

# Digitization for the Shashlyk EMC

**Guang Zhao, Markus Preston, Shengsen Sun**

[zhaog@ihep.ac.cn](mailto:zhaog@ihep.ac.cn)

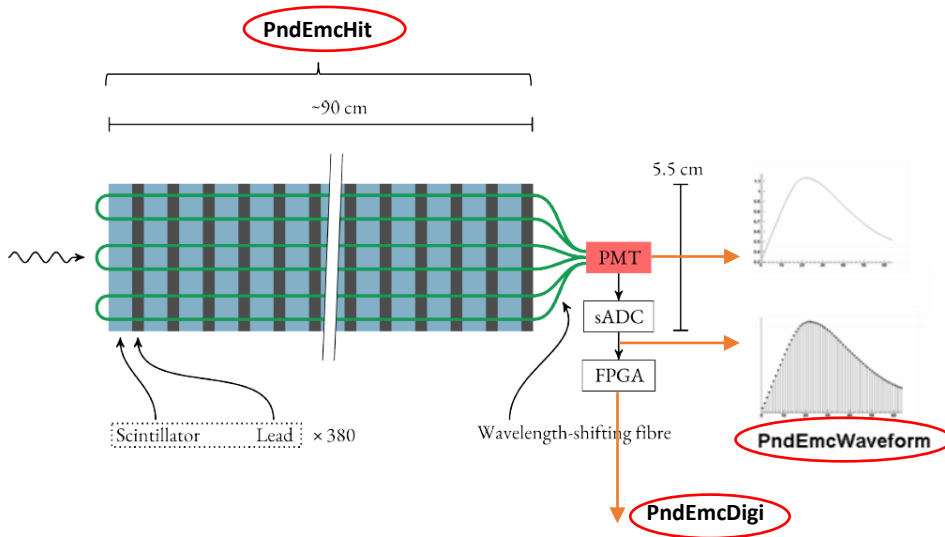
**PANDA Collaboration Meeting 20/03**

**Oct. 27<sup>th</sup>, 2020**

# Outline

- **Introduction**
- **Shashlyk digitization in PandaRoot**
  - Signal generator
  - Feature extraction
- **Summary**

# Digitization process in PandaRoot



## 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

# The Shashlyk EMC Parameters

Table 4.1: Requirements on the EMC detectors in PANDA. Data taken from [23, 48].

Property	Required values			Shashlyk
	Backward endcap	Barrel	Forward endcap	
Relative energy resolution $\sigma_E/E$	$\leq 1\% \oplus \frac{2\%}{\sqrt{E/\text{GeV}}}$	$\leq 1\% \oplus \frac{2\%}{\sqrt{E/\text{GeV}}}$	$\leq 1\% \oplus \frac{2\%}{\sqrt{E/\text{GeV}}}$	$\leq 1\% \oplus \frac{(2-3)\%}{\sqrt{E/\text{GeV}}}$
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) [°]	$\geq 140$	$\geq 22$	$\geq 5$	$\geq 0$
Solid-angle coverage (lab frame) [% $4\pi$ ]	5.5	84.7	3.2	0.74
Hit rate per detector* [MHz]	0.06	0.06	0.5	$\sim 1$
Radiation hardness [Gy $y^{-1}$ ]	10	10	125	1000

\* Hit rate per individual crystal or cell.

## For Shashlyk EMC:

- Larger energy resolution (sampling detector, larger cell size)
- Polar angle coverage: 0-5 deg vertically, 0-10 deg horizontally
- The same single-detector threshold starting at 3 MeV ← Require fine digitization

# Signal Generator

# Signal generator



# Photon statistics

PndEmcHit  
(Energy & Time)

PndEmcWaveform

Scaled  
by  
Energy  
& Q.E.



