Towards a rate-based online MIP peak

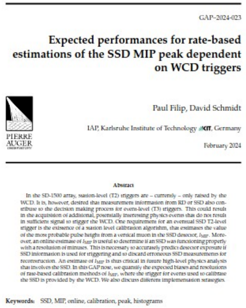
Paul Filip, David Nitz, Ricardo Sato, David Schmidt

Outline

- Recap
- Changes in LS software
- DAQ integration tests
- Summary and outlook

Recap

- GAP2024_023
  - Analyze muon histograms to derive rate-threshold for SSD shower buffer events, predict MIP with it
  - Caveats due to implicit dependence on WCD calib.


GAP–2024-023

Expected performances for rate-based estimations of the SSD MIP peak dependent on WCD triggers

Paul Filip, David Schmidt

IAP, Karlsruhe Institute of Technology KIT, Germany

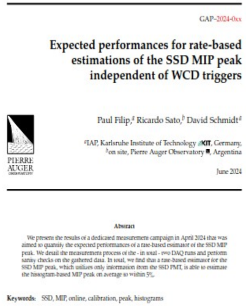
February 2024

Abstract

In the SD-1500 array, station-level (T2) triggers are – currently – only raised by the WCD. It is, however, desired that measurement information from RD or SSD also contribute to the decision making process for event-level (T3) triggers. This could result in the acquisition of additional, potentially interesting physics events that do not result in sufficient signal to trigger the WCD. One requirement for an eventual SSD T2-level trigger is the existence of a station level calibration algorithm, that estimates the value of the most probable pulse height from a vertical muon in the SSD detector, I_{MIP}. Moreover, an online estimate of I_{MIP} is useful to determine if an SSD was functioning properly with a resolution of minutes. This is necessary to accurately predict detector exposure if SSD information is used for triggering and to discard erroneous SSD measurements for reconstruction. An estimate of I_{MIP} is thus critical in future high-level physics analysis that involves the SSD. In this GAP note, we quantify the expected biases and resolutions of rate-based calibration methods of I_{MIP}, where the trigger for events used to calibrate the SSD is provided by the WCD. We also discuss different implementation strategies.

Keywords: SSD, MIP, online, calibration, peak, histograms

- GAP2024_0XX (pending...)
  - Run dedicated tests on Infill stations
  - WCD-independent SSD online calibration feasible
  - Error on rate-based MIP peak < 5% on average


GAP–2024-0xx

Expected performances for rate-based estimations of the SSD MIP peak independent of WCD triggers

Paul Filip,[a] Ricardo Sato,[b] David Schmidt[a]

[a]IAP, Karlsruhe Institute of Technology KIT, Germany,
[b]on site, Pierre Auger Observatory, Argentina

