Description of measurements for MC

Current Release:					
Ver.	Created	Comment	Distribution	Editor	Approver(s)
0.0	2018-01-09	Pre-released version		T. Armstrong, H. Costantini	B. Name

History:					
Ver.	Created	Comment	Distribution	Editor	Approver(s)

Table of Contents

Since the available test setups differ from camera type to camera type and from telescope type to telescope type, one of the essential requirements for making the test measurements useful for MC verification is proper documentation of the test setup, the availability of all relevant calibration measurements, and proper tracking if the devices tested still correspond to the final implementation (and if not, what was changed). The documentation must be accurate and complete enough that a Monte Carlo implementation of the test setup and the measurements carried out with it is possible with current and future MC codes. Data from the test measurements should be available in a data format suitable (via conversion, if necessary) for CTA data processing (ctapipe), identical to processing of the matching MC data. And it should include all relevant pedestal, calibration, etc. data to make full use of the test measurements also with future version of the processing code.

1 Trigger Efficiency

An important measurement needed to verify MC is the trigger efficiency. For this measurement the camera or part of it (pixel disabled or masks put on the camera focal plane to simulate the shower image or muon ring) should be illuminated with homogeneous pulsed light at different illumination levels and the camera should trigger internally. Different measurements should be done changing the illumination level from 1 to 1000 pe/pixel. The information needed from these measurements are:

- The number of injected pulses
- The number of triggered pulses
- The number of triggered pixels (multiplicity)
- The location of the triggered pixels (topology)
- The background light level, ideally done with zero background and increasing illumination level (to be discussed).

The important inputs for MC to reproduce the measurements are:

- The position of the enabled pixels
- The value of the trigger threshold (discriminator threshold)
- The illumination level per pixel and its uncertainty in pe
- The wavelength range of the illumination source

The corresponding MC simulation will be done simulating a uniform illumination source with the same characteristics as in the lab measurement. Different illumination level will be simulated. The output of the simulation will be the ratio between the triggered and generated events (trigger efficiency). The Figure of merit will be the trigger efficiency as a function of the total charge of the image. The total charge of the image can be considered as:

- the total reconstructed charge of the enabled pixels

  or
- the total true charge injected on the enabled pixels

In the first case the reconstructed charge have to be obtained. Both data and MC will be analysed with ctapipe and same charge extraction methods will be applied. The instrument team should provide the data in ctapipe format. This means that a converter from the data format coming out from the EventBuilder to the ctapipe format should be provided. An example of such a converter can be found for GCT (ref github page) written by J. Watson.

In the second case the true charge injected on each pixel can be obtained with a script (adpated from J. Watson code) that takes the geometry of the set up (distance of light source to camera, angular distribution of the light source, camera geometry, pde of sensors) and returns the required photon emission intensity to obtain a requested p.e. per pixel. In the case of MC simulation, the true injected charge can be also obtained directly by the MC truth from the sim_telarray file (details in the following section).

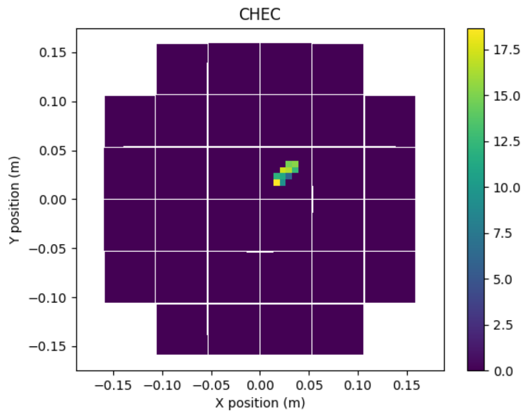
1.1 MC simulation

To simulate the laboratory measurement the illumination package can be used [**?**]. The illumination package can be obtained installing the sim_telarray development version that can be found at: https://www.mpi-hd.mpg.de/hfm/CTA/MC/Software/Testing/corsika6.9_simtelarray.tar.gz, or installing the standalone version: https://www.mpi-hd.mpg.de/hfm/CTA/MC/Software/Testing/LightEmission/

