Manual and trouble shooting of the MVD-Digitizer and MVD-Hitfinder

Version: 25.11.2009

Software by C.Dritsa (GSI/IPHC)

Maintenance: M. Deveaux (Goethe University Frankfurt, m.deveaux(att)gsi.de)

This brief write-up is to provide a short manual for the MVD-Digitizer and some answers to frequently asked questions. It is also available in the CBM-Forum and will regularly be updated.

1. Manual

1.1 Introduction

The MVD-Digitizer and the MVD-Hitfinder provide a reliable and robust detector response model. The model simulates a realistic detector response in terms of cluster multiplicities and charge sharing in the cluster. The spatial resolution provided is realistic for standard tracks but still optimist for very inclined tracks. The Digitizer comes with two simulation models (Digitizer and DigitizerL). It is recommended to use the DigitizerL as it is more precise and closer to physics.

The digitizer allows simulating pile-up in the MVD and delta electrons. To do so, you will have to transport separately a file with central collisions (default input), a file with minimum bias collisions (pile-up) and a file with delta electrons (for delta electron simulation). The digitizer mixes the output of the transport at the level of MCPoints. Important: There is no check for the compatibility of the files. Please make sure that you transported all files with the same version of CBMRoot, the same flavor of LINUX (note lxg1410-1412 have another Linux than the batch farm) and the same geometry and magnetic field.

1.2 Concept of the Digitizer

The Digitizer simulates the signal charge integrated in different pixels starting from MCPoints. It defines the trajectory of the particle in the detector as the straight line between the impact and the exit point of the particle. The software requires a minimum distance between those points. Secondary particles produced in the station itself are therefore typically ignored. Moreover, the Digitizer does typically not react to nuclear fragments, neutrons and gammas.

The digitizer calculates the trajectory of the particle through the sensitive volume of the sensors, which typically much thinner than the station. It is assumed that this sensitive volume is located in the middle of this station, which is a simplified but currently sufficient picture of reality. The interaction between the particle and the material is simulated using an internal random generator, which provides random numbers according to a Landau-Distribution. This solution was chosen as the energy deposit provided by GEANT corresponds to the energy deposit of the particle in the full station, which includes also an energy deposit outside the sensitive layer. Scaling this value according to the thickness of the sensitive layer would provide the correct mean signal charge but the wrong width of the distribution.

The parameters of the Landau-Distribution used in the Digitizer are tuned according to measured data. This data is available for 0° inclination angle only. It is therefore scaled to other inclination angles according to the relative length of the trajectory in the sensor. Hereafter, the trajectory is subdivided into small segments which are treated individually.

The details of this treatment distinguish the Lorentz and the Gauss Digitizer (DigitizerG and DigitizerL). The current default digitizer is the Lorentz Digitizer, which describes best the diffusion process of charge in the undepleted MAPS. The algorithm uses a empirical, two dimensional Lorentz-peak to distribute the charge of the individual segment to the different pixels. This approach realizes a good parameterization of empirical results with different sensors but does not claim to base on an understanding of the underlying transport and diffusion processes.

The Gauss-Digitizer models the charge transport and diffusion in depleted sensors (like strip and hybrid pixel detectors). It assumes that the charge is distributed by diffusion while being attracted by the depletion voltage of the sensor. This is modeled with a Gaussian distribution of the charge which is hereafter integrated over the surface of the pixels. The width of the Gaussian is a parameter in the model, which is tuned according to data. The Gauss-model was initially used but came out to describe the clusters of MAPS with a lower accuracy than the Lorentz-model.

In both models, the charge of the individual segments distributed to the nearby pixels. The pixels sum up the charge of the segments (and of any number of potential additional tracks). This allows simulating cluster formation and cluster merging. Any pixel (and the corresponding MVDdigi) obtains MC-information on the trackID and the MC - impact point of the MCPoint providing the highest charge. If the pixel received charge from more than one particle, the digi will typically the MC-data of the closest track.

Note that the digis contain the charge in units of electrons. This information does not account for any electronic noise of the sensors.

1.3 Concept of the hit finder

The hit finder hosts several functionalities, which are the simulation of the noise of the readout chain, the simulation of the ADC/Discriminator and the cluster finding. Moreover, there is an option to simulate fake hits, which is currently not recommended due to the excessive CPU consumption and the poor accuracy of the corresponding model.

The hit finder uses the MVDdigis provided by the digitizer as input and provides MVDHits, MVDHitInfos and MVDClusters as output. In a first step, the noise of the sensors is simulated by adding a noise according to a Gaussian distribution. Note that by doing so, one obtains negative signal values in some pixels.

In the next step, the charge seen by the pixels is translated into ADC-units. The signal discrimination is then performed based on this ADC-signal.

The hit finder distinguishes two different cuts, which is the cut on seed pixels and the cut on neighbors. A pixel passing the seed cut triggers is used as seed pixel of a cluster (e.g. the hit finder uses it as a starting point for a search for a cluster). A pixel passing the neighbor cut but not passing the seed cut may be added to a cluster, if a seed pixel is already added in this cluster. However, no cluster is formed from pixels passing the neighbor cut in the absence of a seed pixel.

The algorithm of the hit finder starts with a seed pixel and scans its next neighbors for having passed as least the neighbor threshold. Any pixel fulfilling this condition will be added to the cluster and its neighbors will be scanned for potential further pixels passing the threshold.

This algorithm is relatively slow but suited to any cluster charge and to any type of ADC-settings.

