Digitization of the PANDA EMC

Guang Zhao (zhaog@ihep.ac.cn)

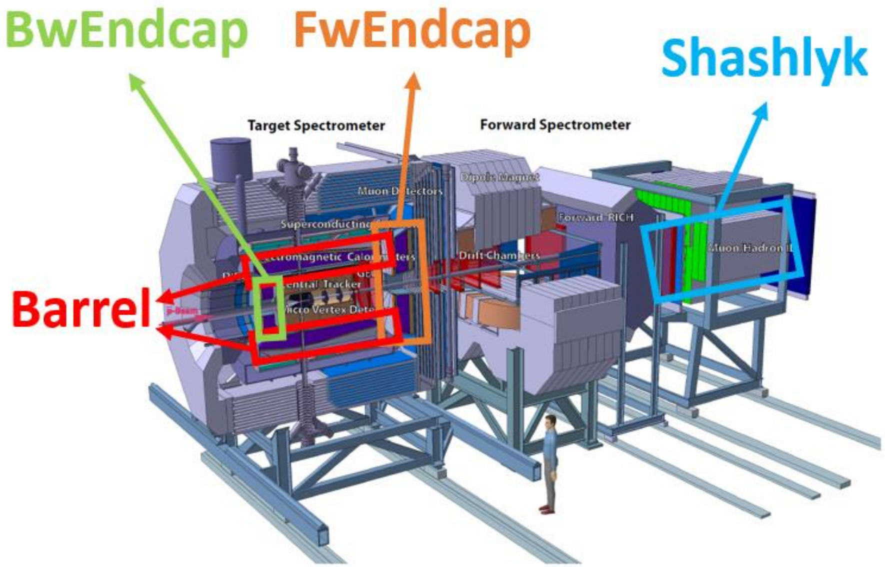
PANDA China Meeting 2021

July 7th, 2021

Outline

- **Introduction**
- **Digitization**
  - Shashlyk EMC digitization
  - Code design
- **Summary**

The PANDA EMC

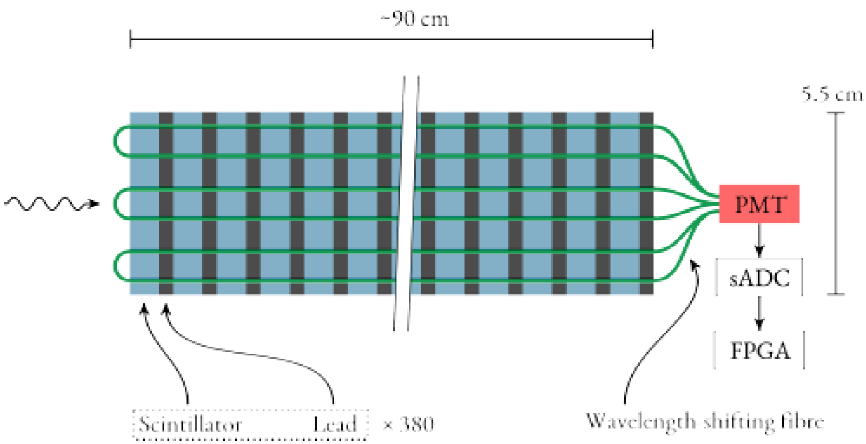


- **Basic requirement:**
  - Detect photons > 10 MeV
  - 4pi solid-angle coverage
- **4 sub-detectors**
  - Homogeneous: $PbWO_4$
    - Barrel
    - Backward endcap
    - Forward endcap
  - Sampling:
    - Shashlyk

Property	Required values			
	Backward endcap	Barrel	Forward endcap	Shashlyk
Relative energy resolution σ_E/E	$\leq 1\% \oplus \frac{2\%}{\sqrt{E/\mathrm{GeV}}}$	$\leq 1\% \oplus \frac{2\%}{\sqrt{E/\mathrm{GeV}}}$	$\leq 1\% \oplus \frac{2\%}{\sqrt{E/\mathrm{GeV}}}$	$\leq 1\% \oplus \frac{(2-3)\%}{\sqrt{E/\mathrm{GeV}}}$
Photon-energy threshold [MeV]	10	10	10	10
Single-detector threshold [MeV]	3	3	3	3
Energy-equivalent noise [MeV]	1	1	1	1
Maximum detectable energy [MeV]	700	7300	14600	15000
Polar-angle coverage (lab frame) [°]	≥ 140	≥ 22	≥ 5	≥ 0
Solid-angle coverage (lab frame) [% 4π]	5.5	84.7	3.2	0.74
Hit rate per detector* [MHz]	0.06	0.06	0.5	~ 1
Radiation hardness [Gy y^{-1}]	10	10	125	1000

* Hit rate per individual crystal or cell.

The shashlyk EMC

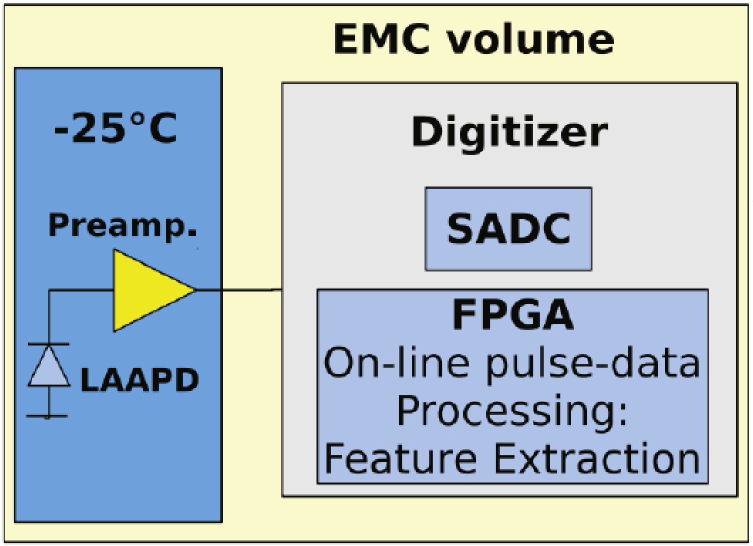


- ✓ **Forward detector: shashlyk**
  - ✓ **Shashlyk cells**
    - ✓ Scintillator + lead
    - ✓ Wavelength-shifting fiber (read out light)
  - ✓ **Photodetectors**
    - ✓ PMT
      - ✓ Higher gain, no preamplifier (absence of magnetic field)

