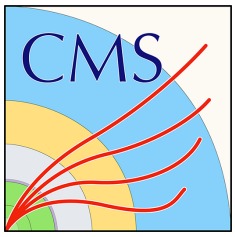


Set up calibration processes for the CMS High-Granularity Detector

Summer student: Alessandro de Robertis
Supervisors: Chiara Amendola, Pedro Viera De Castro Ferreira Da Silva

---

Abstract

The CMS experiment is building a high-granularity calorimeter (HGCAL) to replace the existing endcap calorimeters during the High-Luminosity phase of LHC. Calibration processes must be effectively implemented to ensure meticulous monitoring of the detector in order to achieve the best possible quality of the collected data. The aim of this report is to present the contribution I made toward this objective during the summer school period.

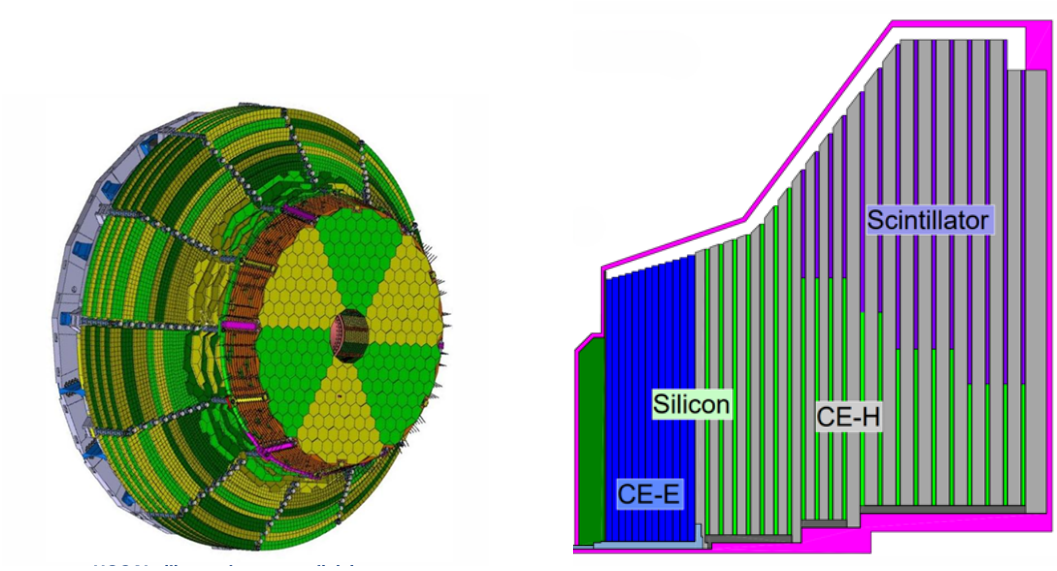
---

1 Detector Overview

The High-Luminosity Large Hadron Collider (HL-LHC) project aims to increase the potential for discoveries after 2029. The purpose is to enhance the integrated luminosity by a factor of 10 beyond the LHC's design value. The instantaneous luminosity, planned to be set at $5 \times 10^{34} cm^{-2} s^{-1}$, will correspond to a mean number of collision per bunch crossing (pileup) equal to $140 - 200$. In comparison, during LHC Run 3, the number of pileup has reached up to 80.

The HL-LHC environment will pose challenges for radiation tolerance and event pileup on detectors, for example for calorimetry in the forward region. As part of its HL-LHC upgrade program, the CMS Collaboration is building a high-granularity

calorimeter (HGCAL) to replace the existing endcap calorimeters. In order to withstand the high radiation levels, silicon sensors were chosen as active material for the bulk of the upgrade of the endcap calorimeters. The detector will consist of an innermost electromagnetic compartment (CE-E) and an outermost hadronic compartment (CE-H) made of both silicon sensors and scintillators as active media (Fig.1).



(a) (b)

Figure 1: (a) HGCAL 3D vision (colors are used to differentiate tiling and cassettes divisions). (b) Longitudinal cross section of the upper half of one endcap calorimeter.

Each endcap will be 5.4 m in diameter and weight about 230 tonnes. The entire detector is designed with fine longitudinal resolution. The electromagnetic part (CE-E) has thin absorber layers of lead and CuW interleaved between the active layers. The hadronic part (CE-H) has thick stainless-steel absorbers between sensors. In order to reduce the electronic noise, the full volume of the detector will operate at $-35°$C. The high segmentation of the detector results in excellent resolution in energy, space, and time. The time resolution is further enhanced by the use of fast silicon-based sensors and front-end electronics designed for rapid performance. For this reason the detector belongs to the category of the often-called "5D detectors". The high granularity of the detector, and so the high space-time precision, is essential to separate different interactions and pileup in the High-Luminosity environment.