Photon statistics in PMT:  
Importing the photon statistics by smearing the energy\*:

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

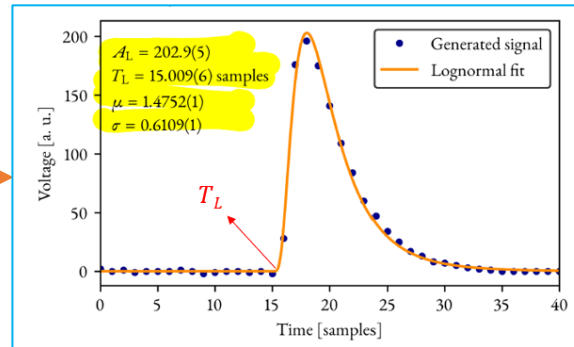
(Q.E. = 0.15, F = 1.3)

\* EMC TDR

# Idea pulse

Idea pulse generation by sampling:

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



**PndEmcHit**  
(Energy & Time)

**PndEmcWaveform**

Scaled  
by  
Energy  
& Q.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)

$N_{p.e.}$

\* EMC TDR



# Real pulse

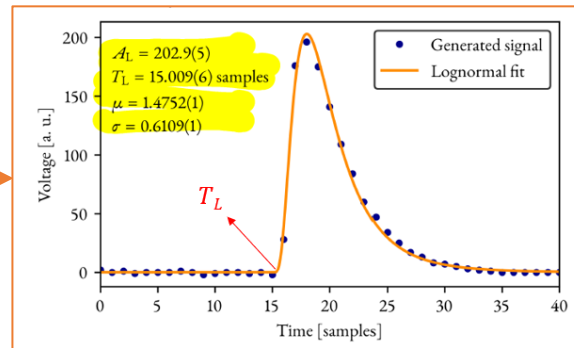
Idea pulse generation by sampling:

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

ADC pulse generation:

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

PndEmcHit  
(Energy & Time)



PndEmcWaveform

Scaled  
by  
Energy  
& Q.E.

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)

$N_{p.e.}$

\* EMC TDR

# Pulses in simulation

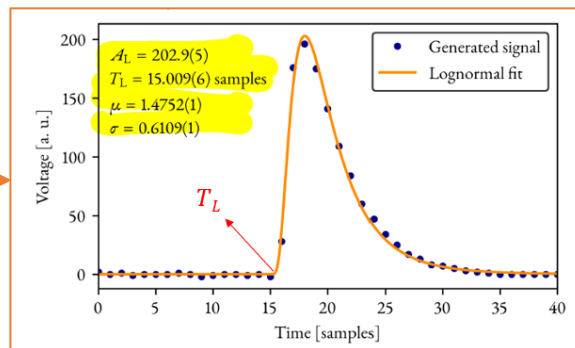
Idea pulse generation by sampling:

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

ADC pulse generation:

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

PndEmcHit  
(Energy & Time)



PndEmcWaveform

Scaled  
by  
Energy  
& Q.E.

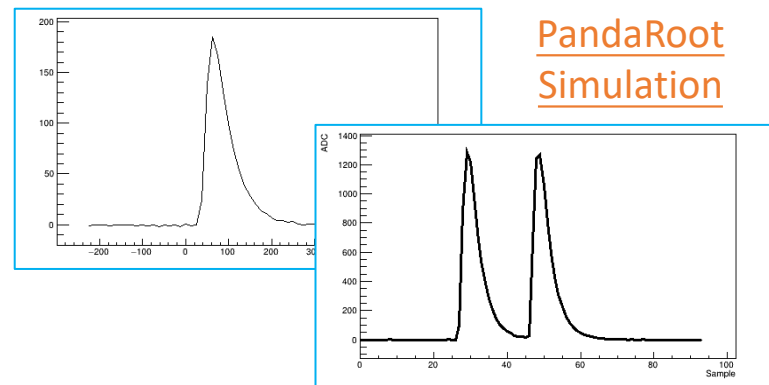
Photon statistics in PMT:  
Importing the photon statistics by  
smearing the energy\*:

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

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

\* EMC TDR

$N_{p.e.}$

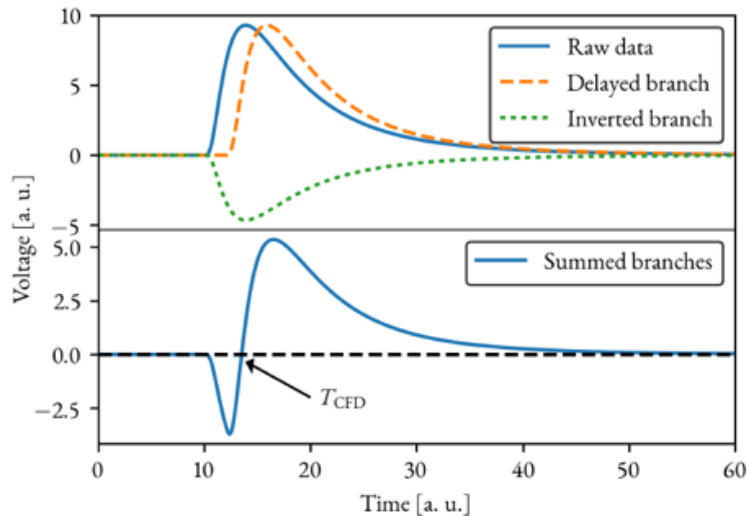
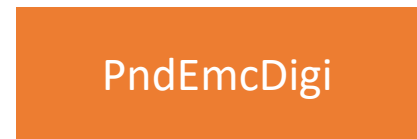


# Feature Extraction

# Feature Extraction



# Time determination: CFD



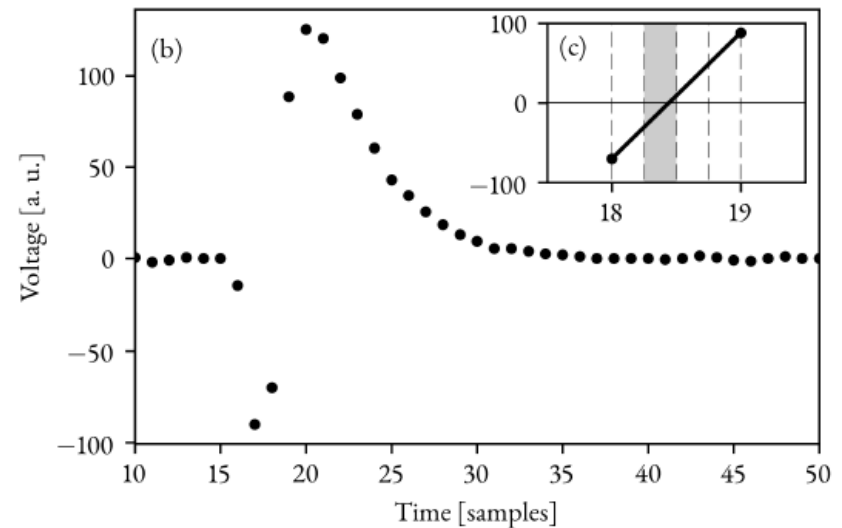
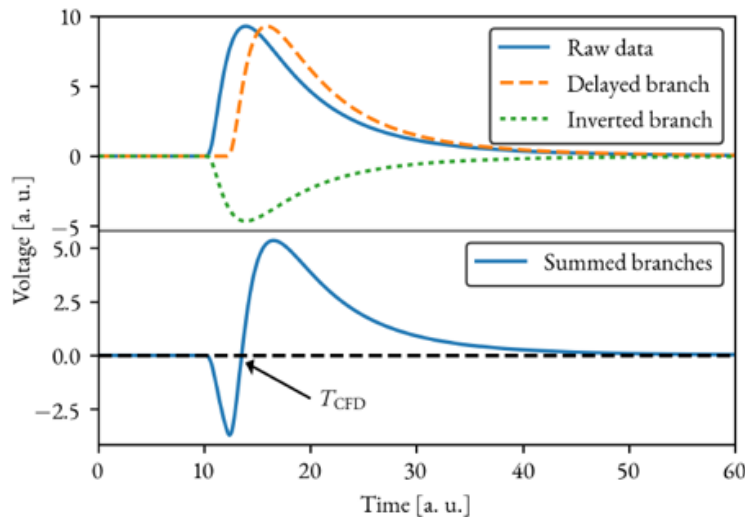
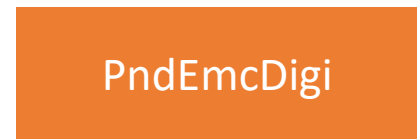
## Constant Fraction Discriminator (CFD)

