GEM digitization and tracking with GEp

E. Fuchey

April 15, 2019

Contents

1 GEM digitization

This part describes the simulation of GEM response starting from the g4sbs simulation information, as well as the background superimposition method.

1.1 Simulation input

The simulation input for the digitization algorithm is produced by g4sbs, a GEANT4 based simulation of the SBS experiments. When a particle is going through a GEM layer, the following information is recorded:

- the energy deposited by the particle in the GEM drift gas layer E_{dep};
- the hit average time t_{hit} and position $\vec{x}_{\text{hit}}$ in the GEM drift gas layer (weighed with energy deposit), as well as the entrance and exit position of the particle in the GEM drift gas layer, $\vec{x}_{\text{in}}$ and $\vec{x}_{\text{out}}$;
- the PDG particle ID (PID), track ID (TID) and mother track ID (MID);
- and of course, the GEM detector ID.

The energy and positions are necessary and sufficient to implement the digitization steps that are described in the next section. The different IDs of the Monte Carlo particle are useful to identify the primary hits during further analysis steps.

1.2 Digitization of Monte Carlo information

1.2.1 ionization and avalanche model

The first step for the digitization is the simulation of the avalanche, using the energy deposited, and the entrance and exit position of the particle in the GEM drift gas. The number of ions generated for the avalanche by the particle is:

$$N_{ions} = \text{Poisson}(E_{dep}/[\text{GasIonWidth} = 26\text{eV}]). \tag{1}$$

The position of each ion is generated uniformly between the path made by the vector $\vec{v}_{path} = \vec{x}_{\text{out}} - \vec{x}_{\text{in}}$. For each ion, the spread of the avalanche is

$$\sigma_{ava} = t_{drift} \times [\text{GasDiffusion} = 10^5 \text{mm}^2\text{s}^{-1}], \tag{2}$$

with $t_{drift} = d_{ion-readout}/[\text{GasDriftVelocity} = 5.510^7 \text{mm}\,\text{s}^{-1}]$ the avalanche drift time. The total avalanche charge is determined for each ion by

$$Q_{ava} = \text{Gaus}([\text{GainMain} = 8000], [\text{Gain0} = 20]) \tag{3}$$

The avalanche is then numerically integrated for all ions, over an area equal to $\sigma_{ava} \times [\text{AvalancheFiducialBand} = 10]$.

During this process, the list of strips which will be hit by the avalanche is evaluated for each coordinate, and the energy deposited in each of the strips hit is calculated as a function of the surface of the strip covered by the avalanche convoluted by the surfacic density of ions, determined by the avalanche distribution function, which is a Cauchy-Lorentz function. This signal is then used to evaluate the ADC samples.

1.2.2 calculation of ADC samples

The avalanche charge deposited in each strip is sampled in intervals of time corresponding to the sampling periods of the APV25 chip with, for samples $i = 1$ to 6, the amplitude a_i:

$$a_i = \text{Max}(0, A\frac{t_i - t_0}{\tau^2}\exp(-\frac{t - t_0}{\tau})) \tag{4}$$

with A the total amplitude of the signal received by the strip, $t_i = (i-1) \times [\text{EleSamplingPeriod} = 25\text{ns}]$, t_0 the event trigger time, and $\tau = [\text{PulseShapeTau} = 56\text{ns}]$ the pulse shape constant. We will describe in more detail the determination of the trigger time in Subsec. 1.3. Each amplitude sample is converted into an ADC value such as:

$$ADC_i = \text{Min}((a_i - [\text{ADCOffset} = 0])/[\text{ADCGain} = 1], 2^{[\text{ADCBits}=12]}). \tag{5}$$

> *Question*: I am describing the code as it is, but should not we multiply by the gain instead of dividing? Is that even relevent provided that, to my understanding, the APV do not amplify the signal readout from the strips?

Just before saving the strips in the output file (meaning after background addition, if any), each ADC sample is reevaluated to account for the APV25 chip ADC sample pedestal:

$$ADC_i = \text{Min}(\text{Gaus}(ADC_i, [\text{PulseNoiseSigma} = 20]), 4000). \tag{6}$$

(Comment: this needs to be squared up: we are not adding the common mode, but it should be accounted in the saturation. If the common mode is around 100 ADC values, then the hardcoded value should be about fine - except it should not be hardcoded anyway. Also, the saturation should only be made in this last step.)

Finally, for each strip filled with some signal, a cross-talk is simulated 32 strips apart, with an amplitude of 10 % of what the signal strip.

1.3 background superimposition

The main interest of this study is to evaluate the tracking capabilities in the presence of beam-on-target background. However, g4sbs does not handle yet a multigenerator which would allow to generate both signal and beam-on-target background simultaneously in a single simulation. The GEM digitization, however, is able to handle the superimposition of several types of simulation files (usually signal and background), in proportions that can be decided by the user (usually many background events on top of one signal event).

In practice, for each background event, the strips hit and the ADC samples are evaluated for each background hit, in a similar way to the signal hits, and are then added on top of the preexisting strips/samples. In other words, if a strip has already been hit before, the ADC samples are simpled added to the already existing values.

The only steps which differs between the signal and the background is the evaluation of the event trigger time t_0, which had already been mentioned in Eq. 4.

For the signal, this trigger time is evaluated as:

$$t_0 = t_{\rm hit} + t_{\rm trigoffset}(GEMplane) + t_{\rm driftmin} + {\rm Gaus}(0, [{\rm TriggerJitter} = 9ns]) \tag{7}$$

with $t_{\rm hit}$ already defined in Sec. 1.1, $t_{\rm trigoffset}(GEMplane)$ the trigger offset to apply to obtain a primary hit time distribution centered to $t = 0$ for each plane, $t_{\rm driftmin}$ the minimal drift time for the ion produced the closest to the readout. The TriggerJitter accounts for both the actual trigger jitter, but mostly for the fact that this trigger is out-of-sync with the APV chip clock.