Both Endcaps	**Silicon**	**Scintillator**
Area	$\sim 620m^2$	$\sim 370m^2$
Channel size	$0.5 - 1.2cm^2$	$4 - 30cm^2$
# Channels	$\sim 6M$	$\sim 280k$
# Modules	$\sim 26k$	$\sim 4k$

Per Endcap	**CE-E**	**CE-H**
Absorber	Pb, CuW, Cu	SS, Cu
Depth	$27.7X_0$	10λ
Layers	26	7 (Si) + 14 (Si + Scint.)
Weight	23 t	205 t

Table 1: Summary of important proprieties of endcaps

1.1 Silicon module overview

The silicon sensors for the CE-E and the inner parts of the CE-H are planar DC-coupled hexagonal silicon sensors fabricated on 8 inch (8") wafers. The hexagonal shape of the sensors makes more efficient use of the available area of the circular wafers. Indeed, studies confirmed that, due to more complete use of the silicon wafers, deploying hexagonal sensors would result in a substantial cost reduction. Sensors will have three different active thicknesses (300, 200 and 120 μm) in order to optimize the charge collection and operation conditions over the full lifetime of the HGCAL. Each 8 inch hexagonal module has a "three-fold diamond" configuration and is tiled into symmetric cells. The wafer is built in two configurations by considering two sizes of sensor cells. Low-density modules are characterized by 1.18 cm^2 sensor cells and the high-density modules by 0.52 cm^2 sensor cells (Fig.2).

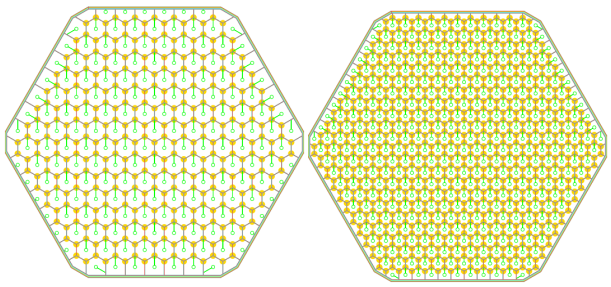


Figure 2: Illustration of the two types of hexagonal wafer modules: low-density (left) and high-density (right).

A single silicon module is composed by a CuW baseplate, essential for the cooling system and also contributes to absorber, a kapton-gold layer, that provides HV bias connection to the sensor and insulation from the baseplate, a silicon sensor and a printed circuit board (PCB) also called *hexaboard* (Fig.3).

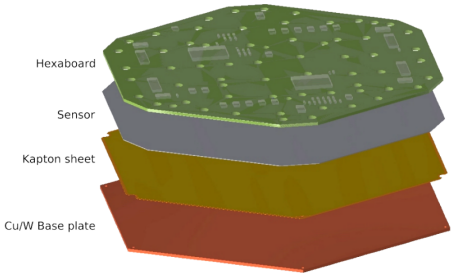


Figure 3: Sketch of the layers of a silicon module.

The electromagnetic compartment consists of 26 sampling layers. Each plane of this structure is subdivided into 60° units called cassettes. Each cassette is double-sided with modules, so 13 layers of these *cassettes* provide the full 26 sampling layers.

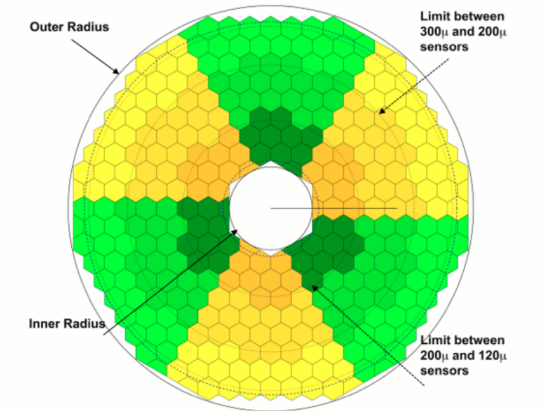


Figure 4: Layout of one layer of CE-E where only silicon sensors are present. The division into six 60° cassettes is shown by the alternating colours. The two radial changes in darkness of colour indicate the changes in silicon thickness.

1.2 Hexaboard

The PCB or hexaboard has the role of front-end electronics (FE). Silicon cells are connected to the input of the ASIC HGCROC, which measures the charge deposited and time of arrival (ToA) at 40 MHz frequency. It also provides, by summing digital signal of neighbouring cells, information to build trigger primitives.