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

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

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

# Time determination: Binary-CFD



## Constant Fraction Discriminator (CFD)

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

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

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

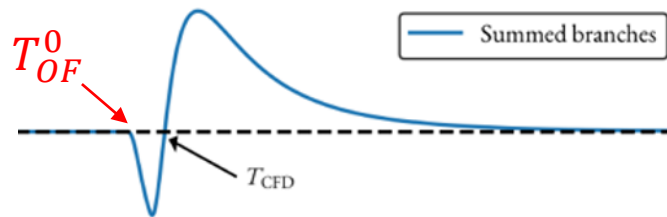
## Binary-Search CFD

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

# Time determination: Correction



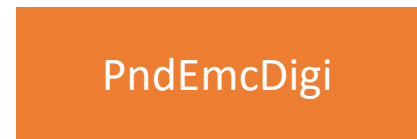
**Time correction:** Correction from the zero-crossing to the actual waveform start time



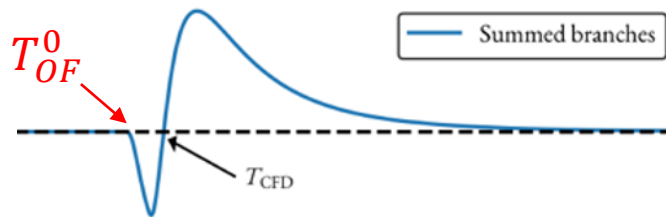
$$T_{OF, i}^0 = T_{B-CFD, i} - \langle T_{B-CFD} - T_L \rangle.$$

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

# Time determination: Simulation



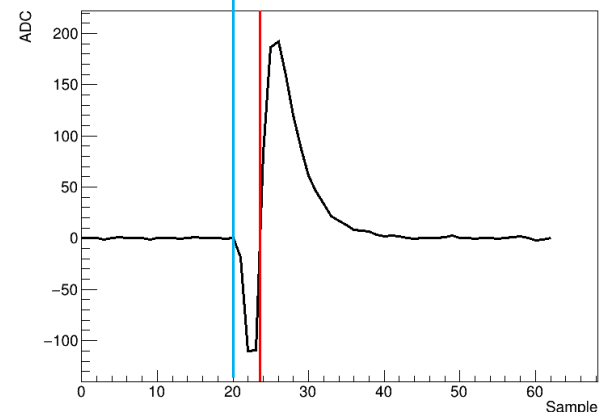
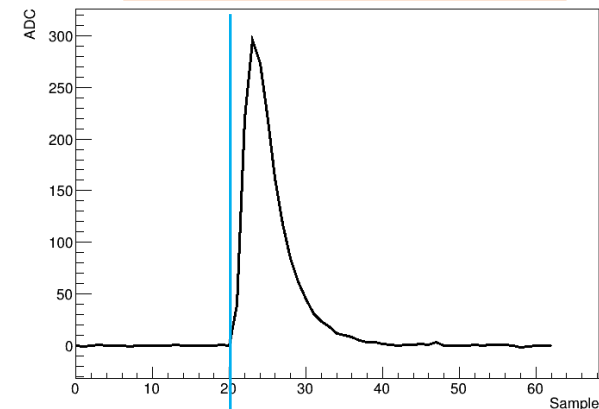
**Time correction:** Correction from the zero-crossing to the actual waveform start time



$$T_{OF, i}^0 = T_{B-CFD, i} - \langle T_{B-CFD} - T_L \rangle.$$

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

## PandaRoot Simulation



$T_{OF}^0$

$T_{B-CFD}$



# 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  $\chi^2$  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  
 $\tau$ : Time difference to  $T_{OF}^0$   
S: Waveform content