June 2024

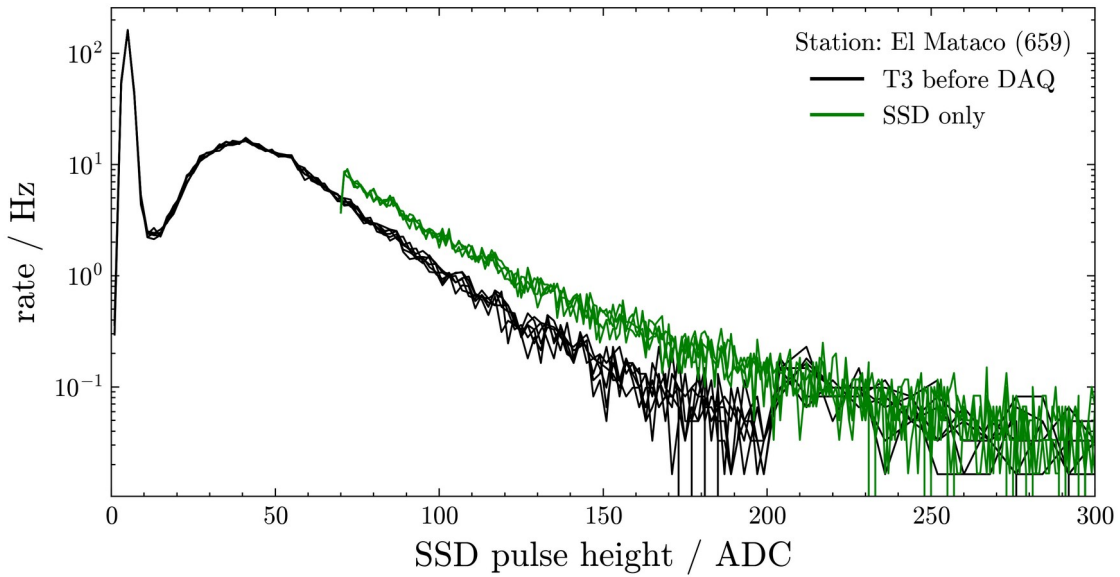
Abstract

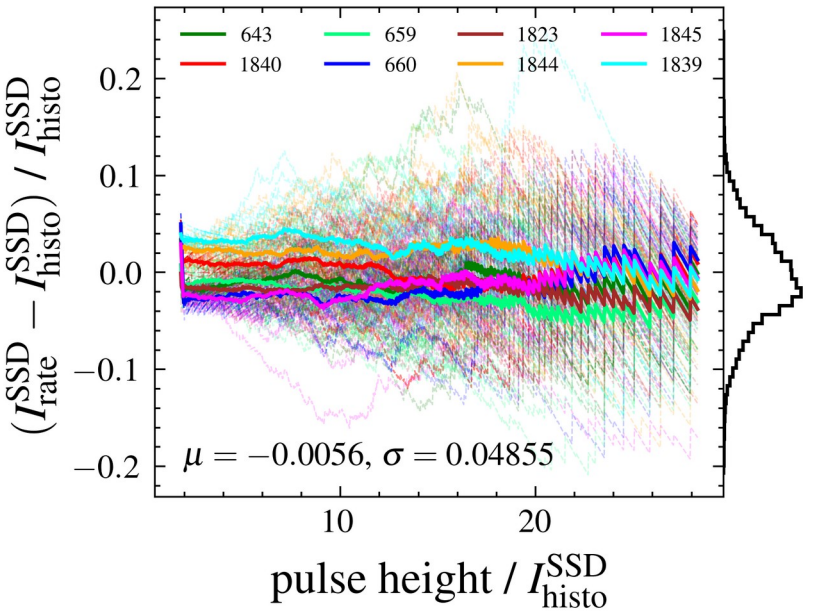
We present the results of a dedicated measurement campaign in April 2024 that was aimed to quantify the expected performances of a rate-based estimator of the SSD MIP peak. We detail the measurement process of the - in total - two DAQ runs and perform sanity checks on the gathered data. In total, we find that a rate-based estimator for the SSD MIP peak, which utilizes only information from the SSD PMT, is able to estimate the histogram-based MIP peak on average to within 5%.

Keywords: SSD, MIP, online, calibration, peak, histograms

Recap

- GAP2024_0XX (pending...)





Changes in LS software

```
Mem: 63880K used, 449204K free, 42000K shrd, 0K buff, 42372K cached
CPU:  0.5% usr  1.0% sys  0.0% nic 98.4% idle  0.0% io  0.0% irq  0.0% sirq
Load average: 0.19 0.07 0.06 2/62 23367
  PID  PPID USER     STAT   VSZ %VSZ CPU %CPU COMMAND
 6304  6303 root     S     2560  0.5   1  0.6 gpsctrl
   48     2 root     SW       0  0.0   0  0.3 [kworker/0:1]
23361 23354 root     R     2948  0.5   1  0.1 top
 6307  6303 root     S     135m 27.0   0  0.0 trigger2
 6306  6303 root     S     7088  1.3   0  0.0 feshwrread
23353   867 root     S     2952  0.5   1  0.0 telnetd -i
 6311  6303 root     S     2064  0.4   0  0.0 monitor
 6309  6303 root     S     130m 26.0   1  0.0 evtsvr
 1093     1 root     S     5988  1.1   1  0.0 /usr/sbin/tcf-agent -d -L- -l0
 6305  6303 root     S     3088  0.6   0  0.0 msgsvr
23354 23353 root     S     3048  0.5   0  0.0 -sh
  867     1 root     S     3028  0.5   1  0.0 /sbin/inetd
 6308  6303 root     S     2956  0.5   1  0.0 muonfill
 6312  6303 root     S     2924  0.5   0  0.0 spmt_cl
```

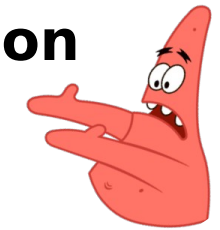
- **trigger2** process
  - reads **`ShowerBuffer`**
    (= WCD-T1 traces)
  - Handles T2s, etc …
  - **online calibration**

- **mounfill** process
  - reads **`MuonBuffer`** (now: 1 LMPT > 30 ADC)
  - Builds all histograms used for offline calibration

