



Universiteit Gent  
Faculteit Wetenschappen  
Vakgroep Fysica en Sterrenkunde

<sup>2</sup> No title yet

<sup>3</sup> No sub-title neither, obviously...

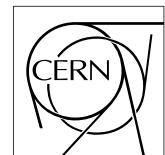
---

<sup>4</sup> Alexis Fagot

5



Thesis to obtain the degree of  
Doctor of Philosophy in Physics  
Academic years 2012-2017





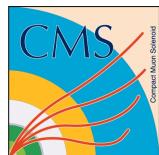


Universiteit Gent  
Faculteit Wetenschappen  
Vakgroep Fysica en Sterrenkunde

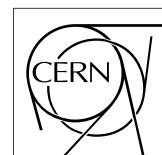
Promotoren: Dr. Michael Tytgat  
Prof. Dr. Dirk Ryckbosch

Universiteit Gent  
Faculteit Wetenschappen  
Vakgroep Fysica en Sterrenkunde  
Proeftuinstraat 86, B-9000 Gent, België  
Tel.: +32 9 264.65.28  
Fax.: +32 9 264.66.97

17



Thesis to obtain the degree of  
Doctor of Philosophy in Physics  
Academic years 2012-2017





## Acknowledgements

<sup>19</sup> Ici on remerciera tous les gens que j'ai pu croiser durant cette aventure et qui m'ont permis de passer  
<sup>20</sup> un bon moment

<sup>21</sup> *Gent, ici la super date de la mort qui tue de la fin d'écriture*  
<sup>22</sup> *Alexis Fagot*



# Table of Contents

|    |   |      |
|----|---|------|
| 24 | <b>Acknowledgements</b>   | i    |
| 25 | <b>Nederlandse samenvatting</b>   | vii  |
| 26 | <b>English summary</b>  | ix   |
| 27 | <b>1 Introduction</b>   | 1-1  |
| 28 | 1.1 A story of High Energy Physics . . . . .                                  | 1-1  |
| 29 | 1.2 Organisation of this study . . . . .                                      | 1-1  |
| 30 | <b>2 Investigating the TeV scale</b>  | 2-1  |
| 31 | 2.1 The Standard Model of Particle Physics . . . . .                          | 2-1  |
| 32 | 2.2 The Large Hadron Collider & the Compact Muon Solenoid . . . . .           | 2-1  |
| 33 | 2.2.1 LHC, the most powerful particle accelerator . . . . .                   | 2-1  |
| 34 | 2.2.1.1 Overview of LHC technology . . . . .                                  | 2-3  |
| 35 | 2.2.2 CMS, a multipurpose experiment . . . . .                                | 2-4  |
| 36 | 2.3 Muon Phase-II Upgrade . . . . .   | 2-4  |
| 37 | <b>3 Physics of Resistive plate chambers</b>                                  | 3-1  |
| 38 | 3.1 Principle . . . . .   | 3-1  |
| 39 | 3.1.1 Electron drift velocity . . . . .                                       | 3-4  |
| 40 | 3.2 Rate capability and time resolution of Resistive Plate Chambers . . . . . | 3-4  |
| 41 | 3.2.1 Operation modes . . . . .   | 3-4  |
| 42 | 3.2.2 Detector designs and performance . . . . .                              | 3-6  |
| 43 | 3.2.2.1 Double-gap RPC . . . . .  | 3-6  |
| 44 | 3.2.2.2 Multigap RPC (MRPC) . . . . .   | 3-7  |
| 45 | 3.2.2.3 Charge distribution and performance limitations . . . . .             | 3-8  |
| 46 | 3.3 Signal formation . . . . .  | 3-11 |
| 47 | 3.4 Gas transport parameters . . . . .  | 3-11 |
| 48 | <b>4 Longevity studies and Consolidation of the present CMS RPC subsystem</b> | 4-1  |
| 49 | 4.1 Resistive Plate Chambers at CMS . . . . .                                 | 4-1  |
| 50 | 4.1.1 Overview . . . . .  | 4-1  |
| 51 | 4.1.2 The present RPC system . . . . .  | 4-2  |
| 52 | 4.1.3 Pulse processing of CMS RPCs . . . . .                                  | 4-3  |
| 53 | 4.2 Testing detectors under extreme conditions . . . . .                      | 4-3  |
| 54 | 4.2.1 The Gamma Irradiation Facilities . . . . .                              | 4-6  |
| 55 | 4.2.1.1 GIF . . . . .   | 4-6  |
| 56 | 4.2.1.2 GIF++ . . . . .   | 4-8  |
| 57 | 4.3 Preliminary tests at GIF . . . . .  | 4-10 |
| 58 | 4.3.1 Resistive Plate Chamber test setup . . . . .                            | 4-10 |

---

|     |          |   |            |
|-----|----------|---|------------|
| 59  | 4.3.2    | Data Acquisition . . . . .  | 4-12       |
| 60  | 4.3.3    | Geometrical acceptance of the setup layout to cosmic muons . . . . .    | 4-12       |
| 61  | 4.3.3.1  | Description of the simulation layout . . . . .                          | 4-13       |
| 62  | 4.3.3.2  | Simulation procedure . . . . .  | 4-15       |
| 63  | 4.3.3.3  | Results . . . . .   | 4-16       |
| 64  | 4.3.4    | Photon flux at GIF . . . . .  | 4-16       |
| 65  | 4.3.4.1  | Expectations from simulations . . . . .                                 | 4-16       |
| 66  | 4.3.4.2  | Dose measurements . . . . .   | 4-21       |
| 67  | 4.3.5    | Results and discussions . . . . .                                       | 4-22       |
| 68  | 4.4      | Longevity tests at GIF++ . . . . .                                      | 4-23       |
| 69  | 4.4.1    | Description of the Data Acquisition . . . . .                           | 4-26       |
| 70  | 4.4.2    | RPC current, environmental and operation parameter monitoring . . . . . | 4-27       |
| 71  | 4.4.3    | Measurement procedure . . . . .   | 4-28       |
| 72  | 4.4.4    | Longevity studies results . . . . .                                     | 4-28       |
| 73  | <b>5</b> | <b>Investigation on high rate RPCs</b>                                  | <b>5-1</b> |
| 74  | 5.1      | Rate limitations and ageing of RPCs . . . . .                           | 5-1        |
| 75  | 5.1.1    | Low resistivity electrodes . . . . .                                    | 5-1        |
| 76  | 5.1.2    | Low noise front-end electronics . . . . .                               | 5-1        |
| 77  | 5.2      | Construction of prototypes . . . . .                                    | 5-1        |
| 78  | 5.3      | Results and discussions . . . . .                                       | 5-1        |
| 79  | <b>6</b> | <b>Conclusions and outlooks</b>   | <b>6-1</b> |
| 80  | 6.1      | Conclusions . . . . .   | 6-1        |
| 81  | 6.2      | Outlooks . . . . .  | 6-1        |
| 82  | <b>A</b> | <b>A data acquisition software for CAEN VME TDCs</b>                    | <b>A-1</b> |
| 83  | A.1      | GIF++ DAQ file tree . . . . .   | A-1        |
| 84  | A.2      | Usage of the DAQ . . . . .  | A-2        |
| 85  | A.3      | Description of the readout setup . . . . .                              | A-3        |
| 86  | A.4      | Data read-out . . . . .   | A-3        |
| 87  | A.4.1    | V1190A TDCs . . . . .   | A-4        |
| 88  | A.4.2    | DataReader . . . . .  | A-6        |
| 89  | A.4.3    | Data quality flag . . . . .   | A-10       |
| 90  | A.5      | Communications . . . . .  | A-12       |
| 91  | A.5.1    | V1718 USB Bridge . . . . .  | A-13       |
| 92  | A.5.2    | Configuration file . . . . .  | A-13       |
| 93  | A.5.3    | WebDCS/DAQ intercommunication . . . . .                                 | A-17       |
| 94  | A.5.4    | Example of inter-process communication cycle . . . . .                  | A-18       |
| 95  | A.6      | Software export . . . . .   | A-18       |
| 96  | <b>B</b> | <b>Details on the offline analysis package</b>                          | <b>B-1</b> |
| 97  | B.1      | GIF++ Offline Analysis file tree . . . . .                              | B-1        |
| 98  | B.2      | Usage of the Offline Analysis . . . . .                                 | B-2        |
| 99  | B.2.1    | Output of the offline tool . . . . .                                    | B-3        |
| 100 | B.2.1.1  | ROOT file . . . . .   | B-3        |
| 101 | B.2.1.2  | CSV files . . . . .   | B-5        |
| 102 | B.3      | Analysis inputs and information handling . . . . .                      | B-6        |
| 103 | B.3.1    | Dimensions file and IniFile parser . . . . .                            | B-6        |
| 104 | B.3.2    | TDC to RPC link file and Mapping . . . . .                              | B-7        |

---

|     |         |  |      |
|-----|---------|--|------|
| 105 | B.4     | Description of GIFT++ setup within the Offline Analysis tool . . . . . | B-9  |
| 106 | B.4.1   | RPC objects . . . . .  | B-9  |
| 107 | B.4.2   | Trolley objects . . . . .  | B-10 |
| 108 | B.4.3   | Infrastructure object . . . . .  | B-11 |
| 109 | B.5     | Handling of data . . . . .   | B-12 |
| 110 | B.5.1   | RPC hits . . . . .   | B-13 |
| 111 | B.5.2   | Clusters of hits . . . . .   | B-14 |
| 112 | B.6     | DAQ data Analysis . . . . .  | B-15 |
| 113 | B.6.1   | Determination of the run type . . . . .                                | B-16 |
| 114 | B.6.2   | Beam time window calculation for efficiency runs . . . . .             | B-17 |
| 115 | B.6.3   | Data loop and histogram filling . . . . .                              | B-18 |
| 116 | B.6.4   | Results calculation . . . . .  | B-19 |
| 117 | B.6.4.1 | Rate normalisation . . . . .   | B-19 |
| 118 | B.6.4.2 | Rate and activity . . . . .  | B-21 |
| 119 | B.6.4.3 | Strip masking tool . . . . .   | B-23 |
| 120 | B.6.4.4 | Output CSV files filling . . . . .                                     | B-25 |
| 121 | B.7     | Current data Analysis . . . . .  | B-29 |



<sup>122</sup>

## Nederlandse samenvatting –Summary in Dutch–

<sup>124</sup> Le resume en Neerlandais (j'aurais peut-être pu apprendre la langue juste pour ça...).

<sup>123</sup>



## English summary

<sup>126</sup> Le meme résume mais en Anglais (on commencera par la hein!).



# List of Figures

127

|     |     |  |     |
|-----|-----|--|-----|
| 128 | 2.1 | CERN accelerator complex. . . . .  | 2-2 |
| 129 | 2.2 | . . . . .  | 2-3 |
| 130 | 2.3 | . . . . .  | 2-4 |
| 131 | 2.4 | Absorbed dose in the CMS cavern after an integrated luminosity of 3000 fb. R is the transverse distance from the beamline and Z is the distance along the beamline from the Interaction Point at Z=0. . . . .  | 2-5 |
| 134 | 2.5 | A quadrant of the muon system, showing DTs (yellow), RPCs (light blue), and CSCs (green). The locations of new forward muon detectors for Phase-II are contained within the dashed box and indicated in red for GEM stations (ME0, GE1/1, and GE2/1) and dark blue for improved RPC (iRPC) stations (RE3/1 and RE4/1). . . . .   | 2-5 |
| 138 | 2.6 | RMS of the multiple scattering displacement as a function of muon $p_T$ for the proposed forward muon stations. All of the electromagnetic processes such as bremsstrahlung and magnetic field effect are included in the simulation. . . . .  | 2-6 |
| 141 | 3.1 | Different phases of the avalanche development in the RPC gas volume subjected to a constant electric field $E_0$ . a) An avalanche is initiated by the primary ionisation caused by the passage of a charged particle through the gas volume. b) Due to its growing size, the avalanche starts to locally influence the electric field. c) The electrons, lighter than the cations reach the anode first. d) The ions reach the cathode. While the charges have not recombined, the electric field in the small region around the avalanche stays affected and locally blind the detector. . . . . | 3-2 |
| 148 | 3.2 | Effeciency (circles and stars with 30 mV and 100 mV thresholds respectively) and streamer probability (opened circles) as function of the operating voltatge of a 2 mm single gap HPL RPC flushed with a gas mixture containing (a) 5%, (b) 2%, (c) 1% and (d) no $SF_6$ [33]. . . . .   | 3-3 |
| 152 | 3.3 | Movement of the charge carriers in an RPC. Figure 3.3a: Voltage across an RPC whose electrode have a relative permittivity of 5 at the moment the tension s applied. Figure 3.3b: After the charge carriers moved, the electrodes are charged and there is no voltage drop over the electrodes anymore. The full potential is applied on the gas gap only. Figure 3.3c: The streamer discharge initiated by a charged particle transports electrons and cations towards the anode and cathode respectively. . . . .  | 3-5 |
| 158 | 3.4 | Typical oscilloscope pulses in streamer mode (Figure 3.4a) and avalanche mode(Figure 3.4b). In the case of streamer mode, the very small avalanche signal is visible. . . . .  | 3-5 |
| 160 | 3.5 | Rate capability comparison for the streamer and avalanche mode of operation. An order of magnitude in rate capability for a maximal efficiency drop of 10% is gained by using the avalanche mode over the streamer mode. . . . .   | 3-6 |

|     |   |      |
|-----|---|------|
| 163 | 3.6 Possible double-gap RPC layouts: a) "standard" 1D double-gap RPC, as used in CMS experiment, where the anodes are facing each other and a 1D read-out plane is sandwiched in between them, b) double read-out double-gap RPC as used in ATLAS experiment, where the cathodes are facing each other and 2 read-out planes are used on the outer surfaces. This last layout can offer the possibility to use a 2D reconstruction by using orthogonal read-out planes. . . . .   | 3-7  |
| 164 |   |      |
| 165 |   |      |
| 166 |   |      |
| 167 |   |      |
| 168 |   |      |
| 169 | 3.7 Comparison of performance of CMS double and single gap RPCs using cosmic muons [44]. Figure 3.7a: Comparison of efficiency sigmoids. Figure 3.7b: Voltage distribution at 95% of maximum efficiency. Figure 3.7c: $\Delta_{10\%}^{90\%}$ distribution. . . . .  | 3-7  |
| 170 |   |      |
| 171 |   |      |
| 172 | 3.8 Presentation of ALICE MRPC using 250 $\mu\text{m}$ gas gaps, 620 $\mu\text{m}$ outer glass electrodes and 550 $\mu\text{m}$ inner floating electrodes. More details on the labels are given in [45]. . . . .  | 3-8  |
| 173 |   |      |
| 174 | 3.9 Particle identification applied to electrons in the STAR experiment. The identification is performed combining ToF and $dE/dx$ measurements [50]. . . . .   | 3-9  |
| 175 |   |      |
| 176 | 3.10 Comparison of the detector performance of ALICE ToF MRPC [51] at fixed applied voltage (in blue) and at fixed effective voltage (in red). The effective voltage is kept fixed by increasing the applied voltage accordingly to the current drawn by the detector.  | 3-9  |
| 177 |   |      |
| 178 |   |      |
| 179 | 3.11 Ratio between total induced and drifting charge have been simulated for single gap, double-gap and multigap layouts [52]. The total induced charge for a double-gap RPC is a factor 2 higher than for a multigap. . . . .  | 3-10 |
| 180 |   |      |
| 181 |   |      |
| 182 | 3.12 Charge spectra have been simulated for single gap, double-gap and multigap layouts [52]. It appears that when single gap shows a decreasing spectrum, double and multigap layouts exhibit a spectrum whose peak is detached from the origin. The detachment gets stronger as the number of gaps increases. . . . .   | 3-10 |
| 183 |   |      |
| 184 |   |      |
| 185 |   |      |
| 186 | 3.13 The maximal theoretical efficiency is simulated for single gap, double-gap and multigap layouts [52] at a constant gap thickness of 2 mm and using an effective Townsend coefficient of 9 $\text{mm}^{-1}$ . . . . .   | 3-11 |
| 187 |   |      |
| 188 |   |      |
| 189 | 4.1 Signals from the RPC strips are shaped by the FEE described on Figure 4.1a. Output LVDS signals are then read-out by a TDC module connected to a computer or converted into NIM and sent to scalers. Figure 4.1b describes how these converted signals are put in coincidence with the trigger. . . . .   | 4-3  |
| 190 |   |      |
| 191 |   |      |
| 192 |   |      |
| 193 | 4.2 Description of the principle of a CFD. A comparison of threshold triggering (left) and constant fraction triggering (right) is shown in Figure 4.2a. Constant fraction triggering is obtained thanks to zero-crossing technique as explained in Figure 4.2b. The signal arriving at the input of the CFD is split into three components. A first one is delayed and connected to the inverting input of a first comparator. A second component is connected to the noninverting input of this first comparator. A third component is connected to the noninverting input of another comparator along with a threshold value connected to the inverting input. Finally, the output of both comparators is fed through an AND gate. . . . . | 4-4  |
| 194 |   |      |
| 195 |   |      |
| 196 |   |      |
| 197 |   |      |
| 198 |   |      |
| 199 |   |      |
| 200 |   |      |
| 201 |   |      |
| 202 | 4.3 (4.3a) Extrapolation from 2016 data of single hit rate per unit area in the barrel region. (4.3b) Extrapolation from 2016 data of single hit rate per unit area in the endcap region. . . . .   | 4-5  |
| 203 |   |      |
| 204 |   |      |
| 205 | 4.4 Background Fluka simulation compared to 2016 Data at $L = 10^{34} \text{cm}^{-2} \cdot \text{s}^{-1}$ in the fourth endcap disk region. A mismatch in between simulation and data can be observed. [To be understood.] . . . . .  | 4-6  |
| 206 |   |      |
| 207 |   |      |

|     |      |   |      |
|-----|------|---|------|
| 208 | 4.5  | Layout of the test beam zone called X5c GIF at CERN. Photons from the radioactive source produce a sustained high rate of random hits over the whole area. The zone is surrounded by 8 m high and 80 cm thick concrete walls. Access is possible through three entry points. Two access doors for personnel and one large gate for material. A crane allows installation of heavy equipment in the area. . . . .  | 4-7  |
| 213 | 4.6  | $^{137}\text{Cs}$ decays by $\beta^-$ emission to the ground state of $^{137}\text{Ba}$ (BR = 5.64%) and via the 662 keV isomeric level of $^{137}\text{Ba}$ (BR = 94.36%) whose half-life is 2.55 min. . . . .   | 4-8  |
| 215 | 4.7  | Floor plan of the GIF++ facility. When the facility downstream of the GIF++ takes electron beam, a beam pipe is installed along the beam line (z-axis). The irradiator can be displaced laterally (its center moves from $x = 0.65$ m to 2.15 m), to increase the distance to the beam pipe. . . . .  | 4-8  |
| 219 | 4.8  | Simulated unattenuated current of photons in the xz plane (Figure 4.8a) and yz plane (Figure 4.8b) through the source at $x = 0.65$ m and $y = 0$ m. With angular correction filters, the current of 662 keV photons is made uniform in xy planes. . . . .  | 4-9  |
| 222 | 4.9  | Description of the RPC setup. Dimensions are given in mm. A tent containing RPCs is placed at 1720 mm from the source container. The source is situated in the center of the container. RE-4-2-BARC-161 chamber is 160 mm inside the tent. This way, the distance between the source and the chambers plan is 2060 mm. Figure 4.9a provides a side view of the setup in the xz plane while Figure 4.9b shows a top view in the yz plane. . . . .  | 4-10 |
| 228 | 4.10 | RE-4-2-BARC-161 chamber is inside the tent as described in Figure 4.9. In the top right, the two scintillators used as trigger can be seen. This trigger system has an inclination of 10° relative to horizontal and is placed above half-partition B2 of the RPCs. PMT electronics are shielded thanks to lead blocks placed in order to protect them without stopping photons from going through the scintillators and the chamber. . . . .   | 4-11 |
| 233 | 4.11 | Hit distributions over all 3 partitions of RE-4-2-BARC-161 chamber is showed on these plots. Top, middle and bottom figures respectively correspond to partitions A, B, and C. These plots show that some events still occur in other half-partitions than B2, which corresponds to strips 49 to 64, in front of which the trigger is placed, contributing to the inefficiency of detection of cosmic muons. In the case of partitions A and C, the very low amount of data can be interpreted as noise. On the other hand, it is clear that a little portion of muons reach the half-partition B1, corresponding to strips 33 to 48. . . . . | 4-12 |
| 241 | 4.12 | Results are derived from data taken on half-partition B2 only. On the 18 <sup>th</sup> of June 2014, data has been taken on chamber RE-2-BARC-161 at building 904 (Prevessin Site) with cosmic muons providing us a reference efficiency plateau of $(97.54 \pm 0.15)\%$ represented by a black curve. A similar measurement has been done at GIF on the 21 <sup>st</sup> of July with the same chamber giving a plateau of $(78.52 \pm 0.94)\%$ represented by a red curve. . . . .  | 4-13 |
| 247 | 4.13 | Representation of the layout used for the simulations of the test setup. The RPC is represented as a yellow trapezoid while the two scintillators as blue cuboids looking at the sky. A green plane corresponds to the muon generation plane within the simulation. Figure 4.9a shows a global view of the simulated setup. Figure 4.9b shows a zommed view that allows to see the 2 scintillators as well as the full RPC plane. . . . .   | 4-14 |
| 252 | 4.14 | $\gamma$ flux $F(D)$ is plot using values from table 4.1. As expected, the plot shows similar attenuation behaviours with increasing distance for each absorption factors. . . . .  | 4-17 |

|     |      |  |      |
|-----|------|--|------|
| 254 | 4.15 | Figure 4.15a shows the linear approximation fit done via formulae 4.7 on data from<br>table 4.2. Figure 4.15b shows a comparison of this model with the simulated flux<br>using a and b given in figure 4.15a in formulae 4.4 and the reference value $D_0 =$<br>$50\text{cm}$ and the associated flux for each absorption factor $F_0^{ABS}$ from table 4.1 . . . . .   | 4-19 |
| 258 | 4.16 | Dose measurements has been done in a plane corresponding to the tents front side.<br>This plan is 1900 mm away from the source. As explained in the first chapter, a<br>lens-shaped lead filter provides a uniform photon flux in the vertical plan orthogonal<br>to the beam direction. If the second line of measured fluxes is not taken into account<br>because of lower values due to experimental equipments in the way between the<br>source and the tent, the uniformity of the flux is well showed by the results. . . . .  | 4-21 |
| 264 | 4.17 | . . . . .  | 4-22 |
| 265 | 4.18 | Evolution of the maximum efficiency for RE2 (4.18a) and RE4 (4.18b) chambers<br>with increasing extrapolated $\gamma$ rate per unit area at working point. Both irradiated<br>(blue) and non irradiated (red) chambers are shown. . . . .  | 4-24 |
| 266 | 4.19 | Evolution of the working point for RE2 (4.19a) and RE4 (4.19b) with increasing<br>extrapolated $\gamma$ rate per unit area at working point. Both irradiated (blue) and non<br>irradiated (red) chambers are shown. . . . .  | 4-24 |
| 271 | 4.20 | Evolution of the maximum efficiency at HL-LHC conditions, i.e. a background hit<br>rate per unit area of $300\text{ Hz/cm}^2$ , with increasing integrated charge for RE2 (4.20a)<br>and RE4 (4.20b) detectors. Both irradiated (blue) and non irradiated (red) chambers<br>are shown. The integrated charge for non irradiated detectors is recorded during test<br>beam periods and stays small with respect to the charge accumulated in irradiated<br>chambers. . . . .  | 4-25 |
| 277 | 4.21 | Comparison of the efficiency sigmoid before (triangles) and after (circles) irradiation<br>for RE2 (4.21a) and RE4 (4.21b) detectors. Both irradiated (blue) and non irradiated<br>(red) chambers are shown. . . . .   | 4-25 |
| 280 | 4.22 | Evolution of the Bakelite resistivity for RE2 (4.22a) and RE4 (4.22b) detectors. Both<br>irradiated (blue) and non irradiated (red) chambers are shown. . . . .  | 4-26 |
| 282 | 4.23 | Evolution of the noise rate per unit area for the irradiated chamber RE2-2-BARC-9<br>only. . . . .   | 4-26 |
| 284 | A.1  | (A.1a) View of the front panel of a V1190A TDC module [58]. (A.1b) View of the<br>front panel of a V1718 Bridge module [59]. (A.1c) View of the front panel of a 6U<br>6021 VME crate [60]. . . . .  | A-3  |
| 287 | A.2  | Module V1190A <i>Trigger Matching Mode</i> timing diagram [58]. . . . .  | A-4  |
| 288 | A.3  | Structure of the ROOT output file generated by the DAQ. The 5 branches ( <code>EventNumber</code> ,<br><code>number_of_hits</code> , <code>Quality_flag</code> , <code>TDC_channel</code> and <code>TDC_TimeStamp</code> ) are visible on<br>the left panel of the ROOT browser. On the right panel is visible the histogram cor-<br>responding to the variable <code>nHits</code> . In this specific example, there were approximately<br>50k events recorded to measure the gamma irradiation rate on the detectors. Each<br>event is stored as a single entry in the <code>TTree</code> . . . . . | A-10 |
| 294 | A.4  | The effect of the quality flag is explained by presenting the content of <code>TBranch</code><br><code>number_of_hits</code> of a data file without <code>Quality_flag</code> in Figure A.4a and the con-<br>tent of the same <code>TBranch</code> for data corresponding to a <code>Quality_flag</code> where all TDCs<br>were labelled as <code>GOOD</code> in Figure A.4b taken with similar conditions. It can be noted<br>that the number of entries in Figure A.4b is slightly lower then in Figure A.4a due<br>to the excluded events. . . . .  | A-12 |