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

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

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

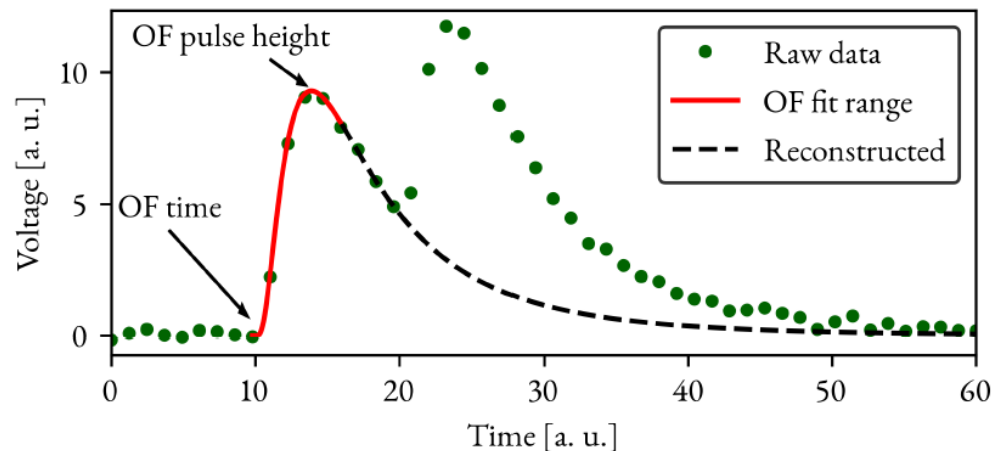
- The coefficients a and b can be analytically solved, which gives the **A** and  **$\tau$**
- The OF can provide an amplitude and a more accurate time as it used more information of the waveform