Changes in LS software

```
Mem: 63880K used, 449204K free, 42000K shrd, 0K buff, 42372K cached
CPU:  0.5% usr  1.0% sys  0.0% nic 98.4% idle  0.0% io  0.0% irq  0.0% sirq
Load average: 0.19 0.07 0.06 2/62 23367
  PID  PPID USER     STAT   VSZ %VSZ CPU %CPU COMMAND
 6304  6303 root     S     2560  0.5   1  0.6 gpsctrl
   48     2 root     SW       0  0.0   0  0.3 [kworker/0:1]
23361 23354 root     R     2948  0.5   1  0.1 top
 6307  6303 root     S     135m 27.0   0  0.0 trigger2
 6306  6303 root     S     7088  1.3   0  0.0 feshwrread
23353   867 root     S     2952  0.5   1  0.0 telnetd -i
 6311  6303 root     S     2064  0.4   0  0.0 monitor
 6309  6303 root     S     130m 26.0   1  0.0 evtsvr
 1093     1 root     S     5988  1.1   1  0.0 /usr/sbin/tcf-agent -d -L- -l0
 6305  6303 root     S     3088  0.6   0  0.0 msgsvr
23354 23353 root     S     3048  0.5   0  0.0 -sh
  867     1 root     S     3028  0.5   1  0.0 /sbin/inetd
 6308  6303 root     S     2956  0.5   1  0.0 muonfill
 6312  6303 root     S     2924  0.5   0  0.0 spmt_cl
```

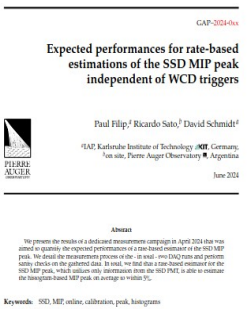
- **trigger2** process
  - reads **`ShowerBuffer`**
    (= WCD-T1 traces)
  - Handles T2s, etc …
  - **online calibration**



- **mounfill** process
  - reads **`MuonBuffer`** (now: 1 LMPT > 30 ADC)
  - Builds all histograms used for offline calibration
  - . . .

Changes in LS software

- Changes in **muonfill** program logic
  - Check for each trace & PMT if
    - It satisfies WCD T1 && PMT > 1.75 Peak (WCD)
    - It satisfies a SB trigger w/ PMT > 2.64 (SSD)
  - Add to T70 counter if conditions are met
  - Use T70 counter to set trigger thresholds


GAP-2024-0xx

Expected performances for rate-based estimations of the SSD MIP peak independent of WCD triggers

Paul Filip,[1] Ricardo Sato,[3] David Schmidt[1]

[1]IAP, Karlsruhe Institute of Technology (KIT, Germany; [3]on site, Pierre Auger Observatory, Argentina

June 2024

Abstract

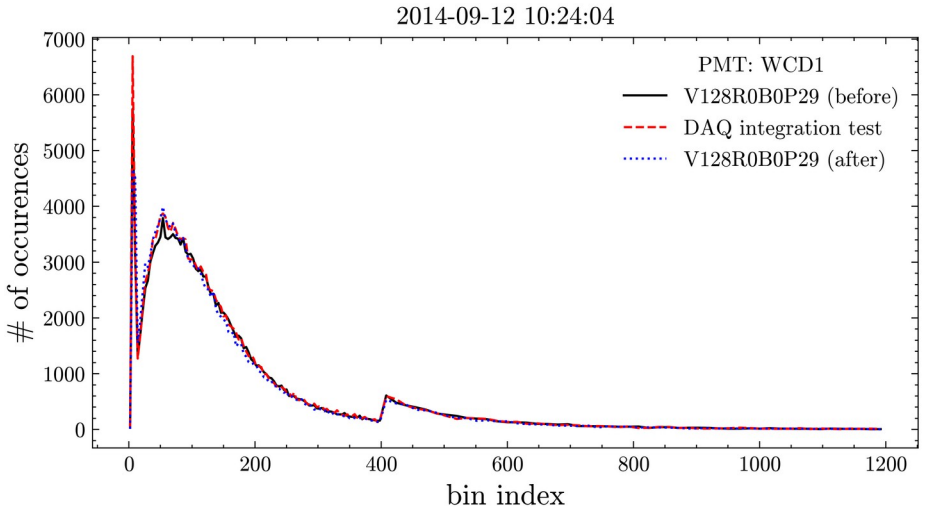
We present the results of a dedicated measurements campaign in April 2024 that was aimed to quantify the expected performances of a rate-based estimator of the SSD MIP peak. We detail the measurements process of the ~ in total ~ two DAQ runs and perform sanity checks on the gathered data. It is...

