

Thresholds/ADC Collection and Implementation

Kolja Kauder (BNL NPPS)

BG TF Meeting
October 31, 2023

Contacts and Values

- List of [github contacts](#) is all but complete
 - (missing: Low-Q2 Tagger (CAL), ZDC)
- [Values](#) are pretty much complete - but see later slides
 - (missing: Low-Q2 Tagger (CAL), ZDC (have values, need the technology choice), Roman Pots (no implementation))

Silicon-style detectors

- All PRs are created

 ElCrecon #1092
updated LOWQ2 threshold

 ElCrecon #1090
updated BTOF threshold

 ElCrecon #1091
updated ECTOF threshold

 ElCrecon #1089
updated MAPS threshold

 ElCrecon #1093
updated B0 LGAD threshold

Calorimeters

- Created one [PR for FEMC](#), to see if it answers questions from Gerard
- In the process, found that it's actually more complicated than I thought, and we'd actually need more parameters still
- Also, logic is not right, but there's no complete agreement on what would be correct → active discussion, considering inviting developers and DSCs to Thursday Simulation meeting

Calorimeters - Current digitization

Energy Summation:

// Make merge map, then

```
for (const auto &[id, ix] : merge_map) {
    for (size_t i = 0; i < ix.size(); ++i) {
        auto hit = simhits[ix[i]];
        edep += hit.getEnergy();
    }
}
```

Now energy cut: `m_cfg.threshold`;

And energy smearing: `m_cfg.eRes`;

```
if ( edep > m_cfg.threshold)
    eResRel = smeared edep
```

ADC conversion: `m_cfg.corrMeanScale`,
`m_cfg.dyRangeADC`, `m_cfg.capADC`,
`m_cfg.pedMeanADC`;

Also smeared: `m_cfg.pedSigmaADC`

```
ped = m_cfg.pedMeanADC +
m_normDist(generator) *
m_cfg.pedSigmaADC;
```

```
adc = std::llround(ped + edep *
(m_cfg.corrMeanScale + eResRel) /
m_cfg.dyRangeADC * m_cfg.capADC);
```

```
adc = min(adc, m_cfg.capADC)
```

Calorimeters - Current ADC

Zero suppression:
 m_cfg.pedMeanADC,
 m_cfg.thresholdFactor,
 m_cfg.pedSigmaADC,
 m_cfg.thresholdValue;

```
thresholdADC =
m_cfg.thresholdFactor *
m_cfg.pedSigmaADC +
m_cfg.thresholdValue;
for (const auto &rh: rawhits) {
    if (rh.getAmplitude() <
m_cfg.pedMeanADC + thresholdADC)
continue;
```

ADC to energy:
 m_cfg.pedMeanADC,
 m_cfg.capADC,
 m_cfg.dyRangeADC,
 m_cfg.sampFrac;

```
energy = ( rh.getAmplitude() -
m_cfg.pedMeanADC) /
m_cfg.capADC *
m_cfg.dyRangeADC /
m_cfg.sampFrac;
```

Calorimeters - What I think the logic should be

- convert to ADC
 - smear
 - apply adc cut
-
- In the reco class maybe have an optional additional energy cut

Intermediate Solution

- Implement all energy thresholds
- Set all ADC thresholds zero-suppression to 0
- This sets thresholds as indicated at first by the DSCs, but it does not reflect ADC fluctuation bringing a value above or below threshold

Supplementary slides