# 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

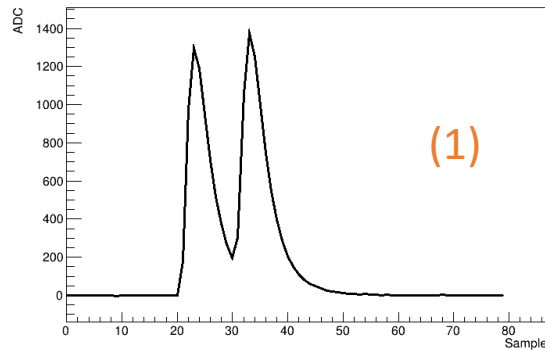


$B0 = -3$   
 $M = 4$

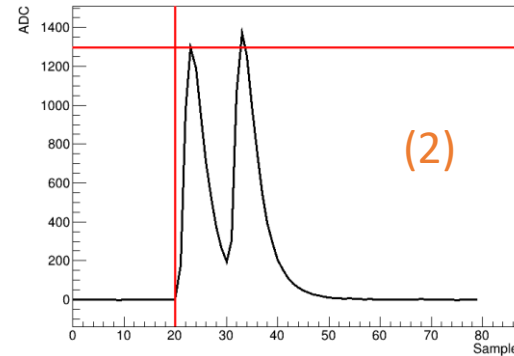
# PandaRoot simulation



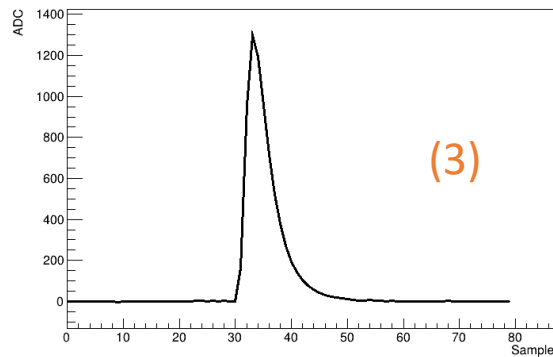
## PandaRoot Simulation



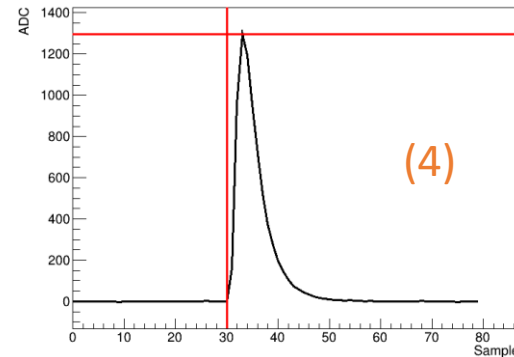
A pile-up waveform



First waveform detected

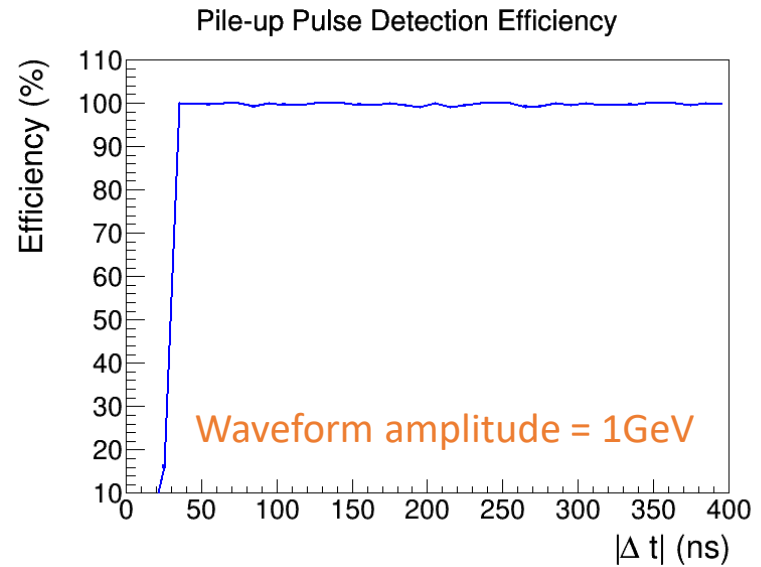
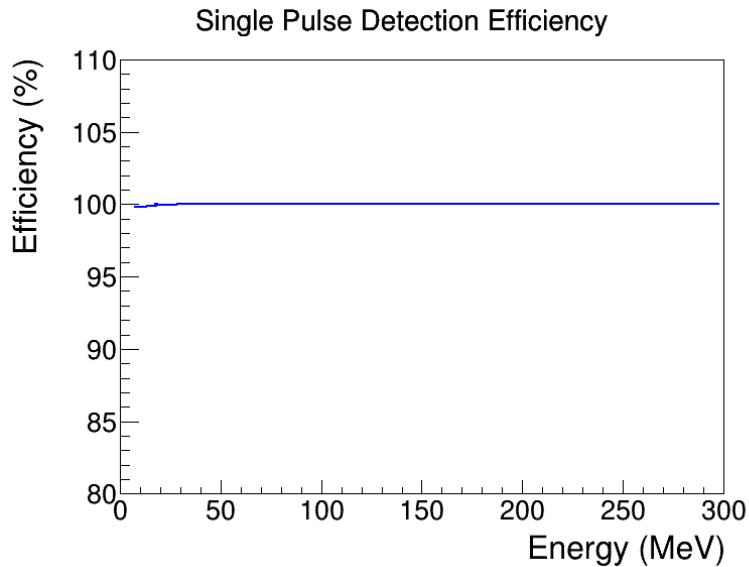