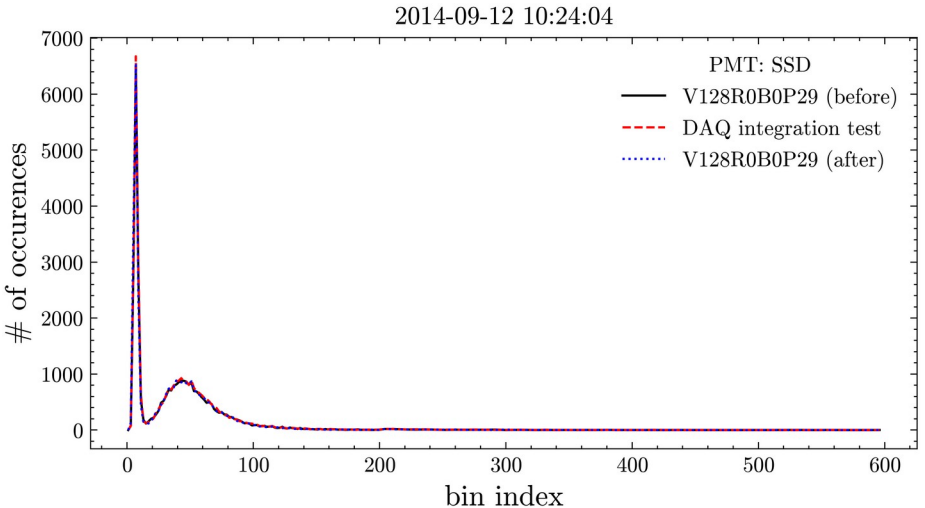
Keywords: SSD, MIP, online, calibration, peak, histograms

- Test new LS branch on Didi (136)
  - Verify histograms remain unchanged
  - Verify rate-based algorithm converges
  - Compare VEM estimates with **trigger2**

DAQ integration tests

- Test new LS branch on Didi (136)
  - Verify histograms remain unchanged (✓)
  - Verify rate-based algorithm converges
  - Compare VEM estimates with **trigger2**



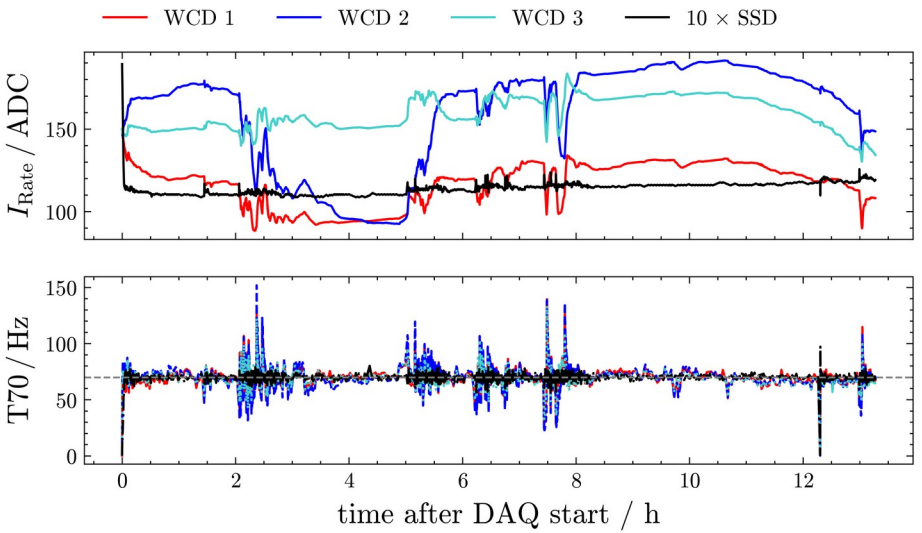


offsets sometimes wrong! Fix developed, but not yet tested!

DAQ integration tests

- Test new LS branch on Didi (136)
  - Verify histograms remain unchanged
  - Verify rate-based algorithm converges
  - Compare VEM estimates with **trigger2**