The HGCROC has 78 channels (72 reading out standard cells, 2 reading out calibration cells, and 4 channels not connected to any sensor cells for common-mode (CM) noise estimation). When the charge deposited in the sensor is above $\sim 50fC$ a time over threshold (ToT) technique is used extending the dynamic range up to $10pC$.

A low-density module contains 3 HGCROC and has 192 *Si* cells. On the other hand an high-density module has 432 *Si* cells managed by 6 chips.

Data are transmitted over two paths: one for the standard data acquisition (ADC, ToA and ToT) and one for the trigger decision (digital ADC sums).

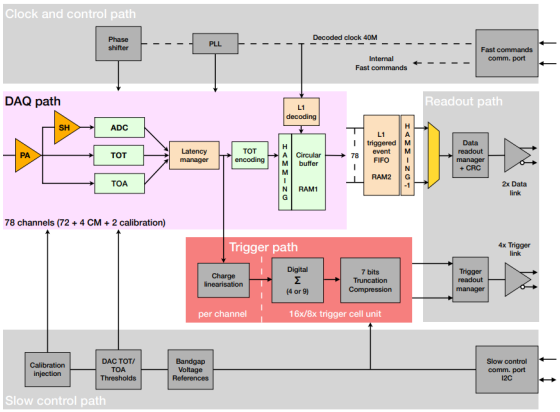


Figure 5: Sketch of the HGCROC logic

2 Single-module calibration

During the HL-LHC the higher instantaneous luminosity will lead to an increase of the aging effects. This is one of the many reasons why the frequency of the required calibrations will need to be increased.

Prompt Calibration Loop (PCL) are currently used in CMS to calculate calibration corrections on basis of *Express stream data* and provide them within a time window of

48 hours to Prompt reconstruction of the same run. Despite PCLs have been a success during LHC Run-1 and Run-2, different approaches were proposed during the LHC Long Shutdown 2. The idea is to make use of new software tools to automatically execute the calibrations workflows on a daily basis during the data taking. The same idea want to be used in HGCAL.

Calibration and data-taking of a single Hexaboard were carried out using an FPGA called single-module tester through which all the scripts were run. The single module calibration for lab tests is divided into several steps. Various variables, such as the DC level of the shaper, the phase parameter, the injection value and the DC levels of the ToA and ToT DACs, are scanned to achieve uniform pedestal values across channels and to verify the correct behavior of the module. When the module is intended for use in a beam test, other scans need to be considered, which are not discussed here. An almost complete discussion of all steps of the single module calibration for lab tests is discussed in this presentation [1].

3 Validation

As mentioned before, in a generic prompt calibration, right after the data are collected, a dedicated stream, called express stream, is first processed, providing a quick feedback about the detector status, physics performance and yielding the input for the calibration workflows. My work was to check if the calibration of a single module was reasonable or not, flag modules or cells that present anomalies and highlight the reasons for these anomalies.

I had access to the calibration data of six low-density hexaboard modules. The calibration of these modules led to a single JSON file containing: pedestals (*ADC_ped*), common-mode pedestals (*CM_ped*), the slope of the regression line comparing the data coming from CM pedestals and reading out channel pedestals (*CM_slope*) and noise per channel (*Noise*).

I was able to plot all of these variables using the pyROOT framework (example in Figure 6)

In order to understand why a channel or the entire module should be flagged and how anomalies should be spotted, a selection can be made of the pedestal of each channel or the mean pedestal among channels related to entire module. The original idea is showed in Figure 7.

