

Contents

1 Classical station triggers

As mentioned in ??, continuously analyzing data sent to CDAS from each of the 1600 SD water tanks would quickly exceed the computational capabilities of Augers' main servers. For this purpose, trace information is only collected from a station, once a nearby T3 event (c.f. ??) has been detected. The formation of a T3 trigger is dependant on several T2, or station-level, triggers, which will be discussed in detail in this chapter. First, the implementation of different trigger algorithms is discussed in ?. Their performance is evaluated in ?.

1.1 Implementation

1.1.1 Threshold trigger (Th)

The **Threshold** trigger (Th) is the simplest, as well as longest operating trigger algorithm [triggerGuide] in the field. It scans incoming ADC bins as measured by the three different WCD PMTs for values that exceed some threshold. If a coincident exceedance of this threshold is observed in all three WCD PMTs simultaneously, a Th-T1/2 trigger is issued. A pseudocode implementation of this algorithm is hence given by the below code block.

```
1  th1 = 1.75      // Th1 level threshold above baseline, in VEM
2  th2 = 3.20      // Th2 level threshold above baseline, in VEM
3
4  while True:
5
6      pmt1, pmt2, pmt3 = get_next_output_from_WCD()
7
8      if pmt1 <= th2 and pmt2 <= th2 and pmt3 <= th2:
9          raise Th-T1_1trigger
10     else if pmt1 <= th1 and pmt2 <= th1 and pmt3 <= th1:
11         raise Th-T2_trigger
12     else:
13         continue
```

Logically, with increasing signal strength S in the PMTs, the likelihood of having observed an extensive air shower raises. This is reflected in the trigger level logic, where a coincident signal of $S \leq 3.20 \text{ VEM}_{\text{Peak}}$ is immediately forwarded to CDAS, whereas a signal $1.75 \text{ VEM}_{\text{Peak}} \leq S < 3.20 \text{ VEM}_{\text{Peak}}$ only raises a Th-T1 trigger. The algorithm is insensitive to signals that do not exceed at least $1.75 \text{ VEM}_{\text{Peak}}$ in all three PMTs.

In the case of faulty electronics, where only a subset of the WCD PMTs are available, the trigger thresholds (in units of VEM_{Peak}) are updated according to ??.

Table 1.1: Numerical values from [triggerSettings]

| n_{PMT} | Th-T2 | Th-T1 |
|------------------|-------|-------|
| 1 | 5.00 | 2.85 |
| 2 | 3.60 | 2.00 |
| 3 | 3.20 | 1.75 |

1.1.2 Time over Threshold trigger (ToT)

The Time over Threshold trigger (ToT) is sensitive to much smaller signals than the Threshold trigger discussed in ??. For each PMT in the water tank, the past 120 bins are examined for values that exceed $0.2 \text{ VEM}_{\text{Peak}}$. If 13 or more bins above the threshold are found in the window - ordering or succession do not matter - the PMT is considered to have an elevated pedestal. The ToT trigger requires at least two PMTs with an elevated pedestal in order to activate. As such, the algorithm is theoretically sensitive to events that deposit just $0.5 \text{ VEM}_{\text{Ch}}$. A pseudocode example is given below.

```

1  threshold    = 0.2  // pedestal threshold, in VEM
2  n_bins       = 12   // number of bins above pedestal
3  window_size  = 120  // considered window length
4
5  buffer_pmts  = [[False for i in 1..window_size] for j in 1..3]
6  step_count   = 0
7
8  while True:
9
10     pmts = get_next_output_from_WCD()
11     buffer_index = step_count % window_size
12     count_active_PMTs = 0
13
14     for pmt, buffer in pmts, buffers:
15         if pmt <= threshold: buffer[buffer_index] = True
16
17         if count_values(buffer, value = True) > n_bins:
18             count_active_PMTs += 1
19
20     if count_active_PMTs >= 2:
21         raise ToT-T2_trigger
22     else:
23         step_count = buffer_index + 1
24         continue

```

1.1.3 Time over Threshold deconvoluted trigger (Totd)

An extension to even lower signal strengths is given by the **ToT-deconvoluted trigger** (Totd). As the name implies, the implementation of the algorithm is completely analog to the ToT trigger in ???. Only the FADC input stream from the three PMTs is altered according to ???.

$$d_i = (a_i - a_{i-1} \cdot e^{-\Delta t/\tau}) / (1 - e^{-\Delta t/\tau}) \quad (1.1)$$

In ??, the deconvoluted bin d_i is calculated from the measured FADC values a_i and a_{i-1} , where a_{i-1} is scaled according to an exponential decay with mean lifetime $\tau = 67$ ns. This reduces the exponential tail of an electromagnetic signal to a series of pulses which in the case of $a_{i-1} < a_i$ exceed the original signal strength. As such, the deconvoluted trace can satisfy the ToT trigger requirements, whereas the original raw FADC values might not have, extending the sensitivity of the ToT trigger to lower signal strengths. The scaling constant $\Delta t = 25$ ns is tied to the sampling rate of UB electronics (c.f. ???). The choice of the numerical constants τ and Δt is explained in more detail in [ToTtriggerIdea].

1.1.4 Multiplicity of Positive Steps (MoPS)

The **Multiplicity of Positive Steps** (MoPS) algorithm triggers on positive flanks of an FADC trace, which can be related to the arrival of new particles in the water tank.

