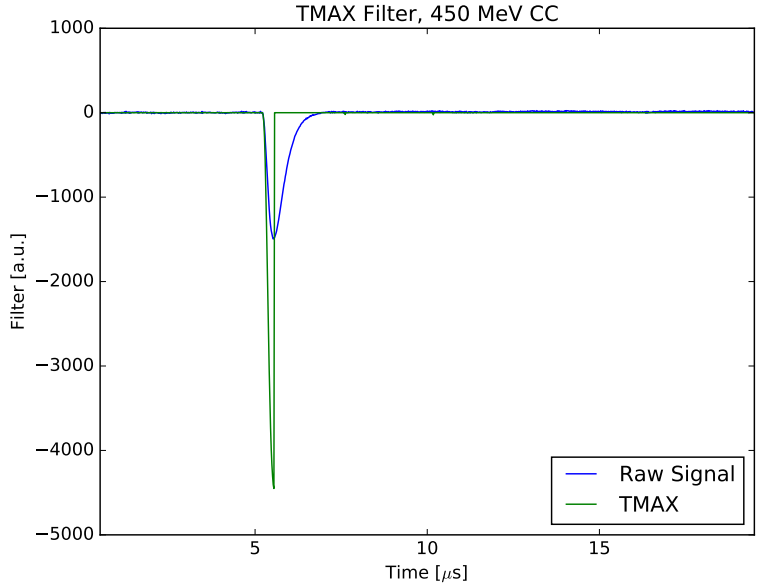
$$D[i] = T[i + r] - T[i]$$

Heaviside function Θ:

$$x \mapsto \begin{cases} 0: & x < 0 \\ 1: & x \geq 0 \end{cases}$$

TMAX:

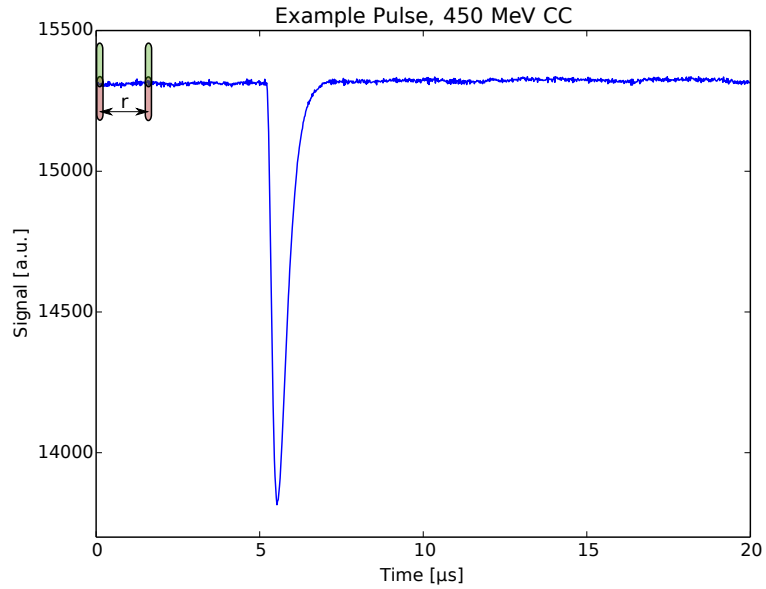
$$F_{TMAX} = \sum_{i=0}^{N} D[i] - \Theta[-D[i]] \cdot D[i]$$



TMAX Filter

TMAX Filter

- Sensitive on rising edge
- Cancels out falling edge
- No overshoot
- Short dead time



Derivative:

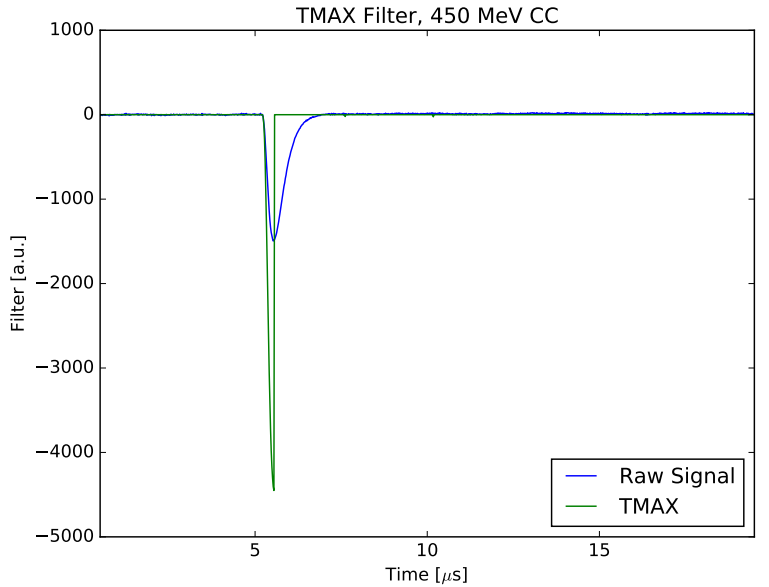
$$D[i] = T[i + r] - T[i]$$

Heaviside function Θ:

$$x \mapsto \begin{cases} 0: & x < 0 \\ 1: & x \geq 0 \end{cases}$$

TMAX:

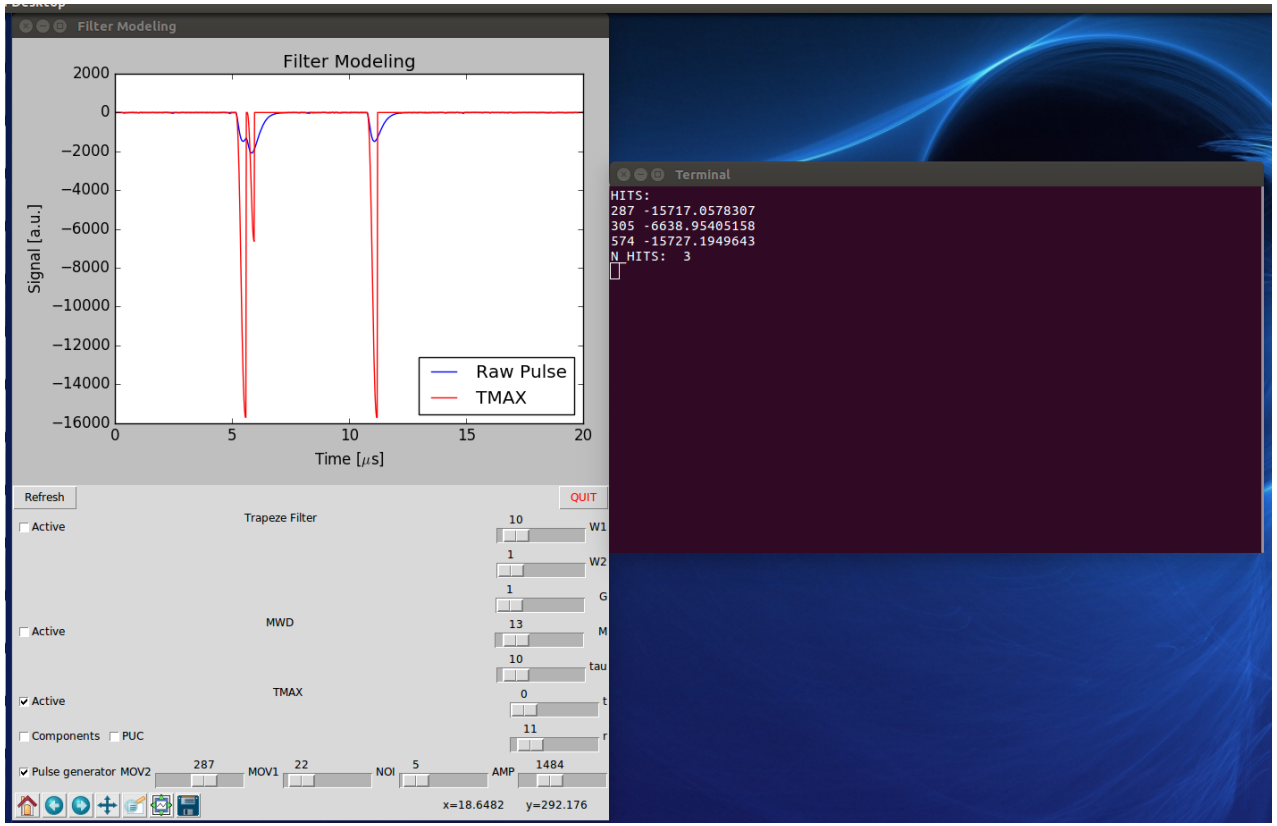
$$F_{TMAX} = \sum_{i=0}^{N} D[i] - \Theta[-D[i]] \cdot D[i]$$



Filter Performance Tests

- Filter Modeling (Parameters)
- Filter Tests with Event Generator
- Filter Tests with Detector Data

Filter Modeling Software



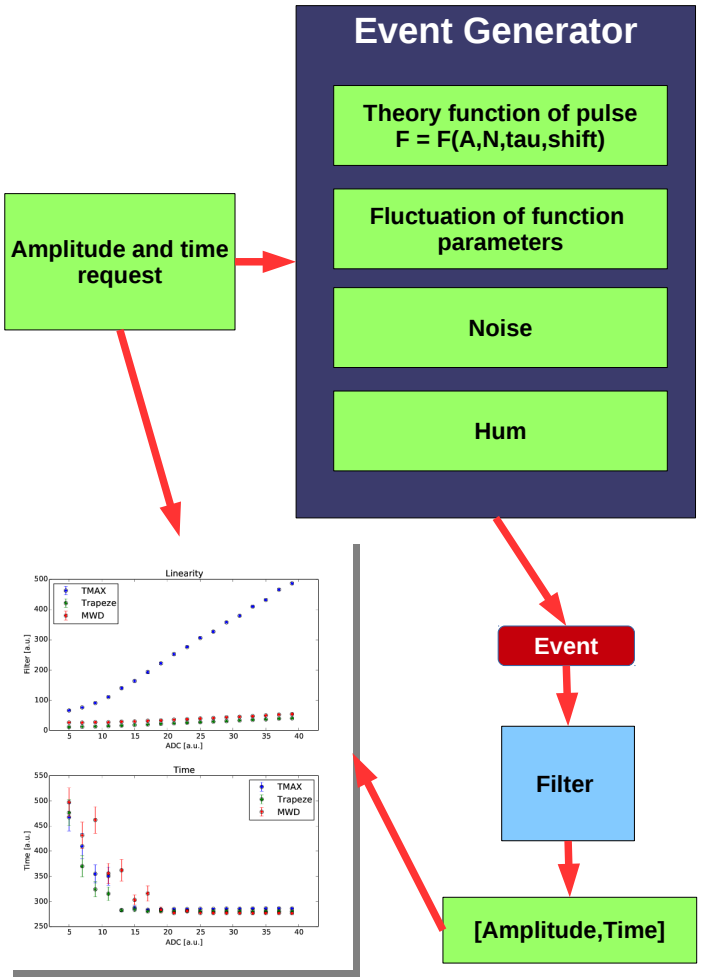
Get a Quick Idea

- See the influence of parameters directly
- Change pulse amplitude and noise
- Simulate pile up
- Find promising parameters for analysis