During the calibration, the ADC value per channel is measured many times. The

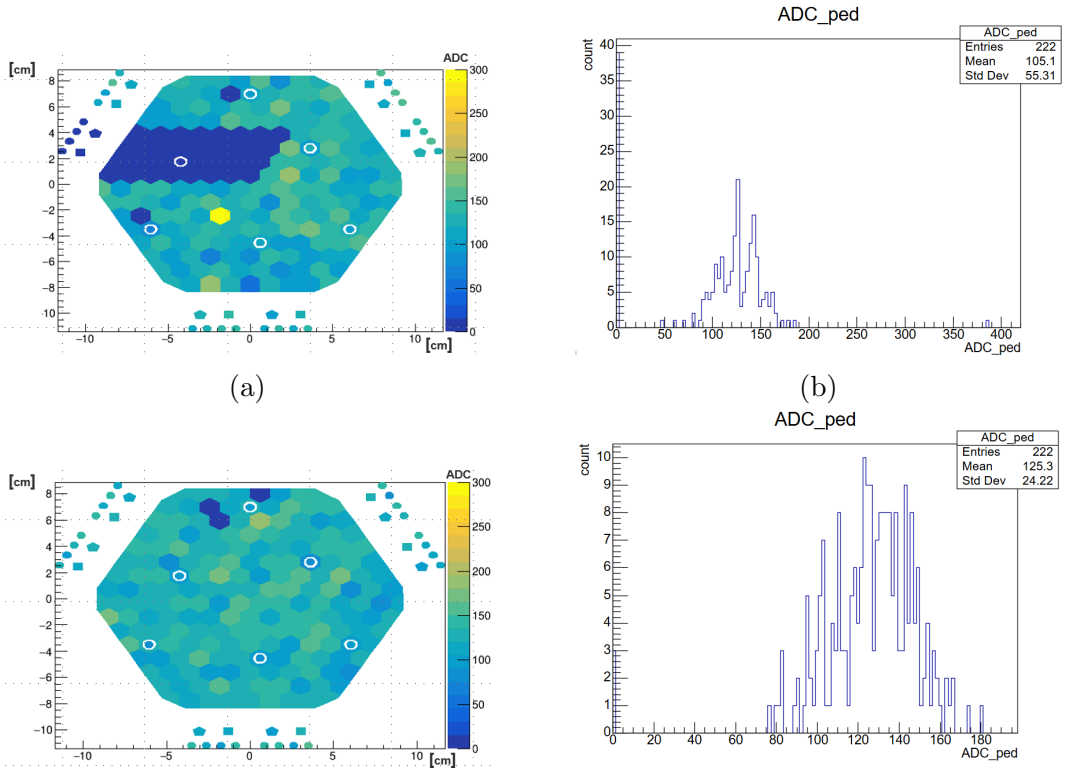


Figure 6: (a), (c) Hexplot of 2 low-density modules. The color scale represents the magnitude of the ADC pedestal for each channel. From the Hexplot it can be noted that for every chip there are 4 Common Mode cells, 2 calibration cells (depicted as circle on the module) and 8 pins not connected to any sensor (depicted as circles on the side of the hexagon). (b), (d) Histograms corresponding to ADC pedestal for each channel. It could be noted that one of the two halves of an HGCROC is broken, for the first module. This leads to a bigger value of RMS and a lower value of the ADC pedestal.

mean of the distribution and the RMS[1] are extracted and stored in the JSON file as respectively the *ADC_ped* and *Noise*. For each module a distribution among channels of these two parameters can be plotted (as depicted in Figure 7). The idea is to compute the mean of the histograms and flag the module only if that value is out of a specific interval. The same selection has been used for the RMS of the same

[1]in this document it is used the abuse of notation according to which RMS and Std.Dev. have the same definition

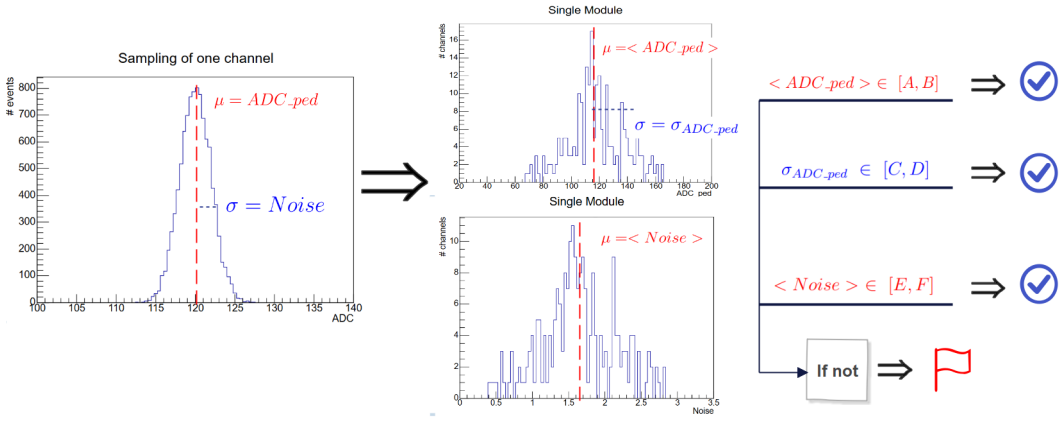


Figure 7: Sketch of the flagging idea.

distributions. If at least one of these four conditions is not satisfied, a flag is placed on the module to narrow down the number of modules that require inspection with higher granularity.

