Realistic calorimeter hit digitisation in the `ILDCaloDigi` processor

Daniel Jeans, The University of Tokyo,
Oskar Hartbrich, DESY.

March 11, 2015

Abstract

Description of quasi-realistic calorimeter hit digitisation in ILD, as implemented in the `ILDCaloDigi` processor in the `MarlinReco` package.

1 Introduction

Possibilities for more realistic treatment of calorimeter hits from silicon– and scintillator–based calorimeters have been implemented in the `ILDCaloDigi` processor within `MarlinReco`. The aim of these is to allow the study of the effects of various detector "defects" such as mis-calibrations, limited dynamic ranges, and signal fluctuations, and also to allow more robust comparisons between technologies under more realistic conditions. This notes discusses the implementation as defined by rev. 4777 of the `MarlinReco` package, available at `https://svnsrv.desy.de/viewvc/marlinreco/MarlinReco/`.

The energy of `SimCalorimeterHits` produced by the `Mokka` simulation is the energy deposited in the detection element (silicon or scintillator) as calculated by `GEANT4`. They therefore take account of Landau fluctuations in the energy deposits. The role of the digitisation is to simulate the behaviour of the process which converts this energy deposit into the reconstructed energy of hits used in detector reconstruction: this includes effects due to the detection medium (e.g. creation of electron-hole pairs in silicon), the read-out system (e.g. pixelated photo-detectors (PPD - SiPM/MPPC)) used to

1

readout scintillation light, and the electronics which treat these signals (e.g. limited dynamic range).

This note describes a parameterised model which to model such effects. The parameters to be used should be decided by the detector groups, by comparing to data collected during test beam campaigns.

`ILDCaloDigi` also applies a threshold on the energy of hits, as well as possibilities for requirements on the timing of energy deposits. This pre-existing functionality was extended from the original `ILDCaloDigi` on some points.

2 General

The various effects implemented for realistic digitisation are controlled by parameters of the `ILDCaloDigi` processor which can be set at run time via the `Marlin` steering file (i.e. without recompiling). Most parameters are duplicated, with one set for the ECAL, and the other for the HCAL, allowing different parameters to be used for the each of the systems. In this note, we denote there duplicated parameters as "*§CAL*": "§" should be replace by "E" for the ECAL, and "H" for the HCAL.

Which type of digitisation to apply to ECAL hits is controlled by the `ECAL_apply_realistic_digi` parameter: a value of 0 (the default) turns off the realistic effects described in this note, and a value of 1 (2) applies the silicon– (scintillator–) specific effects. In the case of the scintillator HCAL, the parameter `HCAL_apply_realistic_digi` plays a similar role, with a value of 0 (1) turning off (on) the simulation of realistic digitisation effects.

Several digitisation parameters are specified in terms of MIP units, so the `ILDCaloDigi` processor requires factors with which to convert the deposited energy (in GeV) to MIP units: these are passed by the `Calib§CALMIP` parameters.

Various detector parameters are taken from the `gear` file, in particular the layer layout and number of virtual cells per scintillator strip in the case of a scintillator strip-based ECAL. If these are not available in the `gear` file, they can be specified via the parameters `ECAL_default_layerConfig` and `StripEcal_default_nVirtualCells` (if values *are* found in the `gear` file, they take precedence over the value of these parameters).

3 Timing and energy thresholds

Mokka provides a list of single energy deposition contributions from Geant4, including deposited energy in GeV and time stamp in ns for each active detector volume (cell or strip). To increase realism of the simulation, cuts on energy and timing of these hits can be applied.

3.1 Energy threshold

Hits in once detector cell are only kept after the digitisation stage if the hit amplitude is above a threshold. This threshold can be configured by the §`calThreshold` steering parameters. The unit used for interpreting the given threshold is chosen by setting §`CALThresholdUnit` to either `GeV` for deposited energy in GeV (default), `MIP` for deposited energy in MIP equivalents, or `px` for deposited energy in SiPM pixels. `MIP` and `px` require appropriate settings of the digitisation parameters explained in section 6.

3.2 Hit timing

It is imperative to run the preceding Mokka simulation with `/Mokka/init/lcioDetailedShowerMode true`. Otherwise, detailed subhit information will not be stored to the output file, which is necessary for hit timing selections to work properly (see also [3]).

Two different algorithms for hit timing selection can be chosen via §`CALSimpleTimingCut`.