Filter Tests with Event Generator

Event Generator

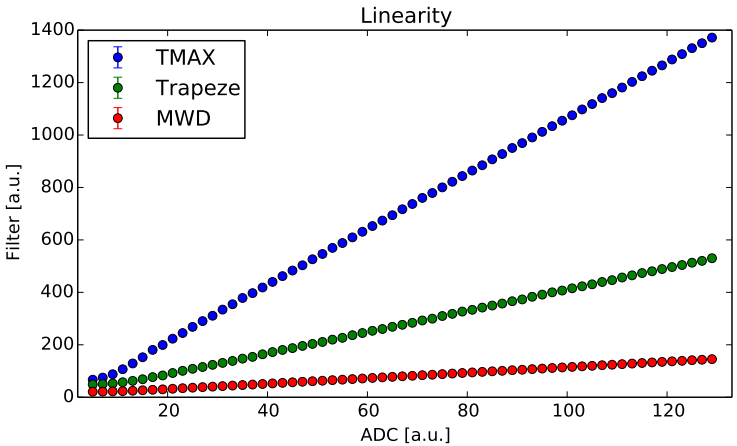
- Generate signals:
  - Amplitude
  - Time
  - Parameter fluctuation
  - Noise
- Verify:
  - Linearity
  - Efficiency
  - Lowest threshold

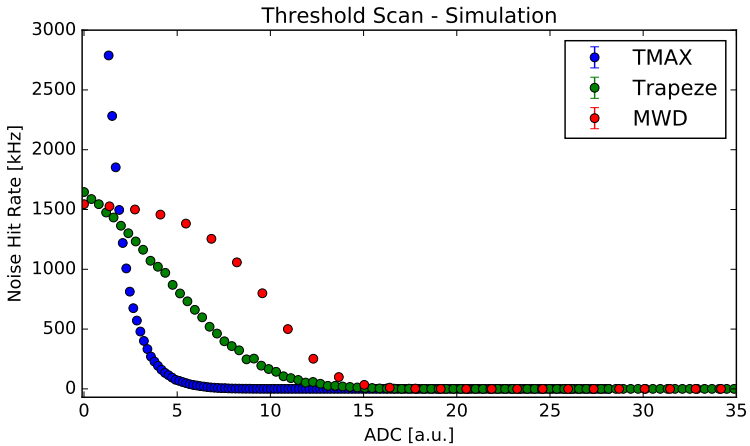


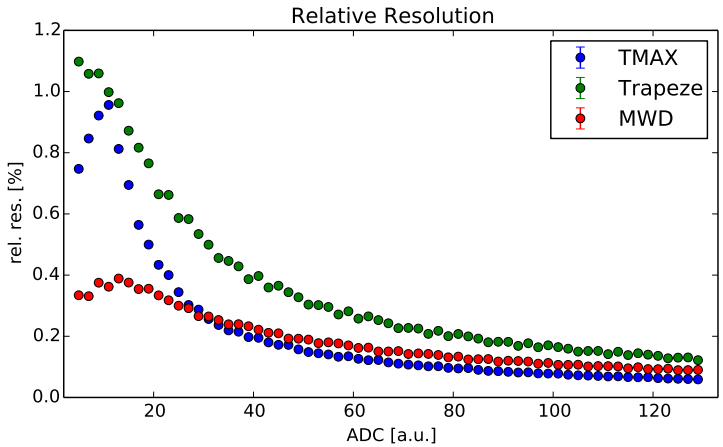
Noise

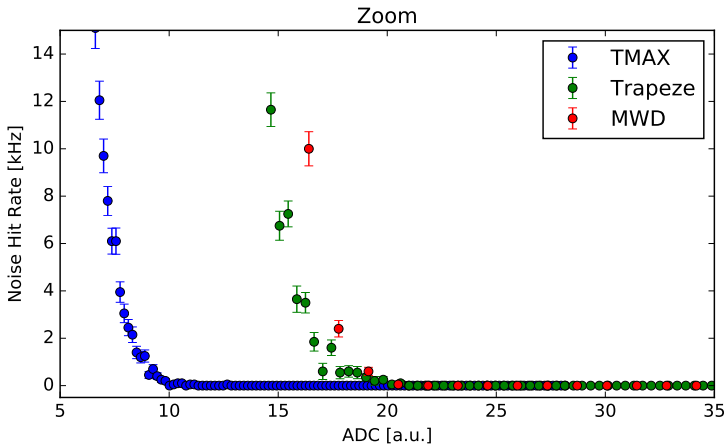
$$\text{Noise} = \text{SMA}_{\tau=300ns}[\text{White Noise}] + \text{White Noise}$$