However if the entire module is flagged, it can be useful understand the reason of this malfunction. The idea now is to scan all the ADC_ped and $Noise$ values per channel comparing these values with the "ideal" ones. Assuming that we already know the expected pedestal for each channel (ADC_ped_{ref}) it can be defined the Δ_{ped} as:

$$\Delta_{ped} = ADC\_ped_{ref} - ADC\_ped \tag{1}$$

In the ideal case, the random variable Δ_{ped} is about zero (due to statistical fluctuation). But if this is not true it could mean that the channel is not working well. For example if a channel is broken and returns a zero value of the ADC, the variable Δ_{ped} will assume a value far from zero. The same it would be obtained if, for some reason, a channel returns an ADC value bigger than expected. The identical idea could be exploited by introducing another "flag" variables that is Δ_{Noise}.

$$\Delta_{Noise} = Noise_{ref} - Noise \tag{2}$$

where $Noise_{ref}$ is the reference value of the Noise for a single channel. So, if one of these scenarios occur the channel could be flagged.

3.1 Selection and definition of acceptance region

In the previous section it has been discussed how the flag of a module happens when a parameter is inside or outside a specific interval. In this section, definitions of these

intervals will be presented.

3.2 Module flagging

As depicted in Figure 7, from the ADC_ped and Noise distributions we can extract four variables: $< ADC\_ped >$, σ_{ADC_ped}, $< Noise >$ and σ_{Noise}. Let's assume that there are reference values for each of these variables. For instance, the $< ADC\_ped >$ should fluctuate according to a gaussian distribution. Since the reference value ($< ADC\_ped >_{ref}$) is assumed to be the expectation value of this distribution, it could be a good choice to define the "acceptance" interval of $< ADC\_ped >$ as:

$$[\ < ADC\_ped >_{ref} \pm \alpha \ \ \sigma^{ref}_{ADC\_ped}\ ] \tag{3}$$

where α is a free parameter that could be set in order to increase or decrease the range of the interval and $\sigma^{ref}_{ADC\_ped}$ represents the reference value of the variable σ_{ADC_ped}. It could be noted that if α is set equal to one, the interval has the usual probabilistic meaning of confidence interval with a confidence level of 68%.

Following this idea, if the $< ADC\_ped >$ of the module is not in that interval a flag will be attached to the module, otherwise the module will be considered as "good".

The same idea was used also for the other variables. The $< Noise >$ interval has the same structure of Eq.(3):

$$[\ < Noise >_{ref} \pm \alpha \ \ \sigma^{ref}_{Noise}\ ] \tag{4}$$

However σ_{ADC_ped} and σ_{Noise} variables do not distribute as a gaussian variables but as a χ^2 random variables, so the intervals would be in the form of :

$$\left[\ \sqrt{\frac{(n-1)s^2}{\chi^2_{\frac{1-\beta}{2}}}}, \sqrt{\frac{(n-1)s^2}{\chi^2_{\frac{\beta}{2}}}}\ \right] \tag{5}$$

where n is the number of measurements (in this case the number of "good" modules, this will be explained later), s is the sample Std. Dev. ($\sigma^{ref}_{ADC\_ped}$, σ^{ref}_{Noise} were taken as s), $\chi^2_{\frac{1-\beta}{2}}$ and $\chi^2_{\frac{\beta}{2}}$ were obtained using the *scipy* library (both of them were determined starting from a confidence level that was set as default equal to $\beta = 0.7$).

3.3 Scan over channels

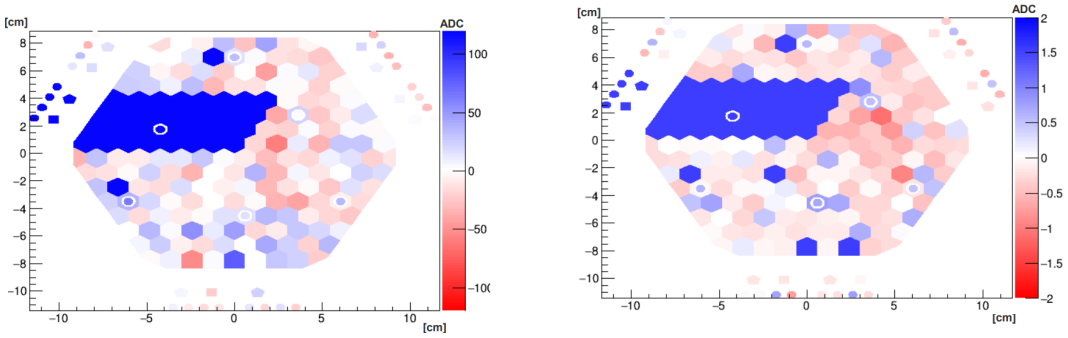
Once it has been established which modules are away from the reference behaviour, a scan over channels of these modules has to be done. As mentioned before, Δ_{ped} and Δ_{noise} variables are introduced as Eq (1), (2). It is clear that ADC_ped_{ref} and $Noise_{ref}$ are the same quantity introduced in the previous section where they are called $< ADC\_ped >_{ref}$ and $< Noise >_{ref}$.