If §`CALSimpleTimingCut` is `true`, all hit contributions with hit time between §`CALTimeWindowMin` and §`CALBarrelTimeWindowMax`/§`CALEndcapTimeWindowMax` are summed into the digitised hit. The hit time of the output is set to the earliest subhit within that time window. This timing mode resembles test beam measurements with the CALICE physics prototypes using external trigger and a fixed acquisition time window.

If §`CALSimpleTimingCut` is `false`, a more complex algorithm is applied. It aims to cluster hits in time into windows of §`CALDeltaTimeHitResolution` and generates individual output hits for each of these time clusters. This may be useful to simulate more complex aqcuisition electronics, but needs very careful setup of parameters. Especially the SPIROC chip currently used in the CALICE AHCAL and ScECAL prototypes cannot be well modeled by

this. Also with the current implementation, it is possible to lose contributions within the general time window defined by §CALTimeWindowMin and §CALTimeWindowMax. It is thus advised to use §CALSimpleTimingCut true at the moment.

4   Technology-blind effects

4.1   Mis-calibrations

The effect of imperfect energy calibrations can be simulated by the use of the parameters §CAL_miscalibration_uncorrel and §CAL_miscalibration_correl, which causes hit energies to be smeared as

$$E' = E \times$$
$$\text{RandGauss}(1, \S\texttt{CAL\_miscalibration\_uncorrel}) \times$$
$$\text{RandGauss}(1, \S\texttt{CAL\_miscalibration\_correl}),$$

where $\text{RandGauss}(\mu, \sigma)$ represents a random number taken from a Gaussian distribution of mean μ and standard deviation σ. In the case of §CAL_miscalibration_uncorrel, a new random number is taken for each calorimeter hit (simulating completely uncorrelated mis-calibrations), while in the case of §CAL_miscalibration_correl, a single random number is used for all §CAL hits in a given event (simulating completely correlated mis-calibrations).

The uncorrelated miscalibrations induced by §CAL_miscalibration_uncorrel of each detector cell can be chosen to be either the same from event to event, or newly chosen for each event. This is controlled by setting the parameter §CAL_miscalibration_uncorrel_memorise = true or false respectively. The first approach is closer to reality, however in the case of a calorimeter with many cells, can lead to large memory consumption; in the case of typical physics events randomly spread across the whole ILD detector, the second, more memory-efficient, approach is almost certainly sufficient. The first approach is probably only necessary in the case of repeated injection into the same detector region, as occurs, for example, in test beams.

4.2 Dead detector cells

The effect of dead detector cells can be simulated by use of the parameter §CAL_deadCellRate, which causes the energy of hits to be set to zero if a random number taken from a uniform distribution in the range [0, 1] is smaller than the value of the §CAL_deadCellRate parameter. Similar to the miscalibration effects, the parameter §CAL_deadCell_memorise selects whether dead cells are randomised for each event or constant for the whole run.

4.3 Dynamic range of readout electronics

The saturation of the readout electronics can be simulated by setting the parameter §CAL_maxDynamicRange_MIP, in which case the energy of the hit is limited to this value of this parameter (specified in MIP units):

$$E'_{MIP} = min(\S\texttt{CAL\_maxDynamicRange\_MIP}, E_{MIP}).$$

4.4 Noise

Uncorrelated, random noise can be simulated by the parameter §CAL_elec_noise_mips, which alters the hit energy by

$$E'_{MIP} = E_{MIP} + \texttt{RandGaus}(0, \S\texttt{CAL\_elec\_noise\_mips}).$$

5 Silicon ECAL hits

A rather simple approach is followed in the case of silicon readout: the energy deposit in the silicon is converted into a number of electron-hole (e-h) pairs, using the parameter energyPerEHpair which gives the energy required to create an e-h pair (in eV). This number of e-h pairs is then used to define the mean of a Poisson distribution, from which a random number is taken to get a statistically smeared number of e-h pairs. This approach is an over-simplification: it ignores, for example, the Fano effect which reduces the fluctuation of e-h pairs with respect to this simple Poisson approximation. Since these fluctuations are anyway much smaller than the Landau fluctuations in energy deposit, they have an almost negligible effect.

6 Scintillator hits

In the case of the scintillator §CAL, several effects are included: non-uniformity of response along the strip length, and the finite number of photo-electrons and PPD pixels. These processes typically have much larger effects than in the case of silicon-based readout. For the scintillator-based HCAL, the same effects, except the strip non-uniformity, are included. The names of the relevant parameters for the HCAL have "`ECAL_`" replaced by "`HCAL_`".