Frond-end digitizer

- **A vital role in processing detector signals in real time**
- **A FPGA-based digitizer module**

Forward end-cap readout

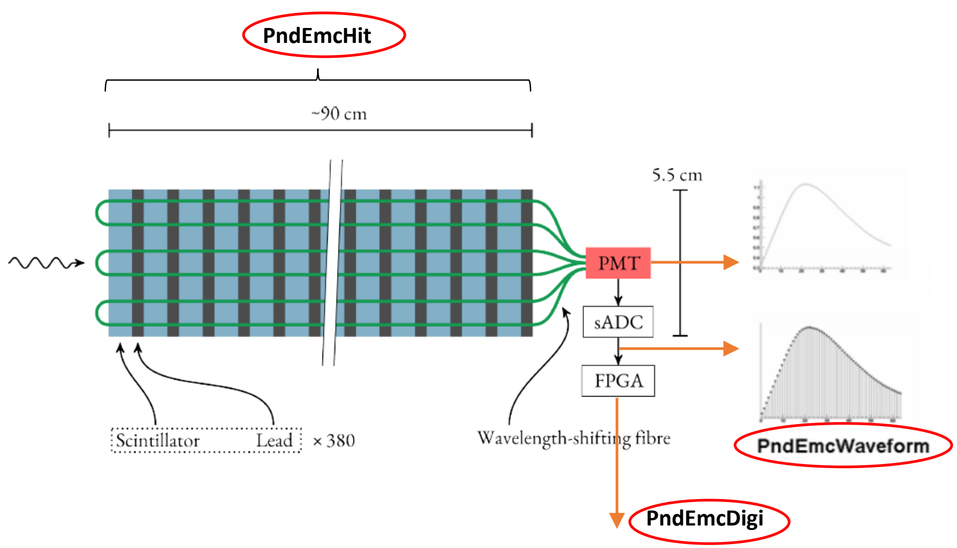


For shashlyk:

- PMT instead of LAAPD
- No cooled detector volume
- 22 boards in total
- For each board:
  - 14-bit ADCs (125 MHz)
  - 2 x Xilinx Kintex-7 FPGAs

Digitization in PandaRoot

Event Generation	Detector Simulation	Digitization	Reconstruction	Rootification
EVTGEN	HITS	DIGITS	SHOWERS	TTree/THist



Signal Generator (SG)

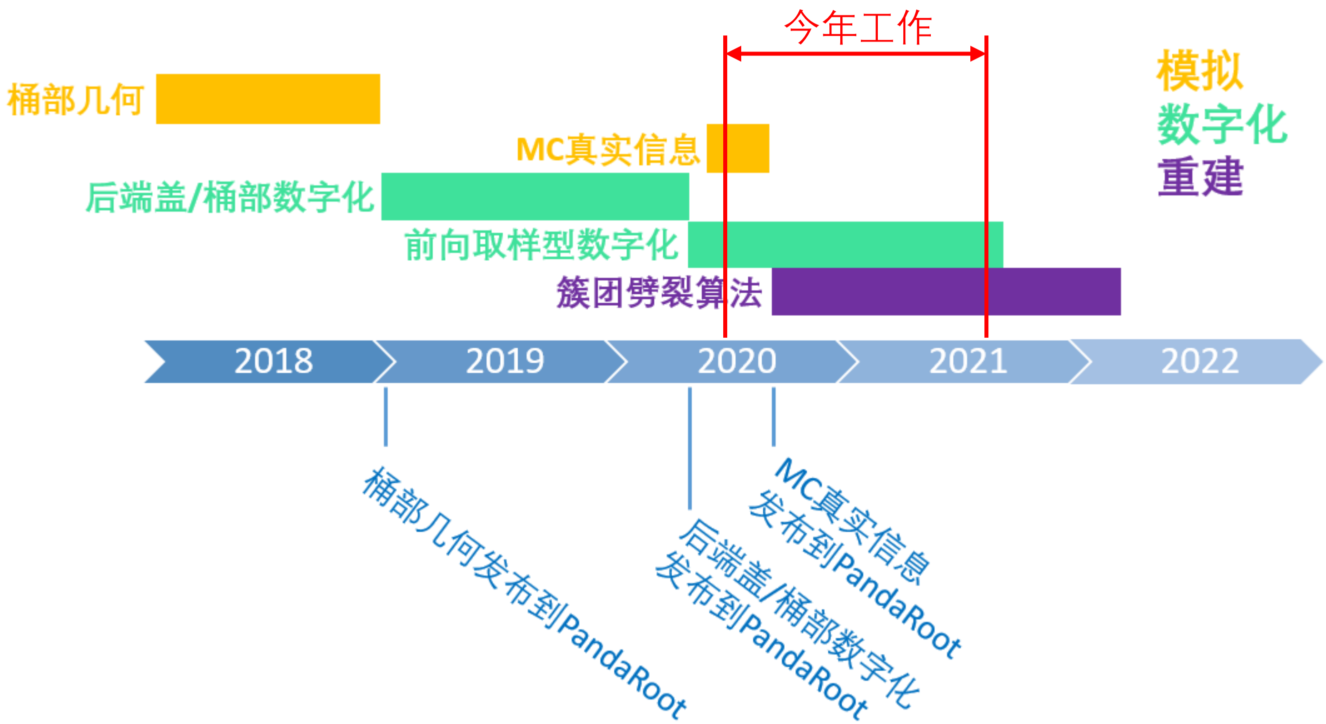
- **Analog waveforms creation**
- **Noises generation**
- **Digitization**
- **Pile-up waveforms creation**