---

|     |     |  |      |
|-----|-----|--|------|
| 300 | A.5 | Using the same data as previously showed in Figure A.4, the effect of the quality flag is explained by presenting the reconstructed hit multiplicity of a data file without <code>Quality_flag</code> in Figure A.5a and the reconstructed content of the same RPC partition for data corresponding to a <code>Quality_flag</code> where all TDCs were labelled as <code>GOOD</code> in Figure A.5b taken with similar conditions. The artificial high content of bin 0 is completely suppressed. . . . .  | A-12 |
| 301 |     |  |      |
| 302 |     |  |      |
| 303 |     |  |      |
| 304 |     |  |      |
| 305 |     |  |      |
| 306 | A.6 | WebDCS DAQ scan page. On this page, shifters need to choose the type of scan (Rate, Efficiency or Noise Reference scan), the gamma source configuration at the moment of data taking, the beam configuration, and the trigger mode. These information will be stored in the DAQ ROOT output. Are also given the minimal measurement time and waiting time after ramping up of the detectors is over before starting the data acquisition. Then, the list of HV points to scan and the number of triggers for each run of the scan are given in the table underneath. . . . .   | A-14 |
| 307 |     |  |      |
| 308 |     |  |      |
| 309 |     |  |      |
| 310 |     |  |      |
| 311 |     |  |      |
| 312 |     |  |      |
| 313 | B.1 | Example of expected hit time distributions in the cases of efficiency (Figure B.1a) and noise/gamma rate per unit area (Figure B.1b) measurements as extracted from the raw ROOT files. The unit along the x-axis corresponds to ns. The fact that "the" muon peak is not well defined in Figure B.1a is due to the contribution of all the RPCs being tested at the same time that don't necessarily have the same signal arrival time. Each individual peak can have an offset with the ones of other detectors. The inconsistancy in the first 100 ns of both time distributions is an artefact of the TDCs and are systematically rejected during the analysis. . . . .  | B-16 |
| 314 |     |  |      |
| 315 |     |  |      |
| 316 |     |  |      |
| 317 |     |  |      |
| 318 |     |  |      |
| 319 |     |  |      |
| 320 |     |  |      |
| 321 | B.2 | The effect of the quality flag is explained by presenting the reconstructed hit multiplicity of a data file without <code>Quality_flag</code> . The artificial high content of bin 0 is the effect of corrupted data. . . . .  | B-19 |
| 322 |     |  |      |
| 323 |     |  |      |
| 324 | B.3 | Display of the masking tool page on the webDCS. The window on the left allows the shifter to edit <code>ChannelsMapping.csv</code> . To mask a channel, it only is needed to set the 3rd field corresponding to the strip to mask to 0. It is not necessary for older mapping file formats to add a 1 for each strip that is not masked as the code is versatile and the default behaviour is to consider missing mask fields as active strips. The effect of the mask is directly visible for noisy channels as the corresponding bin turns red. The global effect of masking strips will be an update of the rate value showed on the histogram that will take into consideration the rejected channels. . . . . | B-24 |
| 325 |     |  |      |
| 326 |     |  |      |
| 327 |     |  |      |
| 328 |     |  |      |
| 329 |     |  |      |
| 330 |     |  |      |
| 331 |     |  |      |



## List of Tables

332

|     |  |      |
|-----|--|------|
| 333 | 3.1 Properties of the most used electrode materials for RPCs. . . . .                                | 3-4  |
| 334 | 4.1 Total photon flux ( $E\gamma \leq 662$ keV) with statistical error predicted considering a       |      |
| 335 | $^{137}\text{Cs}$ activity of 740 GBq at different values of the distance $D$ to the source along    |      |
| 336 | the x-axis of irradiation field [55]. . . . .  | 4-16 |
| 337 | 4.2 Correction factor $c$ is computed thanks to formulae 4.5 taking as reference $D_0 =$             |      |
| 338 | 50 cm and the associated flux $F_0^{ABS}$ for each absorption factor available in table 4.1. .       | 4-18 |
| 339 | 4.3 The data at $D_0$ in 1997 is taken from [55]. In a second step, using Equations 4.8              |      |
| 340 | and 4.9, the flux at $D$ can be estimated in 1997. Then, taking into account the                     |      |
| 341 | attenuation of the source activity, the flux at $D$ can be estimated at the time of the              |      |
| 342 | tests in GIF in 2014. Finally, assuming a sensitivity of the RPC to $\gamma$ $s = 2 \cdot 10^{-3}$ , |      |
| 343 | an estimation of the hit rate per unit area is obtained. . . . .                                     | 4-20 |
| 344 | A.1 Inter-process communication cycles in between the webDCS and the DAQ through                     |      |
| 345 | file string signals. . . . .   | A-19 |



346

## List of Acronyms

347

### List of Acronyms

348

349

#### A

350

351

352 AFL

Almost Full Level

353

354

#### B

355

356

357 BARC

Bhabha Atomic Research Centre

358 BLT

Block Transfer

359 BR

Branching Ratio

360

361

#### C

362

363

364 CAEN

Costruzioni Apparecchiature Elettroniche Nucleari S.p.A.

365 CERN

European Organization for Nuclear Research

366 CFD

Constant Fraction Discriminator

367 CMS

Compact Muon Solenoid

368 CSC

Cathode Strip Chamber

369

370

#### D

371

372

373 DAQ

Data Acquisition

374 DCS

Detector Control Software

375 DQM

Data Quality Monitoring

376 DT

Drift Tube

377

378

#### F

379

380

|     |          |  |
|-----|----------|--|
| 381 | FEE      | Front-End Electronics  |
| 382 | FEB      | Front-End Board  |
| 383 |          |  |
| 384 |          |  |
| 385 | <b>G</b> |  |
| 386 |          |  |
| 387 | GE-/-    | Find a good description  |
| 388 | GE1/1    | Find a good description  |
| 389 | GE2/1    | Find a good description  |
| 390 | GEANT    | GEometry ANd Tracking - a series of software toolkit platforms developed by CERN |
| 391 |          |  |
| 392 | GEM      | Gas Electron Multiplier  |
| 393 | GIF      | Gamma Irradiation Facility   |
| 394 | GIF++    | new Gamma Irradiation Facility   |
| 395 |          |  |
| 396 |          |  |
| 397 | <b>H</b> |  |
| 398 |          |  |
| 399 | HL-LHC   | High Luminosity LHC  |
| 400 | HPL      | High-pressure laminate   |
| 401 | HV       | High Voltage   |
| 402 |          |  |
| 403 |          |  |
| 404 | <b>I</b> |  |
| 405 |          |  |
| 406 | iRPC     | improved RPC   |
| 407 | IRQ      | Interrupt Request  |
| 408 | ISR      | Intersecting Storage Rings   |
| 409 |          |  |
| 410 |          |  |
| 411 | <b>L</b> |  |
| 412 |          |  |
| 413 | LEP      | Large Electron-Positron  |
| 414 | LHC      | Large Hadron Collider  |
| 415 | LS1      | First Long Shutdown  |
| 416 | LS3      | Third Long Shutdown  |
| 417 | LV       | Low Voltage  |
| 418 | LVDS     | Low-Voltage Differential Signaling   |
| 419 |          |  |
| 420 |          |  |
| 421 | <b>M</b> |  |
| 422 |          |  |
| 423 | MC       | Monte Carlo  |

|     |              |  |
|-----|--------------|--|
| 424 | <b>MCNP</b>  | Monte Carlo N-Particle                       |
| 425 | <b>ME-/</b>  | Find good description                        |
| 426 | <b>ME0</b>   | Find good description                        |
| 427 | <b>MRPC</b>  | Multigap RPC                                 |
| 428 |              |  |
| 429 |              |  |
| 430 | <b>N</b>     |  |
| 431 |              |  |
| 432 | <b>NIM</b>   | Nuclear Instrumentation Module logic signals |
| 433 |              |  |
| 434 |              |  |
| 435 | <b>P</b>     |  |
| 436 |              |  |
| 437 | <b>PS</b>    | Proton Synchrotron                           |
| 438 | <b>PMT</b>   | PhotoMultiplier Tube                         |
| 439 |              |  |
| 440 |              |  |
| 441 | <b>R</b>     |  |
| 442 |              |  |
| 443 | <b>RE-/</b>  | Find a good description                      |
| 444 | <b>RE2/2</b> | Find a good description                      |
| 445 | <b>RE3/1</b> | Find a good description                      |
| 446 | <b>RE3/2</b> | Find a good description                      |
| 447 | <b>RE4/1</b> | Find a good description                      |
| 448 | <b>RE4/2</b> | Find a good description                      |
| 449 | <b>RE4/3</b> | Find a good description                      |
| 450 | <b>RMS</b>   | Root Mean Square                             |
| 451 | <b>ROOT</b>  | a framework for data processing born at CERN |
| 452 | <b>RPC</b>   | Resistive Plate Chamber                      |
| 453 |              |  |
| 454 |              |  |
| 455 | <b>S</b>     |  |
| 456 |              |  |
| 457 | <b>SC</b>    | Synchrocyclotron                             |
| 458 | <b>SPS</b>   | Super Proton Synchrotron                     |
| 459 |              |  |
| 460 |              |  |
| 461 | <b>T</b>     |  |
| 462 |              |  |
| 463 | <b>TDC</b>   | Time-to-Digital Converter                    |
| 464 | <b>ToF</b>   | Time-of-flight                               |
| 465 |              |  |
| 466 |              |  |
| 467 | <b>W</b>     |  |
| 468 |              |  |

<sup>469</sup> webDCS      Web Detector Control System

# 1

## Introduction

470

471

<sup>472</sup> **1.1 A story of High Energy Physics**

<sup>473</sup> **1.2 Organisation of this study**



# 2

474

475

## Investigating the TeV scale

### 476 2.1 The Standard Model of Particle Physics

### 477 2.2 The Large Hadron Collider & the Compact Muon Solenoid

478 Throughout its history, CERN has played a leading role in high energy particle physics. Large  
479 regional facilities such as CERN were thought after the second world war in an attempt to increase  
480 international scientific collaboration and allows scientists to share the forever increasing costs of  
481 experiment facilities required due to the need for increasing the energy in the center of mass to  
482 deeper probe matter. The construction of the first accelerators at the end of the 50s, the Synchro-  
483 cyclotron (SC) and the Proton Synchrotron (PS), was directly followed by the first observation of  
484 antinuclei in 1965 [1]. Strong from the experience of the Intersecting Storage Rings (ISR), the very  
485 first proton-proton collider that showed hints that protons are not elementary particles, the Super  
486 Proton Synchrotron (SPS) was built in the 70s to investigate the structure of protons, the preference  
487 for matter over antimatter, the state of matter in the early universe or exotic particles, and lead to  
488 the discovery in 1983 of the W and Z bosons [2–5]. These newly discovered particles and the elec-  
489 troweak interaction would then be studied in details by the Large Electron-Positron (LEP) collider  
490 that will help to prove in 1989 that there only are three generations of elementary particles [6]. The  
491 LEP would then be dismantled in 2000 to allow for the LHC to be constructed in the existing tunnel.

#### 492 2.2.1 LHC, the most powerful particle accelerator

493 The LHC has always been considered as an option to the future of CERN. At the moment of the  
494 construction of the LEP beneath the border between France and Switzerland, the tunnel was built in  
495 order to accomodate what would be a Large Hadron Collider with a dipole field of 10 T and a beam  
496 energy in between 8 and 9 TeV [7] directly followed in 1985 with the creation of a 'Working Group  
497 on the Scientific and Technological Future of CERN' to investigate such a collider [8]. The decision  
498 was finally taken almost 10 years later, in 1994, to construct the LHC in the LEP tunnel [9] and the

499 approval of the 4 main experiments that would take place at the 4 interaction points would come in  
500 1997 [10] and 1998 [11]:

- 501 • ALICE [12] has been designed in the purpose of studying quark-gluon plasma that is believed  
502 to have been a state of matter that existed in the very first moment of the universe.
- 503 • ATLAS [13] and CMS [14] are general purpose experiments that have been designed with  
504 the goal of continuing the exploration of the Standard Model and investigate new physics.
- 505 • LHCb [15] has been designed to investigate the preference of matter over antimatter in the  
506 universe through the CP violation.

507 These large scale experiments, as well as the full CERN accelerator complex, are displayed on  
508 Figure 2.1.

### CERN's Accelerator Complex

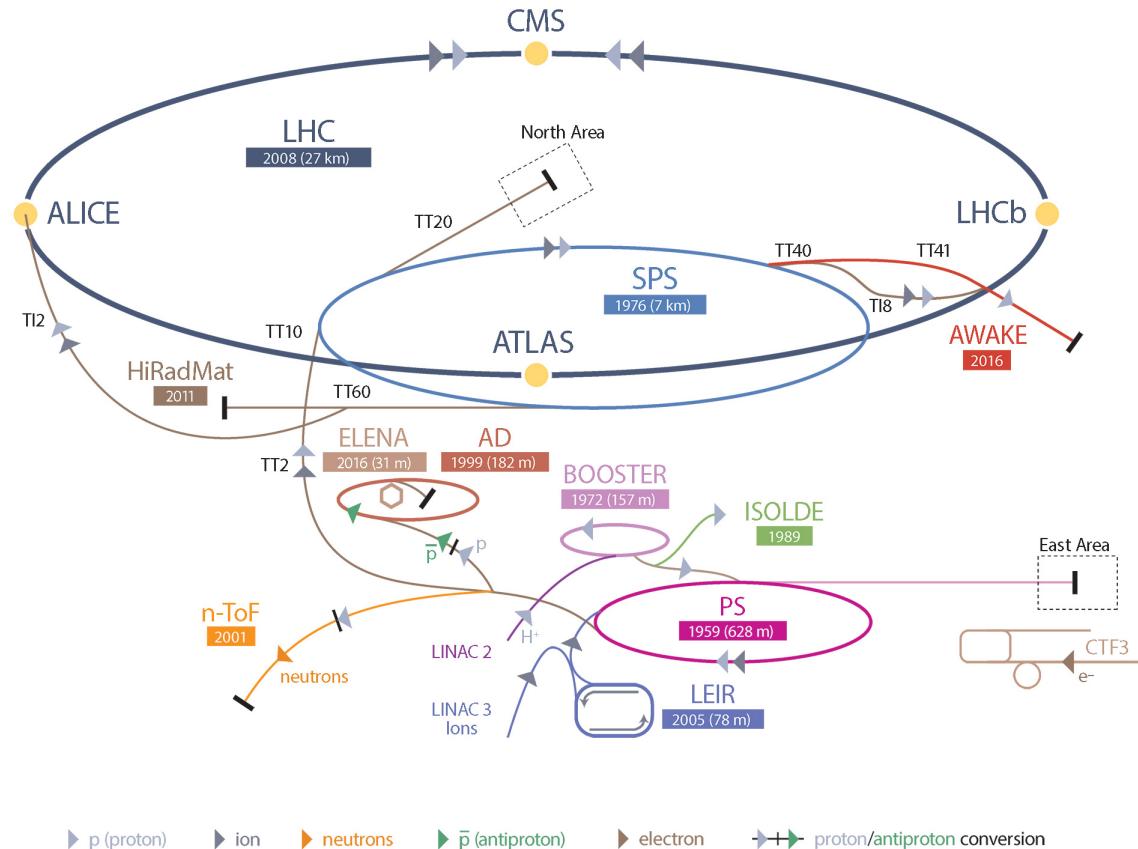


Figure 2.1: CERN accelerator complex.

509 The LHC is a 27 km long hadron collider and the most powerful accelerator used for particle  
510 physics since 2008 [16]. The LHC was originally designed to collide protons at a center-of-mass

511 energy of 14 TeV and luminosity of  $10^{34} \text{ cm}^{-2}\text{s}^{-1}$ , as well as  $Pb$  ions at a center-of-mass energy of  
512 2.8 TeV/A with a peak luminosity of  $10^{27} \text{ cm}^{-2}\text{s}^{-1}$ . Run 1 of LHC was enough for both CMS and  
513 ATLAS to discover the Higgs boson [17] and for LHCb to discover pentaquarks [18] and confirm the  
514 existance of tetraquarks [19]. Nevertheless, after the Third Long Shutdown (LS3) (2024-2026), the  
515 accelerator will be in the so called High Luminosity LHC (HL-LHC) configuration [20], increasing  
516 its instantaneous luminosity to  $10^{35} \text{ cm}^{-2}\text{s}^{-1}$  for  $pp$  collisions and to  $4.5 \times 10^{27} \text{ cm}^{-2}\text{s}^{-1}$ , boosting  
517 the discovery potential of the LHC.

518 **2.2.1.1 Overview of LHC technology**

519 The two parrallel LHC beams are contained in a single twin-bore magnet because of the space re-  
520 striction in the LEP tunnel that couldn't allow 2 completely separate proton rings to be built next  
521 to each other. The dipoles of the twin-bore magnets are showed in Figure 2.2 alongside the mag-  
522 netic field generated along the dipole section to accelerate the particles. Quadrupoles, presented in  
523 Figure 2.3, are also used to focus to the beams.

524 **2.2.2 CMS, a multipurpose experiment**

525 **2.3 Muon Phase-II Upgrade**

526 After the more than two years lasting First Long Shutdown (LS1), the Large Hadron Collider (LHC)  
527 delivered its very first Run-II proton-proton collisions early 2015. LS1 gave the opportunity to the  
528 LHC and to the its experiments to undergo upgrades. The accelerator is now providing collisions  
529 at center-of-mass energy of 13 TeV and bunch crossing rate of 40 MHz, with a peak luminosity  
530 exceeding its design value. During the first and upcoming second LHC Long Shutdown, the Compact  
531 Muon Solenoid (CMS) detector is also undergoing a number of upgrades to maintain a high system  
532 performance [21].

533 From the LHC Phase-2 or High Luminosity LHC (HL-LHC) period onwards, i.e. past the Third  
534 Long Shutdown (LS3), the performance degradation due to integrated radiation as well as the average  
535 number of inelastic collisions per bunch crossing, or pileup, will rise substantially and become a  
536 major challenge for the LHC experiments, like CMS that are forced to address an upgrade program  
537 for Phase-II [22]. Simulations of the expected distribution of absorbed dose in the CMS detector  
538 under HL-LHC conditions, show in figure 4.16 that detectors placed close to the beamline will have  
539 to withstand high irradiation, the radiation dose being of the order of a few tens of Gy.

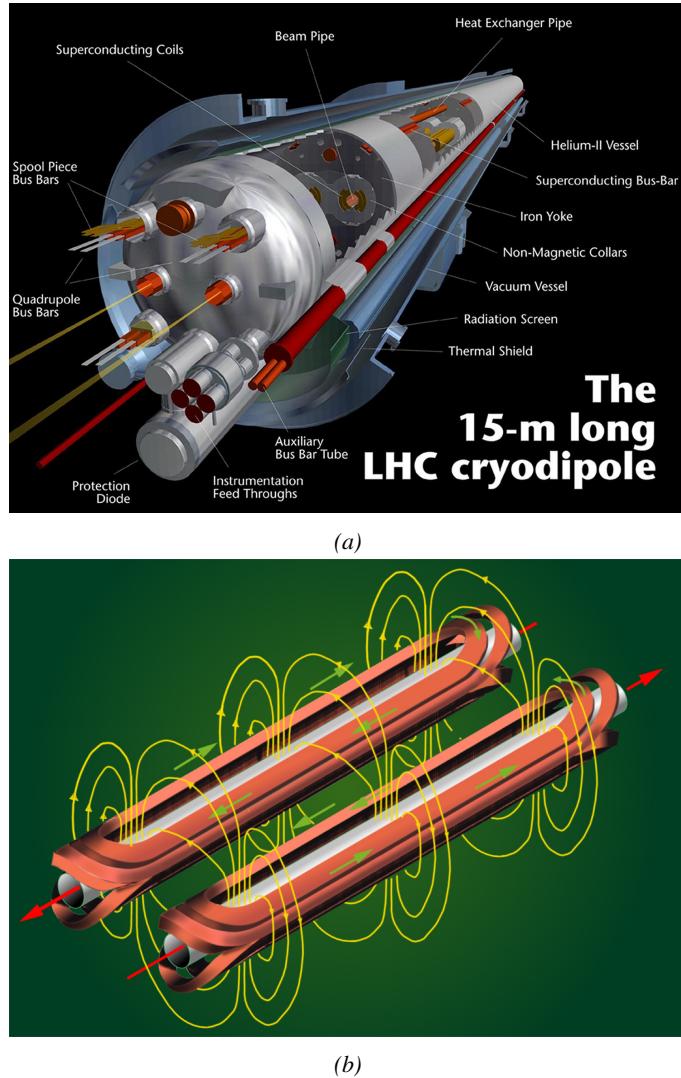


Figure 2.2: Figure 2.2a: schematics of the LHC cryodipoles. Figure 2.2b: magnetic field and resulting motion force applied on the beam particles.

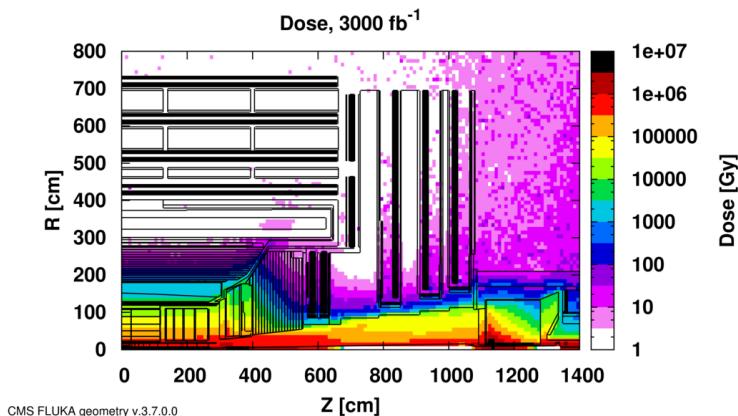


Figure 2.4: Absorbed dose in the CMS cavern after an integrated luminosity of  $3000 \text{ fb}^{-1}$ .  $R$  is the transverse distance from the beamline and  $Z$  is the distance along the beamline from the Interaction Point at  $Z=0$ .

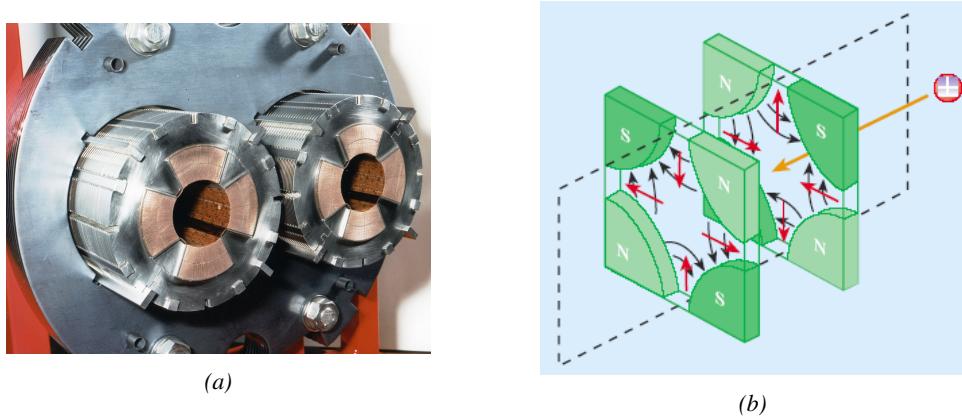


Figure 2.3: Figure 2.3a: picture of the LHC quadrupoles. Figure 2.3b: magnetic fields and resulting focussing force applied on the beam bunches by 2 consecutive quadrupoles.

540 The measurement of small production cross-section and/or decay branching ratio processes, such  
 541 as the Higgs boson coupling to charge leptons or the  $B_s \rightarrow \mu^+ \mu^-$  decay, is of major interest and  
 542 specific upgrades in the forward regions of the detector will be required to maximize the physics  
 543 acceptance on the largest possible solid angle. To ensure proper trigger performance within the  
 544 present coverage, the muon system will be completed with new chambers. In figure 2.5 one can  
 545 see that the existing Cathode Strip Chambers (CSCs) will be completed by Gas Electron Multipliers  
 546 (GEMs) and Resistive Plate Chambers (RPCs) in the pseudo-rapidity region  $1.6 < |\eta| < 2.4$  to  
 547 complete its redundancy as originally scheduled in the CMS Technical Proposal [23].