Non-uniformity along strip length

In Mokka simulation, ECAL scintillator strips can be split along their length into virtual cells, in each of which a `SimCalorimeterHit` can be produced. ILDCaloDigi identifies all virtual-cell hits coming from the same strip, and combines them into a single `CalorimeterHit`. Different weights can be given to the energies of different virtual cells within a strip, to simulate non-uniformity along the strip length. A simple exponential dependence has been implemented, controlled by the parameter `ECAL_strip_absorbtionLength`. The energy of the final `CalorimeterHit` is then given by

$$E'_{MIP} = \sum_i E_i \times exp(\delta x_i / \texttt{ECAL\_strip\_absorbtionLength})$$

where the index i runs over the virtual cells of a strip, and δx_i is the distance between the centres of the virtual cell and the strip. This energy is then treated in the following steps:

Conversion of energy to MIP equivalents

The deposited energy in the scintillator is converted to MIP units using the parameter `Calib§CALMIP`

$$E_{MIP} = \texttt{Calib§CALMIP} \times E_{GeV}.$$

Finite number of photo-electrons and PPD pixels

A finite number of photo-electrons (p.e.) are produced in the PPD by energy deposited in the scintillator. The energy deposited in the scintillator is converted to an average number of photo-electrons in the PPD:

$$n_{pe}^{ave} = E_{MIP} \times \texttt{§CAL\_PPD\_PE\_per\_MIP},$$

.

    The finite number of PPD pixels introduces both a saturation effect on the average response and additional signal fluctuations relevant mostly at high signal levels. To include these statistical effects in photon generation and PPD response two separate models have been proposed, called here the *Stoykov Model* and the *Binomial Model*. Both proposed models of PPD response ignore cross-talk between pixels, after pulses, and similar effects.

Stoykov Model

The *Stoykov Model* follows the calculation steps in [1], by first quantising the number of photons hitting the PPD by taking the actual number of p.e. n_{pe} randomly from a Poisson distribution with mean n_{pe}^{ave}. Using a Poisson distribution is justified, as from the around 4500 photons generated per incident MIP [2] only around 15 (depending on the chosen §CAL_PPD_PE_per_MIP) actually fire pixels in the PPD.

    The average number of fired PPD pixels n_{pix}^{ave} for a given number of input p.e. n_{pe} is modeled as

$$n_{pix}^{ave} = \text{§CAL\_PPD\_N\_Pixels} \times (1 - exp(-n_{pe}/\text{§CAL\_PPD\_N\_Pixels})),$$

and fluctuations in the number of fired PPD pixels due to the limited number of pixels are modeled as [1]:

$$
\begin{aligned}
n_{pix} &= n_{pix}^{ave} + \delta n, \text{ where} \\
\delta n &= \text{RandGauss}(0, w) \\
w &= \sqrt{\text{§CAL\_PPD\_N\_Pixels} \times exp(-\alpha) \times (1 - (1 + \alpha) \times \exp(-\alpha))} \\
\alpha &= n_{pix}^{ave}/\text{§CAL\_PPD\_N\_Pixels}
\end{aligned}
$$

    The terms given above are derived using an approximation of longer terms. The validity of this approximation is shown in B. Modelling this fluctuation with a Gaussian shape is somewhat problematic, as it should not be possible to fire more PPD pixels than the number of incident photons.

Binomial Model

In the *Binomial Model*, first the average number of fired PPD pixels is calculated as

$$n_{pix}^{ave} = \text{§CAL\_PPD\_N\_Pixels} \times (1 - exp(-n_{pe}^{ave}/\text{§CAL\_PPD\_N\_Pixels})),$$

.

    This mean amplitude is then quantised by randomising it from a binomial distribution with $n = \S\texttt{CAL\_PPD\_N\_Pixels}$ and $p = \frac{n_{pix}^{ave}}{\S\texttt{CAL\_PPD\_N\_Pixels}}$

    While the *Stoykov Model* is intuitive in its succession of steps, it suffers from the unknown shape of the fluctuation from finite PPD pixels. The *Binomial Model* is less intuitive in its approach, but has been successfully used in the digitisation of AHCAL physics prototype simulations.

    In appendix A it is shown that the expected resolutions arising from these digitisations are exactly identical for both models. As the *Binomial Model* does not suffer the shape problems of the *Stoykov Model*, the *Binoial Model* is used in this processor.