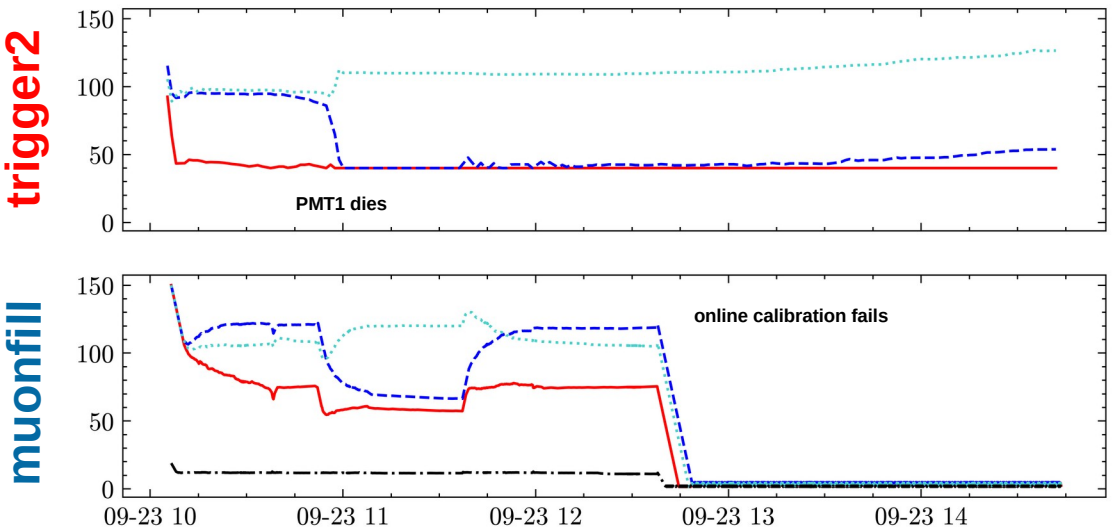
(✓)
✓



- Overnight integration test
- Recovery to 70 Hz rate even with drastic changes

DAQ integration tests

- Test new LS branch on Didi (136)
  - Verify histograms remain unchanged ✓
  - Verify rate-based algorithm converges ✓
  - Compare VEM estimates with **trigger2** ✗



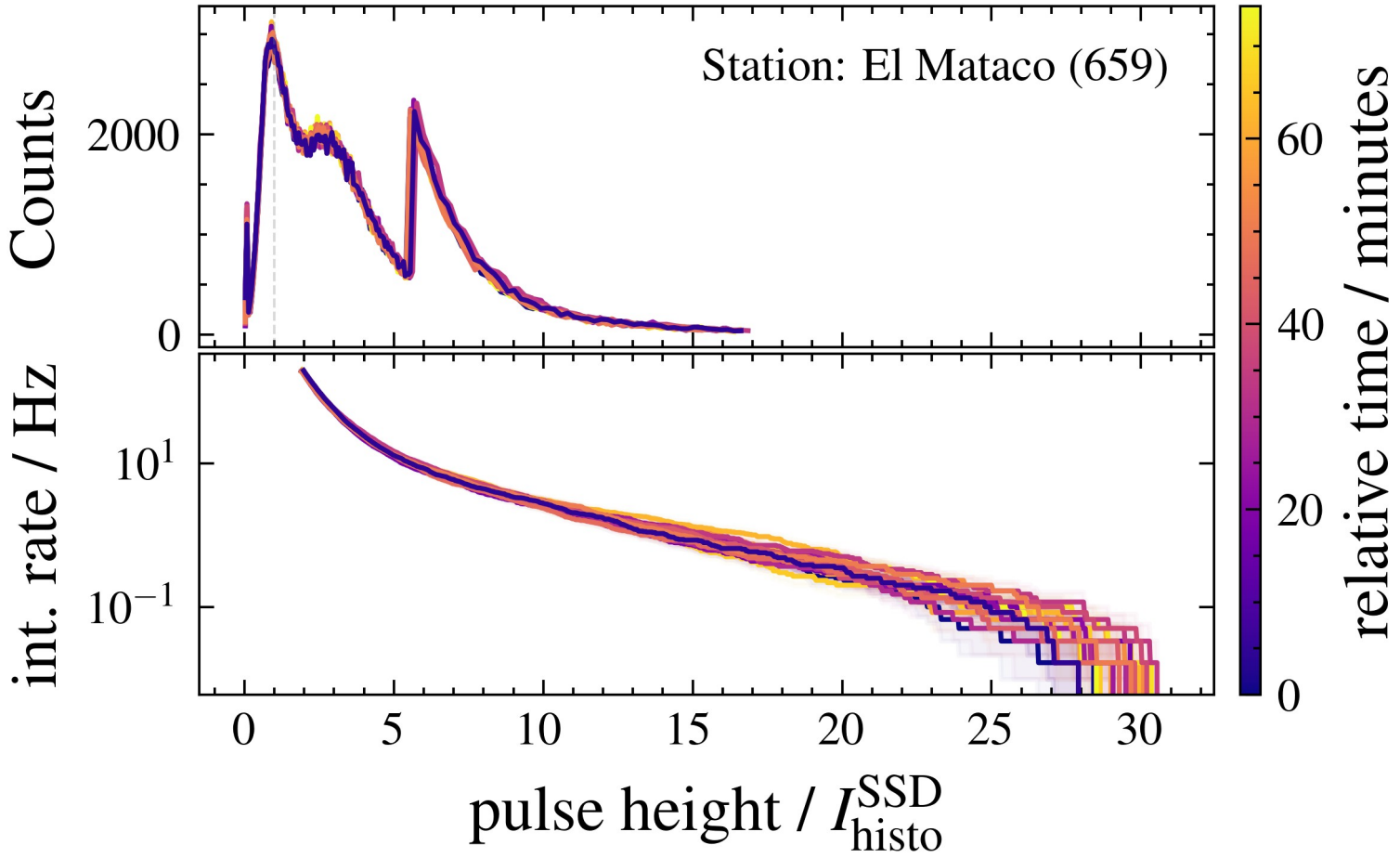
- Dynamic reading of WCD tube mask not implemented yet
- ~30% difference between VEM peak from **muonfill** & **trigger2**