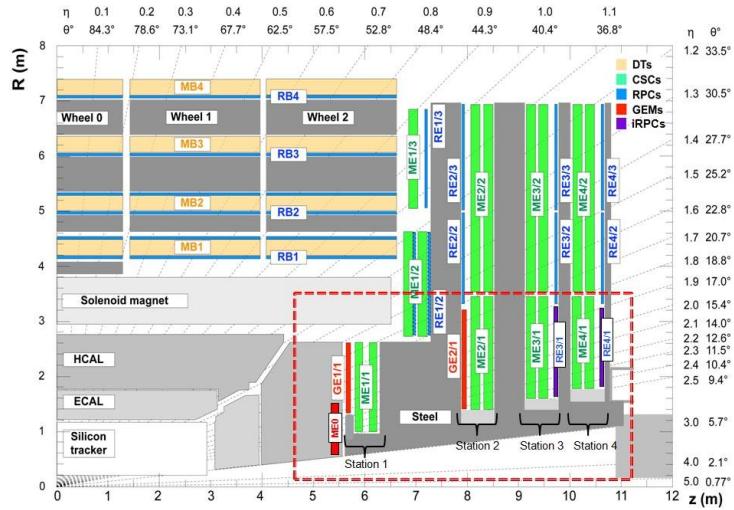


Figure 2.5: A quadrant of the muon system, showing DTs (yellow), RPCs (light blue), and CSCs (green). The locations of new forward muon detectors for Phase-II are contained within the dashed box and indicated in red for GEM stations (ME0, GE1/1, and GE2/1) and dark blue for improved RPC (iRPC) stations (RE3/1 and RE4/1).

548 RPCs are used by the CMS first level trigger for their good timing performances. Indeed, a very

good bunch crossing identification can be obtained with the present CMS RPC system, given their fast response of the order of 1 ns. In order to contribute to the precision of muon momentum measurements, muon chambers should have a spatial resolution less or comparable to the contribution of multiple scattering [21]. Most of the plausible physics is covered only considering muons with  $p_T < 100$  GeV thus, in order to match CMS requirements, a spatial resolution of  $\mathcal{O}(\text{few mm})$  the proposed new RPC stations, as shown by the simulation in figure 2.6. According to preliminary designs, RE3/1 and RE4/1 readout pitch will be comprised between 3 and 6 mm and 5  $\eta$ -partitions could be considered.

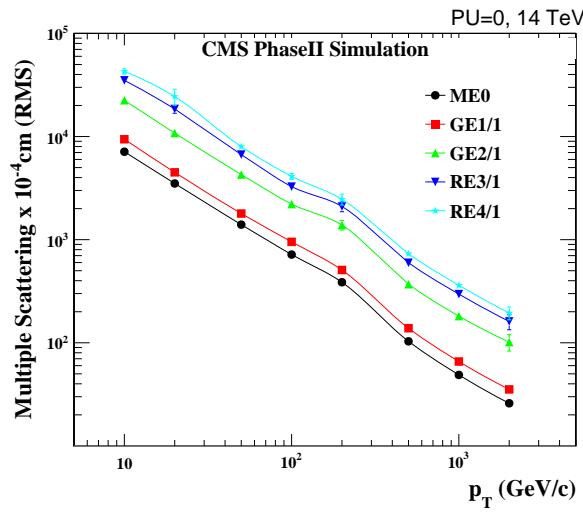


Figure 2.6: RMS of the multiple scattering displacement as a function of muon  $p_T$  for the proposed forward muon stations. All of the electromagnetic processes such as bremsstrahlung and magnetic field effect are included in the simulation.

# 3

557

558

## Physics of Resistive plate chambers

559 A Resistive Plate Chamber (RPC) is a gaseous detector using the same physical processes described  
560 in Chapter 3. It has been developed in 1981 by Santonico and Cardarelli [24], under the name of  
561 *Resistive Plate Counter*, as an alternative to the local-discharge spark counters proposed in 1978  
562 by Pestov and Fedotovich [25, 26]. Working with spark chambers implied using high-pressure gas  
563 and high mechanical precision which the RPC simplified by formerly using a gas mixture of argon  
564 and butane flowed at atmospheric pressure and a constant and uniform electric field propagated  
565 in between two parallel electrode plates. Moreover, a significant increase in rate capability was  
566 introduced by the use of electrode plate material with high bulk resistivity, preventing the discharge  
567 from growing throughout the whole gas gap. Indeed, the effect of using resistive electrodes is that  
568 the constant electric field is locally canceled out by the development of the discharge, limiting its  
569 growth.

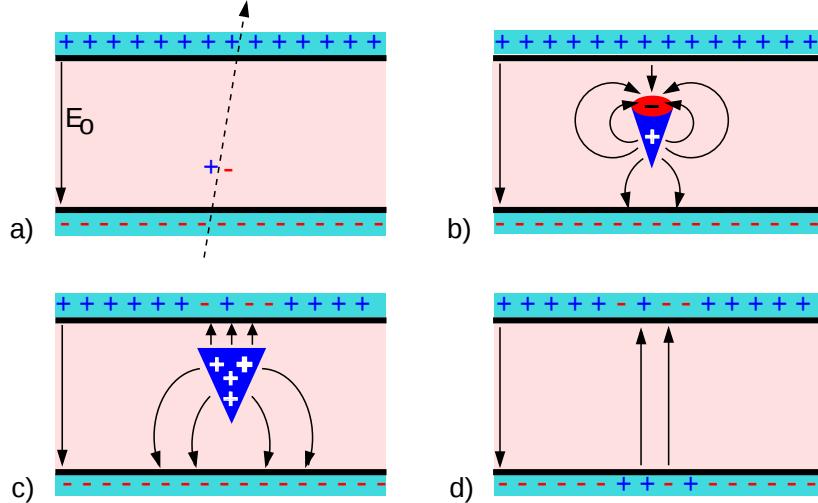
570 Through its development history, different operating modes [27–29] and new detector designs [30–  
571 32] have been discovered, leading to further improvement of the rate capability of such a detector.  
572 Moreover, the addition of  $SF_6$  into the gas mix improved the stability of operation of the RPC [33,  
573 34].

574 The low developing costs and easily achievable large detection areas offered by RPCs, as well as  
575 the wide range of possible designs, made them a natural choice to as muon chambers and/or trigger  
576 detectors in multipurpose experiments such as CMS [21] or ATLAS [35], time-of-flight detectors in  
577 ALICE [36], calorimeter with CALICE [37] or even detectors for volcanic muography with ToMu-  
578 Vol [38].

### 579 3.1 Principle

580 RPCs are ionisation detectors composed of two parallel resistive plate electrodes in between which  
581 a constant electric field is set. The space in between the electrodes, referred as *gap*, is filled with a  
582 dense gas that is used to generate primary ionization into the gas volume. The free charge carriers  
583 (electrons and cations) created by the ionization of the gas molecules are then accelerated towards

584 the electrodes by the electric field, as shown in Figure 3.1 [39].



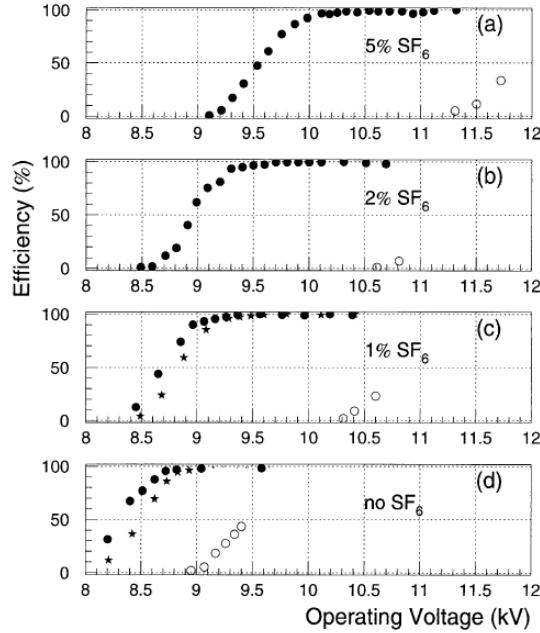
585 *Figure 3.1: Different phases of the avalanche development in the RPC gas volume subjected to a constant  
586 electric field  $E_0$ . a) An avalanche is initiated by the primary ionisation caused by the passage of a charged  
587 particle through the gas volume. b) Due to its growing size, the avalanche starts to locally influence the electric  
588 field. c) The electrons, lighter than the cations reach the anode first. d) The ions reach the cathode. While  
589 the charges have not recombined, the electric field in the small region around the avalanche stays affected and  
590 locally blind the detector.*

585 RPCs being passive detectors, a current on pick-up copper read-out placed outside of the gas  
586 volume is induced by the charge accumulation during the growth of the avalanche. As a result,  
587 the time resolution of the detector is substantially increased as the output signal is generated while  
588 the electrons are still in movement. The advantage of a constant electric field, over multi-wire  
589 proportional chambers, is that the electrons are being fully accelerated from the moment charge  
590 carriers are freed and feel the full strength of the electric field that doesn't depend on the distance to  
591 the readout and that the output signal doesn't need for the electrons to be physically collected.

592 The typical gas mixture RPCs are operated with is generally composed of 3 gas compounds.

- 593 • Tetrafluoroethane ( $C_2F_4H_2$ ), also referred to as *Freon*, is the principal compound of the RPC  
594 gas mixtures, with a typical fraction above 90%. It is used for it's high effective Townsend  
595 coefficient and the great average fast charge that allows to operate the detector with a high  
596 threshold with respect to argon, for example, that has similar effective Townsend coefficient  
597 but suffers from a lower fast charge. To operate with similar conditions, argon would require a  
598 higher electric field leading to a higher fraction of streamers, thus limiting the rate capability  
599 of the detector [40].
- 600 • Isobutane (i- $C_4H_{10}$ ), only present in a few percent in the gas mixtures, is used for its UV  
601 quenching properties [41] helping to prevent streamers due to UV photon emission during the  
602 avalanche growth.
- 603 • Sulfur hexafluoride, ( $SF_6$ ), referred to simply as *SF6*, is used in very little quantities for its  
604 high electronegativity. Excess of electrons are being absorbed by the compound and streamers

are suppressed [34]. Nevertheless, a fraction of  $SF_6$  higher than 1% will not bring any extra benefit in terms of streamer cancelation power but will lead to higher operating voltage [33], as can be understood through Figure 3.2.



*Figure 3.2: Effeciency (circles and stars with 30 mV and 100 mV thresholds respectively) and streamer probability (opened circles) as function of the operating voltatge of a 2 mm single gap HPL RPC flushed with a gas mixture containing (a) 5%, (b) 2%, (c) 1% and (d) no  $SF_6$  [33].*

After an avalanche developed in the gas, a time long compared to the development of a discharge is needed to recombine the charge carriers in the electrode material due to their resistivity. This property has the advantage of affecting the local electric field and avoiding sparks in the detector but, on the other hand, the rate capability is intrinsically limited by the time constant  $\tau_{RPC}$  of the detector. Using a quasi-static approximation of Maxwell's equations for weakly conducting media, it can be shown that the time constant  $\tau_{RPC}$  necessary to the charge recombination at the interface in between the electrode and the gas volume is given by the Formula 3.1 [42].

$$\tau_{RPC} = \frac{\epsilon_{electrode} + \epsilon_{gas}}{\sigma_{electrode} + \sigma_{gas}} \quad (3.1)$$

A gas can be assimilated to vacuum, leading to  $\epsilon_{gas} = \epsilon_0$  and  $\sigma_{gas} = 0$ , and the electrodes permittivity and conductivity can be written as  $\epsilon_{electrode} = \epsilon_r \epsilon_0$  and  $\sigma_{electrode} = 1/\rho_{electrode}$ , showing the strong dependence of the time constant to the electrodes resistivity in Formula 3.2.

$$\tau_{RPC} = (\epsilon_r + 1)\epsilon_0 \times \rho_{electrode} \quad (3.2)$$

Very few materials with a low enough resistivity exist in nature. The resistivity targeted to build RPCs ranges from  $10^9$  to  $10^{12} \Omega \cdot \text{cm}$ . The most common RPC electrode materials are displayed in Table 3.1. When the doped glass and ceramics can offer short time constants of the order of 1 ms,

the developing cost of such materials is quite high due to the very low demand. Thus, High-pressure laminate (HPL) is often the choice for high rate experiments using very large RPC detection areas. Other experiments working at cosmic muon fluxes can safely operate with ordinary float glass.

| Material                 | $\rho_{electrode}$ ( $\Omega \cdot \text{cm}$ ) | $\epsilon_r$ | $\tau_{RPC}$ (ms)  |
|--------------------------|---|--------------|--------------------|
| Float glass              | $10^{12}$                                       | $\sim 7$     | $\sim 700$         |
| High-pressure laminate   | $10^{10}$ to $10^{12}$                          | $\sim 6$     | $\sim 6$ to 600    |
| Doped glass (LR S)       | $10^9$ to $10^{11}$                             | $\sim 10$    | $\sim 1$ to 100    |
| Doped ceramics (SiN/SiC) | $10^9$  | $\sim 8.5$   | $\sim 1$           |
| Doped ceramics (Ferrite) | $10^8$ to $10^{12}$                             | $\sim 20$    | $\sim 0.2$ to 2000 |

Table 3.1: Properties of the most used electrode materials for RPCs.

### 3.1.1 Electron drift velocity

Talk about the electron drift velocity and mention the time resolution of RPCs.

## 3.2 Rate capability and time resolution of Resistive Plate Chambers

As already previously discussed, the electrode material plays a key role in the max intrinsic rate capability of RPCs. R&D is being done to develop at always cheaper costs material with lower resistivity. Nevertheless, the amount of charge released, i.e. the size of the discharge, if reduced leads to a smaller blind area in the detector, increasing the rate capability of the detector.

### 3.2.1 Operation modes

RPCs where developed early 1980s. At that time it was using an operating mode now referred to as *streamer mode*. Streamers are large discharges that develop in between the 2 electrodes enough to locally discharge the electrodes. If the electric field inside of the gas volume is strong enough, with electrons being fast compared to ions, a large and dense cloud of positive ions will develop nearby the anode and extend toward the cathode while the electrons are being collected, eventually leading to a streamer discharge due to the increase of field seen at the cathode. the field is then strong enough so that electrons are pulled out of the cathode. Electrodes, though they are a unique volume of resistive material, can be assimilated to capacitors. At the moment an electric field is applied in between their outer surfaces, the charge carriers inside of the volume will start moving leading to a situation where there is no voltage across the electrodes and a higher density of negative charges, i.e. electrons, on the inner surface of the cathode. Finally, when a streamer discharge occurs, these electrons are partially released in the gas volume contributing to increase the discharge strength until the formation of a conductive plasma, the streamer. This can be understood through Figure 3.3 [27]. Streamer signals are very convenient in terms of read-out as no amplification is required with output pulses amplitudes of the order of a few tens to few hundreds of mV as can be seen on Figure 3.4.

When the electric field is reduced though, the electronic gain is small until the electrons get close enough to the anode and the positive ion cloud is much smaller. The electric field cannot rise to the point a field emission of electrons on the cathode is possible. The resulting signal is weak, of the

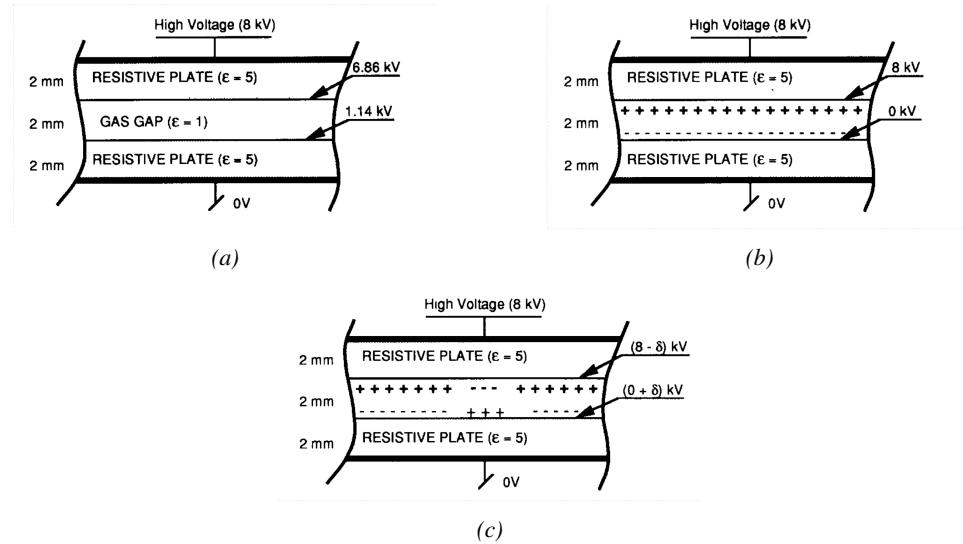


Figure 3.3: Movement of the charge carriers in an RPC. Figure 3.3a: Voltage across an RPC whose electrode have a relative permittivity of 5 at the moment the tension is applied. Figure 3.3b: After the charge carriers moved, the electrodes are charged and there is no voltage drop over the electrodes anymore. The full potential is applied on the gas gap only. Figure 3.3c: The streamer discharge initiated by a charged particle transports electrons and cations towards the anode and cathode respectively.

order of a few mv as shown on Figure 3.4, and requires amplification. This is the *avalanche mode* of RPC operation.

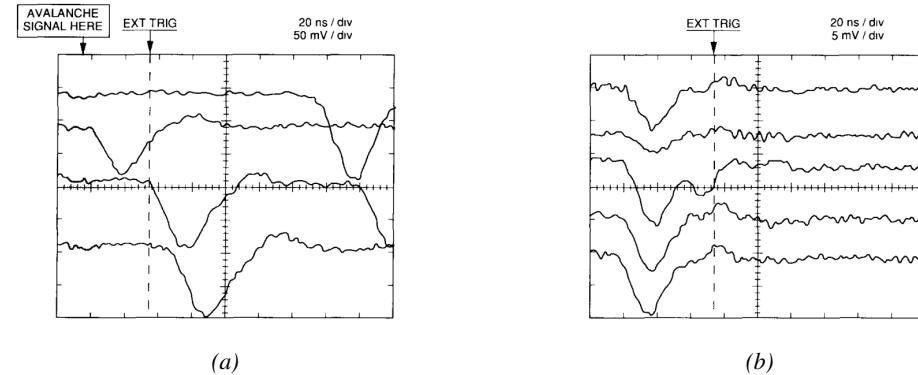


Figure 3.4: Typical oscilloscope pulses in streamer mode (Figure 3.4a) and avalanche mode (Figure 3.4b). In the case of streamer mode, the very small avalanche signal is visible.

This mode offers a higher rate capability by providing smaller discharges that don't affect the electrodes charge and are more locally contained in the gas volume as was demonstrated by Crotty with Figure 3.5 [27]. The detector only stays locally blind the time the charge carriers are recombined and there is no need for electrode recharge which is a long process affecting a large portion of the detector. Another advantage of avalanche signals over streamer is the great time consistency. Figure 3.4 shows very clearly that avalanche signals have a very small time jitter. This is a natural choice for high rate experiments.

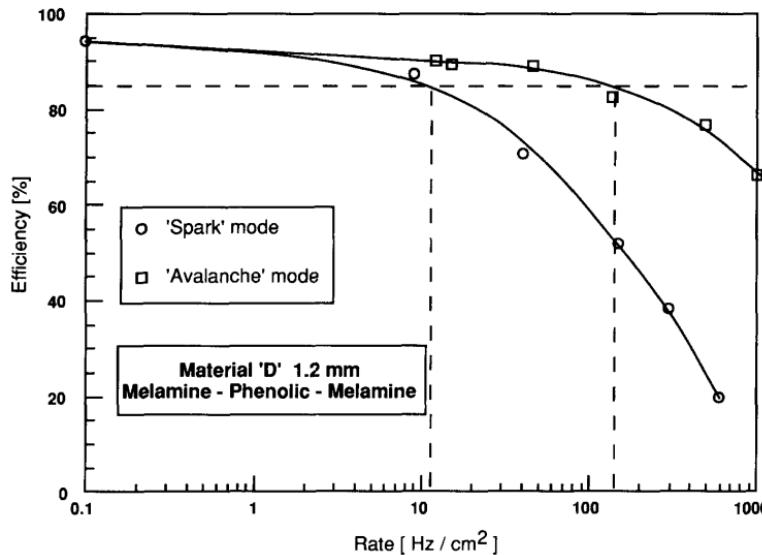


Figure 3.5: Rate capability comparison for the streamer and avalanche mode of operation. An order of magnitude in rate capability for a maximal efficiency drop of 10% is gained by using the avalanche mode over the streamer mode.

### 660 3.2.2 Detector designs and performance

661 Different RPC design have been used and each of them present its own advantages. Historically,  
 662 the first type of RPC to have been developed is what is referred now to *no narrow gap* RPC [24, 43].  
 663 After the avalanche mode has been discovered [27], it has been proven that increasing the width  
 664 of the gas gap lead to higher rate capability, due to lower charge deposition per avalanche, and  
 665 lower power dissipation [43]. Nevertheless, by increasing the gas gap width, the time resolution of  
 666 the detector decreases. This is a natural result if the increase of active gas volume in the detector is  
 667 taken into account. Indeed, for a given threshold, only the small fraction of gas closest to the cathode  
 668 will provide enough gain to have a detectable signal. In the case of a wider gas volume, the active  
 669 region is then larger and a larger time jitter is introduced with the variation of starting position of the  
 670 avalanche, as discussed in [30]. To solve improve both the time resolution and the rate capability,  
 671 different methods were used trying to get advantages of both narrow and wide gap RPCs.

#### 672 3.2.2.1 Double-gap RPC

673 Double-gap RPCs are made out of 2 narrow RPC detectors. The 2 RPC gaps are stacked on top of  
 674 each other as shown in Figure 3.6. This detector layout, popularized by the two multipurpose experiments  
 675 CMS [21] and ATLAS [35] at LHC, can be used as an OR system in which each individual  
 676 chamber participates in the output signal. The gain of such a detector is greatly reduced with respect  
 677 to single-gap RPCs with an efficiency plateau reached at lower voltage, as visible on Figure 3.7.

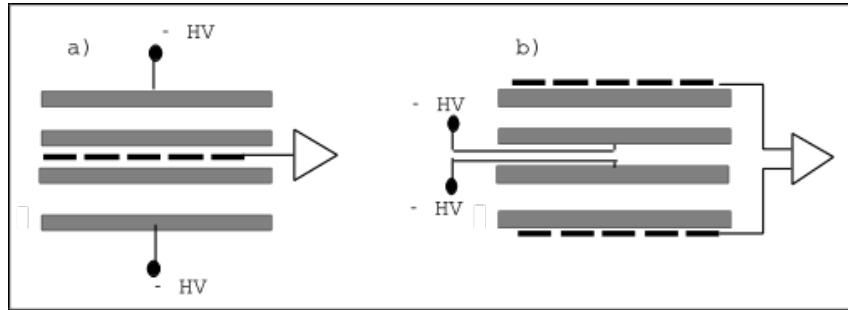


Figure 3.6: Possible double-gap RPC layouts: a) "standard" 1D double-gap RPC, as used in CMS experiment, where the anodes are facing each other and a 1D read-out plane is sandwiched in between them, b) double read-out double-gap RPC as used in ATLAS experiment, where the cathodes are facing each other and 2 read-out planes are used on the outer surfaces. This last layout can offer the possibility to use a 2D reconstruction by using orthogonal read-out planes.