Instead of just flag the channel, in the calibration workflow, it could be helpful visualize directly through an *Hexplot*. This is exactly what has been done as it can be seen in Figure 8.

In addition another flag variable could be defined to easily spot channels which have pedestals far from the reference value.

$$Pull_{ped} = \frac{\Delta_{ped}}{< Noise >_{ref}} \tag{6}$$

By default in the script if one channel of a bad module has $Pull_{ped} > 1$, it will be flagged as a "bad" channel.



(a) (b)

Figure 8: (a) Hexplot of the Δ_{ped} flag variable for the first Module inspected showed in Figure 6 (a). (b) Hexplot of the Δ_{Noise} of the same module. In both cases, if the cell is depicted as red means that its Noise or ADC_ped is bigger than the reference value, otherwise if it is blue it means that Noise or ADC_ped is lower than expected.

4 Initialization

In the previous sections the reference values were assumed to be known. In this section it will be explained how to get these values.

If we know the set of *flagged* modules from the previous Validation step, one possibility is to define the four reference values by computing the mean of all "good" modules from the previous Validation step. Regarding the two RMSs, the mean of variances was computed and then it was taken the square root. Using this definition of reference values, now it should be clear why in Eq.5 n is exactly the number of "good"modules coming from the previous Validation step.

One of the problem of this idea is the Initialization. The algorithm takes as an input the set of "bad"modules coming from the previous Validation step. We don't know a priori which are the "good" modules and which are the "bad"modules at the beginning. However we can follow guidelines to identify good modules from the start.

First of all the distribution of noise of a "good" module must be consistent with that expected from the sensor capacitance and the pre-amplifier gain. Moreover, the leakage current is expected to increase due to ageing effect and so will the noise. In addition, the distribution of pedestals should have minimal dispersion, consistent with the noise. Finally, accumulating data from several modules will soon tell us how the real system behaves and how these margins should be updated.

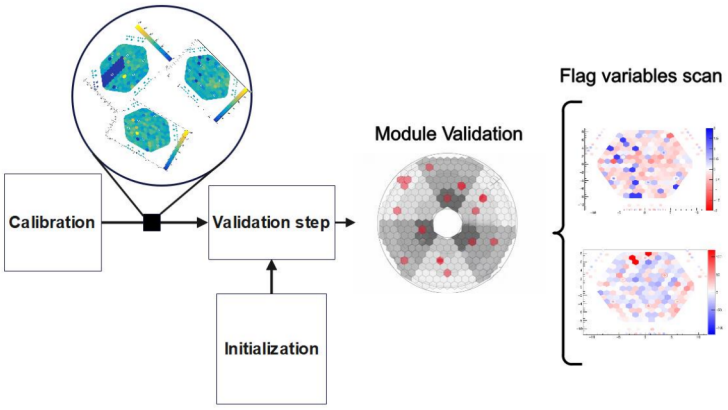


Figure 9: Data coming from the Calibration step are used in the Validation step to flag non-working well modules. After that, some hexplots, showing flag variables trend of channels in each flagged module, are produced.

5 Conclusion

Precise calibration and monitoring of the HGCAL detector will be a key ingredient in achieving the excellent HGCAL performance required by many physics analyses. My contribution fits perfectly in the idea of automatic execution of calibration workflows during data taking. The algorithm takes as an input the set of *flagged* modules from the previous Validation step and then it returns a new set of *flagged* modules of the current step. Then, each "bad" module is inspected by computing a *flag variables scan*, during which hexplots of these flag variables are plotted in order to easily spot malfunctions of the modules.

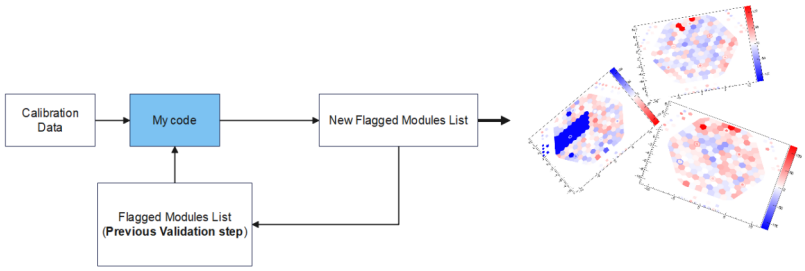


Figure 10: The algorithm takes as an input the list of modules flagged from the previous Validation step and returns the new one.