Feature Extraction (FE)

- **Hit detection**
- **Amplitude/time extraction**
- **Pile-up recovery**

Time-based simulation

Road Map



今年工作
模拟
数字化
重建
桶部几何
MC真实信息
后端盖/桶部数字化
前向取样型数字化
簇团劈裂算法
2018
2019
2020
2021
2022
桶部几何发布到PandaRoot
后端盖/桶部数字化
发布到PandaRoot
MC真实信息
发布到PandaRoot

Shashlyk EMC digitization

- **In collaboration with Markus Preston from Stockholm University**

Signal Generator

Signal generator

Photon statistics

PndEmcHit
(Energy & Time)

Scaled by *Energy* & *Q.E.*

$N_{p.e.}$

Photon statistics in PMT:

Importing the photon statistics by smearing the energy*:

$$\frac{\sigma_E}{E} = \frac{1}{\sqrt{E}} \times \sqrt{\frac{F}{N_{p.e.}}}$$

(*Q.E.* = 0.15, *F* = 1.3)

* EMC TDR

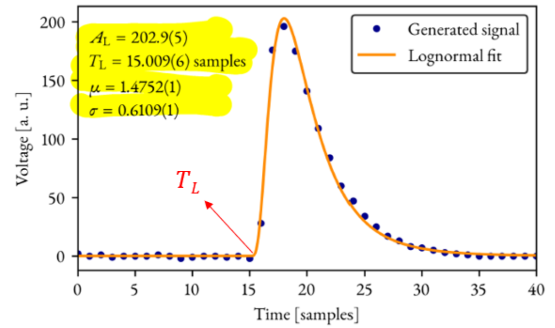
PndEmcWaveform

Idea pulse

Idea pulse generation by sampling:

$$f_{\mathrm{L}}(t) = \begin{cases} \frac{A_{\mathrm{L}} \exp(\mu - 0.5\sigma^2)}{t - T_{\mathrm{L}}} \exp\left[\frac{[\log(t - T_{\mathrm{L}}) - \mu]^2}{2\sigma^2}\right], & \text{if } t > T_{\mathrm{L}} \\ 0, & \text{otherwise,} \end{cases}$$

PndEmcHit
(Energy & Time)



PndEmcWaveform

Scaled by *Energy* & *Q.E.*

$N_{p.e.}$

Photon statistics in PMT:

Importing the photon statistics by smearing the energy*:

$$\frac{\sigma_E}{E} = \frac{1}{\sqrt{E}} \times \sqrt{\frac{F}{N_{p.e.}}}$$

($Q.E.$ = 0.15, F = 1.3)

* EMC TDR

Real pulse

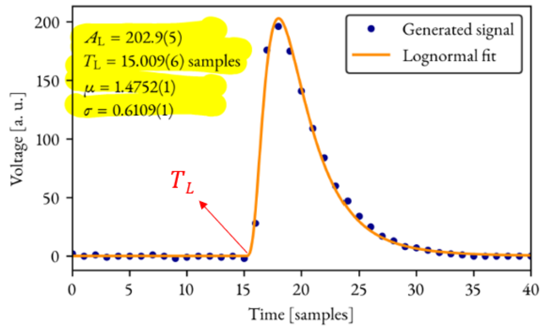
Idea pulse generation by sampling:

$$f_L(t) = \begin{cases} \frac{A_L \exp(\mu - 0.5\sigma^2)}{t - T_L} \exp\left[\frac{[\log(t - T_L) - \mu]^2}{2\sigma^2}\right], & \text{if } t > T_L \\ 0, & \text{otherwise,} \end{cases}$$

ADC pulse generation:

- Adding electronics noise: 1 ADC
- Digitizing: 125 MHz

PndEmcHit (Energy & Time) → PndEmcWaveform



Scaled by *Energy* & *Q.E.*

$N_{p.e.}$

Photon statistics in PMT:

Importing the photon statistics by smearing the energy*:

$$\frac{\sigma_E}{E} = \frac{1}{\sqrt{E}} \times \sqrt{\frac{F}{N_{p.e.}}}$$

($Q.E.$ = 0.15, F = 1.3)

* EMC TDR

Pulses in simulation

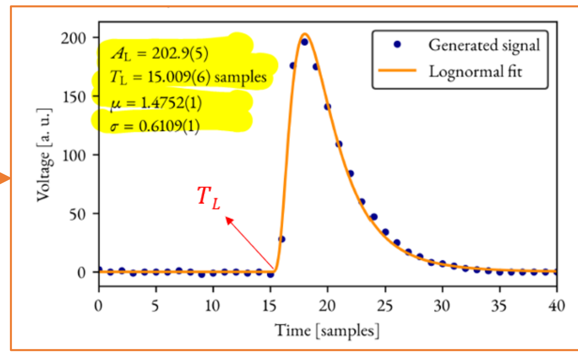
Idea pulse generation by sampling:

$$f_L(t) = \begin{cases} \frac{A_L \exp(\mu - 0.5\sigma^2)}{t - T_L} \exp\left[\frac{[\log(t-T_L)-\mu]^2}{2\sigma^2}\right], & \text{if } t > T_L \\ 0, & \text{otherwise,} \end{cases}$$

ADC pulse generation:

- Adding electronics noise: 1 ADC
- Digitizing: 125 MHz

PndEmcHit
(Energy & Time)



PndEmcWaveform

Scaled by *Energy* & *Q.E.*

$N_{p.e.}$

Photon statistics in PMT:

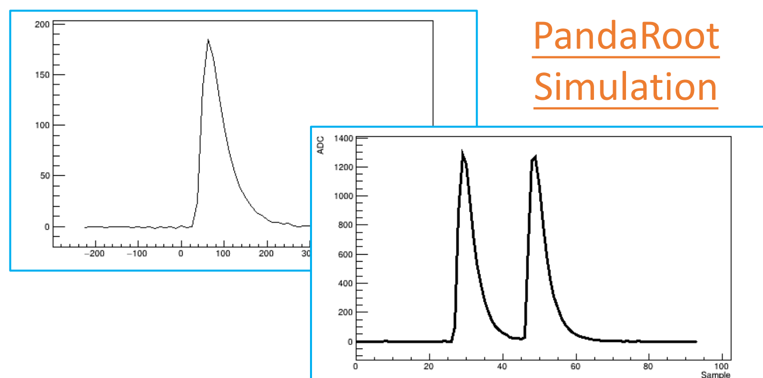
Importing the photon statistics by smearing the energy*:

$$\frac{\sigma_E}{E} = \frac{1}{\sqrt{E}} \times \sqrt{\frac{F}{N_{p.e.}}}$$

($Q.E.$ = 0.15, F = 1.3)

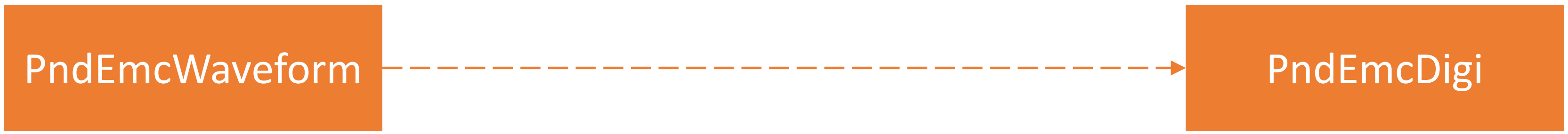
* EMC TDR

PandaRoot Simulation



Feature Extraction

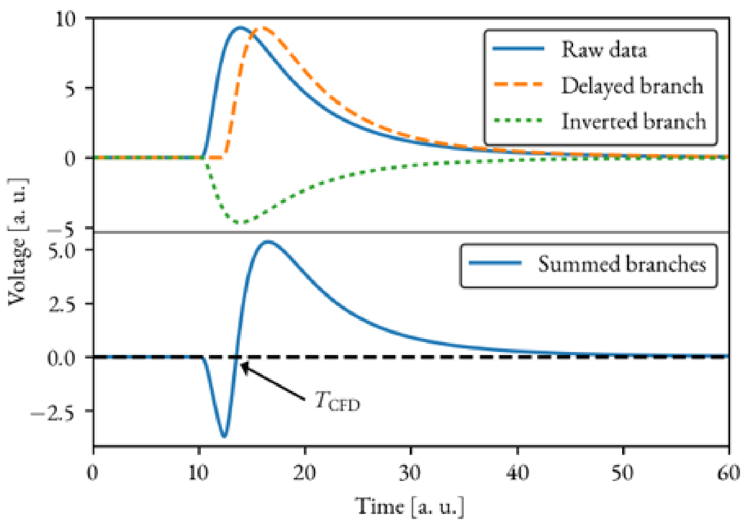
Feature Extraction



Time determination: CFD

PndEmcWaveform → (CFD) → T0

PndEmcDigi



Constant Fraction Discriminator (CFD)

- Extract time at a fixed fraction of the maximum height
- To reduce the time-walk

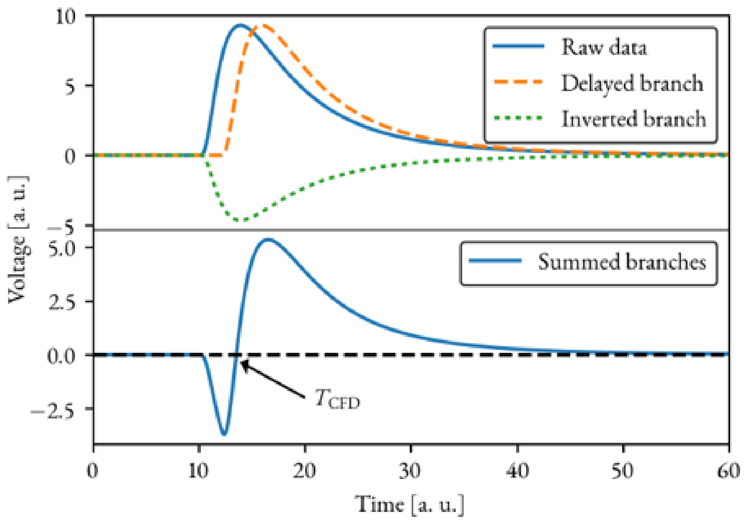
$$V_{\mathrm{CFD}}(t) = (V(t - t_d) - V_0(t - t_d)) - f(V(t) - V_0(t))$$

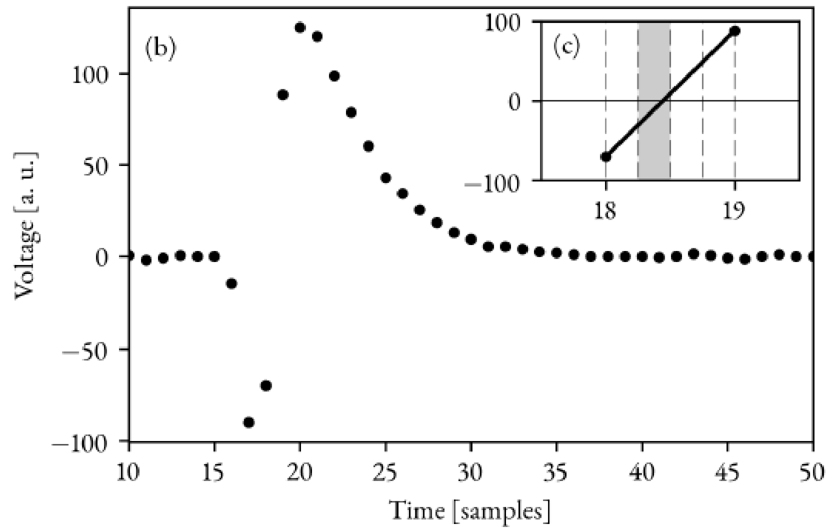
CFD parameters: $t_d = 2$, $f = 0.5$

Time determination: Binary-CFD

PndEmcWaveform → (CFD) → T0

PndEmcDigi





Constant Fraction Discriminator (CFD)