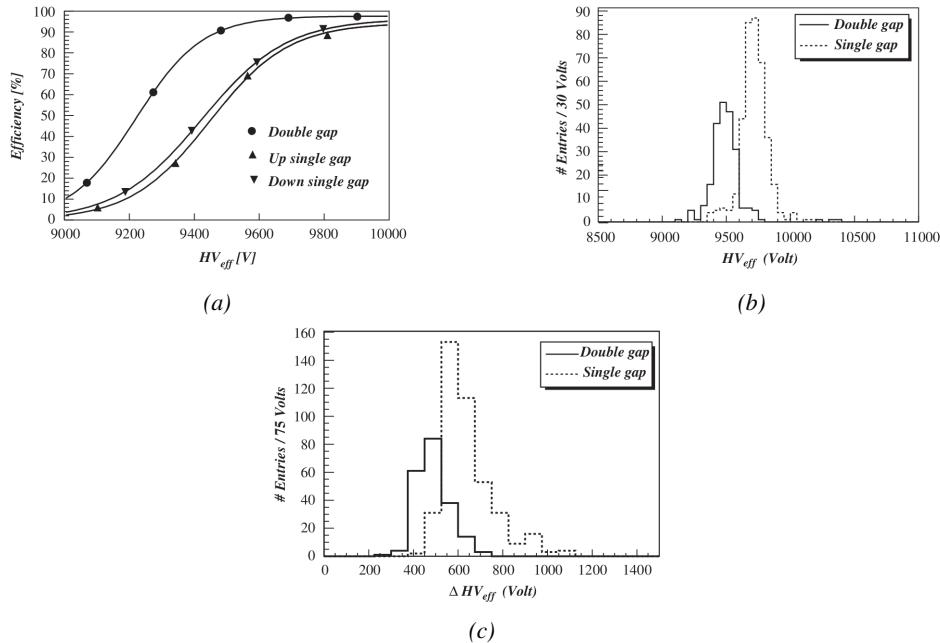
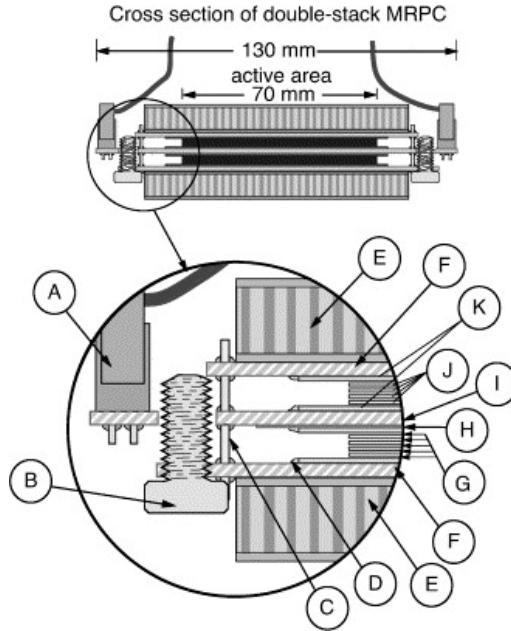


Figure 3.7: Comparison of performance of CMS double and single gap RPCs using cosmic muons [44]. Figure 3.7a: Comparison of efficiency sigmoids. Figure 3.7b: Voltage distribution at 95% of maximum efficiency. Figure 3.7c:  $\Delta_{10\%}^{90\%}$  distribution.

### 678 3.2.2.2 Multigap RPC (MRPC)

679 MRPCs are layouts in which floating sub electrode plates are placed into a wide gap RPC to divide  
 680 the gas volume and create a sum of narrow gaps [30, 31]. The time resolution of such a detector can  
 681 reach of few tens of ps, with gas gaps of the order of a few hundred  $\mu\text{m}$  as shown in Figure 3.8  
 682 representing ALICE Time-of-flight (ToF) MRPCs.

683 Sometimes used as a double multigap RPC, taking advantage of the OR of double gap RPCs,  
 684 the MRPC is mainly used as ToF detector [45–49] due to its excellent timing properties that allow



*Figure 3.8: Presentation of ALICE MRPC using 250  $\mu\text{m}$  gas gaps, 620  $\mu\text{m}$  outer glass electrodes and 550  $\mu\text{m}$  inner floating electrodes. More details on the labels are given in [45].*

685 to perform particle identification as explained by Williams in [50]. The principle of particle iden-  
 686 tification using ToF is simply the measurement of the velocity of a particle. Indeed, particles are  
 687 defined by their mass (for the parameter of interest here, their electric charge being measured using  
 688 the bending angle of the particles traveling through a magnetic field) and this mass can be calculated  
 689 by measuring the velocity  $\beta$  and momentum of the particle:

$$\beta = \frac{p}{\sqrt{p^2 + m^2}} \quad (3.3)$$

690 Intuitively, it is trivial to understand that 2 different particles having the same momentum will  
 691 have a different velocity due to the mass difference and thus a different flight time  $T_1$  and  $T_2$  through  
 692 the detector and this is used to separate and identify particles. The better the time resolution of the  
 693 ToF system used, the stronger will the separation be:

$$T = \frac{L}{v} = \frac{L}{c \cdot \beta}, \quad \Delta T = T_1 - T_2 = \frac{L}{c} \left( \sqrt{1 + m_1^2/p^2} - \sqrt{1 + m_2^2/p^2} \right) \cong (m_1^2 - m_2^2) \frac{L}{2cp^2} \quad (3.4)$$

694 An example of particle identification is given for the case of STAR experiment in Figure 3.9.

695 Another benefit of using such small gas gaps is the strong reduction of the average avalanche  
 696 volume and thus of the blind spot on MRPCs leading to an improved rate capability. Multigaps can  
 697 sustain backgrounds of several kHz/cm<sup>2</sup> as demonstrated in Figure 3.10.

### 698 3.2.2.3 Charge distribution and performance limitations

699 The direct consequence of the different RPC layouts is a variation of intrinsic time resolution of the  
 700 RPC as the gap size decreases. An advantage is given to multigaps whose design use sub-millimeter

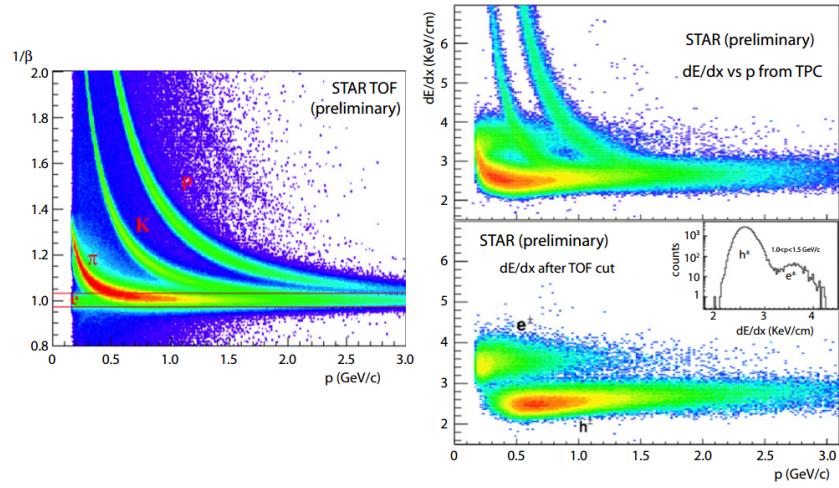


Figure 3.9: Particle identification applied to electrons in the STAR experiment. The identification is performed combining ToF and  $dE/dx$  measurements [50].

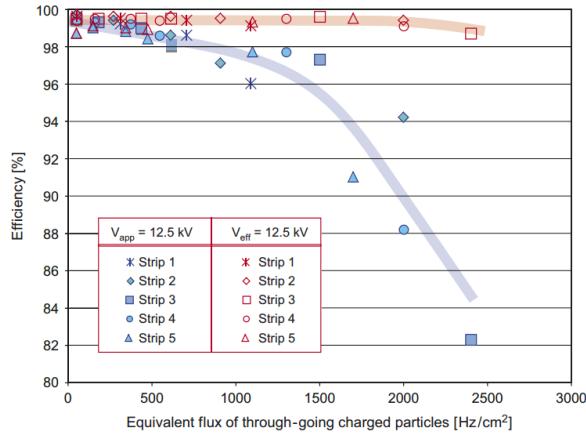


Figure 3.10: Comparison of the detector performance of ALICE ToF MRPC [51] at fixed applied voltage (in blue) and at fixed effective voltage (in red). The effective voltage is kept fixed by increasing the applied voltage accordingly to the current drawn by the detector.

701 gas volumes providing very consistent signals.

702 On the charge spectrum point of view, each layout has its own advantages. When the double-gap  
 703 has the highest induced over drifting charge ratio, as seen in Figure 3.11, the multigap has a charge  
 704 spectrum strongly detached from the origin, as visible in Figure 3.12. A high induced over drifting  
 705 charge ratio means that the double gap can be safely operated at high threshold or that at similar  
 706 threshold it can be operated with a twice smaller drifting charge, meaning a higher rate capability.  
 707 On the other hand, the strong detachment of the charge spectrum from the origin in the MRPC case  
 708 allows to reach a higher efficiency with increasing threshold as most of the induced charge is not low  
 709 due to the convolution of several single gap spectra. The range of stable efficiency increases with  
 710 the number of gap, as presented in Figure 3.13.

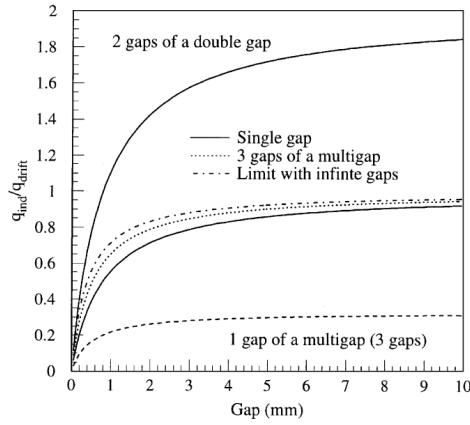


Figure 3.11: Ratio between total induced and drifting charge have been simulated for single gap, double-gap and multigap layouts [52]. The total induced charge for a double-gap RPC is a factor 2 higher than for a multigap.

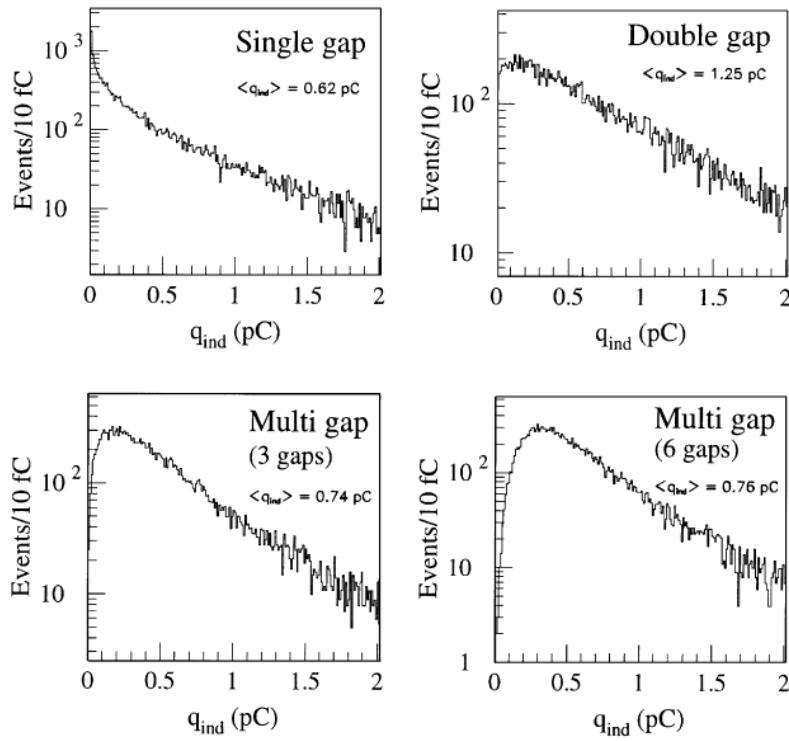


Figure 3.12: Charge spectra have been simulated for single gap, double-gap and multigap layouts [52]. It appears that when single gap shows a decreasing spectrum, double and multigap layouts exhibit a spectrum whose peak is detached from the origin. The detachment gets stronger as the number of gaps increases.

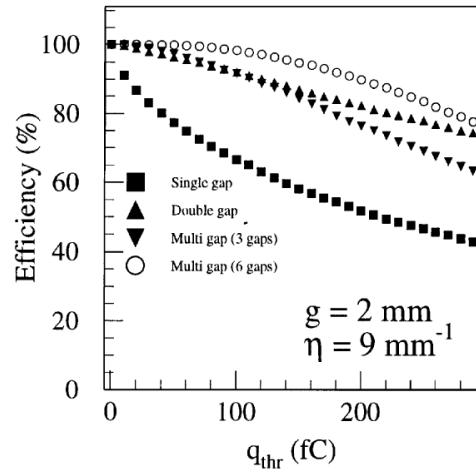


Figure 3.13: The maximal theoretical efficiency is simulated for single gap, double-gap and multigap layouts [52] at a constant gap thickness of 2 mm and using an effective Townsend coefficient of  $9 \text{ mm}^{-1}$ .

### 711 3.3 Signal formation

### 712 3.4 Gas transport parameters



# 4

713

714

715

## Longevity studies and Consolidation of the present CMS RPC subsystem

### 716 4.1 Resistive Plate Chambers at CMS

#### 717 4.1.1 Overview

718 The Resistive Plate Chambers (RPC) system, located in both barrel and endcap regions, provides a  
719 fast, independent muon trigger with a looser  $p_T$  threshold over a large portion of the pseudorapidity  
720 range ( $|\eta| < 1.6$ ) [add reconstruction].

721

722 During High-Luminosity LHC (HL-LHC) operations the expected conditions in terms of back-  
723 ground and pile-up will make the identification and correct  $P_T$  assignment a challenge for the Muon  
724 system. The goal of RPC upgrade is to provide additional hits to the Muon system with precise tim-  
725 ing. All these informations will be elaborated by the trigger system in a global way enhancing the  
726 performance of the trigger in terms of efficiency and rate control. The RPC Upgrade is based on two  
727 projects: an improved Link Board System and the extension of the RPC coverage up to  $|\eta| = 2.4$ .  
728 [FIXME 2.4 or 2.5?]

729 The Link Board system, that will be described in section xxx, is responsible to process, syn-  
730 chronize and zero-suppress the signals coming from the RPC front end boards. The Link Board  
731 components have been produced between 2006 and 2007 and will be subjected to aging and failure  
732 in the long term. The upgraded Link Board system will overcome the aging problems described in  
733 section xxx and will allow for a more precise timing information to the RPC hits from 25 to 1 ns [ref  
734 section xxx].

735 The extension of the RPC system up to  $|\eta| = 2.1$  was already planned in the CMS TDR [ref  
736 cmstdr] and staged because of budget limitations and expected background rates higher than the rate  
737 capability of the present CMS RPCs in that region. An extensive R&D program has been done in  
738 order to develop an improved RPC that fulfills the CMS requirements. Two new RPC layers in the  
739 innermost ring of stations 3 and 4 will be added with benefits to the neutron-induced background

740 reduction and efficiency improvement for both trigger and offline reconstruction.

### 741 4.1.2 The present RPC system

742 The RPC system is organized in 4 stations called RB1 to RB4 in the barrel region, and RE1 to RE4  
 743 in the endcap region. The innermost barrel stations, RB1 and RB2, are instrumented with 2 layers  
 744 of RPCs facing the innermost (RB1in and RB2in) and outermost (RB1out and RB2out) sides of the  
 745 DT chambers. Every chamber is then divided from the read-out point of view into 2 or 3  $\eta$  partitions  
 746 called “rolls”. The RPC system consist of 480 barrel chambers and 576 endcap chambers. Details  
 747 on the geometry are discussed in the paper [ref to geo paper].

748 The CMS RPC chamber is a double-gap, operated in avalanche mode to ensure reliable operation  
 749 at high rates. Each RPC gap consists of two 2-mm-thick resistive High-Pressure Laminate (HPL)  
 750 plates separated by a 2-mm-thick gas gap. The outer surface of the HPL plates is coated with a thin  
 751 conductive graphite layer, and a voltage is applied. The RPCs are operated with a 3-component,  
 752 non-flammable gas mixture consisting of 95.2% freon ( $C_2H_2F_4$ , known as R134a), 4.5% isobutane  
 753 ( $i-C_4H_{10}$ ), and 0.3% sulphur hexafluoride ( $SF_6$ ) with a relative humidity of 40% - 50%. Readout  
 754 strips are aligned in  $\eta$  between the 2 gas gaps. [\[Add a sentence on FEBs.\]](#)

755 The discriminated signals coming from the Front End boards feed via twisted cables (10 to 20 m  
 756 long) the Link Board System located in UXC on the balconies around the detector. The Link System  
 757 consist of the 1376 Link Boards (LBs) and the 216 Control Boards (CBs), placed in 108 Link Boxes.  
 758 The Link Box is a custom crate (6U high) with 20 slots (for two CBs and eighteen LBs). The Link  
 759 Box contains custom backplane to which the cables from the chambers are connected, as well as the  
 760 cables providing the LBs and CBs power supply and the cables for the RPC FEBs control with use  
 761 of the I2C protocol (trough the CB). The backplane itself contains only connectors (and no any other  
 762 electronic devices).

763 The Link Board has 96 input channels (one channel corresponds to one RPC strip). The input  
 764 signals are the  $\sim 100$  ns binary pulses which are synchronous to the RPC hits, but not to the LHC  
 765 clock (which drives the entire CMS electronics). Thus the first step of the FEB signals processing  
 766 is synchronization, i.e. assignment of the signals to the BXes (25 ns periods). Then the data are  
 767 compressed with a simple zero-suppressing algorithm (the input channels are grouped into 8 bit  
 768 partitions, only the partitions with at least one nonzero bit are selected for each BX). Next, the non-  
 769 empty partitions are time-multiplexed i.e. if there are more than one such partition in a given BX,  
 770 they are sent one-by-one in consecutive BXes. The data from 3 neighbouring LBs are concentrated  
 771 by the middle LB which contains the optical transmitter for sending them to the USC over a fiber at  
 772 1.6 Gbps.

773 The Control Boards provide the communication of the control software with the LBs via the  
 774 FEC/CCU system. The CBs are connected into token rings, each ring consists of 12 CBs of one  
 775 detector tower and a FEC mezzanine board placed on the CCS board located in the VME crate in  
 776 the USC. In total, there are 18 rings in the entire Link System. The CBs also perform automatic  
 777 reloading of the LB's firmware which is needed in order to avoid accumulation of the radiation  
 778 induced SEUs in the LBs firmware.

779 Both LBs and CB are based on the Xilinx Spartan III FPGAs, the CB additionally contains  
 780 radiation-tolerant (FLASH based) FPGA Actel ProAsicPlus.

781 The High Voltage power system is located in USC, not exposed to radiation and easily accessible  
 782 for any reparation. A single HV channel powers 2 RPC chambers both in the barrel and endcap  
 783 regions. The Low Voltage boards are located in UXC on the balconies and provide the voltage to the

784 front end electronics.

### 785 4.1.3 Pulse processing of CMS RPCs

786 Signals induced by cosmic particle in the RPC strips are shaped by standard CMS RPC Front-End  
 787 Electronics (FEE) following the scheme of Figure 4.1. On a first stage, analogic signals are amplified  
 788 and then sent to the Constant Fraction Discriminator (CFD) described in Figure 4.2. At the end of  
 789 the chain, 100 ns long pulses are sent in the LVDS output. These output signal are sent on one side to  
 790 a V1190A Time-to-Digital Converter (TDC) module from CAEN and on the other to an OR module  
 791 to count the number of detected signals. Trigger and hit coïncidences are monitored using scalers.  
 792 The TDC is used to store the data into ROOT files. These files are thus analysed to understand the  
 793 detectors performance.

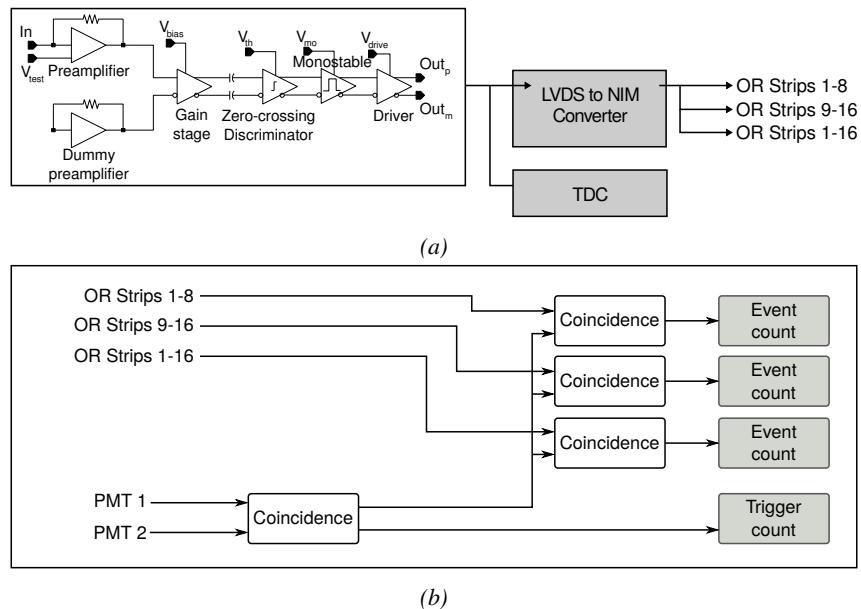
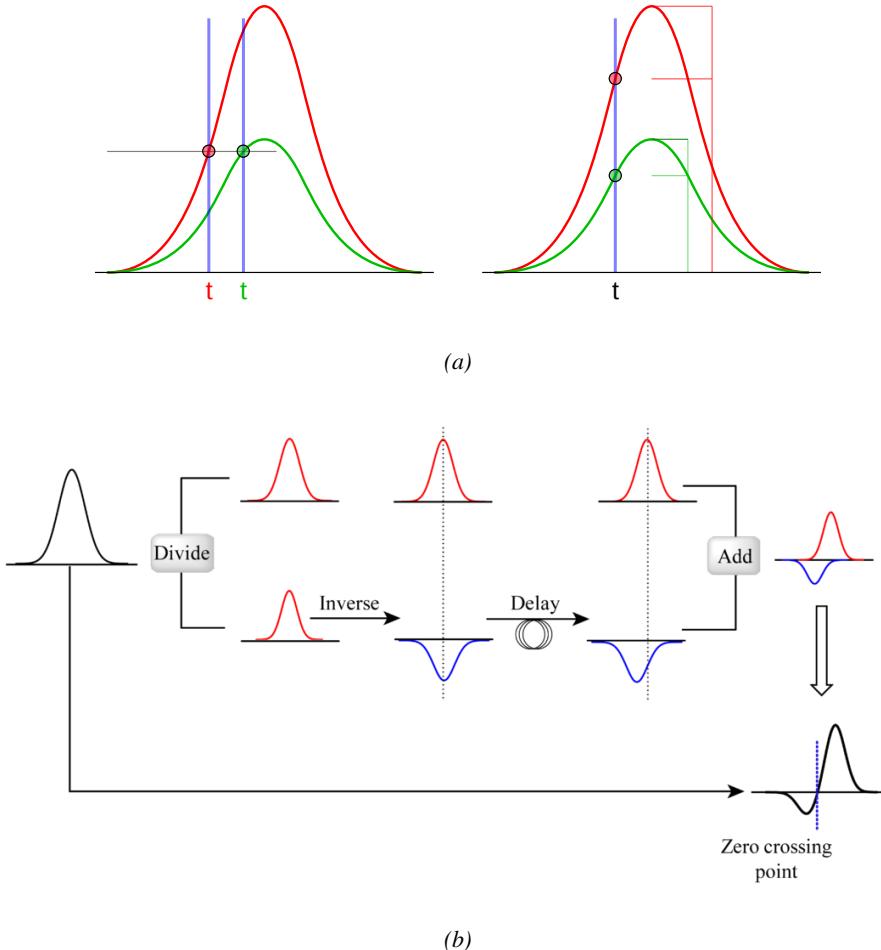


Figure 4.1: Signals from the RPC strips are shaped by the FEE described on Figure 4.1a. Output LVDS signals are then read-out by a TDC module connected to a computer or converted into NIM and sent to scalers. Figure 4.1b describes how these converted signals are put in coincidence with the trigger.

## 794 4.2 Testing detectors under extreme conditions

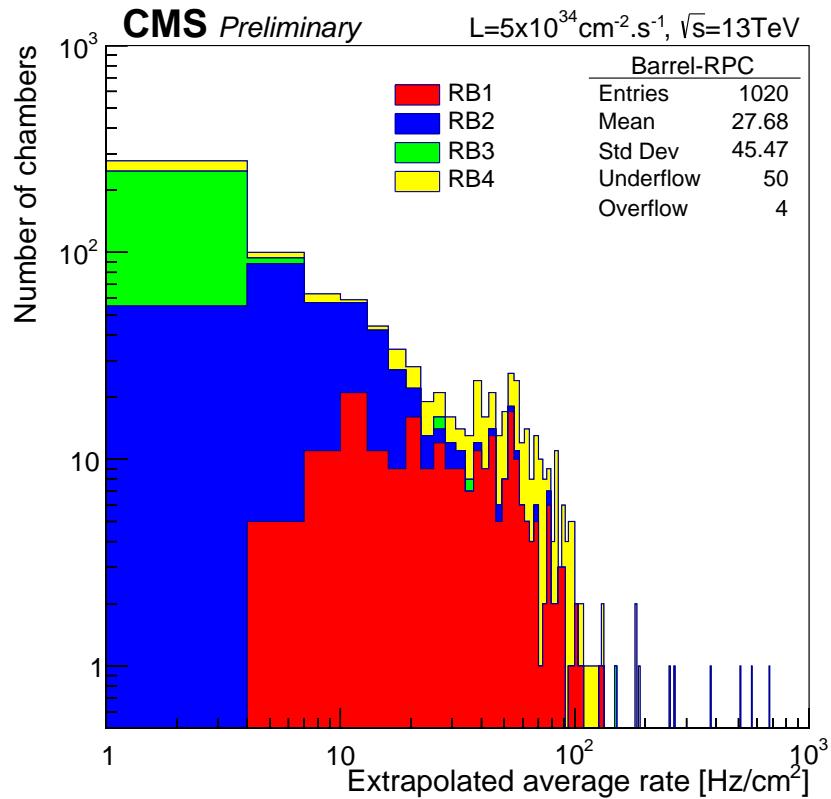
795 The upgrade from LHC to HL-LHC will increase the peak luminosity from  $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$  to reach  
 796  $5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ , increasing in the same way the total expected background to which the RPC  
 797 system will be subjected to. Composed of low energy gammas and neutrons from  $p$ - $p$  collisions, low  
 798 momentum primary and secondary muons, puch-through hadrons from calorimeters, and particles  
 799 produced in the interaction of the beams with collimators, the background will mostly affect the  
 800 regions of CMS that are the closest to the beam line, i.e. the RPC detectors located in the endcaps.  
 801 [To update.]