For the background on the other hand, the trigger time has almost the same expression, except $t_{\rm hit}$, (*Comment*: I understand the relevance of spreading background from -200 ns - which is the time when a background hit can start contributing to the signal. On the other hand, I'm really not convinced we should extend the spread up to +200 ns. A particle depositing energy in the gas drift at the same time the $6^{\rm th}$ sample is past *i.e.* at +150 ns is not going to give any signal.)

2 GEM clustering and (pre-2019) analysis

The clustering aims to reconstruct the GEM hits with which the tracks will be reconstructed. This section describes the decoding of the GEM strips, the selection of "good" strips, and clustering, that has been used until early 2019. The clustering is made independently in each readout coordinates, then the hits are being recombined after some tracking steps.

2.1 Strip decoding and selection

The GEM data are decoded strip-per-strip independently for each readout plane. For each strip all 6 ADC samples are read and the sum of ADC samples is evaluated. For the strip to be selected as a potentially good signal strip, this sum has to reach a threshold configurable by the database. We had set it to 180 which corresponds to the equivalent of the sum over the 6 samples of 2 standard deviations of a sample pedestal. While the samples are read, we check the variable noted "MaxADC" which is the maximal ADC values over the second, third, fourth samples (which is where the signal peak should tyipcally be). This maximum has to be at least 40 ADC (configurable in the database) for the strip to be selected as a potentially good signal strip. Finally, the strips samples are being fitted by a function of the form:

$$f(t) = A\frac{t - t_0}{\tau} \times \exp\frac{t - t_0}{\tau}, \tag{8}$$

to detemine the peak of the distribution. The peak of the fitted function has to be located within the time range covered by the 6 samples. As a recap, a strip will be considered as good if:

- The sum of all 6 ADC samples is ≥ 180 ADC channels;
- One sample among the second, third, and fourth has an ADC value ≥ 40;
- the peak of the function described in equation 8 fitted over the 6 samples is within the time span covered by those 6 samples.

Only the strips meeting this requirement are used for the clustering described in the next subsection.

2.2 Clustering, cluster selection

By the algorithm we are going to describe, a cluster is a group of "good" adjacent strips (*i.e.* a group of strips meeting the conditions stated in the previous subsection).

When the strips are grouped together, the algorithm compares the size of the group of strips with a limit (noted "MaxClustSize") defined in the database (in our case, 6 strips -*need to check that as well*). If the strip group size is smaller than the limit, the algorithm performs the clustering directly. This is described in subsubsection 2.2.1. If it is larger, the algorithm attempts to split the cluster in size and time. This is described in subsubsection 2.2.2.

2.2.1 single cluster analysis

In the simple case of a single cluster (*i.e.* with a number of strips below "MaxClustSize"), the amplitude is calculated as the sum of the strips ADC; the position is calculated as the ADC weighted average of the position of the strips involved in the cluster; the time of the hit is the time of the strip with the largest "MaxADC" value.

2.2.2 cluster splitting

In the event where there is a consecutive number of valid strips above "MaxClustSize", a "peak-valley-peak" pattern is searched. In practice, a first ADC maximum is searched, then the strips with a minimum ADC value is searched for. If this minimum is less than a fraction defined by the database parameter "" and is followed by another maximum, then the strip is split between the two consecutive clusters. What that means is the strip will be accounted for in both clusters, and the ADC value will be split between both clusters *i.e.* the ADC sum of this strip is divided between both clusters. If the group the strips after the first minimum is larger than "MaxClustSize", the algorithm repeats this procedure until it has scanned all the group of strips. All the clusters obtained can now be treated as single clusters for the calculation of their position, amplitude and time. If in a group of strips larger than "MaxClustSize" a "peak-valley-peak" pattern cannot be found, the algorithm considers it as a "large" cluster, which in practice is treated as a single cluster for the calculation of its position, amplitude and time.

3 GEM clustering - post-2019 analysis

The new clustering algorithm itself is in development. For the moment this section just means to put on the table what ideas can be realized.

This section describes the new GEM decoding analysis we set up after the realization that the old clustering method described in section 2 was suffering inefficiencies. One of the features we want to exploit is the correlation in time and amplitude between both readout coordinates. We also want to exploit the relatively well known shape of the spread avalanche. The main challenge is that we want to consider clusters, and not strips, as the first degrees of freedom, but we still have to deal with strips as they are the only information we have available from the DAQ.

An option could be to scan the strips in both coordinates, and start to group together strips with a similar sampling pattern (indicating a similar time). If two strips with a similar sampling pattern and a detectable (but not *large*) signal are separated with one strip which does not seem to show this same pattern, the middle strip should be integrated into the cluster nonetheless.

Indeed the signal amplitude we could have for the proton in GEp could be fairly small, and it is fairly easy for a strip to be distorted enough by the pedestal that we can hardly match its sampling pattern with the others.

While scanning the strips, if we scan at least a couple strips with a similar sampling pattern, the one with the maximal amplitude should be considered as the "center" of the hit, and the limits of the hit could be extended to the few neighboring strips.

After obtaining such patterns in both coordinates, those would have to be associated together based on time and amplitude criteria (one criterion could take precedence the other, e.g. time on amplitude). The hits in each projection shall be retained only if the have found a match on the other coordinate. *Assuming this is*

the way to go, how to handle background ?

Another option might be to scan the strips and retain the local ADC maxima with times on both coordinates. After obtaining these ADC maxima on both coordinates, one should match them coordinate to coordinate, and build the hits from the successfully matched strips using the typical avalanche profile.