- Extract time at a fixed fraction of the maximum height
- To reduce the time-walk

$$V_{\mathrm{CFD}}(t) = (V(t-t_d) - V_0(t-t_d)) - f(V(t) - V_0(t))$$

CFD parameters: $t_d = 2$, $f = 0.5$

Binary-Search CFD

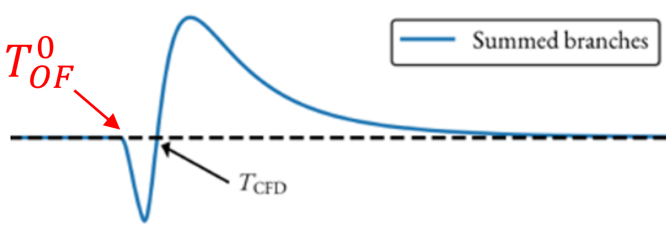
- Binary search the zero-crossing quarter-sample wide window
- $T_{B\text{-}CFD}$: Center of the window
- Arithmetic:
  - One-bit shift: (V(1) – V(0))/2
  - Much faster than division

Time determination: Correction

PndEmcWaveform → (CFD) → T0

PndEmcDigi

Time correction: Correction from the zero-crossing to the actual waveform start time



$$T^0_{\mathrm{OF},\,i} = T_{\mathrm{B\text{-}CFD},\,i} - \langle T_{\mathrm{B\text{-}CFD}} - T_{\mathrm{L}} \rangle.$$

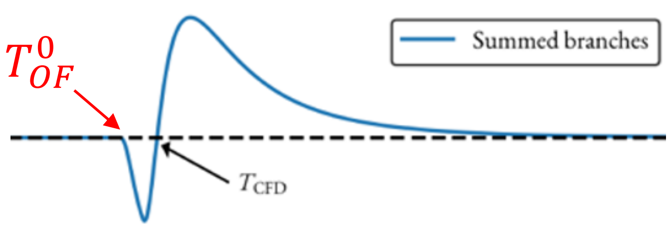
B-CFD window	$\langle T_{\mathrm{B\text{-}CFD}} - T_{\mathrm{L}} \rangle$ [samples]
1	3.454
2	3.460
3	3.417
4	3.413

Time determination: Simulation

PndEmcWaveform → (CFD) → T0

PndEmcDigi

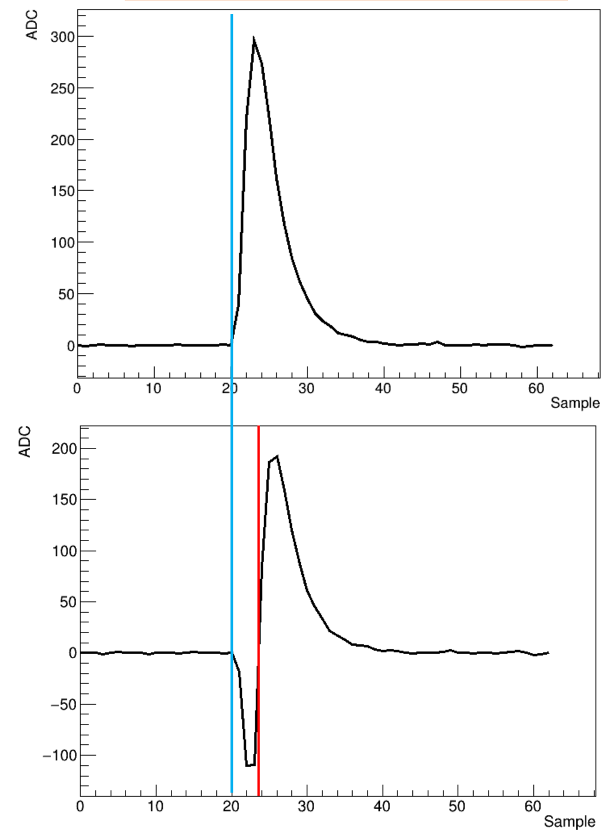
Time correction: Correction from the zero-crossing to the actual waveform start time



$$T^0_{\mathrm{OF},\,i} = T_{\mathrm{B\text{-}CFD},\,i} - \langle T_{\mathrm{B\text{-}CFD}} - T_{\mathrm{L}} \rangle.$$

B-CFD window	$\langle T_{\mathrm{B\text{-}CFD}} - T_{\mathrm{L}} \rangle$ [samples]
1	3.454
2	3.460
3	3.417
4	3.413

PandaRoot Simulation



T^0_{OF} T_{B-CFD}

Optimal Filter: Fine time/amplitude determination

Optimal Filter (OF)

- The process of OF is equivalent to fitting the incoming data with a linearized version of the known pulse shape in a χ^2 fit

$$\chi^2 = \sum_{i=1}^{M} \sum_{j=1}^{M} (S_i - Ag(t_i - \tau) V_{ij} (S_j - Ag(t_j - \tau))$$

> g(t): Pulse function
> A: Amplitude
> τ: Time difference to T_{OF}^{0}
> S: Waveform content