802

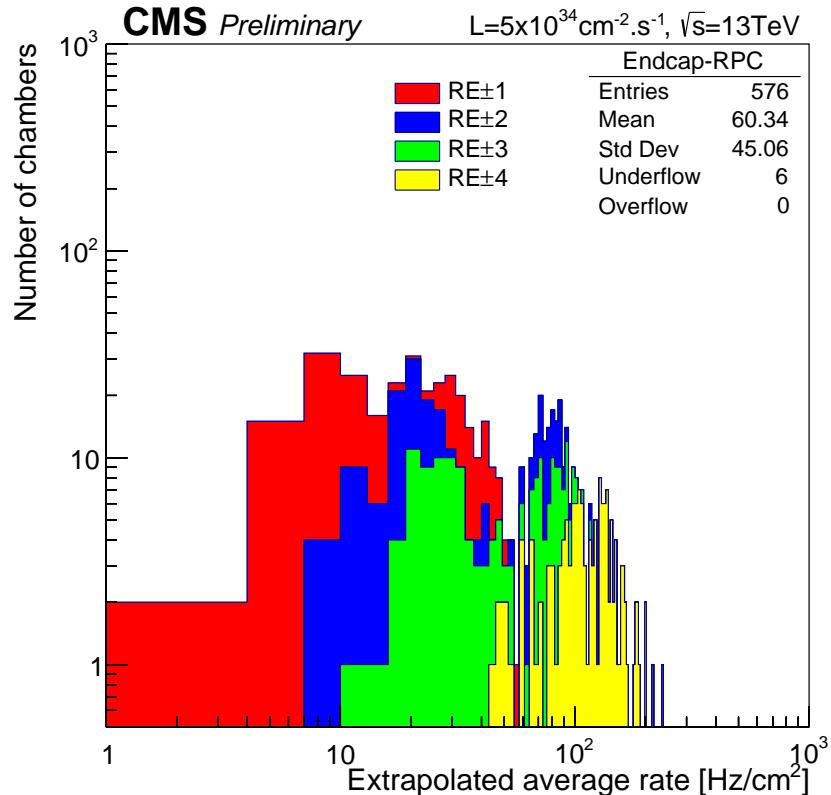


*Figure 4.2: Description of the principle of a CFD. A comparison of threshold triggering (left) and constant fraction triggering (right) is shown in Figure 4.2a. Constant fraction triggering is obtained thanks to zero-crossing technique as explained in Figure 4.2b. The signal arriving at the input of the CFD is split into three components. A first one is delayed and connected to the inverting input of a first comparator. A second component is connected to the noninverting input of this first comparator. A third component is connected to the noninverting input of another comparator along with a threshold value connected to the inverting input. Finally, the output of both comparators is fed through an AND gate.*

803     The 2016 data allowed to study the values of the background rate in all RPC system. In Figure  
 804     4.3, the distribution of the chamber background hit rate per unit area is shown at a luminosity  
 805     of  $5 \times 10^{34} \text{ cm}^{-2} \cdot \text{s}^{-1}$  linearly extrapolating from data collected in 2016 [ref mentioning the linear  
 806     dependency of rate vs lumi]. The maximum rate per unit area at HL-LHC conditions is expected to  
 807     be of the order of  $600 \text{ Hz/cm}^2$  (including a safety factor 3). Nevertheless, Fluka simulations have  
 808     conducted in order to understand the background at HL-LHC conditions. The comparison to the  
 809     data has shown, in Figure 4.4, a discrepancy of a factor 2 even though the order of magnitude is  
 810     consistent. [Understand mismatch.]



(a)



(b)

Figure 4.3: (4.3a) Extrapolation from 2016 data of single hit rate per unit area in the barrel region. (4.3b) Extrapolation from 2016 data of single hit rate per unit area in the endcap region.

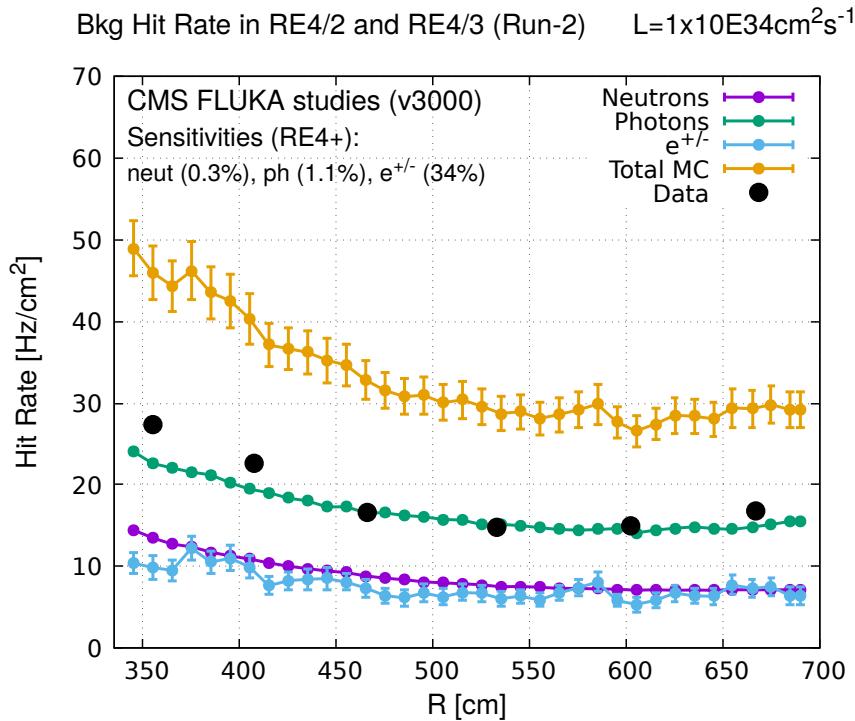


Figure 4.4: Background Fluka simulation compared to 2016 Data at  $L = 10^{34}\text{cm}^{-2}\cdot\text{s}^{-1}$  in the fourth endcap disk region. A mismatch in between simulation and data can be observed. [\[To be understood.\]](#)

In the past, extensive long-term tests were carried out at several gamma and neutron facilities certifying the detector performance. Both full size and small prototype RPCs have been irradiated with photons up to an integrated charge of  $\sim 0.05C/\text{cm}^2$  and  $\sim 0.4C/\text{cm}^2$ , respectively [53, 54]. During Run-I, the RPC system provided stable operation and excellent performance and did not show any aging effects for integrated charge of the order of  $0.01C/\text{cm}^2$ . Projections on currents from 2016 Data, has allowed to determine that the total integrated charge, by the end of HL-LHC, would be of the order of  $1C/\text{cm}^2$  (including a safety factor 3). [\[Corresponding figure needed.\]](#)

819

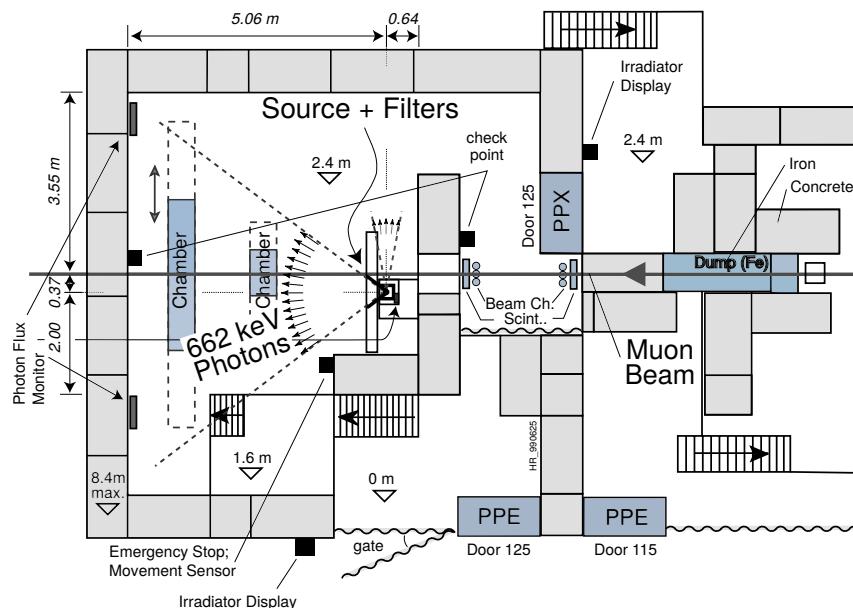
## 820 4.2.1 The Gamma Irradiation Facilities

### 821 4.2.1.1 GIF

822 Located in the SPS West Area at the downstream end of the X5 test beam, the Gamma Irradiation  
823 Facility (GIF) was a test area in which particle detectors were exposed to a particle beam in presence  
824 of an adjustable gamma background [55]. Its goal was to reproduce background conditions these  
825 detectors would suffer in their operating environment at LHC. GIF layout is shown in Figure 4.5.  
826 Gamma photons are produced by a strong  $^{137}\text{Cs}$  source installed in the upstream part of the zone  
827 inside a lead container. The source container includes a collimator, designed to irradiate a  $6 \times 6 \text{m}^2$   
828 area at 5 m maximum to the source. A thin lens-shaped lead filter helps providing with a uniform  
829 outgoing flux in a vertical plane, orthogonal to the beam direction. The principal collimator hole  
830 provides a pyramidal aperture of  $74^\circ \times 74^\circ$  solid angle and provides a photon flux in a pyramidal vol-

ume along the beam axis. The photon rate is controled by further lead filters allowing the maximum rate to be limited and to vary within a range of four orders of magnitude. Particle detectors under test are then placed within the pyramidal volume in front of the source, perpendicularly to the beam line in order to profit from the homogeneous photon flux. Adjusting the background flux of photons can then be done by using the filters and choosing the position of the detectors with respect to the source.

836



*Figure 4.5: Layout of the test beam zone called X5c GIF at CERN. Photons from the radioactive source produce a sustained high rate of random hits over the whole area. The zone is surrounded by 8 m high and 80 cm thick concrete walls. Access is possible through three entry points. Two access doors for personnel and one large gate for material. A crane allows installation of heavy equipment in the area.*

As described on Figure 4.6, the  $^{137}\text{Cs}$  source emits a 662 keV photon in 85% of the decays. An activity of 740 GBq was measured on the 5<sup>th</sup> March 1997. To estimate the strength of the flux in 2014, it is necessary to consider the nuclear decay through time assiciated to the Cesium source whose half-life is well known ( $t_{1/2} = (30.05 \pm 0.08)$  y). The GIF tests where done in between the 20<sup>th</sup> and the 31<sup>st</sup> of August 2014, i.e. at a time  $t = (17.47 \pm 0.02)$  y resulting in an attenuation of the activity from 740 GBq in 1997 to 494 GBq in 2014.

843

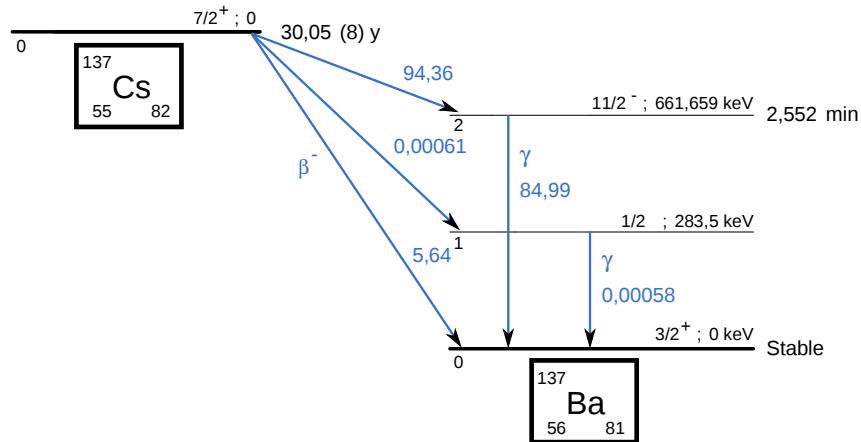


Figure 4.6:  $^{137}\text{Cs}$  decays by  $\beta^-$  emission to the ground state of  $^{137}\text{Ba}$  ( $\text{BR} = 5.64\%$ ) and via the 662 keV isomeric level of  $^{137}\text{Ba}$  ( $\text{BR} = 94.36\%$ ) whose half-life is 2.55 min.

#### 844 4.2.1.2 GIF++

845 The new Gamma Irradiation Facility (GIF++), located in the SPS North Area at the downstream end  
 846 of the H4 test beam, has replaced its predecessor during LS1 and has been operational since spring  
 847 2015 [56]. Like GIF, GIF++ features a  $^{137}\text{Cs}$  source of 662 keV gamma photons, their fluence being  
 848 controlled with a set of filters of various attenuation factors. The source provides two separated large  
 849 irradiation areas for testing several full-size muon detectors with continuous homogeneous irradiation,  
 850 as presented in Figure 4.7.

851

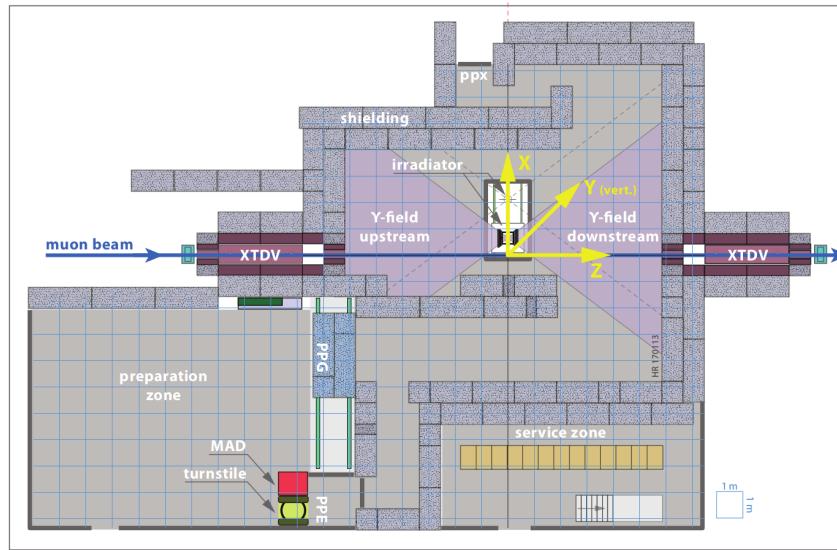


Figure 4.7: Floor plan of the GIF++ facility. When the facility downstream of the GIF++ takes electron beam, a beam pipe is installed along the beam line ( $z$ -axis). The irradiator can be displaced laterally (its center moves from  $x = 0.65 \text{ m}$  to  $2.15 \text{ m}$ ), to increase the distance to the beam pipe.

852 The source activity was measured to be about 13.5 TBq in March 2016. The photon flux being  
 853 far greater than HL-LHC expectations, GIF++ provides an excellent facility for accelerated aging  
 854 tests of muon detectors.

855

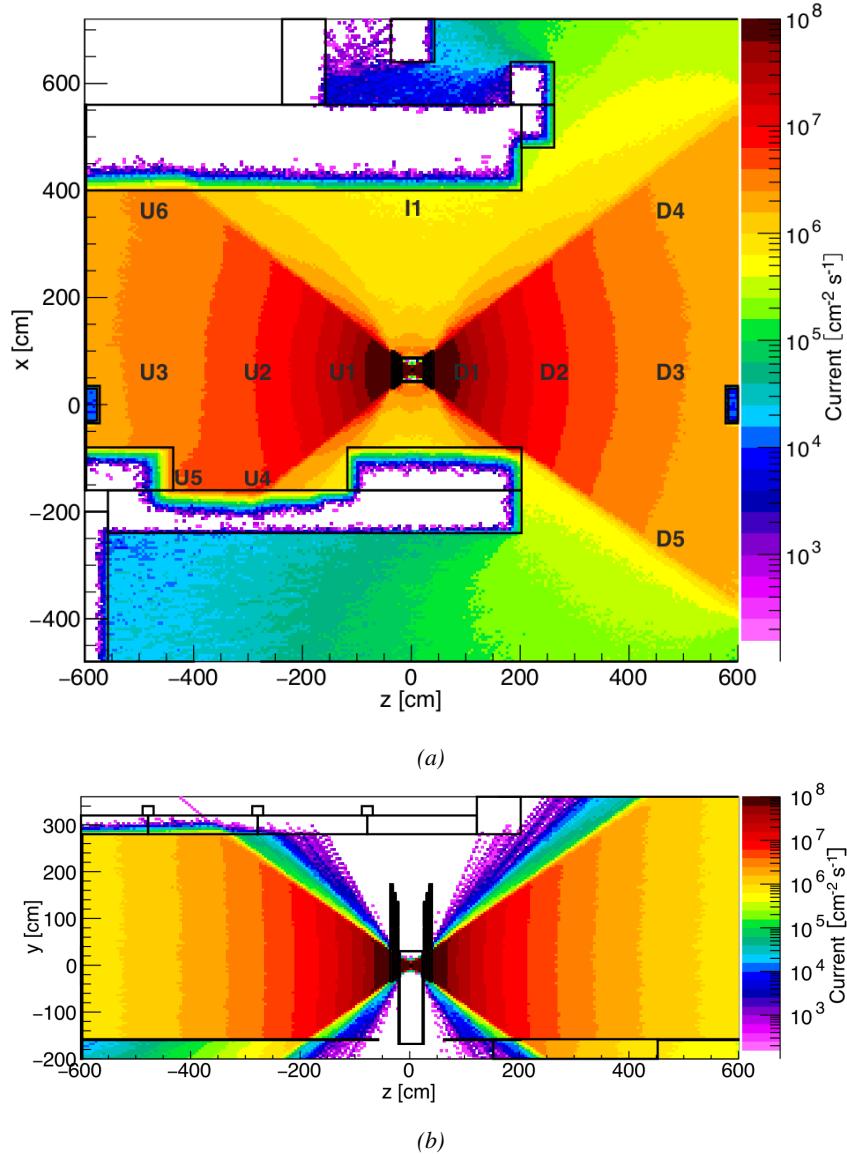


Figure 4.8: Simulated unattenuated current of photons in the xz plane (Figure 4.8a) and yz plane (Figure 4.8b) through the source at  $x = 0.65$  m and  $y = 0$  m. With angular correction filters, the current of 662 keV photons is made uniform in xy planes.

856 The source is situated in the muon beam line with the muon beam being available a few times a  
 857 year. The H4 beam, composed of muons with a momentum of about 150 GeV/c, passes through the  
 858 GIF++ zone and is used to study the performance of the detectors. Its flux is of 104 particles/ $\text{cm}^2$

859 focused in an area similar to  $10 \times 10 \text{ cm}^2$ . Therefore, with properly adjusted filters, one can imitate  
 860 the HL-LHC background and study the performance of muon detectors with their trigger/readout  
 861 electronics in HL-LHC environment.

862

## 863 4.3 Preliminary tests at GIF

### 864 4.3.1 Resistive Plate Chamber test setup

865 During summer 2014, preliminary tests have been conducted in the GIF area on a newly produced  
 866 RE4/2 chamber labelled RE-4-2-BARC-161. This chamber has been placed into a trolley covered  
 867 with a tent. The position of the RPC inside the tent and of the tent related to the source is described  
 868 in Figure 4.9. To test this CMS RPC, three different absorber settings were used. First of all,  
 869 measurements were done with fully opened source. Then, to complete this preliminary study, the  
 870 gamma flux has been attenuated from a factor 2 and a factor 5. The expected gamma flux at the level  
 871 of our detector will be discussed in subsection ??.

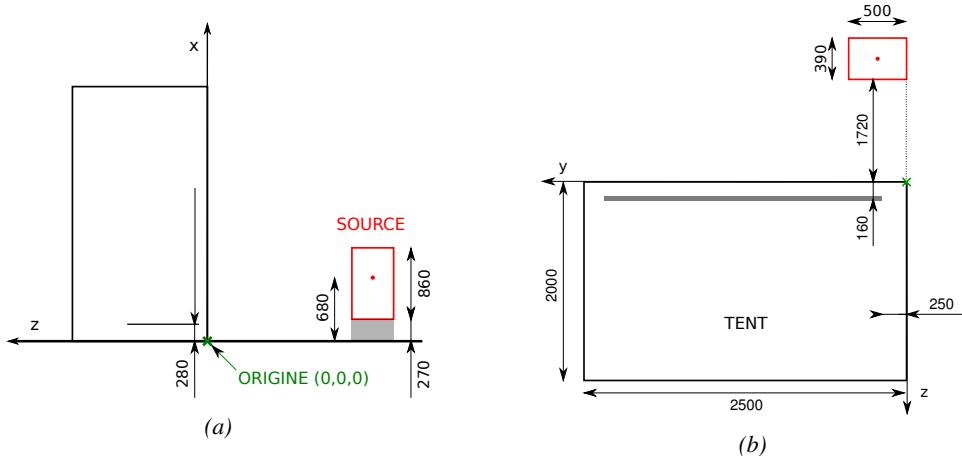
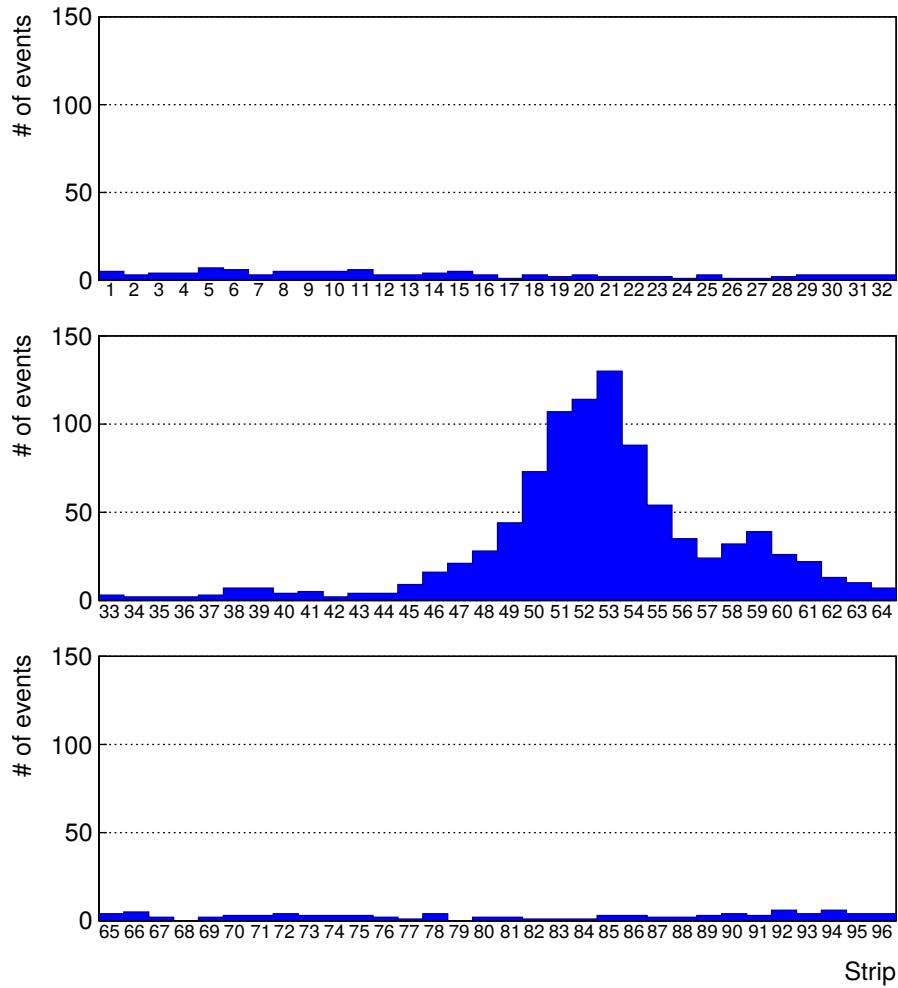


Figure 4.9: Description of the RPC setup. Dimensions are given in mm. A tent containing RPCs is placed at 1720 mm from the source container. The source is situated in the center of the container. RE-4-2-BARC-161 chamber is 160 mm inside the tent. This way, the distance between the source and the chambers plan is 2060 mm. Figure 4.9a provides a side view of the setup in the xz plane while Figure 4.9b shows a top view in the yz plane.