Simulation - Examples





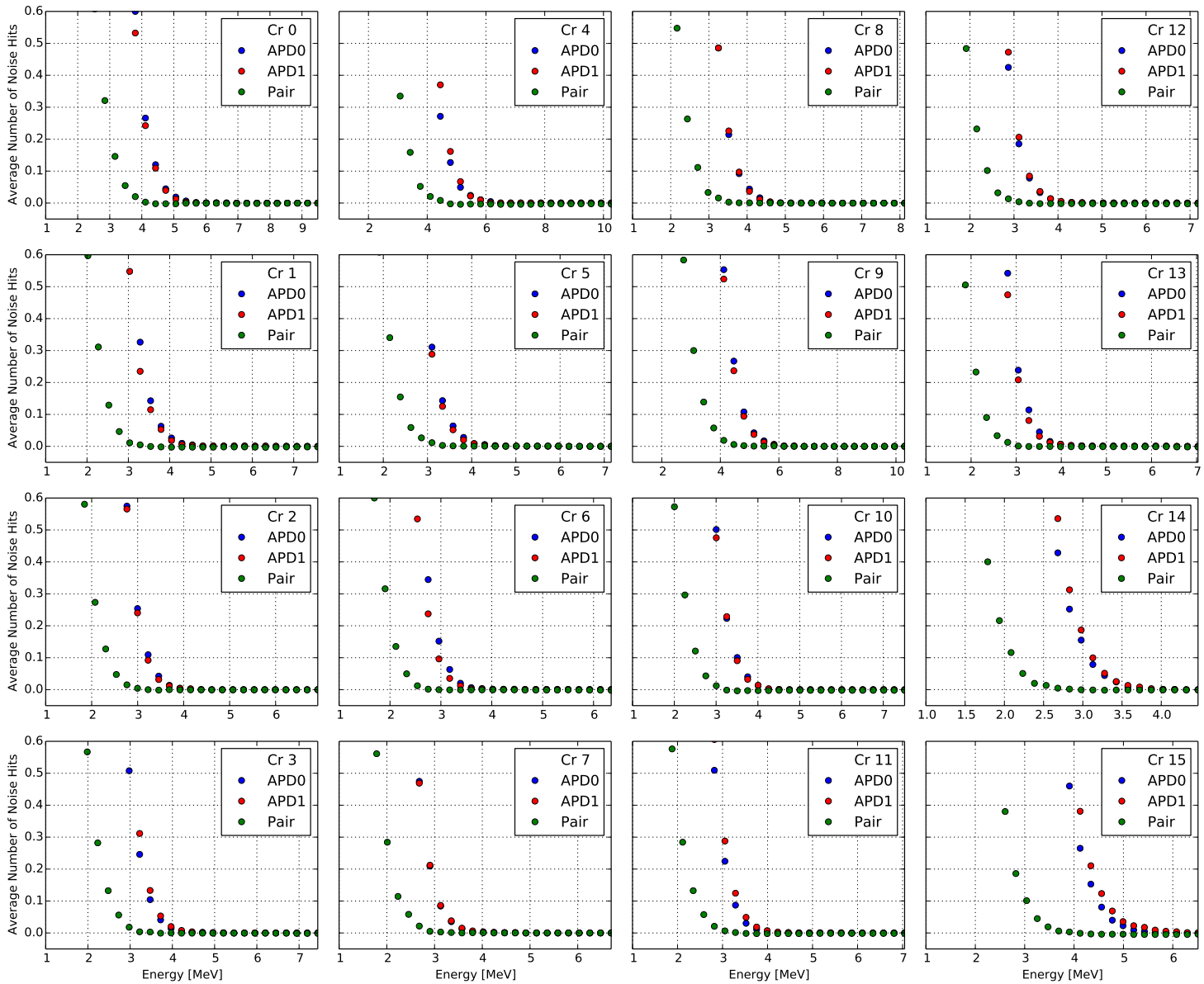




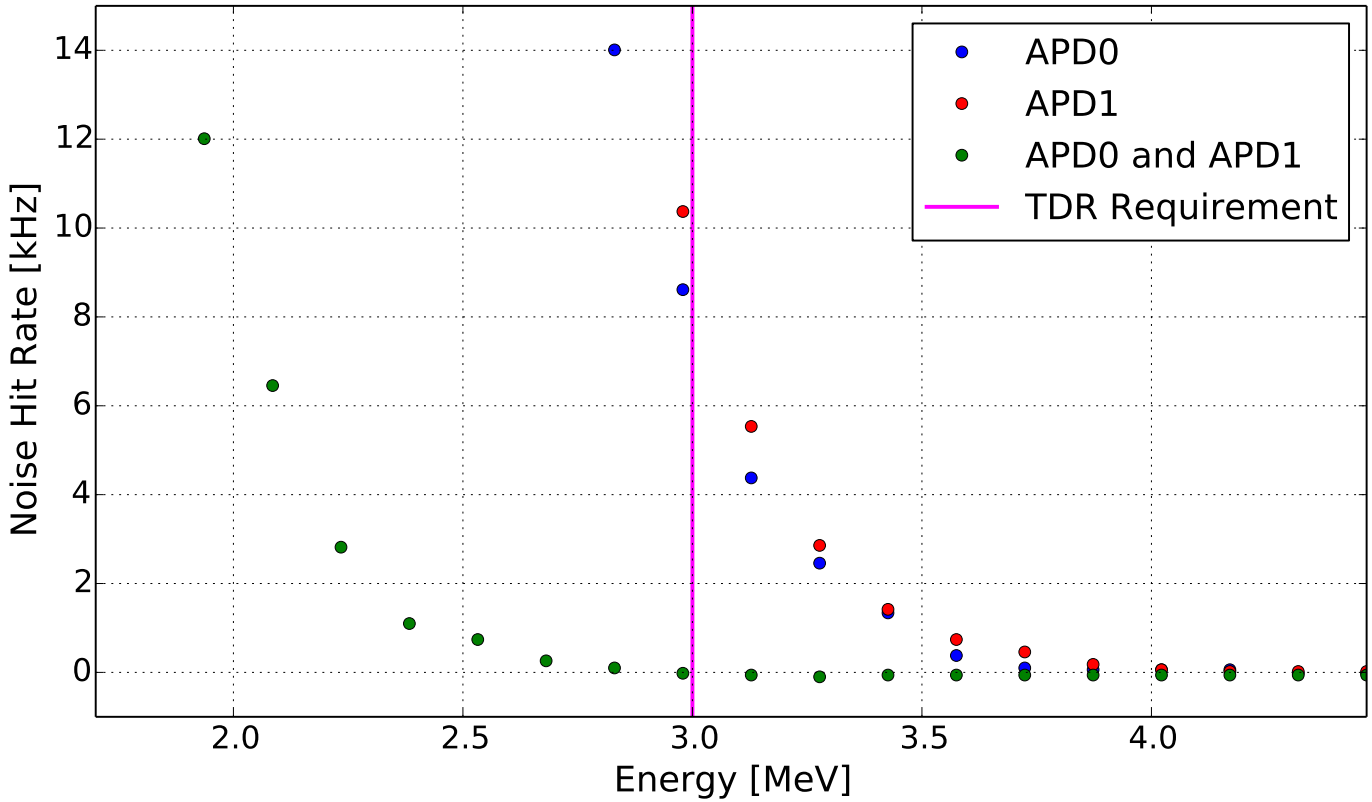
Energy

15 a.u. $\approx$ 3 MeV

Filter Tests with Detector Data - Trapeze Filter



Filter Tests with Detector Data - Trapeze Filter



Noise

Analysis with other filters is ongoing.

Summary and Outlook - Feature Extraction

Summary and Outlook

- Mainz will take care about the ASIC signal FE
- Hardware and software framework is working
- Promising filters available
- Further tests needed
  - Simulations