- By solving this linear problem, the ***A*** and ***Aτ*** can be written in the following form, which are two FIR filters:

$$\alpha_1 \equiv A = \sum_{i=1}^{M} a_i S_i$$

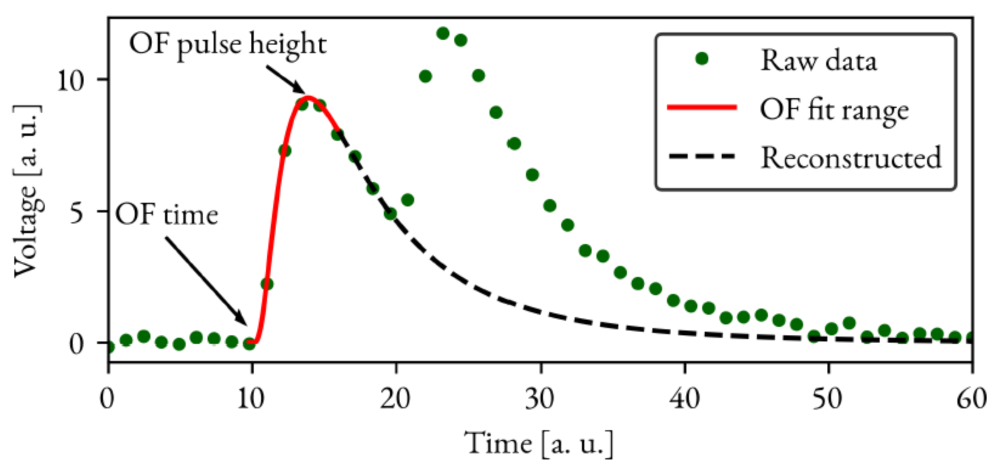
$$\alpha_2 \equiv A\tau = \sum_{i=1}^{M} b_i S_i$$

- The coefficients a and b can be analytically solved, which gives the ***A*** and ***τ***
- The OF can provide an amplitude and a more accurate time as it used more information of the waveform

Optimal Filter: Pile-up recovery

Pileup recovery:

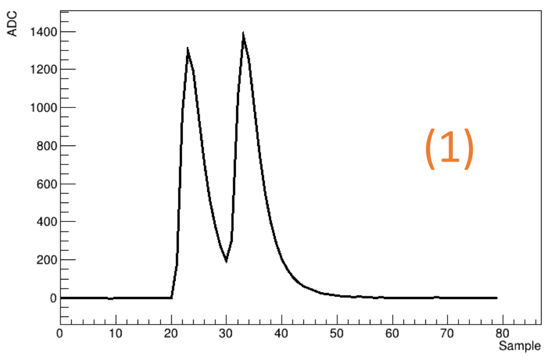
- To reduce the contamination from upcoming pile-up waveforms, a truncated pulse shape are used in the OF (B0, B0+M)
- The previous detected pulse are subtracted as the baseline
- Perform the CFD + OF for the remaining waveform