*Figure 4.10: RE-4-2-BARC-161 chamber is inside the tent as described in Figure 4.9. In the top right, the two scintillators used as trigger can be seen. This trigger system has an inclination of 10° relative to horizontal and is placed above half-partition B2 of the RPCs. PMT electronics are shielded thanks to lead blocks placed in order to protect them without stopping photons from going through the scintillators and the chamber.*

872 At the time of the tests, the beam not being operationnal anymore, a trigger composed of 2 plastic  
 873 scintillators has been placed in front of the setup with an inclination of 10 deg with respect to the  
 874 detector plane in order to look at cosmic muons. Using this particular trigger layout, shown on Fig-  
 875 ure 4.10, leads to a cosmic muon hit distribution into the chamber similar to the one in Figure 4.11.  
 876 Measured without gamma irradiation, two peaks can be seen on the profil of partition B, centered  
 877 on strips 52 and 59. Section ?? will help us understand that these two peaks are due respectively to  
 878 forward and backward coming cosmic particles where forward coming particles are first detected by  
 879 the scintillators and then the RPC while the backward coming muons are first detected in the RPC.



*Figure 4.11: Hit distributions over all 3 partitions of RE-4-2-BARC-161 chamber is showed on these plots. Top, middle and bottom figures respectively correspond to partitions A, B, and C. These plots show that some events still occur in other half-partitions than B2, which corresponds to strips 49 to 64, in front of which the trigger is placed, contributing to the inefficiency of detection of cosmic muons. In the case of partitions A and C, the very low amount of data can be interpreted as noise. On the other hand, it is clear that a little portion of muons reach the half-partition B1, corresponding to strips 33 to 48.*

### 4.3.2 Data Acquisition

### 4.3.3 Geometrical acceptance of the setup layout to cosmic muons

In order to profit from a constant gamma irradiation, the detectors inside of the GIF bunker need to be placed in a plane orthogonal to the beam line. The muon beam that used to be available was meant to test the performance of detectors under test. This beam not being active anymore, another solution to test detector performance had to be used. Thus, it has been decided to use cosmic muons detected through a telescope composed of two scintillators. Lead blocks were used as shielding to

887 protect the photomultipliers from gammas as can be seen from Figure 4.10.

888 An inclination has been given to the cosmic telescope to maximize the muon flux. A good com-  
 889 promise had to be found between good enough muon flux and narrow enough hit distribution to  
 890 be sure to contain all the events into only one half partitions as required from the limited available  
 891 readout hardware. Nevertheless, a consequence of the misplaced trigger, that can be seen as a loss  
 892 of events in half-partition B1 in Figure 4.11, is an inefficiency. Nevertheless, the inefficiency of ap-  
 893 proximately 20 % highlighted in Figure 4.12 by comparing the performance of chamber BARC-161  
 894 in 904 and at GIF without irradiation seems too important to be explained only by the geometri-  
 895 cal acceptance of the setup itself. Simulations have been conducted to show how the setup brings  
 896 inefficiency.

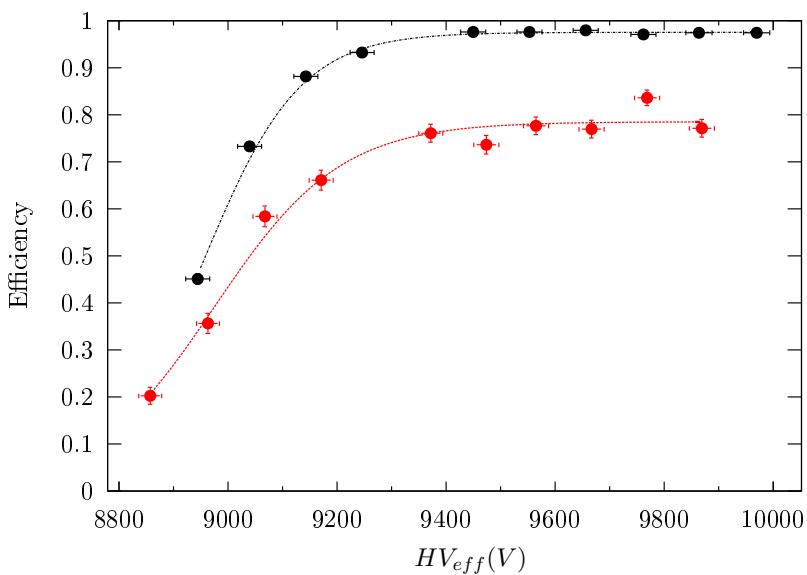
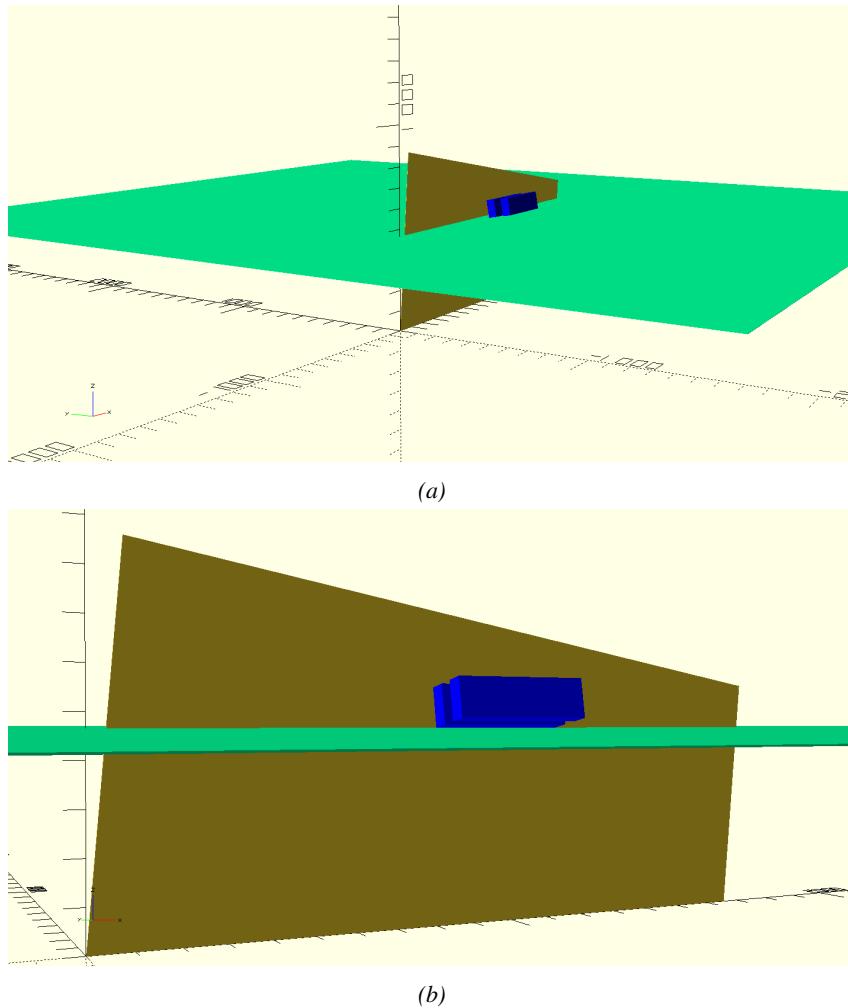


Figure 4.12: Results are derived from data taken on half-partition B2 only. On the 18<sup>th</sup> of June 2014, data has been taken on chamber RE-2-BARC-161 at building 904 (Prevessin Site) with cosmic muons providing us a reference efficiency plateau of  $(97.54 \pm 0.15)\%$  represented by a black curve. A similar measurement has been done at GIF on the 21<sup>st</sup> of July with the same chamber giving a plateau of  $(78.52 \pm 0.94)\%$  represented by a red curve.

#### 897 4.3.3.1 Description of the simulation layout

898 The layout of GIF setup has been reproduced and incorporated into a Monte Carlo (MC) simulation  
 899 to study the influence of the disposition of the telescope on the final distribution measured by the  
 900 RPC. A 3D view of the simulated layout is given into Figure 4.13. Muons are generated randomly  
 901 in a horizontal plane located at a height corresponding to the lowest point of the PMTs. This way,  
 902 the needed size of the plane in order to simulate events happening at very big azimuthal angles (i.e.  
 903  $\theta \approx \pi$ ) can be kept relatively small. The muon flux is designed to follow the usual  $\cos^2\theta$  distribution  
 904 for cosmic particle. The goal of the simulation is to look at muons that pass through the muon  
 905 telescope composed of the two scintillators and define their distribution onto the RPC plane. During  
 906 the reconstruction, the RPC plane is then divided into its strips and each muon track is assigned to a  
 907 strip.



*Figure 4.13: Representation of the layout used for the simulations of the test setup. The RPC is represented as a yellow trapezoid while the two scintillators as blue cuboids looking at the sky. A green plane corresponds to the muon generation plane within the simulation. Figure 4.9a shows a global view of the simulated setup. Figure 4.9b shows a zoomed view that allows to see the 2 scintillators as well as the full RPC plane.*

908 In order to further refine the quality of the simulation and understand deeper the results the  
 909 dependance of the distribution has been studied for a range of telescope inclinations. Moreover,  
 910 the threshold applied on the PMT signals has been included into the simulation in the form of a  
 911 cut. In the approximation of uniform scintillators, it has been considered that the threshold can be  
 912 understood as the minimum distance particles need to travel through the scintillating material to give  
 913 a strong enough signal. Particles that travel a distance smaller than the set "threshold" are thus not  
 914 detected by the telescope and cannot trigger the data taking. Finally, the FEE threshold also has  
 915 been considered in a similar way. The mean momentum of horizontal cosmic rays is higher than  
 916 those of vertical ones but the stopping power of matter for momenta ranging from 1 GeV to 1 TeV  
 917 stays comparable. It is then possible to assume that the mean number of primary  $e^-/ion$  pairs per  
 918 unit length will stay similar and thus, depending on the applied discriminator threshold, muons with

919 the shortest path through the gas volume will deposit less charge and induce a smaller signal on the  
 920 pick-up strips that could eventually not be detected. These two thresholds also restrain the overall  
 921 geometrical acceptance of the system.

922 **4.3.3.2 Simulation procedure**

923 The simulation software has been designed using C++ and the output data is saved into ROOT  
 924 histograms. Simulations start for a threshold  $T_{scint}$  varying in a range from 0 to 45 mm in steps  
 925 of 5 mm, where  $T_{scint} = 0$  mm corresponds to the case where there isn't any threshold apply on  
 926 the input signal while  $T_{scint} = 45$  mm, which is the scintillator thickness, is the case where muons  
 927 cannot arrive orthogonally onto the scintillator surface. For a given  $T_{scint}$ , a set of RPC thresholds  
 928 are considered. The RPC threshold,  $T_{RPC}$  varies from 2 mm, the thickness of the gas volume, to  
 929 3 mm in steps of 0.25 mm. For each  $(T_{scint}; T_{RPC})$  pair,  $N_\mu = 10^8$  muons are randomly generated  
 930 inside the muon plane described in the previous paragraph with an azimuthal angle  $\theta$  chosen to follow  
 931 a  $\cos^2\theta$  distribution.

932 Planes are associated to each surface of the scintillators. Knowing muon position into the muon  
 933 plane and its direction allows us, by assuming that muons travel in a straight line, to compute the  
 934 intersection of the muon track with these planes. Applying conditions to the limits of the surfaces  
 935 of the scintillator faces then gives us an answer to whether or not the muon passed through the  
 936 scintillators. In the case the muon has indeed passed through the telescope, the path through each  
 937 scintillator is computed and muons whose path was shorter than  $T_{scint}$  are rejected and are thus  
 938 considered as having not interacted with the setup.

939 On the contrary, if the muon is labeled as good, its position within the RPC plane is computed  
 940 and the corresponding strip, determined by geometrical tests in the case the distance through the  
 941 gas volume was enough not to be rejected because of  $T_{RPC}$ , gets a hit and several histograms  
 942 are filled in order to keep track of the generation point on the muon plane, the intersection points  
 943 of the reconstructed muons within the telescope, or on the RPC plane, the path traveled through  
 944 each individual scintillator or the gas volume, as well as other histograms. Moreover, muons fill  
 945 different histograms whether they are forward or backward coming muons. They are discriminated  
 946 according to their direction components. When a muon is generated, an  $(x, y, z)$  position is assigned  
 947 into the muon plane as well as a  $(\theta; \phi)$  pair that gives us the direction it's coming from. This way,  
 948 muons satisfying the condition  $0 \leq \phi < \pi$  are designated as backward coming muons while muons  
 949 satisfying  $\pi \leq \phi < 2\pi$  as forward coming muons.

950 This simulation is then repeated for different telescope inclinations ranging in between 4 and 20°  
 951 and varying in steps of 2°. Due to this inclination and to the vertical position of the detector under  
 952 test, the muon distribution reconstructed in the detector plane is asymmetrical. The choice as been  
 953 made to chose a skew distribution formula to fit the data built as the multiplication of gaussian and  
 954 sigmoidal curves together. A typical gaussian formula is given as 4.1 and has three free parameters  
 955 as  $A_g$ , its amplitude,  $\bar{x}$ , its mean value and  $\sigma$ , its root mean square. Sigmoidal curves as given by  
 956 formula 4.2 are functions converging to 0 and  $A_s$  as  $x$  diverges. The inflection point is given as  $x_i$   
 957 and  $\lambda$  is proportional to the slope at  $x = x_i$ . In the limit where  $\lambda \rightarrow \infty$ , the sigmoid becomes a  
 958 step function.

$$g(x) = A_g e^{-\frac{(x-\bar{x})^2}{2\sigma^2}} \quad (4.1)$$

$$s(x) = \frac{A_s}{1 + e^{-\lambda(x-x_i)}} \quad (4.2)$$

Finally, a possible representation of a skew distribution is given by formula 4.3 and is the product of 4.1 and 4.2. Naturally, here  $A_{sk} = A_g \times A_s$  and represents the theoretical maximum in the limit where the skew tends to a gaussian function.

$$sk(x) = g(x) \times s(x) = A_{sk} \frac{e^{\frac{-(x-\bar{x})^2}{2\sigma^2}}}{1 + e^{-\lambda(x-x_i)}} \quad (4.3)$$

### 4.3.3.3 Results

#### Influence of $T_{scint}$ on the muon distribution

#### Influence of $T_{RPC}$ on the muon distribution

#### Influence of the telescope inclination on the muon distribution

#### Comparison to data taken at GIF without irradiation

### 4.3.4 Photon flux at GIF

#### 4.3.4.1 Expectations from simulations

In order to understand and evaluate the  $\gamma$  flux in the GIF area, simulations had been conducted in 1999 and published by S. Agosteo et al [55]. Table 4.1 presented in this article gives us the  $\gamma$  flux for different distances  $D$  to the source. This simulation was done using GEANT and a Monte Carlo N-Particle (MCNP) transport code, and the flux  $F$  is given in number of  $\gamma$  per unit area and unit time along with the estimated error from these packages expressed in %.

| Nominal ABS | Photon flux $F$ [ $s^{-1}cm^{-2}$ ] |                             |                             |                             |
|-------------|-------------------------------------|-----------------------------|-----------------------------|-----------------------------|
|             | at $D = 50$ cm                      | at $D = 155$ cm             | at $D = 300$ cm             | at $D = 400$ cm             |
| 1           | $0.12 \cdot 10^8 \pm 0.2\%$         | $0.14 \cdot 10^7 \pm 0.5\%$ | $0.45 \cdot 10^6 \pm 0.5\%$ | $0.28 \cdot 10^6 \pm 0.5\%$ |
| 2           | $0.68 \cdot 10^7 \pm 0.3\%$         | $0.80 \cdot 10^6 \pm 0.8\%$ | $0.25 \cdot 10^6 \pm 0.8\%$ | $0.16 \cdot 10^6 \pm 0.6\%$ |
| 5           | $0.31 \cdot 10^7 \pm 0.4\%$         | $0.36 \cdot 10^6 \pm 1.2\%$ | $0.11 \cdot 10^6 \pm 1.2\%$ | $0.70 \cdot 10^5 \pm 0.9\%$ |

Table 4.1: Total photon flux ( $E\gamma \leq 662$  keV) with statistical error predicted considering a  $^{137}Cs$  activity of 740 GBq at different values of the distance  $D$  to the source along the  $x$ -axis of irradiation field [55].

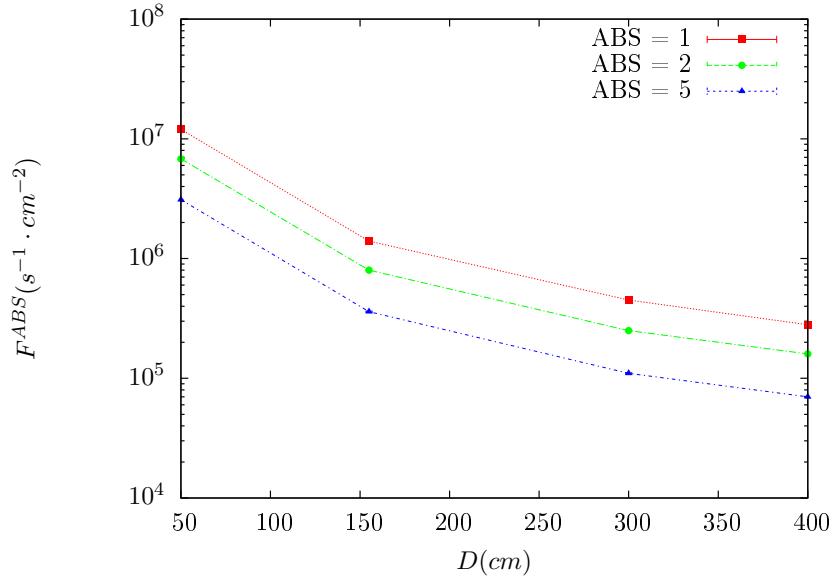


Figure 4.14:  $\gamma$  flux  $F(D)$  is plot using values from table 4.1. As expected, the plot shows similar attenuation behaviours with increasing distance for each absorption factors.

The simulation doesn't directly provide us with an estimated flux at the level of our RPC. First of all, it is needed to extract the value of the flux from the available data contained in the original paper and then to estimate the flux in 2014 at the time the experimentation took place. Figure 4.14 that contains the data from Table 4.1. In the case of a pointlike source emitting isotropic and homogeneous gamma radiations, the gamma flux  $F$  at a distance  $D$  to the source with respect to a reference point situated at  $D_0$  where a known flux  $F_0$  is measured will be expressed like in Equation 4.4, assuming that the flux decreases as  $1/D^2$ , where  $c$  is a fitting factor.

$$F^{ABS} = F_0^{ABS} \times \left( \frac{cD_0}{D} \right)^2 \quad (4.4)$$

By rewriting Equation 4.4, it comes that :

$$c = \frac{D}{D_0} \sqrt{\frac{F^{ABS}}{F_0^{ABS}}} \quad (4.5)$$

$$\Delta c = \frac{c}{2} \left( \frac{\Delta F^{ABS}}{F^{ABS}} + \frac{\Delta F_0^{ABS}}{F_0^{ABS}} \right) \quad (4.6)$$

Finally, using Equation 4.5 and the data in Table 4.1 with  $D_0 = 50$  cm as reference point, we can build Table 4.2. It is interesting to note that  $c$  for each value of  $D$  doesn't depend on the absorption factor.

| Nominal<br>ABS | Correction factor $c$ |                    |                    |
|----------------|-----------------------|--------------------|--------------------|
|                | at $D = 155$ cm       | at $D = 300$ cm    | at $D = 400$ cm    |
| 1              | $1.059 \pm 0.70\%$    | $1.162 \pm 0.70\%$ | $1.222 \pm 0.70\%$ |
| 2              | $1.063 \pm 1.10\%$    | $1.150 \pm 1.10\%$ | $1.227 \pm 0.90\%$ |
| 5              | $1.056 \pm 1.60\%$    | $1.130 \pm 1.60\%$ | $1.202 \pm 1.30\%$ |

Table 4.2: Correction factor  $c$  is computed thanks to formulae 4.5 taking as reference  $D_0 = 50$  cm and the associated flux  $F_0^{ABS}$  for each absorption factor available in table 4.1.

985 For the range of  $D/D_0$  values available, it is possible to use a simple linear fit to get the evolution  
 986 of  $c$ . The linear fit will then use only 2 free parameters,  $a$  and  $b$ , as written in Equation 4.7. This gives  
 987 us the results showed in Figure 4.15. Figure 4.15b confirms that using only a linear fit to extract  $c$  is  
 988 enough as the evolution of the rate that can be obtained superimposes well on the simulation points.

$$c \left( \frac{D}{D_0} \right) = a \frac{D}{D_0} + b \quad (4.7)$$

$$F^{ABS} = F_0^{ABS} \left( a + \frac{b D_0}{D} \right)^2 \quad (4.8)$$

$$\Delta F^{ABS} = F^{ABS} \left[ \frac{\Delta F_0^{ABS}}{F_0^{ABS}} + 2 \frac{\Delta a + \Delta b \frac{D_0}{D}}{a + \frac{b D_0}{D}} \right] \quad (4.9)$$

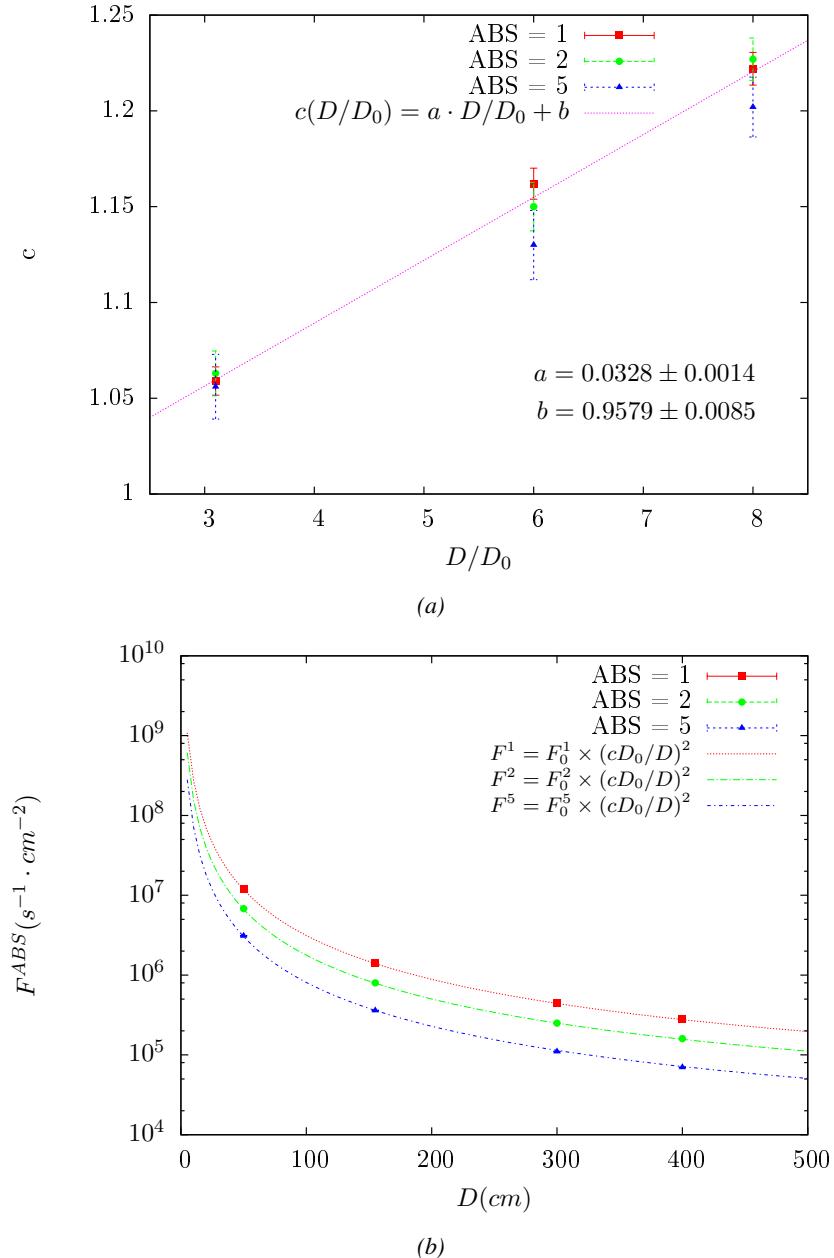


Figure 4.15: Figure 4.15a shows the linear approximation fit done via formulae 4.7 on data from table 4.2. Figure 4.15b shows a comparison of this model with the simulated flux using  $a$  and  $b$  given in figure 4.15a in formulae 4.4 and the reference value  $D_0 = 50\text{cm}$  and the associated flux for each absorption factor  $F_0^{ABS}$  from table 4.1