  - Detector data (beamtime and lightpulser)
- Good cooperation with Groningen (Myroslav)
- Hardware implementation on PANDA ADC

Baseline Adjustment

Bachelor Thesis: Phillip Grasemann

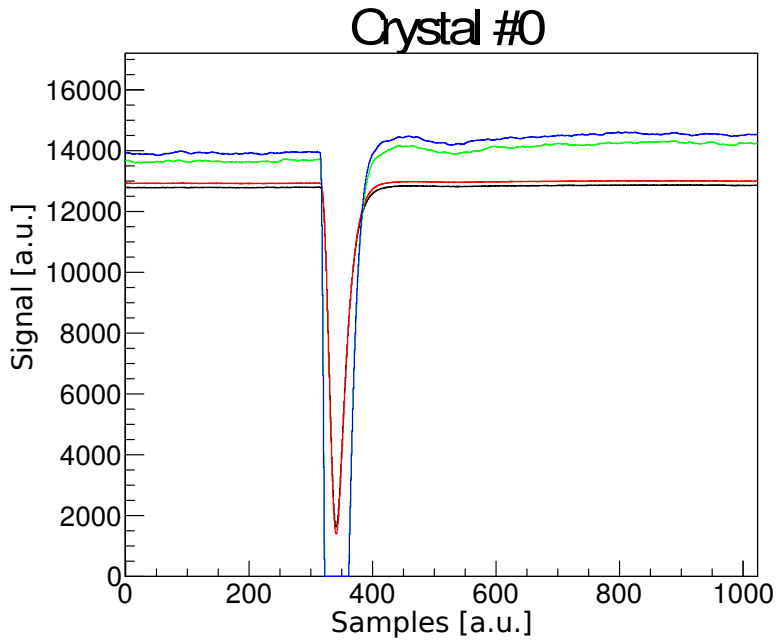
Baseline Adjustment

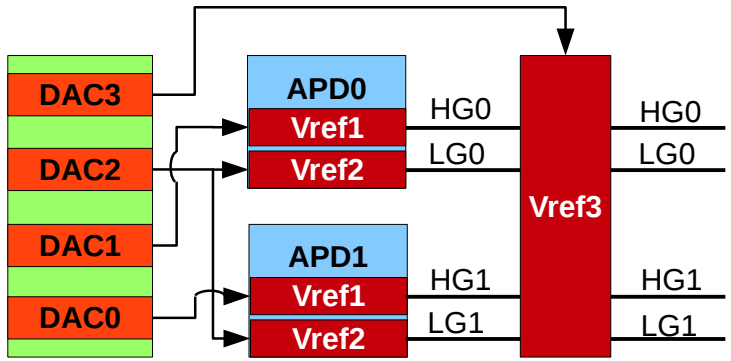
Aim of BA

- Fit ASIC signals in ADC entrance area
- Exploit the full dynamic range

Requirements on BA

- Automatic determination of ASIC DAC values
- Fast procedure (∼ 12000 ASIC x 4 channel)
- Automatic monitoring of baselines (temperature drift etc.)