6 Appendix

6.1 Gaussian assumption

During this report it was assumed that some variables, such as ADC_ped, would be distributed as Gaussian variables. In the plot Fig.11 you can see how the chi-square of the Gaussian fit is consistent with this assumption.

6.2 Reference values

The four reference values discussed above were computed by taking the mean of the same variable of all "good" modules from the previous step. However, in order to

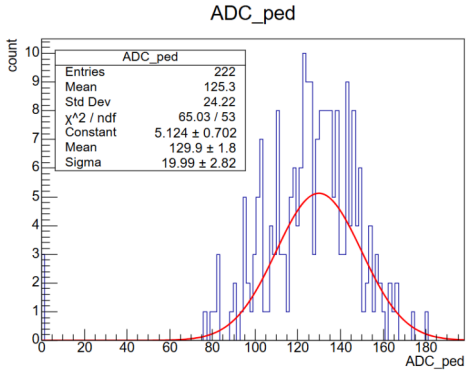


Figure 11: ADC_ped distribution for a "good" module. The red line represents the Gaussian fit, and the fit parameters are shown at the top left of the plot.

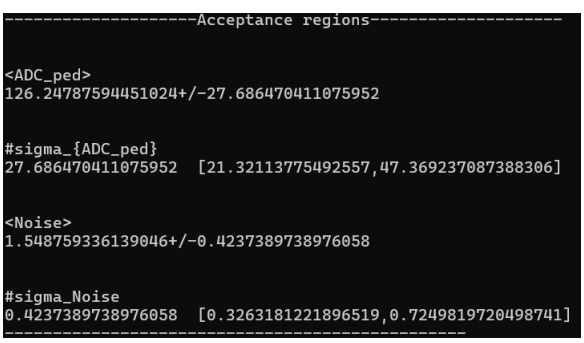
get a more accurate analysis from the statistical point of view, a weighted average should be taken into account (the following could be done also for the *Noise*).

$$< ADC\_ped >_{ref} = \sum_{i=1}^{n} w_i < ADC\_ped >_i \tag{7}$$

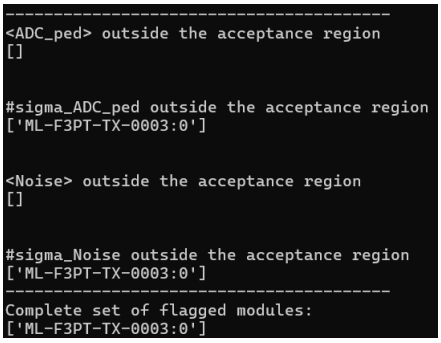
$$w_i = \frac{\frac{1}{\sigma^2_{ADC\_ped_i}}}{\frac{1}{\sigma^2}} \qquad \frac{1}{\sigma^2} = \sum_{i=1}^{n} \frac{1}{\sigma^2_{ADC\_ped_i}}$$

6.3 Script output

As discussed in the previous sections, once the calibration of a module is completed, a JSON file is produced. Then my script returns a set of Flagged Modules starting from the previous one. I analyzed data of six Modules and, since it was not possible to recalibrate them, the script should have returned the same list of "bad" modules that is given as an input. However this is not trivial and, in order to accomplish that, a wise choice of "bad" modules from the start and a right tuning of parameters α and β must be carried out.



(a)



(b)

Figure 12: One among six modules was flagged ("by eye") from the start. (a) Reference values and confidence interval of all variables discussed in the previous sections are displayed ($\alpha = 1$ and $\beta = 0.7$). (b) Display of flagged modules. As expected, the same module that was provided as input is flagged in the output.

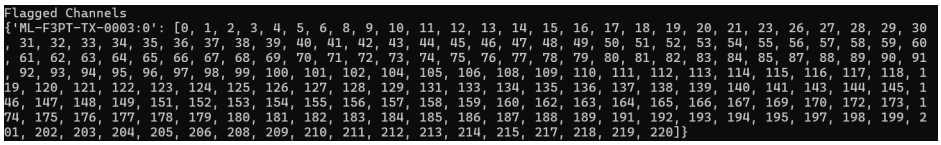


Figure 13: The script returns also a dictionary whose keys are the flagged modules and whose items are the number of channels which satisfy the $Pull_{ped} > 1$ condition.