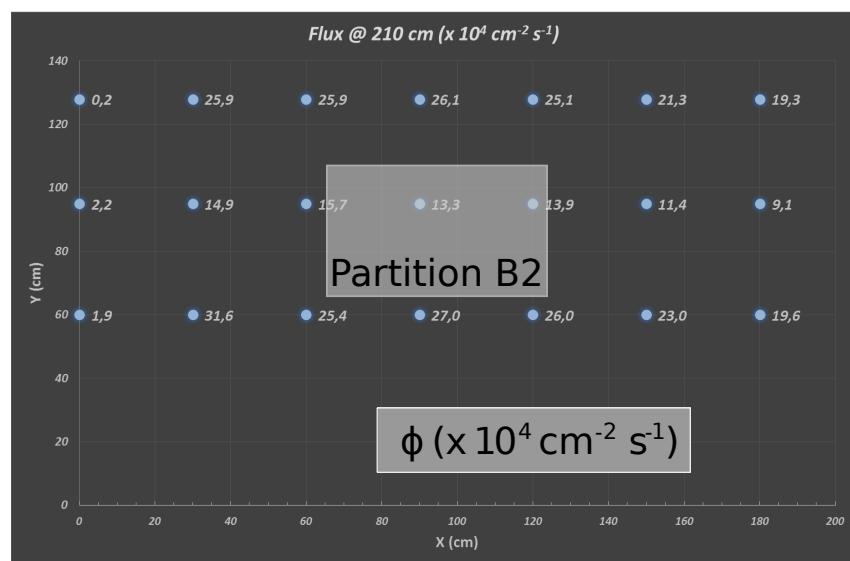
In the case of the 2014 GIF tests, the RPC plane is located at a distance  $D = 206\text{ cm}$  to the source. Moreover, to estimate the strength of the flux in 2014, it is necessary to consider the nuclear decay through time associated to the Cesium source whose half-life is well known ( $t_{1/2} = (30.05 \pm 0.08)\text{ y}$ ). The very first source activity measurement has been done on the 5<sup>th</sup> of March 1997 while the GIF

993 tests were done in between the 20<sup>th</sup> and the 31<sup>st</sup> of August 2014, i.e. at a time  $t = (17.47 \pm 0.02)$  y  
 994 resulting in an attenuation of the activity from 740 GBq in 1997 to 494 GBq in 2014. All the needed  
 995 information to extrapolate the flux through our detector in 2014 has now been assembled, leading  
 996 to the Table 4.3. It is interesting to note that for a common RPC sensitivity to  $\gamma$  of  $2 \cdot 10^{-3}$ , the  
 997 order of magnitude of the estimated hit rate per unit area is of the order of the kHz for the fully  
 998 opened source. Moreover, taking profit of the two working absorbers, it will be possible to scan  
 999 background rates at 0 Hz,  $\sim 300$  Hz as well as  $\sim 600$  Hz. Without source, a good estimate of the  
 1000 intrinsic performance will be available. Then at 300 Hz, the goal will be to show that the detectors  
 1001 fulfill the performance certification of CMS RPCs. Then a first idea of the performance of the  
 1002 detectors at higher background will be provided with absorption factors 2 ( $\sim 600$  Hz) and 1 (no  
 1003 absorption). [Here I will also put a reference to the plot showing the estimated background rate at  
 1004 the level of RE3/1 in the case of HL-LHC but this one being in another chapter, I will do it later.]

| Nominal ABS | Photon flux $F$ [ $s^{-1}cm^{-2}$ ] |                             |                             | Hit rate/unit area [ $Hz cm^{-2}$ ]<br>at $D^{2014} = 206$ cm |
|-------------|-------------------------------------|-----------------------------|-----------------------------|---|
|             | at $D_0^{1997} = 50$ cm             | at $D^{1997} = 206$ cm      | at $D^{2014} = 206$ cm      |   |
| 1           | $0.12 \cdot 10^8 \pm 0.2\%$         | $0.84 \cdot 10^6 \pm 0.3\%$ | $0.56 \cdot 10^6 \pm 0.3\%$ | $1129 \pm 32$   |
| 2           | $0.68 \cdot 10^7 \pm 0.3\%$         | $0.48 \cdot 10^6 \pm 0.3\%$ | $0.32 \cdot 10^6 \pm 0.3\%$ | $640 \pm 19$  |
| 5           | $0.31 \cdot 10^7 \pm 0.4\%$         | $0.22 \cdot 10^6 \pm 0.3\%$ | $0.15 \cdot 10^6 \pm 0.3\%$ | $292 \pm 9$   |

Table 4.3: The data at  $D_0$  in 1997 is taken from [55]. In a second step, using Equations 4.8 and 4.9, the flux at  $D$  can be estimated in 1997. Then, taking into account the attenuation of the source activity, the flux at  $D$  can be estimated at the time of the tests in GIF in 2014. Finally, assuming a sensitivity of the RPC to  $\gamma$   $s = 2 \cdot 10^{-3}$ , an estimation of the hit rate per unit area is obtained.

<sup>1005</sup> **4.3.4.2 Dose measurements**



*Figure 4.16: Dose measurements has been done in a plane corresponding to the tents front side. This plan is 1900 mm away from the source. As explained in the first chapter, a lens-shaped lead filter provides a uniform photon flux in the vertical plan orthogonal to the beam direction. If the second line of measured fluxes is not taken into account because of lower values due to experimental equipments in the way between the source and the tent, the uniformity of the flux is well showed by the results.*

<sup>1006</sup> **4.3.5 Results and discussions**

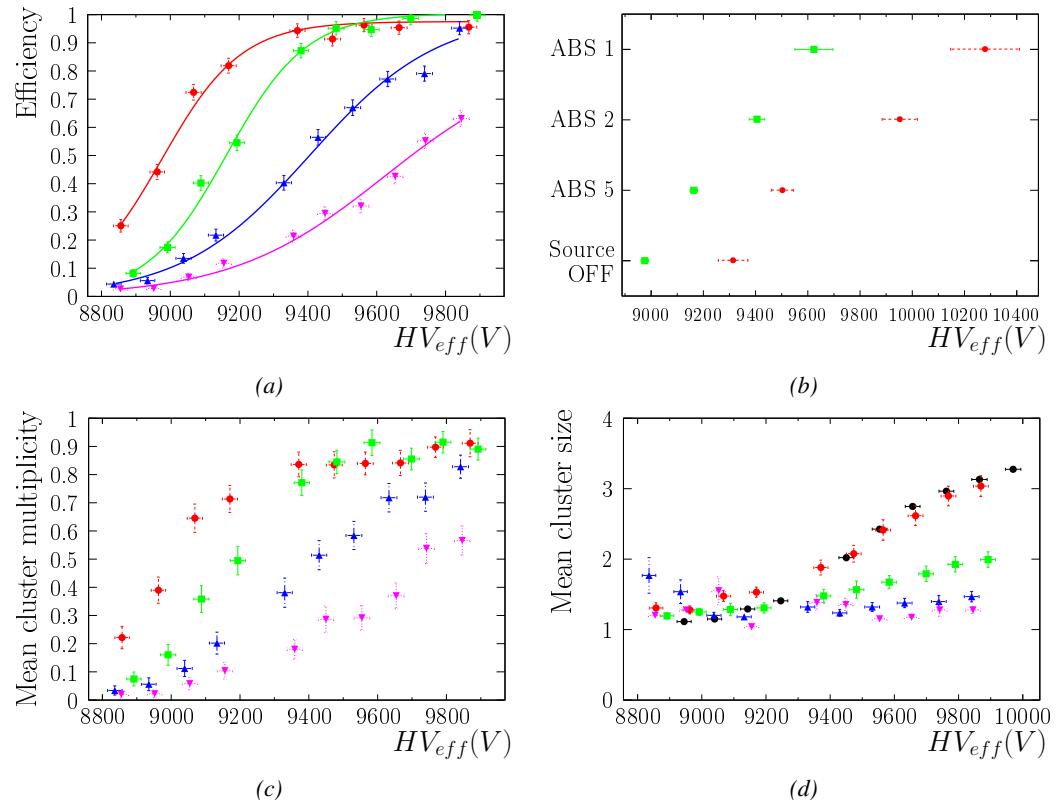


Figure 4.17

## 1007 4.4 Longevity tests at GIF++

1008 Longevity studies imply a monitoring of the performance of the detectors probed using a high inten-  
1009 sity muon beam in a irradiated environment by periodically measuring their rate capability, the dark  
1010 current running through them and the bulk resistivity of the Bakelite composing their electrodes.  
1011 GIF++, with its very intense  $^{137}\text{Cs}$  source, provides the perfect environment to perform such kind  
1012 of tests. Assuming a maximum acceleration factor of 3, it is expected to accumulate the equivalent  
1013 charge in 1.7 years.

1014 As the maximum background is found in the endcap, the choice naturally was made to focus the  
1015 GIF++ longevity studies on endcap chambers. Most of the RPC system was installed in 2007. Nev-  
1016 ertheless, the large chambers in the fourth endcap (RE4/2 and RE4/3) have been installed during  
1017 LS1 in 2014. The Bakelite of these two different productions having different properties, four spare  
1018 chambers of the present system were selected, two RE2,3/2 spares and two RE4/2 spares. Having  
1019 two chambers of each type allows to always keep one of them non irradiated as reference, the per-  
1020 formance evolution of the irradiated chamber being then compared through time to the performance  
1021 of the non irradiated one.

1022 The performance of the detectors under different level of irradiation is measured periodically dur-  
1023 ing dedicated test beam periods using the H4 muon beam. In between these test beam periods, the  
1024 two RE2,3/2 and RE4/2 chambers selected for this study are irradiated by the  $^{137}\text{Cs}$  source in order  
1025 to accumulate charge and the gamma background is monitored, as well as the currents. The two  
1026 remaining chambers are kept non-irradiated as reference detectors. Due to the limited gas flow in  
1027 GIF++, the RE4 chamber remained non-irradiated until end of November 2016 where a new mass  
1028 flow controller has been installed allowing for bigger volumes of gas to flow in the system.

1029 Figures 4.18 and 4.19 give us for different test beam periods, and thus for increasing integrated  
1030 charge through time, a comparison of the maximum efficiency, obtained using a sigmoid-like func-  
1031 tion, and of the working point of both irradiated and non irradiated chambers [**SIGMOID2005**]. No  
1032 aging is yet to see from this data, the shifts in  $\gamma$  rate per unit area in between irradiated and non  
1033 irradiated detectors and RE2 and RE4 types being easily explained by a difference of sensitivity due  
1034 to the various Bakelite resistivities of the HPL electrodes used for the electrode production.

1035 Collecting performance data at each test beam period allows us to extrapolate the maximum effi-  
1036 ciency for a background hit rate of  $300\text{ Hz}/\text{cm}^2$  corresponding to the expected HL-LHC conditions.  
1037 Aging effects could emerge from a loss of efficiency with increasing integrated charge over time,  
1038 thus Figure 4.20 helps us understand such degradation of the performance of irradiated detectors in  
1039 comparison with non irradiated ones. The final answer for an eventual loss of efficiency is given in  
1040 Figure 4.21 by comparing for both irradiated and non irradiated detectors the efficiency sigmoids  
1041 before and after the longevity study. Moreover, to complete the performance information, the Bake-  
1042 lite resistivity is regularly measured thanks to  $Ag$  scans (Figure 4.22) and the noise rate is monitored  
1043 weekly during irradiation periods (Figure 4.23). At the end of 2016, no signs of aging were observed  
1044 and further investigation is needed to get closer to the final integrated charge requirements proposed  
1045 for the longevity study of the present CMS RPC sub-system.

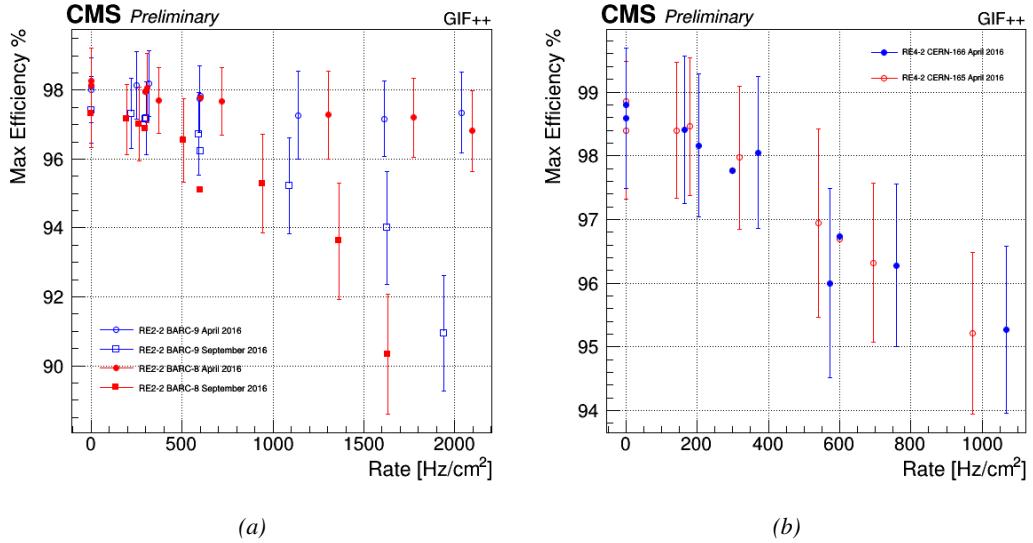


Figure 4.18: Evolution of the maximum efficiency for RE2 (4.18a) and RE4 (4.18b) chambers with increasing extrapolated  $\gamma$  rate per unit area at working point. Both irradiated (blue) and non irradiated (red) chambers are shown.

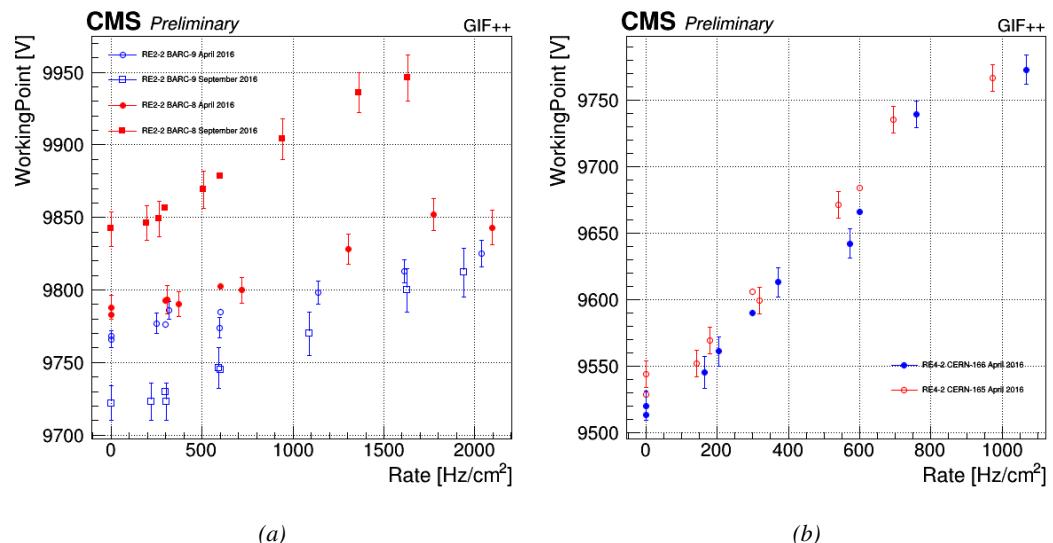
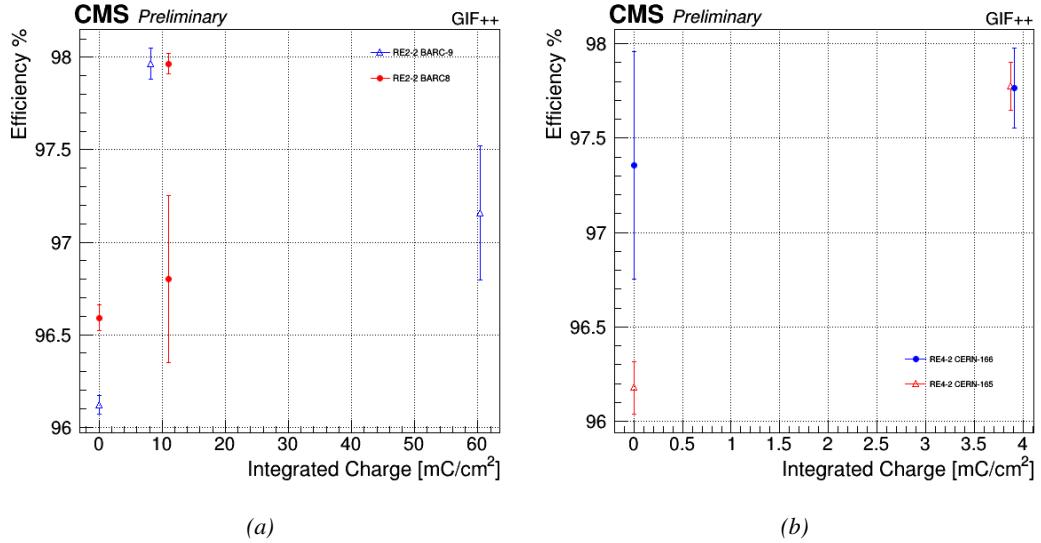
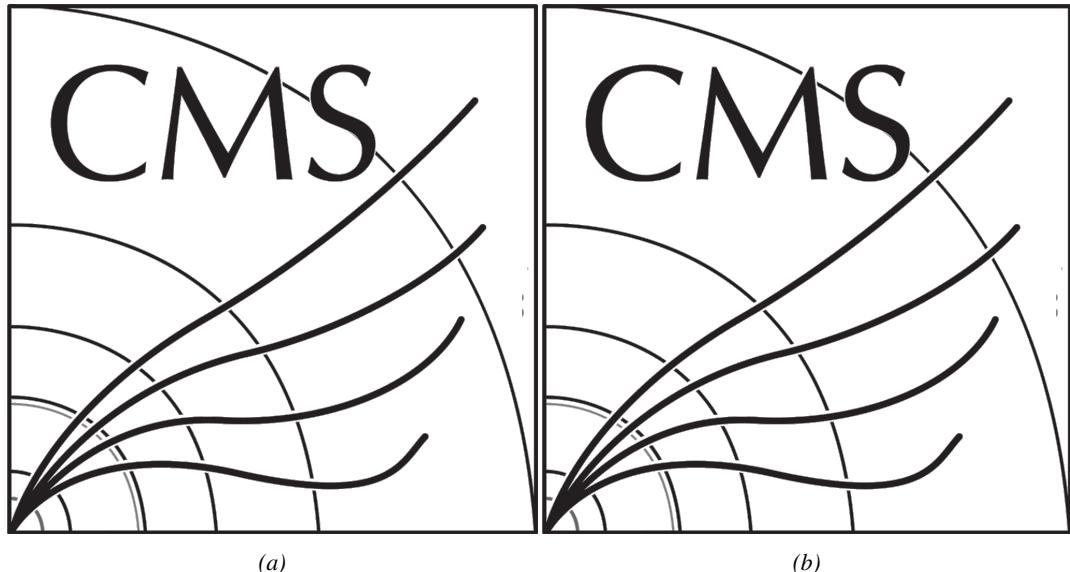


Figure 4.19: Evolution of the working point for RE2 (4.19a) and RE4 (4.19b) with increasing extrapolated  $\gamma$  rate per unit area at working point. Both irradiated (blue) and non irradiated (red) chambers are shown.



*Figure 4.20: Evolution of the maximum efficiency at HL-LHC conditions, i.e. a background hit rate per unit area of 300 Hz/cm<sup>2</sup>, with increasing integrated charge for RE2 (4.20a) and RE4 (4.20b) detectors. Both irradiated (blue) and non irradiated (red) chambers are shown. The integrated charge for non irradiated detectors is recorded during test beam periods and stays small with respect to the charge accumulated in irradiated chambers.*



*Figure 4.21: Comparison of the efficiency sigmoid before (triangles) and after (circles) irradiation for RE2 (4.21a) and RE4 (4.21b) detectors. Both irradiated (blue) and non irradiated (red) chambers are shown.*

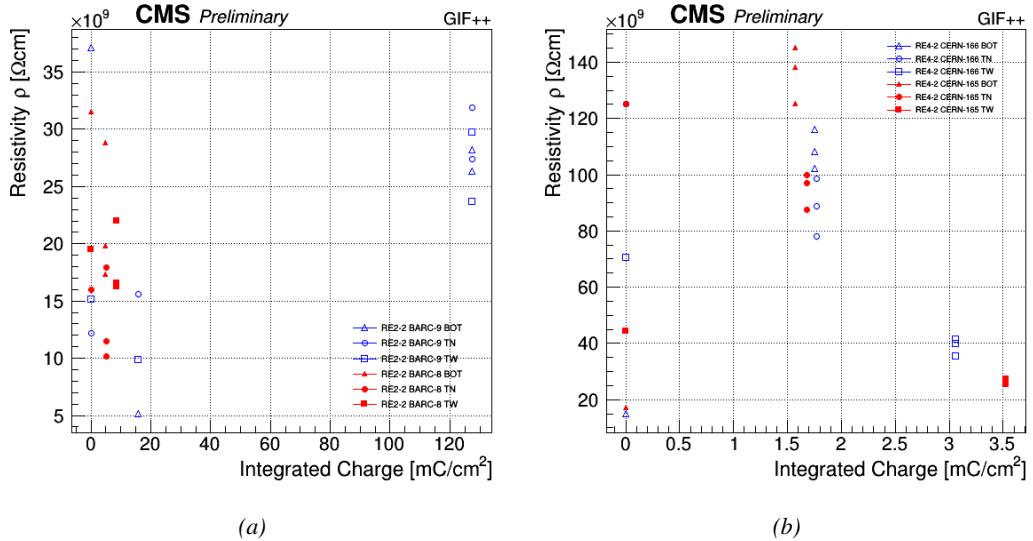


Figure 4.22: Evolution of the Bakelite resistivity for RE2 (4.22a) and RE4 (4.22b) detectors. Both irradiated (blue) and non-irradiated (red) chambers are shown.

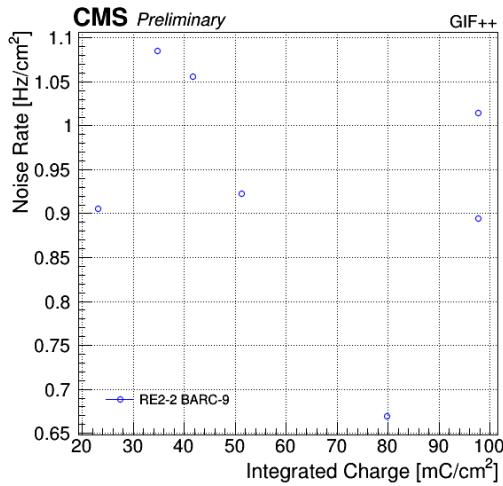


Figure 4.23: Evolution of the noise rate per unit area for the irradiated chamber RE2-2-BARC-9 only.

#### 4.4.1 Description of the Data Acquisition

For the longevity studies, four spare chambers of the present system are used. Two spare RPCs of the RE2,3 stations as well as two spare RPCs from the new RE4 stations have been mounted in a Trolley. Six RE4 gaps are also placed in the trolley. The trolley is placed inside the Gif++ in the upstream region of the bunker, taking the cesium source as a reference. The trolley is oriented for the detection surface of the chambers to be orthogonal to the beam line. The system can be moved along the orthogonal plane in order to have the beam in all  $\eta$ -partitions. For the aging the trolley is

1054 moved outside the beam line and is placed in a distance of 5.2 m to the source, which irradiates the  
 1055 bunker using an attenuation filter of 2.2 which corresponds to a fluence of  $10^7 \text{ gamma/cm}^2$ .

1056 During GIF++ operation, the data collected can be divided into different categories as several  
 1057 parameters are monitored in addition to the usual RPC performance data. On one hand, to know  
 1058 the performance of a chamber, it is need to measure its efficiency and to know the background  
 1059 conditions in which it is operated. To do this, the hit signals from the chamber are recorded and  
 1060 stored in a ROOT file via a Data Acquisition (DAQ) software. On the other hand, it is also very  
 1061 important to monitor parameters such as environmental pressure and temperature, gas temperature  
 1062 and humidity, RPC HV, LV, and currents, or even source and beam status. This is done through the  
 1063 GIF++ web Detector Control Software (DCS) that stores this information in a database.

1064 Two different types of tests are conducted on RPCs via the DAQ. Indeed, the performance of the  
 1065 detectors is measured periodically during dedicated test beam periods using the H4 muon beam. In  
 1066 between these test beam periods, when the beam is not available, the chambers are irradiated by the  
 1067  $^{137}\text{Cs}$  in order to accumulate deposited charge and the gamma background is measured.

1068 RPCs under test are connected through LVDS cables to V1190A Time-to-Digital Converter  
 1069 (TDC) modules manufactured by CAEN. These modules, located in the rack area outside of the  
 1070 bunker, get the logic signals sent by the chambers and save them into their buffers. Due to the  
 1071 limited size of the buffers, the collected data is regularly erased and replaced. A trigger signal is  
 1072 needed for the TDC modules to send the useful data to the DAQ computer via a V1718 CAEN USB  
 1073 communication module.

1074 In the case of performance test, the trigger signal used for data acquisition is generated by the  
 1075 coincidence of three scintillators. A first one is placed upstream outside of the bunker, a second one  
 1076 is placed downstream outside of the bunker, while a third one is placed in front of the trolley, close by  
 1077 the chambers. Every time a trigger is sent to the TDCs, i.e. every time a muon is detected, knowing  
 1078 the time delay in between the trigger and the RPC signals, signals located in the right time window  
 1079 are extracted from the buffers and saved for later analysis. Signals are taken in a time window of  
 1080 400 ns centered on the muon peak (here we could show a time spectrum). On the other hand, in the  
 1081 case of background rate measurement, the trigger signal needs to be "random" not to measure muons  
 1082 but to look at gamma background. A trigger pulse is continuously generated at a rate of 300 Hz using  
 1083 a dual timer. To integrate an as great as possible time, all signals contained within a time window of  
 1084 10us prior to the random trigger signal are extracted form the buffers and saved for further analysis  
 1085 (here another time spectrum to illustrate could be useful, maybe even place both spectrum together  
 1086 as a single Figure).