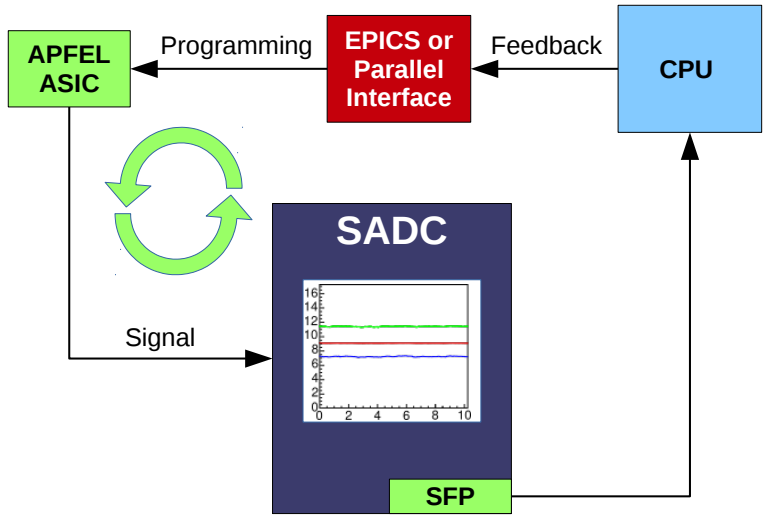




Baseline Adjustment - Current Version

Hardware

- BL readout via MBS DAQ
- Computer with adjustment routine
- ASIC programming:
  - Parallel interface and PCB
  - GSI HADCON via EPICS



Software

- Python script
- Iterative algorithm

PROTO16, 64 Channel

- $\sim$ 50 BL readouts
- Adjustment time $\sim$ 100 s

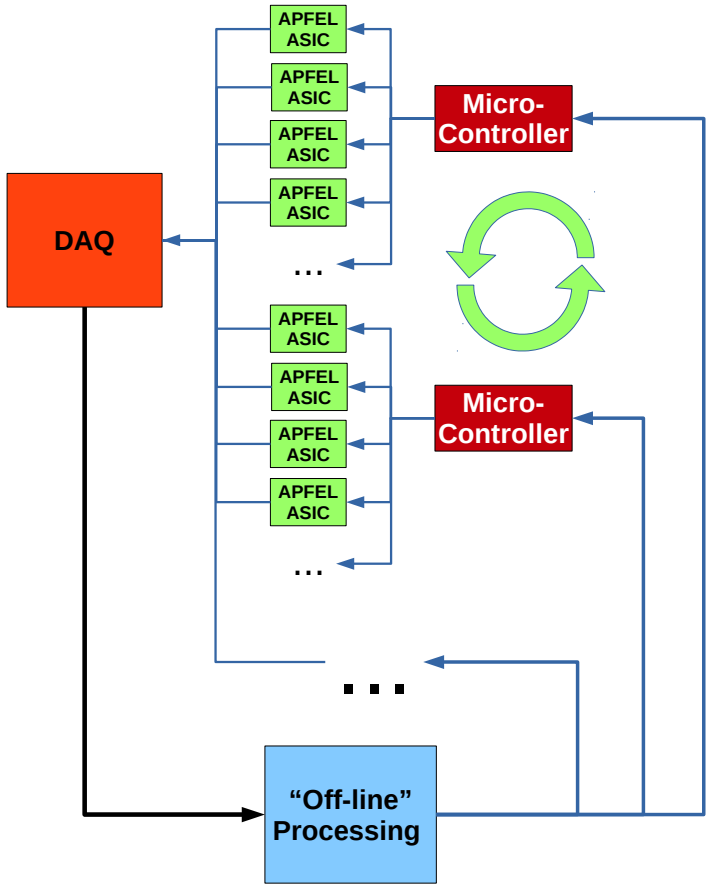
Baseline Adjustment - Scaling: Method 1

Hardware

- BL readout via PANDA DAQ
- Micro controller
  - possibly HADCON

Software

- Adjustment routine adapted to micro controller



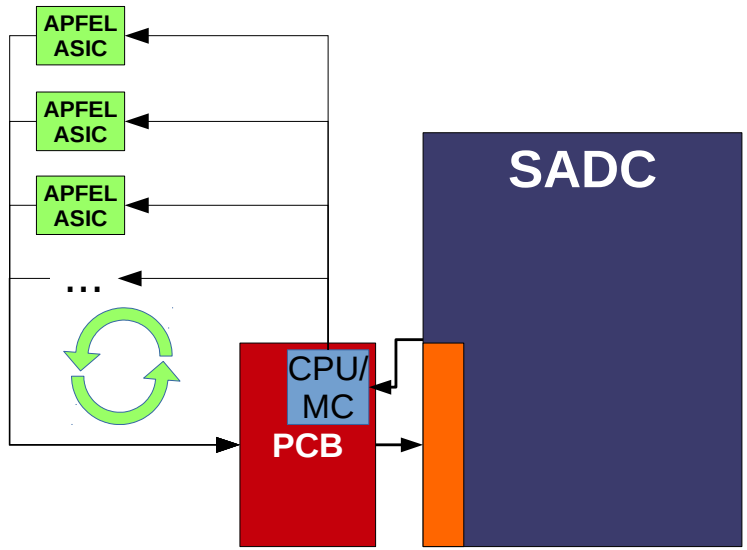
Baseline Adjustment - Scaling: Method 2

Hardware

- Adapter PCB on ADC entrance
  (ADC PCB has to be modified)
- Micro controller or CPU

Software

- Adjustment routine adapted to micro controller
- Or it runs on CPU



Summary and Outlook - Baseline Adjustment

Summary and Outlook

- Mainz will take care about the ASIC baseline adjustment
- PROTO16 operates with first version already
  (Bachelor Thesis: Phillip Grasemann)
- Searching for the best scaling concept