Variations in pixel response

Variations in individual pixel signals (due to e.g. variations in capacitance) can be simulated by using the parameter $\S\texttt{CAL\_pixel\_spread}$, which introduces extra variations in the PPD signal n_{pe}^{sig}

$$n_{pix}^{sig} = n_{pix} \times \texttt{RandGauss}(1, \S\texttt{CAL\_pixel\_spread}/\sqrt{(n_{pix})}).$$

Electronics noise

Electronics noise is then added to n_{pix}^{sig}, as described in section 4.

Mis-calibration of total pixel number, unfolding of average PPD response

A mis-calibration in this number of total PPD pixels can be introduced by the parameter $\S\texttt{CAL\_PPD\_N\_Pixels\_uncertainty}$, in which case the assumed number of PPD pixels is defined to be:

$N_{pix} = \S\texttt{CAL\_PPD\_N\_Pixels} \times \texttt{RandGauss}(1, \S\texttt{CAL\_PPD\_N\_Pixels\_uncertainty}).$

    This mis-calibrated number of total pixels is then used to unfold the PPD response by using the inverse of the saturation curve:

$$n_{pe}^{\text{unfold}} = -N_{pix} \times log(1 - min(n_{pix}^{sig}, N_{pix} - \epsilon)/N_{pix})),$$

where $\epsilon = 1$ is a small number which ensures that the logarithm is well behaved.

Conversion back to energy

The unfolded number of p.e. is then converted back into a scintillator energy deposit using the parameters §`CAL_PPD_PE_per_MIP` and `Calib`§`CALMIP`.

7 Conclusion

Possibilities for realistic modeling of silicon– and scintillator–based calorimeter energy readout have been implemented in `ILDCaloDigi`. The modeling is rather simple, but should be adequate to allow a comparably realistic simulation of the different technologies. Tuning of the processor parameters should be performed by comparisons with data collected by detector prototypes in test beams.

Appendices

A Comparison of parametrisation models

We define a as the number of photons hitting the SiPM, n as the number of different SiPM pixels hit from these a photons and the number of total SiPM pixels m. Furthermore $p = {}^n/_m$ denotes the fraction of fired SiPM pixels. The relation between incoming photons (unsaturated signal) and fired pixels (saturated response) is given by

$$n = f_{sat}(a) = m e^{-\frac{a}{m}} \tag{1}$$

$$a = f_{sat}^{-1}(n) = -m \log\left(1 - \frac{n}{m}\right) \tag{2}$$

$$\frac{\mathrm{d}f_{sat}^{-1}(n)}{\mathrm{d}n} = m\frac{1}{m-n} = e^{\frac{a}{m}}. \tag{3}$$

Beginning with the variance of the binomial distribution:

$$\mathrm{Var}_{sat}^{bino} = mp(1-p) \tag{4}$$

$$= m\left(\frac{n}{m}\right)\left(1 - \frac{n}{m}\right)$$

$$= m\left(e^{-\frac{a}{m}}\right)\left(1 - e^{-\frac{a}{m}}\right) \tag{5}$$

Converting to the unsaturated variance, following [1]:

$$\text{Var}_{unsat}^{bino} = \text{Var}_{sat}^{bino} \left(\frac{\mathrm{d} f_{sat}^{-1}(n)}{\mathrm{d} n} \right)^2 \tag{6}$$

$$= m \left(e^{-\frac{a}{m}} \right) \left(1 - e^{-\frac{a}{m}} \right) \left(e^{\frac{2a}{m}} \right)$$

$$= m \left(e^{\frac{a}{m}} - 1 \right) \tag{7}$$

This yields the resolution of the binomial SiPM model:

$$\text{Res}^{bino} = \frac{\sqrt{\text{Var}_{unsat}^{bino}}}{a} = \frac{\sqrt{m \left(e^{\frac{a}{m}} - 1 \right)}}{a} \tag{8}$$

It is easily seen, that this is equivalent to the parametrised resolution $\text{Reso}^{Stoykov}$ given in [1]:

$$\text{Res}^{Stoykov} = \frac{1}{\sqrt{m}} \sqrt{\frac{1}{\alpha} + \left(\frac{1}{\alpha} \sqrt{e^{\alpha} - 1 - \alpha} \right)^2}, \quad \alpha = \frac{a}{m} \tag{9}$$

$$= \frac{1}{\sqrt{m}} \sqrt{\frac{m}{a} \left(1 + \left(\frac{m}{a} \left(e^{\frac{a}{m}} - 1 - \frac{a}{m} \right) \right) \right)}$$

$$= \frac{\sqrt{m \left(e^{\frac{a}{m}} - 1 \right)}}{a} = \text{Res}^{bino} \tag{10}$$