  FBW ↔ filtered & downsampled?
- Problem with Didi PMTs?
- Further tests needed

Summary and outlook

- Have shown that rate-based online calib. of SSD possible
  - Independent of WCD; Expected errors < 5%
- Implemented online calibration in **muonfill** process
- DAQ integration tests with new LS software
  - Stable operation for > 14 hours (longer tests needed, though)
  - Histograms remain unchanged (in some cases offset is wrong)
  - Needs to be tested on stations in field at some point
  - **Work in progress**

Backup

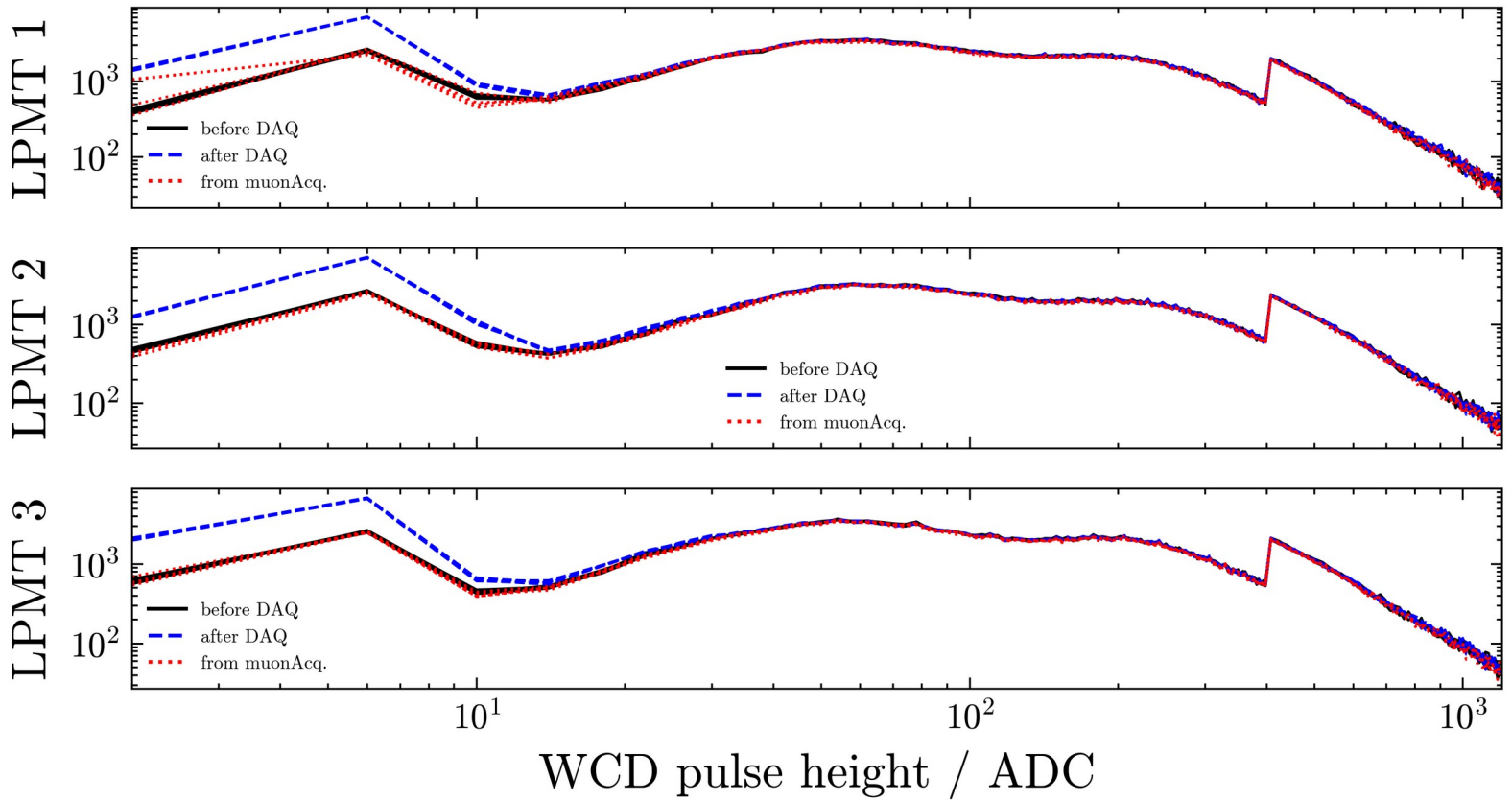

Station: El Mataco (659)
Counts
2000
0
int. rate / Hz
10^1
10^{-1}
0
5
10
15
20
25
30
pulse height / $I^{\mathrm{SSD}}_{\mathrm{histo}}$
60
40
20
0
relative time / minutes