Backup

Smoothing via Simple Moving Average

SMA

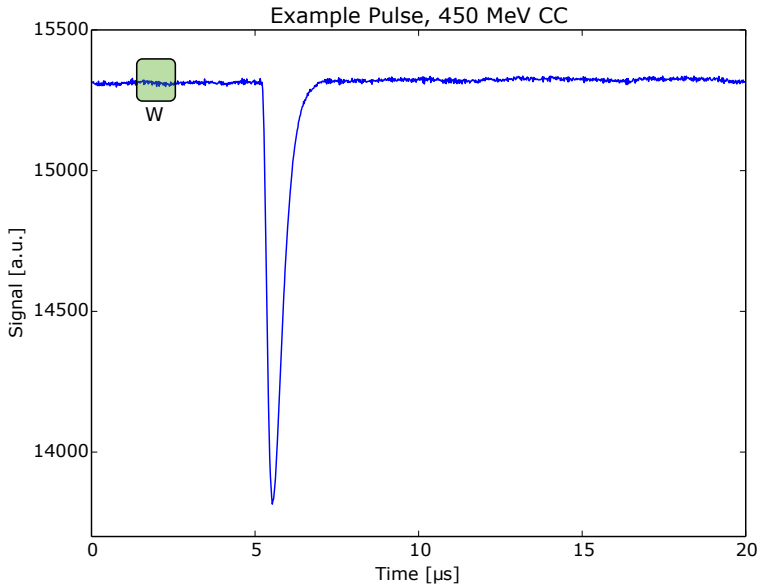
- Smoothing of high frequency noise
- Investigation of influence of baseline averaging

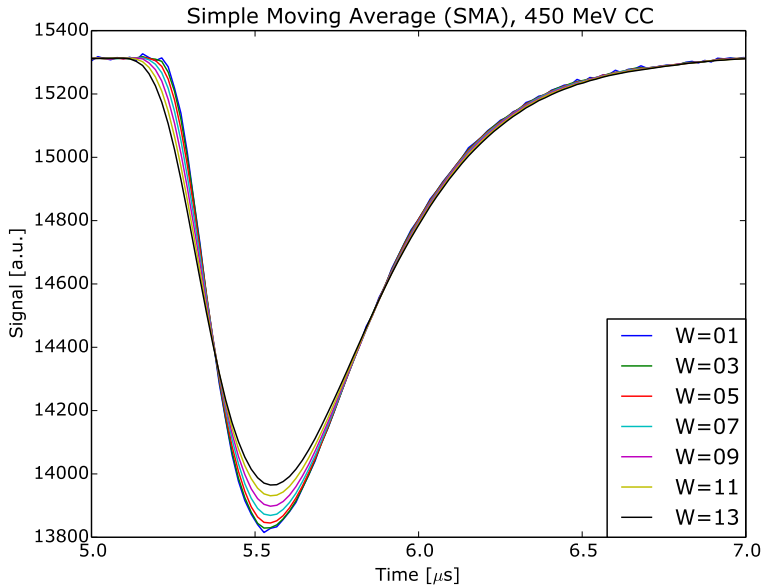
Filter value:

$$F_n = \frac{1}{W} \sum_{i=n}^{n+W} T[i]$$

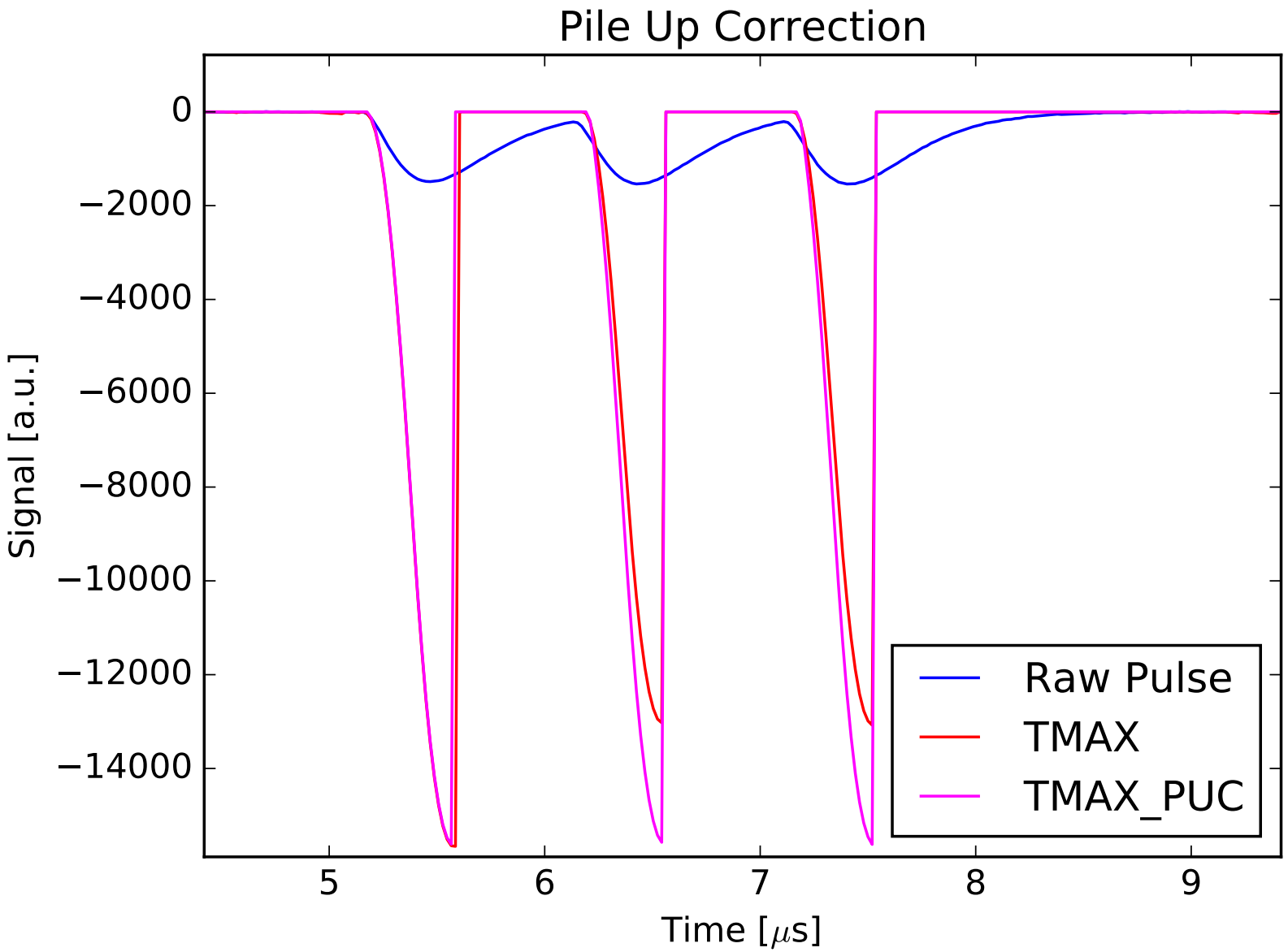
Time delay:

$$t_{delay} = \frac{W-1}{2}$$

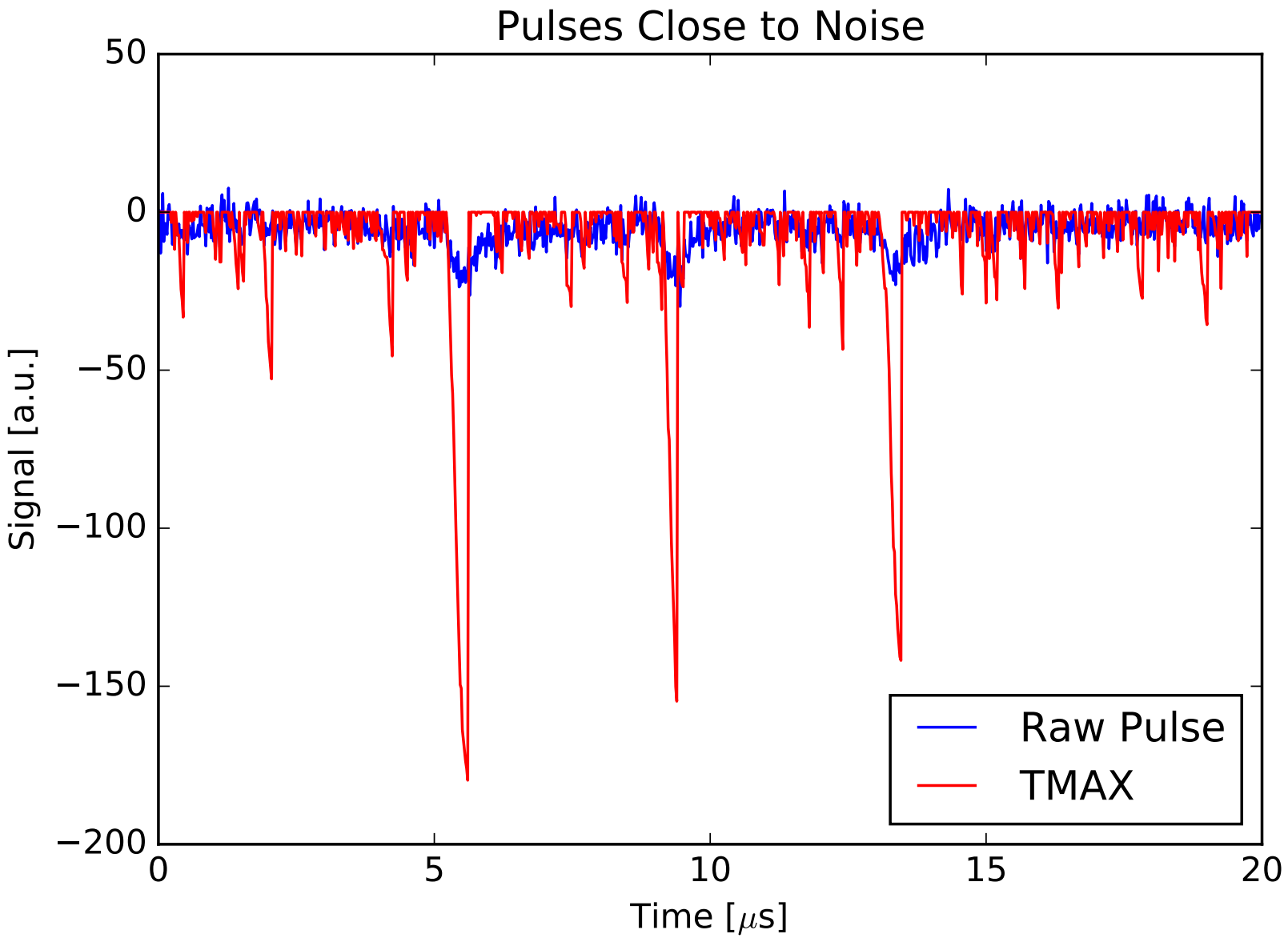




Pile Up Correction 1 μs



Signals Close to Noise



Slopes of HG0 and HG1