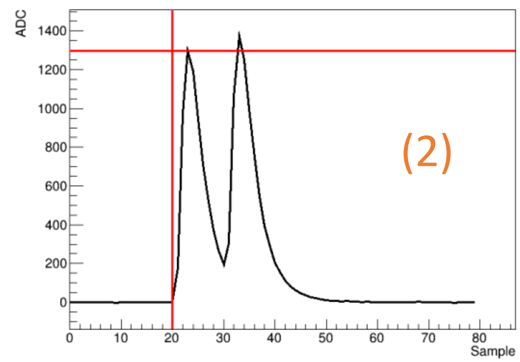
B0 = -3
M = 4

PandaRoot simulation

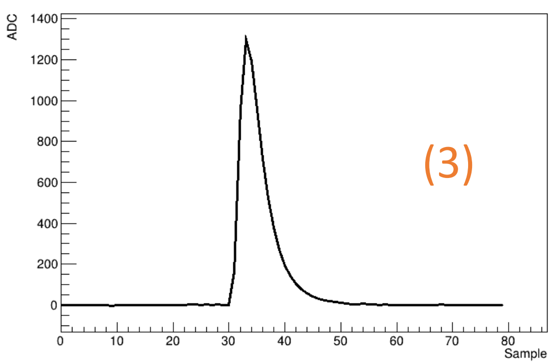
PandaRoot Simulation



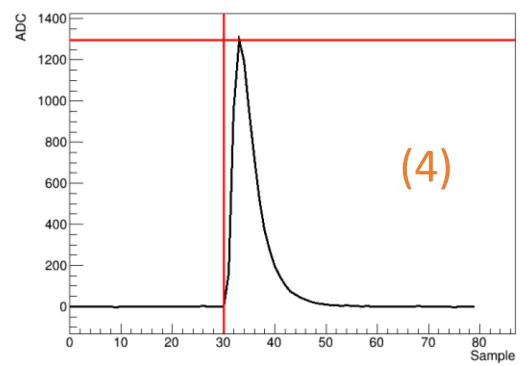
A pile-up waveform



First waveform detected

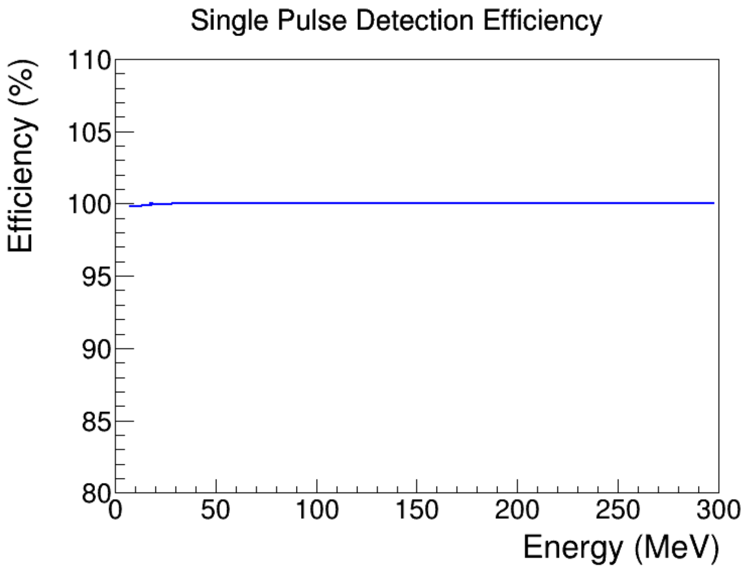


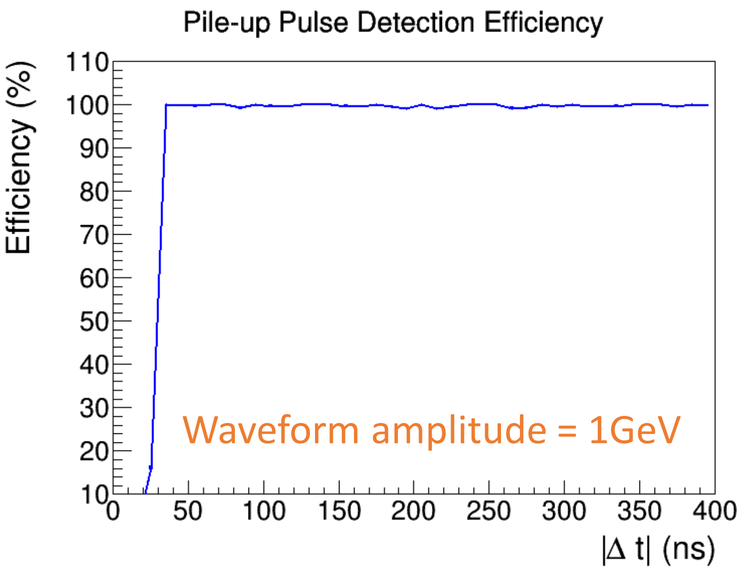
Baseline subtracted



Second waveform detected

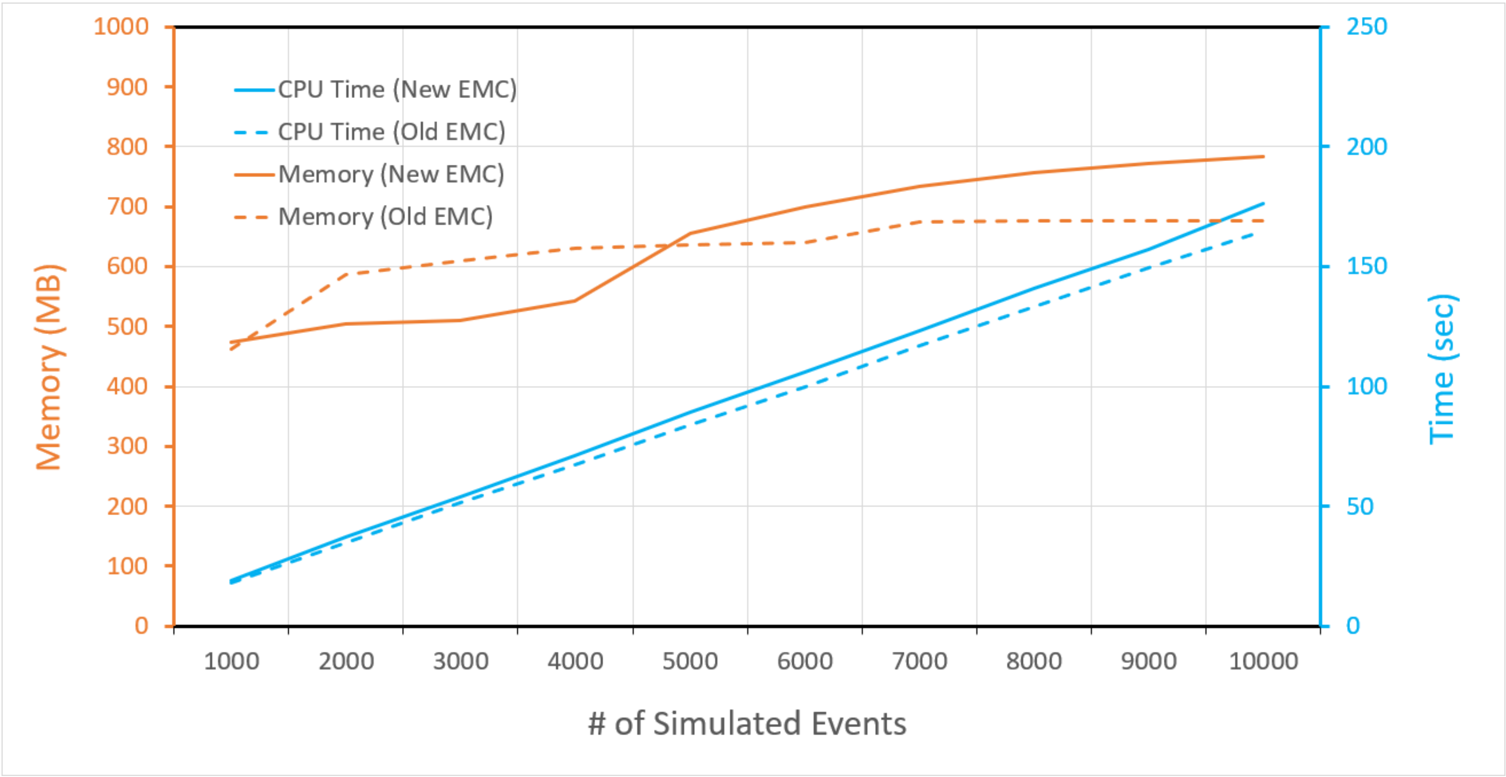
Waveform detection efficiency in simulation