Once a cluster is formed (no further neighbors passing the threshold are found), the centre of gravity of the pixel charge is calculated and used as hit position in the MVDHit. The uncertainty of the hit position is currently NOT calculated. Instead, a value provided by the used is stored in the hit. The hit info class provides contains the trackID of the track, which provided most charge to the pixel, which is in the centre of gravity of the cluster.

The MVDCluster contains more detailed information. It stores the geometry of the cluster into a 7x7 pixel array. This information might be used in future to separate merged clusters. Moreover, it provides information on the hit position of the MC-Point of up to 5 tracks merging in the cluster. Note that all pixels in the 7x7 array are filled with noise according to the Gaussian noise distribution provided that no corresponding digi is available.

1.4 Parameters of the Model

1.4.1 DigitizerL

Setter	Meaning	UD	Recommended value
SetEpiThickness(Double_t epiTh)	Sets the thickness of the sensor.	yes	14 µm
SetSegmentLength(Double_t length)	Sets the length of a segment in the simulation	no	1 µm
SetDiffusionCoef()	Sets the width of the Gaussian (in DigitizerG)	no	Don't touch
SetElectronsPerKeV()	Material constant translating energy into electron hole pairs	no	Don't touch
SetWidthOfCluster()	Maximum distance between a track and a pixel receiving charge	yes	3 x pixel pitch
SetPixelSizeX / Y ()	Modifies the pixel pitch without adapting the simulation parameters Not recommended except for advanced users.	No	Not recommended
SetPixelSize()	Modifies the pixel with reading the simulation parameters from an internal data base. Possible values are: 10, 18.4, 30 (µm)	yes	18.4 µm
SetChargeThreshold()	Minimum number of electrons, a pixel needs to generate a digi (in units of electrons)	Not really	1
SetCutOnDeltaRays	Probably obsolete (to be checked)	--	
SetMvdGeometry	Currently obsolete	--	
SetPileUp	Defines the number of minimum bias collisions added to each event. Use SetBgFileName to define the name of the corresponding file	yes	Pileup - 1
SetDeltaEvents	Defines the number of Ions producing delta electrons added to each event. Use SetDeltaName to define the name of the corresponding file. Make sure that only one ion is transported in each event of this file	yes	Pileup * 100
SetBgFileName	Defines the name of the file containing the minimum bias collisions used for pileup. Not required if SetPileUp is not used.	Yes	--
SetDeltaName	Defines the name of the file containing the delta electrons used for delta electron simulation. Not required if SetDeltaEvents is not used	Yes	--
SetBgBufferSize	Defines the number of events available in the BG-File	Yes	>Pileup * 10
SetDeltaBufferSize	Defines the number of events available in the Delta-File	Yes	>Pileup * 1000
ShowDebugHistograms()	Shows some histograms allowing for quality checks of the digitizer		

1.4.2 HitFinder

Setter	Meaning	UD	Recommended value
SetSigmaNoise(value, simulate)	Sets the noise of the sensor (in electrons). Defines if noise should be simulated	yes	15, kTRUE
SetSeedThreshold	Sets the seed threshold in ADC-units	yes	1 (75 electrons equivalent)
SetNeigbourThreshold	Sets the neighbour threshold in electrons	yes	1 (75 electrons equivalent)
SetAdcDynamic	Sets the dynamic of the ADC (in electrons)	yes	150
SetAdcOffset	Sets the offset of the ADC (in electrons)	yes	0
SetADCBits	Sets the number of bits of the ADC	yes	1 (discriminator)
SetHitPosErrX	Sets the indicated uncertainty of the hit (not the real one)	yes	Pixel pitch / 5
SetHitPosErrY	Sets the indicated uncertainty of the hit (not the real one)	yes	Pixel pitch / 5
SetHitPosErrZ	Sets the indicated uncertainty of the hit (not the real one)	yes	Pixel pitch / 5
SetDebugHistograms	kTrue prints debug histograms. Generates charge distributions for comparision with real data	yes	kFALSE (for production)

2. Trouble shooting

Problem	Origin	Solution
No delta electrons and pile-up	Module is not turned on (default)	Turn the modules on by adding: mvdDigitize-> SetBgFileName(bgFile) mvdDigitize-> SetBgBufferSize (Int_t numberOfEventsInFile) mvdDigitize-> SetPileUp (Int_t numberOfBgEventsToBeAdded) mvdDigitize-> SetBgFileName(deltaFile) mvdDigitize-> SetBgBufferSize (Int_t numberOfEventsInFile) mvdDigitize-> SetPileUp (Int_t numberOfDeltaEventsToBeAdded)

	The delta electrons are present but the normalisation is wrong.	Use the following default values:
		numberOfBgEventsToBeAdded= pileup - 1
		numberOfDeltaEventsToBeAdded= pileup * 100
		Make sure that each event of the delta event file contains one ion impinging the target each.

	Files, geometries or Linux flavours are not compatible	Make sure that you transported your files with the same Linux flavour. Note: lxg1410-1412 are currently not compatible with the batch farm... not even if you use the 32bit queue. Use lxi003 etc. for interactive work.
The setters (like mvdDigitize->SetPileUp) are reported not to exist when starting the MVD-Digitizer / Hitfinder from a reconstruction macro. However, they are in the source code.	You declared the Task as Fairtask in your Macro. The Fairtask does not know about the setters.	Declare the Digitizer as CbmMvdDigitizer* digitizer in your task (not FairTask* digitizer). Do the same for the Hitfinder.
If I use the MVD-Digitizer, the tracking crashes. The problem does not appear when using the hit producer.	An outdated version of the digitizer is used	Update to DEC09 stable
The MVD-Hits are slightly but systematically displaced with respect to the MVD-points.	An outdated version of the digitizer is used	Update to DEC09 stable