- run J. Watson's script to obtain the required photon emission intensity N$_{ph}$ of the light source to obtain a requested number of p.e. per pixel.
- run ff-1m to generate an input file for sim_telarray that simulates the light source used in laboratory to illuminate the camera. Example:

  ```
  ./ff-1m --events 100 --photons Nph --distance 100 --camera-radius 30
  --angular-distribution ang_dist.dat -o beamed_test.dat.gz
  ```

  this example simulates a light source that generates N$_{ph}$ photons per flasher unit (as obtained by previous script) at a distance of 100 cm from the camera, in a fiducial radius of the camera in the detection plane of 30 cm, with an angular distribution as in the ang_dist.dat file.
- run sim_telarray on the generated file. Example:



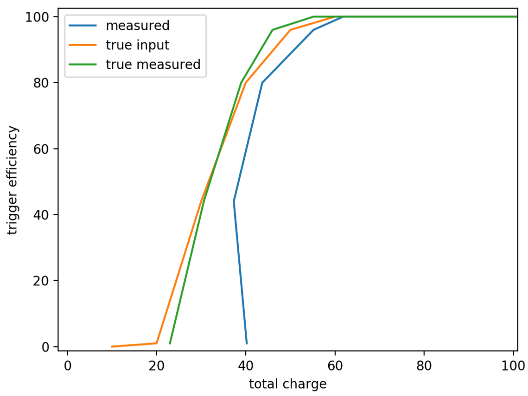


Figure 1.1 – Example of trigger efficiency simulation. The left plot shows the enabled pixels in the simulation. The right plot shows the trigger efficiency as a function of the MC truth charge of the image (true measured), the requested pe per pixel multiplied by the the number of the enabled pixels (true input) and the reconstructed charge of the image (measured).

```
./sim_telarray -c cfg/CTA/CTA-PROD4-SST-GCT-prototype.cfg -DNUM_TELESCOPES=1 -Icfg/CTA
-i beamed_test.dat.gz -C Bypass_Optics=2  -C Altitude=2100 -C iobuf_maximum=1000000000 -C
maximum_telescopes=1  -C atmospheric_transmission=atm_trans_2100_1_10_0_0_2100.dat  -C
telescope_theta=0 -C telescope_phi=0 -C power_law=2.5  -C output_file=ff-1m_optics2.simtel.gz
```

By counting the numbers of "has triggered!" in the sim_telarray output log file one can obtain the number of triggered events.

- To obtain the true total charge of the image (i.e. the sum of the true charge in each pixel) from the MC truth, one should use ctapipe as shown in the example:

```
python get_image_charge.py
```

, TO BE COMPLETED

An example of MC simulated trigger efficiency measurement is shown in Figure 1.1 in the case of the GCT camera. In this test a generic mask is applied to the camera that enables only few pixels (left plot). The result can be seen in the right plot with three different data sets: 'measured' which uses the total reconstructed charge in the whole masked image, 'true input' which is just the number of enables pixels times the requested intensity per pixel, and the 'true measured' which is based on the total MC truth charge in the enabled pixels.

2 Pedestal and waveforms

Pedestal measurements should be performed in dark conditions and with increasing illumination (non pulsed light) simulating an increased NSB (from zero to maximum operational rate in the requirements). The output of those measurements are the digitized traces in ADC channels as they are produced by the EventBuilder (R1 level). The Instrument Team should provide those data together with the calibration parameters (for example ADC to pe conversion factor). The instrument team should provide the data in ctapipe format.

The pulseshape measurement will be done with pulsed light and increasing illumination level. The camera will be triggered externally. The measurements should be done with different simulated NSB levels as for the pedestal measurement. Data should be provided in ctapipe format with all camera specific calibration parameters. For both measurements (Pedestal and pulseshape) the following information should be provided by the instrument team:

- illumination level and uncertainty of nonpulsed background light in pe
- illumintation level and uncertainty of pulsed light in pe
- wavelength range of illumination sources

2.1 MC simulation

To simulate the pedestal measurements, sim_telarray can be run using the DARK_EVENTS option in the case of a simulation of pedestal run with camera lid closed (completely dark, no NSB) or using the PEDESTAL_EVENTS option in the case of a simulation of a pedestal run with camera lid open and a certain NSB as for normal events.

(HAVE TO TEST CALIB SIMULATION!...don't really know how it works....)

Data can be analyzed using ctapipe (raw_data_exploration.py)