GEM digitization and tracking with GEp

E. Fuchey

April 12, 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:

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

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

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

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

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

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

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:

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

with A the total amplitude of the signal received by the strip, $t_i = (i-1) \times [\mathrm{EleSamplingPeriod} = 25\mathrm{ns}]$, t_0 the event trigger time, and $\tau = [\mathrm{PulseShapeTau} = 56\mathrm{ns}]$ 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:

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

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:

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

(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:

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

with t_{hit} already defined in Sec. 1.1, $t_{\text{trigoffset}}(GEMplane)$ the trigger offset to apply to obtain a primary hit time distribution centered to $t = 0$ for each plane, t_{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_{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$^{\text{th}}$ 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.

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 $\geq$ 180 ADC channels;
- One sample among the second, third, and fourth has an ADC value $\geq$ 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 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

2.3 performance for G_E^p