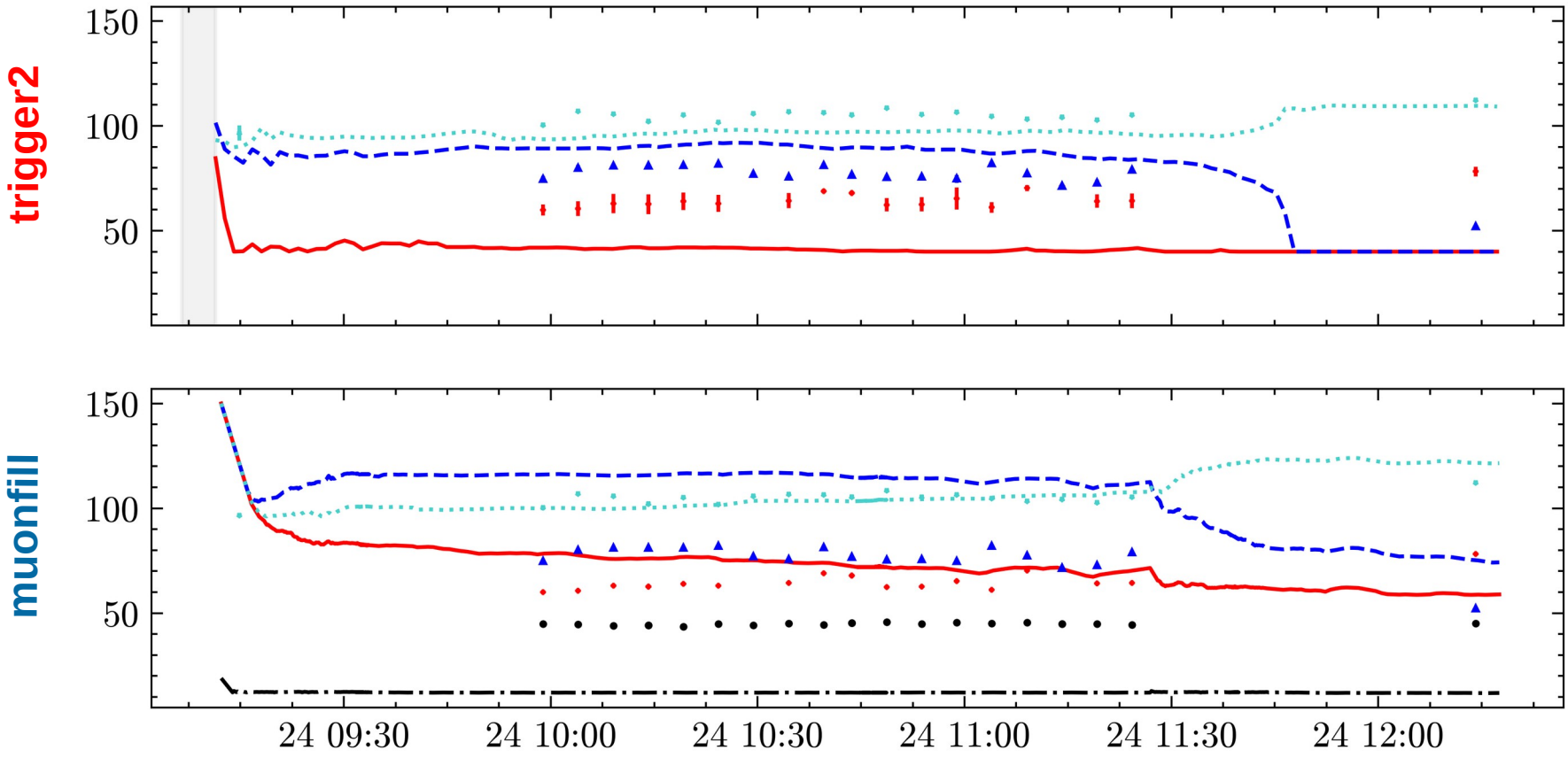
```c
void onlinecalib_init(struct muon_histo_complete *histo)
{
  // hardcoded, but can be changed
  // maybe do something like xbtrig
  histo->extra.peak[0] = 1500;
  histo->extra.peak[1] = 1500;
  histo->extra.peak[2] = 1500;
  histo->extra.peak[3] = 500;
  PrintLog(LOG_INFO, "histo->extra.peak[0] = %u\n", histo->extra.peak[0]);
  PrintLog(LOG_INFO, "histo->extra.peak[1] = %u\n", histo->extra.peak[1]);
  PrintLog(LOG_INFO, "histo->extra.peak[2] = %u\n", histo->extra.peak[2]);
  PrintLog(LOG_INFO, "histo->extra.peak[3] = %u\n", histo->extra.peak[3]);
```

```
/* perform online calibration and (re)set thresholds */
int all_within_limits = 0;
for (i=0; i<4; i++)
{
  int delta = CurrentHisto->extra.countsT70[i] / CurrentHisto->extra.dt_online - TThresh;
  int sign = delta < 0 ? -1 : 1;

  if (abs(delta) <= 2)                                          // have 70 +- 2 Hz rate
  {                                                             // --------------------------
    CurrentHisto->extra.adjust[i] = 1;                          // decrement calib_adjust by ~0.1 ADC
    all_within_limits += 1;
  }
  else                                                          // don't have 70 +- 2 Hz rate
  {                                                             // --------------------------
    if (abs(delta) > 20)
    {
      CurrentHisto->extra.adjust[i] = 20;                       // 70 +- >20 Hz -> calib increment = 2 ADC
      CurrentHisto->extra.dt_online = 10;                       // restart calib from 10s integration window
    }
    else if (abs(delta) > 10) CurrentHisto->extra.adjust[i] = 10;     // 70 +- >10 Hz -> calib increment = 1 ADC
    else if (abs(delta) > 5) CurrentHisto->extra.adjust[i] = 5;       // 70 +- > 5 Hz -> calib increment = 0.5 ADC
    else if (CurrentHisto->extra.adjust[i] > 1) --CurrentHisto->extra.adjust[i];
  }