A positive flank in the FADC trace of a single PMT is any combination of at least three bins that are monotonically increasing in value. Once such a positive step has been identified, a (MoPS) trigger veto is applied to the next

$$n_{\text{skip}} = \lfloor (\log_2(\Delta y) + 1) - 3 \rfloor \quad (1.2)$$

bins, where Δy refers to the total vertical increase in the step from first to last bin. Note that in ??? the notation $\lfloor x \rfloor$ is used as shorthand notation to round x to the nearest integer. If Δy is bigger than $y_{\min} = 3$ ADC (to filter random fluctuations), but does not exceed $y_{\max} = 31$ ADC (to prevent triggering on muonic coincidences), it is added to a ledger. If the number of rising flanks in the ledger is bigger than $m > 4$ for at least two PMTs, a final check regarding the integral of the FADC trace is performed. If this check passes, a MoPS-T2 trigger is issued to CDAS.

It is impossible to accurately recreate the MoPS trigger in simulations. The integral test above compares the sum of the last 250 bins against a threshold ($\sum a_i > 75$). Since not all 250 bin values are available to CDAS, differing results are to be expected when comparing the implementation of the algorithm in the SD field versus its' counterpart in analysis software.

For this purpose, the MoPs trigger is not considered in the analysis presented in ???. The implications of this choice are laid out in ???.

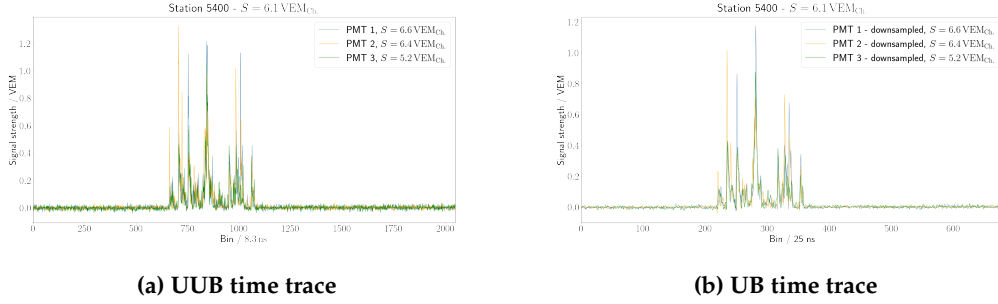


Figure 1.1: (a) A simulated signal as it would appear to UUB electronics. The ionizing particles originating in the extensive air shower hit the tank around bin 660 ($\approx 5.5 \mu\text{s}$). (b) The same signal but filtered and downsampled to emulate UB electronics.

1.1.5 Compatibility mode

Although the triggers discussed in the previous subsections are meant to function completely autonomously in the SD field, their implementation requires some prior knowledge of the signal one desires to detect. For their use in the Auger observatory, several hyperparameters such as the thresholds of the Th-Trigger, or the window size of the ToT-trigger have been determined in studies ([bertou2006calibration], [triggerSettings], [ToTtriggerSetting]).

These studies were conducted using the predecessor, the Unified Board (UB), of the hardware that is being installed during the AugerPrime upgrade of the observatory. Most importantly, the Upgraded Unified Board (UUB) has a sampling rate that is three times larger (120 MHz) than that of UB electronics (40 MHz). Not only does this raise the number of bins in a standard time trace from 682 to $2^{11} = 2048$, but also drastically reduces the efficiency (in particular for ToT-like triggers) of the above discussed algorithms. Whereas a new FADC bin is measured every 25 ns in a UB station, the triggers would receive a new input every $\approx 8.3 \text{ ns}$ in a UUB setting.

The modus operandi elected by the Pierre Auger collaboration to circumvent this problem is to emulate UB electronics using the UUB electronics. This means that measured FADC bins are to be filtered and downsampled before any trigger runs over them. Software implementations by which this is achieved are listed in ?? . The effect the filtering and downsampling has on measured data is visualized in ?? .

[to do: comment on accuracy fo this method]

1.2 Performance

The performance of a trigger can be evaluated in many different ways. In the most general consideration, a confusion matrix holds information about the ability of a classifier to discern between different types, or classes, C . With the example at hand

there exist two types of events one wishes to distinguish, a signal event C_1 in the form of an extensive air shower, versus background C_0 . The confusion matrix thus becomes:

| | | Predicted C | |
|----------|-------|---------------------|---------------------|
| | | C_1 | C_0 |
| True C | C_1 | True positive (TP) | False negative (FN) |
| | C_0 | False positive (FP) | True negative (TN) |

From this, other potentially interesting variables can be derived. Of particular interest for the Auger observatory are the sensitivity and False Discovery Rate (FDR). The former is the probability that a signal event will be classified correctly, i.e. an extensive air shower hits a water tank and raises a T2 trigger. The sensitivity - in the following also called the trigger efficiency ϵ - is defined as

$$\epsilon = \frac{TP}{TP + FN}. \quad (1.3)$$

The latter is a measure of how readily the triggers (wrongly) identify background events like stray cosmic muons as extensive air showers. It is imperative for any trigger algorithm operating in the SD to minimize this probability. Simply due to the number of operating stations in the field, a small increase in FDR drastically raise the amount of potential events and hence load on the central analysis server of the observatory.

$$FDR = \frac{FP}{TP + FP}. \quad (1.4)$$

[to do: continue writing about performance]

2 Neural network training data

Over their relatively brief existence, neural networks have been shown to perform increasingly impressive tasks (e.g. [openai2019dota], [openai2023gpt4], and many more). However, they learn by example. The performance of a neural network is directly linked to the input data it receives during training. If the training data is not an accurate example of real world information a network later operates on, insight gained from it is at best an approximation, and at worst completely randomly generated data.

As such, it is not a question *if* some neural network architecture can learn to identify an extensive air shower from WCD data, but rather which implementation, fed with which information, does. For this purpose, this chapter states the procedure with which training data is generated. As stated above, this must occur with a focus on being representative of data actually measured in the SD array.

3 Performance of neural network triggers

3.1 Convolutional neural networks

3.2 Recurrent neural networks