Reasonable pulse detection efficiencies

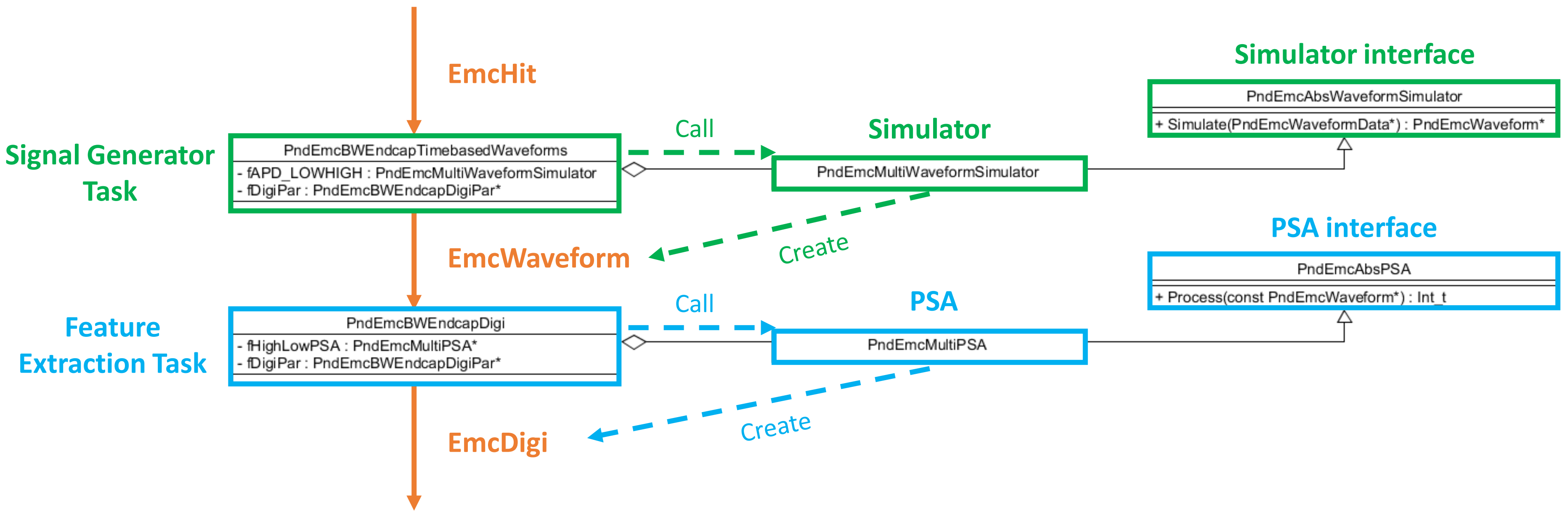
Performance Test



We can obtain quite similar computing performance compared to the old PandaRoot algorithm

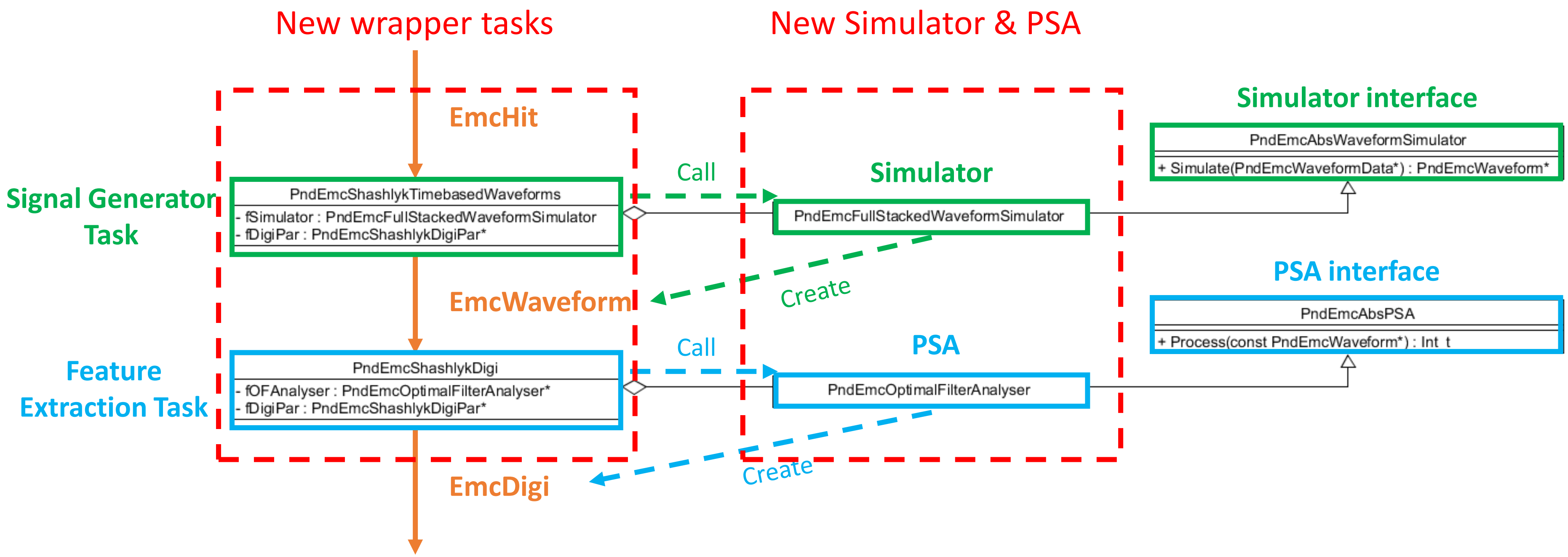
Code design

Code Structure (BWEC)



- Two tasks for signal generator and feature extraction respectively
- Simulator and Pulse Shape Analyzer (PSA) as the “algorithms”
- The algorithms inherit from the “interfaces”

Code Structure (Shashlyk)



- Easy to modify from the bwec package by plugging in two new "algorithms"
- An entirely new Simulator and PSA are implemented for the shashlyk EMC (core algorithms)
- New wrapper tasks (only simple modifications)

Summary

- **Digitization for the shashlyk EMC is almost done**
  - Have implemented Markus' work in PandaRoot, including
    - Pulse generation using a shape template
    - Feature extraction using CFD+OF filters
  - Code is most ready. Need some final checks before checking in

- **Code design**
  - Unified OO framework for the digitization
  - Key functionalities are modularized. Can be easy for migration

- **Plan**
  - To check in the code of shashlyk EMC digitization
  - Will be focusing on the reconstruction
    - Guide Ziyu to study the cluster splitting algorithm

Thank you!