  if (CurrentHisto->extra.peak[i] > CurrentHisto->extra.adjust[i] || sign > 0) CurrentHisto->extra.peak[i] += sign * CurrentHisto->extra.adjust[i];
}

if (all_within_limits == 4) CurrentHisto->extra.dt_online += 5;
if (CurrentHisto->extra.dt_online > gl.integrationInterval) CurrentHisto->extra.dt_online = gl.integrationInterval + 1;
```

```
for(pmt=0; pmt<4; pmt++){
  if(gl.ch_bin_min[pmt] <= 0){
    charge[pmt] = base[pmt];
  } else {
    charge[pmt] = 0;
  }
  peak[pmt] = base[pmt];
  bin_max[pmt] = 0;
  trace[pmt][0] = base[pmt];

  if (pmt < 3)
  {
    T100Threshold[pmt] = (TH100_WCD[flags->wcd_pmt_mask] * peak_estimate[pmt]) / 10.;
    T70Threshold[pmt] = (TH70_WCD[flags->wcd_pmt_mask] * peak_estimate[pmt]) / 10.;
  } else
  {
    T70Threshold[pmt] = (TH70_SSD * peak_estimate[pmt]) / 10.;
  }

  flags->is_T70[pmt] = 0;
}
```

```
int isThT1 = 0;
for(bin=1; bin<NBBIN; bin++ ){
  if((buff1[index + bin ] & 0x80000000) ||
     (buff2[index + bin ] & 0x80000000) ){
    return(2);
  }

  trace[0][bin]= buff1[index + bin]        & 0xFFF;
  trace[1][bin]=(buff1[index + bin] >> 16) & 0xFFF;
  trace[2][bin]= buff2[index + bin]        & 0xFFF;
  trace[3][bin]=(buff2[index + bin] >> 16) & 0xFFF;

  int thT1_multiplicity = 0;
  for(pmt=0; pmt<4; pmt++){
    if(gl.ch_bin_min[ pmt ] <= bin &&
       bin < gl.ch_bin_max[ pmt ]    ){
      charge[pmt] += trace[pmt][bin];
    }

    if( peak[pmt] < trace[pmt][bin] ){
      peak[pmt] = trace[pmt][bin];
      bin_max[pmt] = bin;
    }

    if (pmt==3 || isThT1 || thT1_multiplicity < pmt) continue;
    if (trace[pmt][bin] > base[pmt] + T100Threshold[pmt]) thT1_multiplicity += 1;
  }

  if (thT1_multiplicity == 3) isThT1 = 1;
}

for (pmt=0; pmt<4; pmt++)
{
  if (pmt < 3 && !isThT1) continue;
  flags->is_T70[pmt] = peak[pmt] > base[pmt] + T70Threshold[pmt];
}
```


LPMT 1
LPMT 2
LPMT 3
10^3
10^2
before DAQ
after DAQ
from muonAcq.
10^1
10^2
10^3
WCD pulse height / ADC


trigger2
muonfill
150
100
50
150
100
50
24 09:30
24 10:00
24 10:30
24 11:00
24 11:30
24 12:00