Baseline subtracted



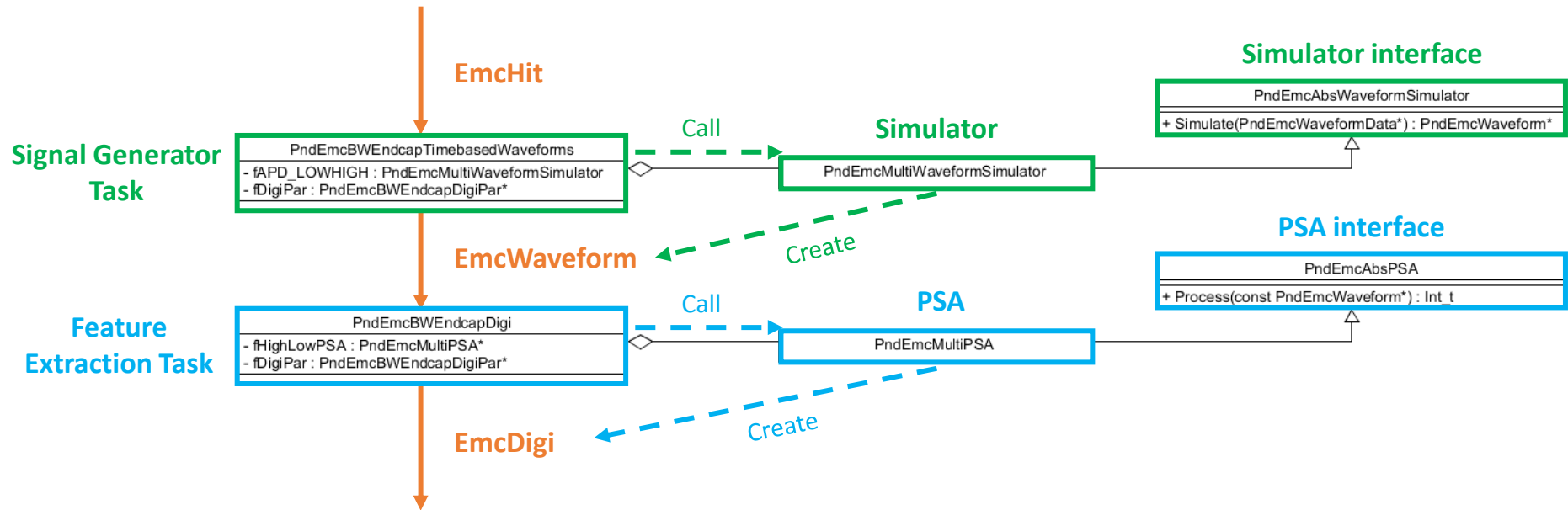
Second waveform detected

# Waveform detection efficiency in simulation



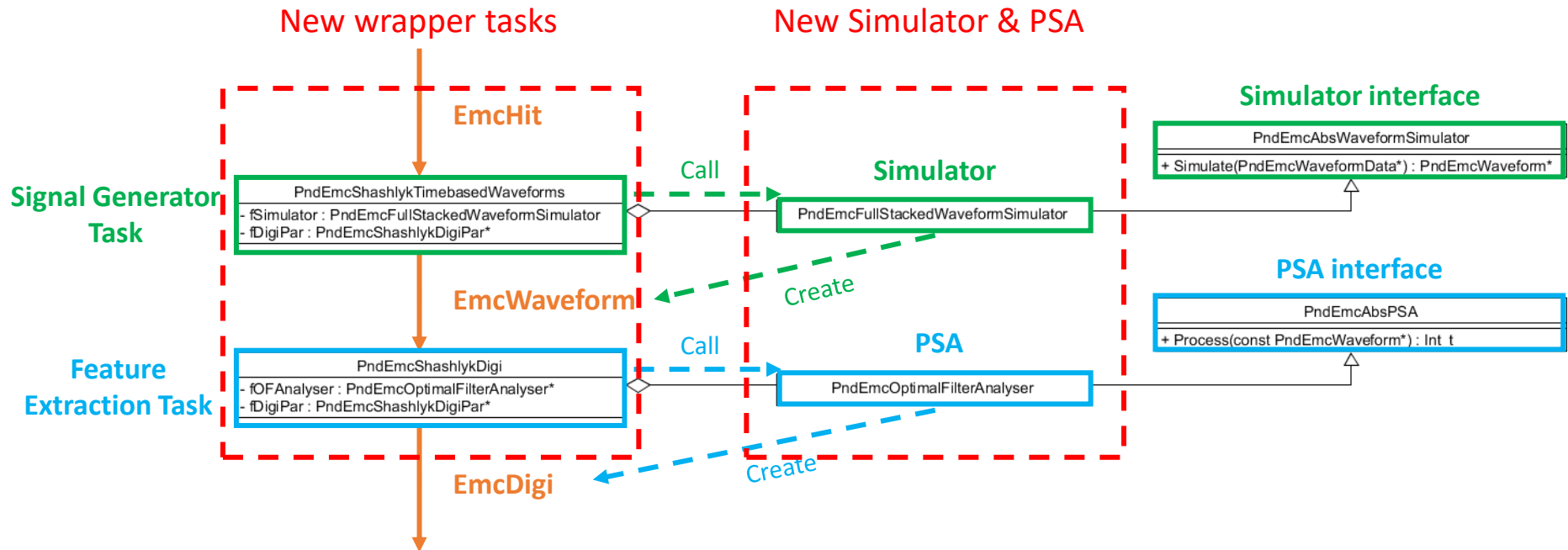
Reasonable pulse detection efficiencies

# Code Structure (BWEC)



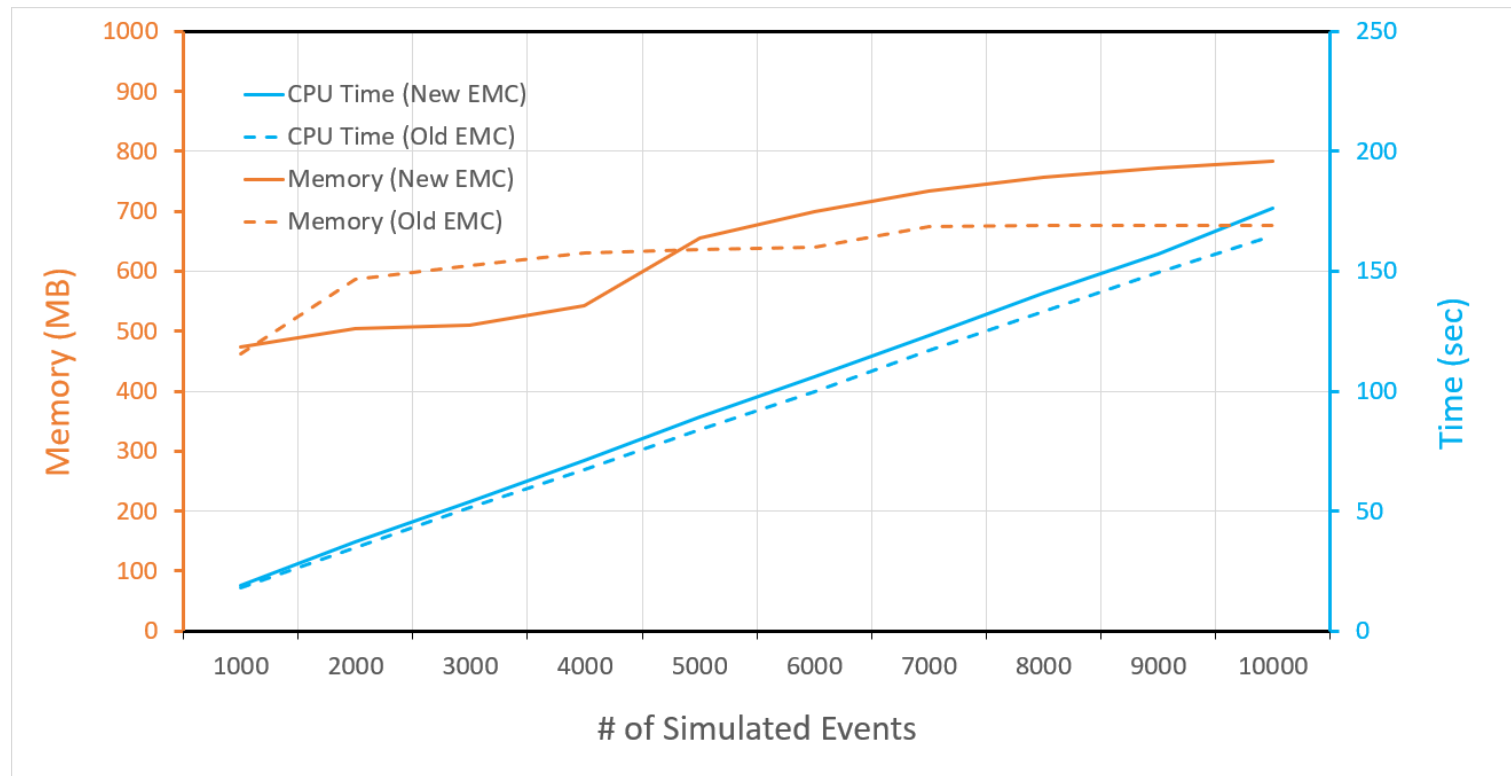
- Two tasks for signal generator and feature extraction respectively
- Simulator and Pulse Shape Analyzer (PSA) as the “algorithms”
- The algorithms inherit from the “interfaces”

# 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)

# Performance Test



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

# Summary

- **Have implemented Markus' work in PandaRoot, including**
  - Pulse generation using a shape template
  - Feature extraction using CFD+OF filters
- **Code is most ready**
  - Using the same framework as the bwec/barrel digitization
  - Key functions are modularized. Can be easy to migrate to the framework that are currently being developed by Ben
  - Need some final checks before checking in

Thank you!