6.4 Hexplots

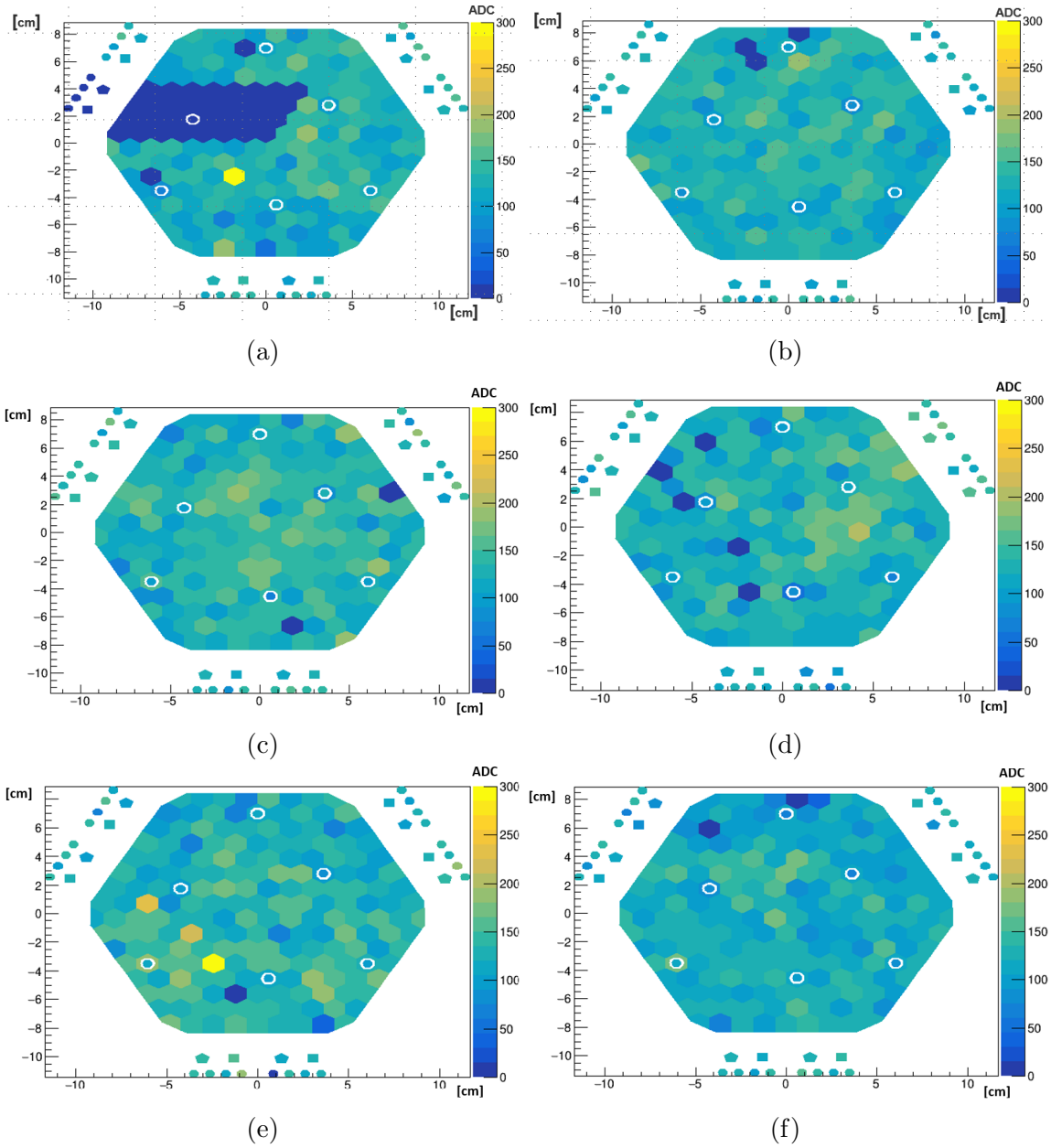


Figure 14: All the hexplots of the six modules under study are displayed. The color scale represents the value of ADC pedestal for each channel.

References

[1] https://cernbox.cern.ch/s/9iqZDAIBpLf8135.

[2] The Phase-2 Upgrade of the CMS Endcap Calorimeter. Technical report, CERN, Geneva, 2017.

[3] Cms Collaboration et al. The cms experiment at the cern lhc. *Journal of instrumentation*, 3(August 2008):1–334, 2008.

[4] Frederic Dulucq. HGCROC3: the front-end readout ASIC for the CMS High Granularity Calorimeter. TWEPP 2021 Topical Workshop on Electronics for Particle Physics. 2021.

[5] Gabriele Milella, CMS Collaboration, et al. An overview of the cms high granularity calorimeter and its current status. In *European Physical Society Conference on High Energy Physics*, number PUBDB-2023-06482. LHC/CMS Experiment, 2023.

[6] Christophe Ochando and on behalf of the CMS Collaboration. Hgcal: A high-granularity calorimeter for the endcaps of cms at hl-lhc. *Journal of Physics: Conference Series*, 928(1):012025, nov 2017.