B   Full vs. approximate Variance [1]

In [1] the variance of the number N of urns containing one or more balls after randomly distributing a (photons) balls into m urns (pixels) is given as

$$\text{Var}^{Stoykov} = m(m-1)(1 - 2m^{-1})^a + m(1 - m^{-1})^a - m^2(1 - m^{-1})^{2a}. \tag{11}$$

This is claimed to be approximated by the shorter form

$$\text{Var}_{approx}^{Stoykov} = m(1 - e^{\frac{a}{m}}) \tag{12}$$

for $m, n \to \infty$ and the ratio $^a/_m$ bounded.

To check the validity of these assumptions, a simple comparison was performed for a sensor with $m = 2000$ pixels. As shown in Figure 1, the difference between the full and approximate variance terms are very small for
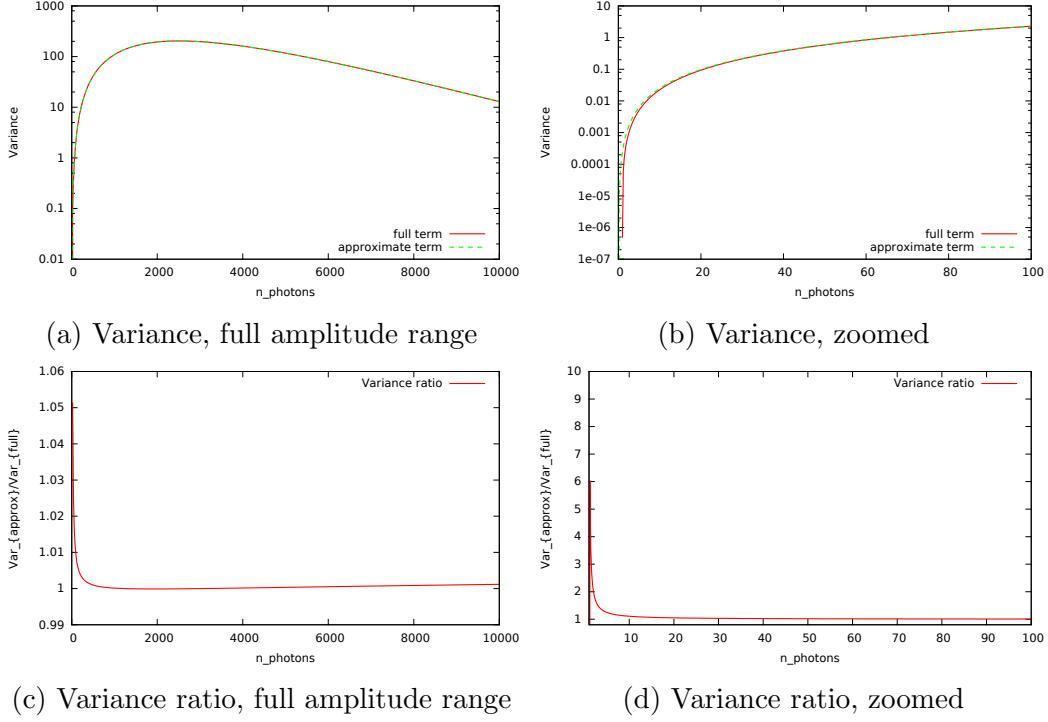
(a) Variance, full amplitude range   (b) Variance, zoomed

(c) Variance ratio, full amplitude range   (d) Variance ratio, zoomed

Figure 1: Comparison of full and approximate variance terms as given in [1]. Only for very small amplitudes <10 pixels relevant deviations are visible.

amplitudes >10 pixels. Even for amplitudes 1ess 0px the observed mismatch is far from relevant, as for the full resolution Poissonian uncertainty is added in quadrature, which is orders of magnitude larger for such amplitudes. Additionally, a threshold of 0.5 MIP (corresponding to about 7.5 pixels fired) is expected for ILD operation. Thus low amplitude hits are not taken into account at all.

From this we conclude, that the approximate variance form is suitable for use in digitisation.

References

[1] A.Stoykov et al., "On the limited amplitude resolution of multipixel Geiger-mode APDs", arXiv:0706.0746.

[2] Lennart Adam, Uni Mainz, private communication with OH, 27.01.2015

[3] Steven Green, Cambridge, presentation for ILD optimisation meeting: https://indico.cern.ch/event/369380/contribution/3/material/slides/0.pdf