1087 The signals sent to the TDCs correspond to hit collections in the RPCs. When a particle hits  
 1088 a RPC, it induce a signal in the pickup strips of the RPC readout. If this signal is higher than the  
 1089 detection threshold, a LVDS signal is sent to the TDCs. The data is then organised into 4 branches  
 1090 keeping track of the event number, the hit multiplicity for the whole setup, and the time and channel  
 1091 profile of the hits in the TDCs.

#### 1092 **4.4.2 RPC current, environmental and operation parameter monitoring**

1093 In order to take into account the variation of pressure and temperature between different data taking  
 1094 periods the applied voltage is corrected following the relationship :

$$1095 \quad HV_{eff} = HV_{app} \times \left( 0.2 + 0.8 \cdot \frac{P_0}{P} \times \frac{T}{T_0} \right) \quad (4.10)$$

1095 where  $T_0$  (=293 K) and  $P_0$  (=990 mbar) are the reference values.

1096 **4.4.3 Measurement procedure**

1097 Insert a short description of the online tools (DAQ, DCS, DQM).

1098 Insert a short description of the offline tools : tracking and efficiency algorithm.

1099 Identify long term aging effects we are monitoring the rates per strip.

1100 **4.4.4 Longevity studies results**

# 5

1101

1102

## Investigation on high rate RPCs

1103 **5.1 Rate limitations and ageing of RPCs**

1104 **5.1.1 Low resistivity electrodes**

1105 **5.1.2 Low noise front-end electronics**

1106 **5.2 Construction of prototypes**

1107 **5.3 Results and discussions**



# 6

## Conclusions and outlooks

1108

1109

### **1110 6.1 Conclusions**

### **1111 6.2 Outlooks**



# A

1112

1113

## A data acquisition software for CAEN VME TDCs

1114

1115 Certifying detectors in the perspective of HL-LHC required to develop tools for the GIF++ expe-  
1116 riment. Among them was the C++ Data Acquisition (DAQ) software that allows to make the com-  
1117 munications in between a computer and TDC modules in order to retrieve the RPC data [57]. In this  
1118 appendix, details about this software, as of how the software was written, how it functions and how  
1119 it can be exported to another similar setup, will be given.

1120

### A.1 GIF++ DAQ file tree

1121 GIF++ DAQ source code is fully available on github at [https://github.com/afagot/GIF\\_](https://github.com/afagot/GIF_DAQ)  
1122 DAQ. The software requires 3 non-optional dependencies:

1123

- CAEN USB Driver, to mount the VME hardware,
- CAEN VME Library, to communicate with the VME hardware, and
- ROOT, to organize the collected data into a TTree.

1126

1127

The CAEN VME library will not be packaged by distributions and will need to be installed man-  
ually. To compile the GIF++ DAQ project via a terminal, from the DAQ folder use the command:

1128

```
1129     make
```

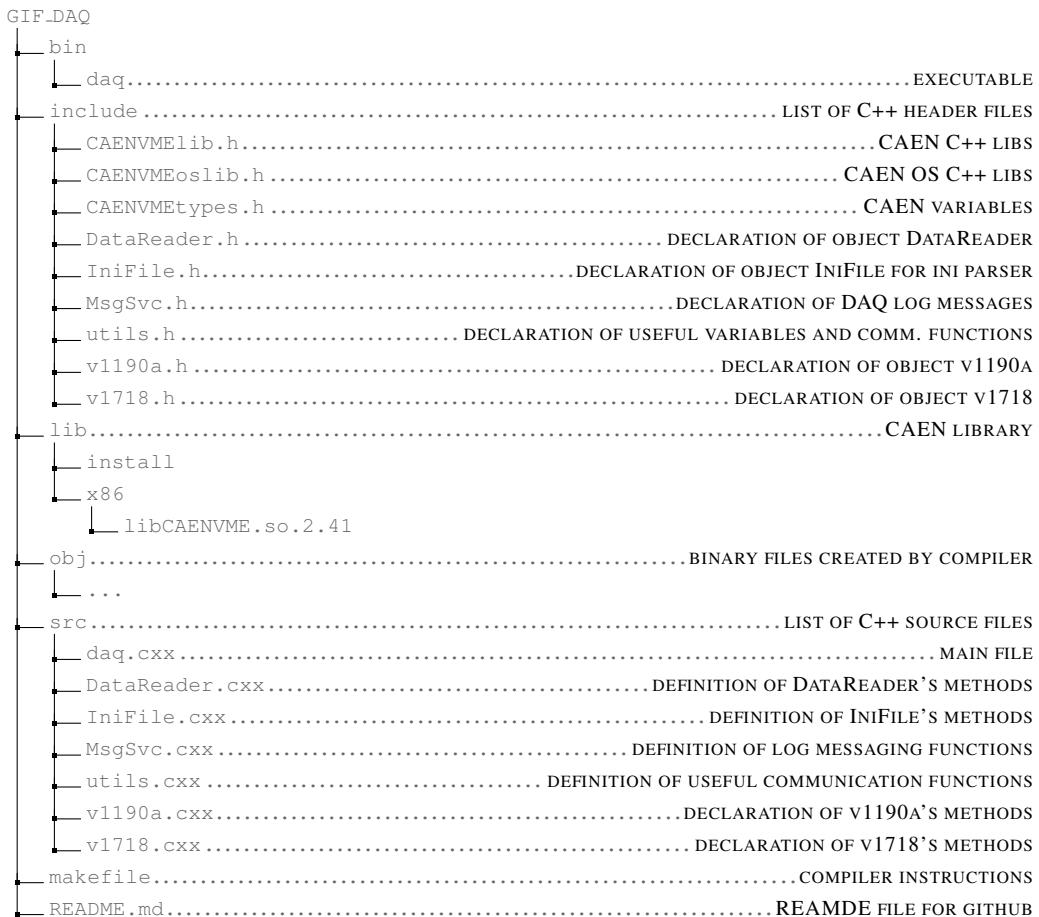
1130

1131

The source code tree is provided below along with comments to give an overview of the files' con-  
tent. The different objects created for this project (`v1718`, `v1190a`, `IniFile` & `DataReader`) will be  
described in details in the following sections.

1132

1133



## 1134 A.2 Usage of the DAQ

1135 GIF++ DAQ, as used in GIF++, is not a standalone software. Indeed, the system being more complexe,  
 1136 the DAQ only is a sub-layer of the software architecture developped to control and monitor  
 1137 the RPCs that are placed into the bunker for performance study in an irradiated environment. The top  
 1138 layer of GIF++ is a Web Detector Control System (webDCS) application. The DAQ is only called  
 1139 by the webDCS when data needs to be acquired. The webDCS operates the DAQ through command  
 1140 line. To start the DAQ, the webDCS calls:

1141

1142   bin/daq /path/to/the/log/file/in/the/output/data/folder

1143 where /path/to/the/log/file/in/the/output/data/folder is the only argument required. This  
 1144 log file is important for the webDCS as this file contains all the content of the communication of the  
 1145 webDCS and the different systems monitored by the webDCS. Its content is constantly displayed  
 1146 during data taking for the users to be able to follow the operations. The communication messages  
 1147 are normally sent to the webDCS log file via the functions declared in file MsgSvc.h, typically  
 1148 MSG\_INFO(string message).

1149

### 1150 A.3 Description of the readout setup

1151 The CMS RPC setup at GIF++ counts 5 V1190A Time-to-Digital Converter (TDC) manufactured  
 1152 by CAEN [58]. V1190A are VME units accepting 128 independent Multi-Hit/Multi-Event TDC  
 1153 channels whose signals are treated by 4 100 ps high performance TDC chips developed by CERN  
 1154 / ECP-MIC Division. The communication between the computer and the TDCs to transfer data is  
 1155 done via a V1718 VME master module also manufactured by CAEN and operated from a USB  
 1156 port [59]. These VME modules are all hosted into a 6U VME 6021 powered crate manufactured by  
 1157 W-Ie-Ne-R than can accommodate up to 21 VME bus cards [60]. These 3 components of the DAQ  
 1158 setup are shown in Figure A.1.

1159

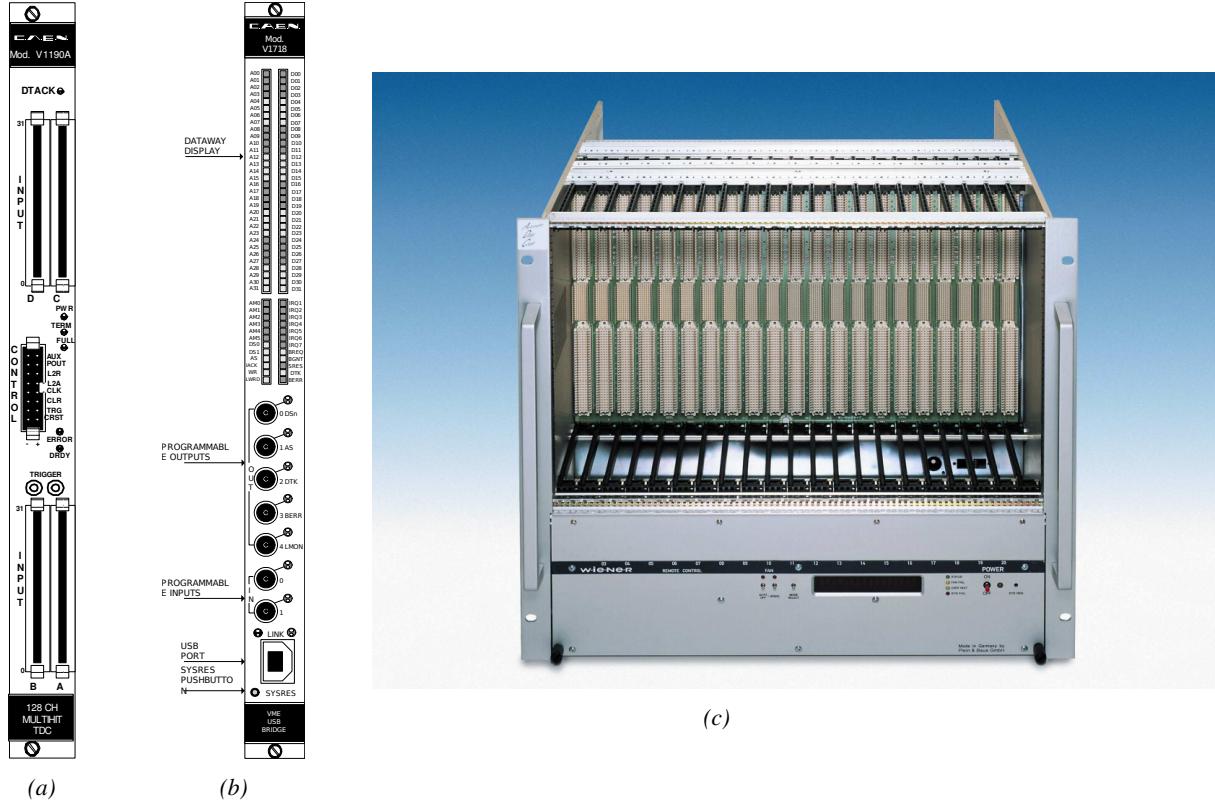


Figure A.1: (A.1a) View of the front panel of a V1190A TDC module [58]. (A.1b) View of the front panel of a V1718 Bridge module [59]. (A.1c) View of the front panel of a 6U 6021 VME crate [60].

1160

### A.4 Data read-out

1161 To efficiently perform a data readout algorithm, C++ objects to handle the VME modules (TDCs  
 1162 and VME bridge) have been created along with objects to store data and read the configuration file

1163 that comes as an input of the DAQ software.

1164

#### 1165 A.4.1 V1190A TDCs

1166 The DAQ used at GIF takes profit of the *Trigger Matching Mode* offered by V1190A modules.  
 1167 This setting is enabled through the method `v1190a::SetTrigMatching (int ntdcs)` where `ntdcs`  
 1168 is the total number of TDCs in the setup this setting needs to be enabled for (Source Code A.1). A  
 1169 trigger matching is performed in between a trigger time tag, a trigger signal sent into the TRIGGER  
 1170 input of the TDC visible on Figure A.1a, and the channel time measurements, signals recorded from  
 1171 the detectors under test in our case. Control over this data acquisition mode, explained through  
 1172 Figure A.2, is offered via 4 programmable parameters:

- 1173 • **match window:** the matching between a trigger and a hit is done within a programmable time  
 1174 window. This is set via the method

1175   `void v1190a::SetTrigWindowWidth(Uint windowHeight, int ntdcs)`

- 1176 • **window offset:** temporal distance between the trigger tag and the start of the trigger matching  
 1177 window. This is set via the method

1178   `void v1190a::SetTrigWindowWidth(Uint windowHeight, int ntdcs)`

- 1179 • **extra search margin:** an extended time window is used to ensure that all matching hits are  
 1180 found. This is set via the method

1181   `void v1190a::SetTrigSearchMargin(Uint searchMargin, int ntdcs)`

- 1182 • **reject margin:** older hits are automatically rejected to prevent buffer overflows and to speed  
 1183 up the search time. This is set via the method

1184   `void v1190a::SetTrigRejectionMargin(Uint rejectMargin, int ntdcs)`

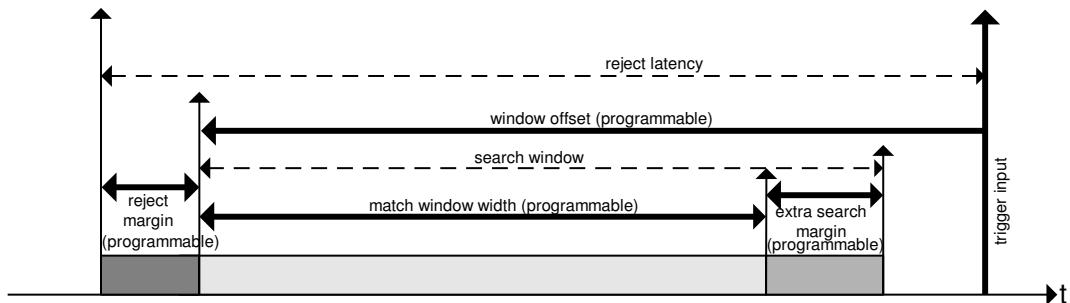


Figure A.2: Module V1190A Trigger Matching Mode timing diagram [58].

1185 Each of these 4 parameters are given in number of clocks, 1 clock being 25 ns long. It is easy to  
 1186 understand at this level that there are 3 possible functioning settings:

- 1187 • **1:** the match window is entirely contained after the trigger signal,

- 1188 • **2:** the match window overlaps the trigger signal, or

- 1189 • **3:** the match window is entirely contained before the trigger signal as displayed on Figure A.2.

In both the first and second cases, the sum of the window width and of the offset can be set to a maximum of 40 clocks, which corresponds to 1  $\mu$ s. Evidently, the offset can be negative, allowing for a longer match window, with the constraint of having the window ending at most 1  $\mu$ s after the trigger signal. In the third case, the maximum negative offset allowed is of 2048 clocks (12 bit) corresponding to 51.2  $\mu$ s, the match window being strictly smaller than the offset. In the case of GIF++, the choice has been made to use this last setting by delaying the trigger signal. During the studies performed in GIF++, both the efficiency of the RPCs, probed using a muon beam, and the noise or gamma background rate are monitored. The extra search and reject margins are left unused.

To probe the efficiency of RPC detectors, the trigger time tag is provided by the coincidence of scintillators when a bunch of muons passes through GIF++ area is used to trigger the data acquisition. For this measurement, it is useful to reduce the match window width only to contain the muon information. Indeed, the delay in between a trigger signal and the detection of the corresponding muon in the RPC being very constant (typically a few tens of ns due to jitter and cable length), the muon signals are very localised in time. Thus, due to a delay of approximately 325 ns in between the muons and the trigger, the settings were chosen to have a window width of 24 clocks (600 ns) centered on the muon peak thanks to a negative offset of 29 clocks (725 ns).

On the other hand, monitoring the rates don't require for the DAQ to look at a specific time window. It is important to integrate enough time to have a robust measurement of the rate as the number of hits per time unit. The triggering signal is provided by a pulse generator at a frequency of 300 Hz to ensure that the data taking occurs in a random way, with respect to beam physics, to probe only the irradiation spectrum on the detectors. The match window is set to 400 clocks (10  $\mu$ s) and the negative offset to 401 clocks as it needs to exceed the value of the match window.

```

1212
1213     class v1190a
1214     {
1215         private :
1216             long           Handle;
1217             vector<Data32>   Address;
1218             CVDataWidth    DataWidth;
1219             CVAddressModifier AddressModifier;
1220
1221         public:
1222
1223             v1190a(long handle, IniFile *inifile, int ntdcs);
1224             ~v1190a();
1225             Data16 write_op_reg(Data32 address, int code, string error);
1226             Data16 read_op_reg(Data32 address, string error);
1227             void Reset(int ntdcs);
1228             void Clear(int ntdcs);
1229             void TestWR(Data16 value,int ntdcs);
1230             void CheckTDCStatus(int ntdcs);
1231             void CheckCommunication(int ntdcs);
1232             void SetTDCTestMode(Data16 mode,int ntdcs);
1233             void SetTrigMatching(int ntdcs);
1234             void SetTrigTimeSubtraction(Data16 mode,int ntdcs);
1235             void SetTrigWindowWidth(Uint windowHeight,int ntdcs);
1236             void SetTrigWindowOffset(Uint windowOffset,int ntdcs);
1237             void SetTrigSearchMargin(Uint searchMargin,int ntdcs);
1238             void SetTrigRejectionMargin(Uint rejectMargin,int ntdcs);
1239             void GetTrigConfiguration(int ntdcs);
1240             void SetTrigConfiguration(IniFile *inifile,int ntdcs);
1241             void SetTDCDetectionMode(Data16 mode,int ntdcs);
1242             void SetTDCResolution(Data16 lsb,int ntdcs);
1243             void SetTDCDeadTime(Data16 time,int ntdcs);
1244             void SetTDCHeadTrailer(Data16 mode,int ntdcs);
1245             void SetTDCEventSize(Data16 size,int ntdcs);
1246             void SwitchChannels(IniFile *inifile,int ntdcs);
1247             void SetIRQ(Data32 level, Data32 count,int ntdcs);
1248             void SetBlockTransferMode(Data16 mode,int ntdcs);
1249             void Set(IniFile *inifile,int ntdcs);
1250             void CheckStatus(CVErrorCodes status) const;
1251             int ReadBlockD32(Uint tdc, const Data16 address,
1252                             Data32 *data, const Uint words, bool ignore_berr);
1253             Uint Read(RAWData *DataList,int ntdcs);
1254     };

```

1214                 *Source Code A.1: Description of C++ object v1190a.*

1215     The v1190a object, defined in the DAQ software as in Source Code A.1, offers the possibility to  
1216     concatenate all TDCs in the readout setup into a single object containing a list of hardware addresses  
1217     (addresses to access the TDCs' buffer through the VME crate) and each constructor and method acts  
1218     on the list of TDCs.  
1219

#### 1220     A.4.2 DataReader

1221     Enabled thanks to v1190a::SetBlockTransferMode(Data16 mode, int ntdcs), the data transfer  
1222     is done via Block Transfer (BLT). Using BLT allows to tranfer a fixed number of events called a  
1223     block. This is used together with an Almost Full Level (AFL) of the TDCs' output buffers, defined

1224 through `v1190a::SetIRQ(Data32 level, Data32 count, int ntdcs)`. This AFL gives the maximum amount of 32735 words (16 bits, corresponding to the depth of a TDC output buffer) that can  
 1225 written in a buffer before an Interrupt Request (IRQ) is generated and seen by the VME Bridge,  
 1226 stopping the data acquisition to transfer the content of each TDC buffers before resuming. For each  
 1227 trigger, 6 words or more are written into the TDC buffer:

- 1229     • a **global header** providing information of the event number since the beginning of the data  
       acquisition,
- 1231     • a **TDC header**,
- 1232     • the **TDC data** (*if any*), 1 for each hit recorded during the event, providing the channel and the  
       time stamp associated to the hit,
- 1234     • a **TDC error** providing error flags,
- 1235     • a **TDC trailer**,
- 1236     • a **global trigger time tag** that provides the absolute trigger time relatively to the last reset,  
       and
- 1238     • a **global trailer** providing the total word count in the event.

1239 As previously described in Section ??, CMS RPC FEEs provide us with 100 ns long LVDS out-  
 1240 put signals that are injected into the TDCs' input. Any avalanche signal that gives a signal above the  
 1241 FEEs threshold is thus recorded by the TDCs as a hit within the match window. Each hit is assigned  
 1242 to a specific TDC channel with a time stamp, with a precision of 100 ps. The reference time,  $t_0 = 0$ ,  
 1243 is provided by the beginning of the match window. Thus for each trigger, coming from a scintillator  
 1244 coïncidence or the pulse generator, a list of hits is stored into the TDCs' buffers and will then be  
 1245 transferred into a ROOT Tree.

1246 When the BLT is used, it is easy to understand that the maximum number of words that have  
 1247 been set as ALF will not be a finite number of events or, at least, the number of events that would  
 1248 be recorded into the TDC buffers will not be a multiple of the block size. In the last BLT cycle to  
 1249 tranfer data, the number of events to transfer will most probably be lower than the block size. In that  
 1250 case, the TDC can add fillers at the end of the block but this option requires to send more data to the  
 1251 computer and is thus a little slower. Another solution is to finish the transfer after the last event by  
 1252 sending a bus error that states that the BLT reached the last event in the pile. This method has been  
 1253 chosen in GIF++.

1255 Due to irradiation, an event in GIF++ can count up to 300 words per TDC. A limit of 4096 words  
 1256 (12 bits) has been set to generate IRQ which represent from 14 to almost 700 events depending on  
 1257 the average of hits collected per event. Then the block size has been set to 100 events with enabled  
 1258 bus errors. When an AFL is reached for one of the TDCs, the VME bridge stops the acquisition by  
 1259 sending a BUSY signal.

1261

1262     The data is then transferred one TDC at a time into a structure called `RAWData` (Source Code A.2).

```
1263
1264     struct RAWData{
1265         vector<int>           *EventList;
1266         vector<int>           *NHitsList;
1267         vector<int>           *QFlagList;
1268         vector<vector<int>> > *ChannelList;
1269         vector<vector<float>> > *TimeStampList;
1270     };
```

1265                                  *Source Code A.2: Description of data holding C++ structure RAWData.*

1266     In order to organize the data transfer and the data storage, an object called `DataReader` was  
1267     created (Source Code A.3). On one hand, it has `v1718` and `v1190a` objects as private members for  
1268     communication purposes, such as VME modules settings via the configuration file `*iniFile` or data  
1269     read-out through `v1190a::Read()` and on the other hand, it contains the structure `RAWData` that allows  
1270     to organise the data in vectors reproducing the tree structure of a ROOT file.

```
1271
1272     class DataReader
1273     {
1274         private:
1275             bool          StopFlag;
1276             IniFile *iniFile;
1277             Data32 MaxTriggers;
1278             v1718 *VME;
1279             int          nTDCs;
1280             v1190a *TDCs;
1281             RAWData TDCData;
1282
1283         public:
1284             DataReader();
1285             virtual ~DataReader();
1286             void SetIniFile(string inifilename);
1287             void SetMaxTriggers();
1288             Data32 GetMaxTriggers();
1289             void SetVME();
1290             void SetTDC();
1291             int GetQFlag(Uint it);
1292             void Init(string inifilename);
1293             void FlushBuffer();
1294             void Update();
1295             string GetFileName();
1296             void WriteRunRegistry(string filename);
1297             void Run();
1298     };
```

1273                                  *Source Code A.3: Description of C++ object DataReader.*

1274     Each event is transferred from `TDCData` and saved into branches of a ROOT `TTree` as 3 integers  
1275     that represent the event ID (`EventCount`), the number of hits read from the TDCs (`nHits`), and the  
1276     quality flag that provides information for any problem in the data transfer (`qflag`), and 2 lists of  
1277     `nHits` elements containing the fired TDC channels (`TDCCh`) and their respective time stamps (`TDCTS`),  
1278     as presented in Source Code A.4. The ROOT file is named using information contained into  
1279     the configuration file, presented in section A.5.2. The needed information is extracted using method  
1280     `DataReader::GetFileName()` and allow to build the output filename format `ScanXXXXXX_HVX_DAQ.root`

1281 where ScanXXXXXX is a 6 digit number representing the scan number into GIFT++ database and HVX  
 1282 the HV step within the scan that can be more than a single digit. An example of ROOT data file is  
 1283 provided with Figure A.3.

```
1284
RAWData TDCData;
TFile *outputFile = new TFile(outputFileName.c_str(),"recreate");
TTree *RAWDataTree = new TTree("RAWData","RAWData");

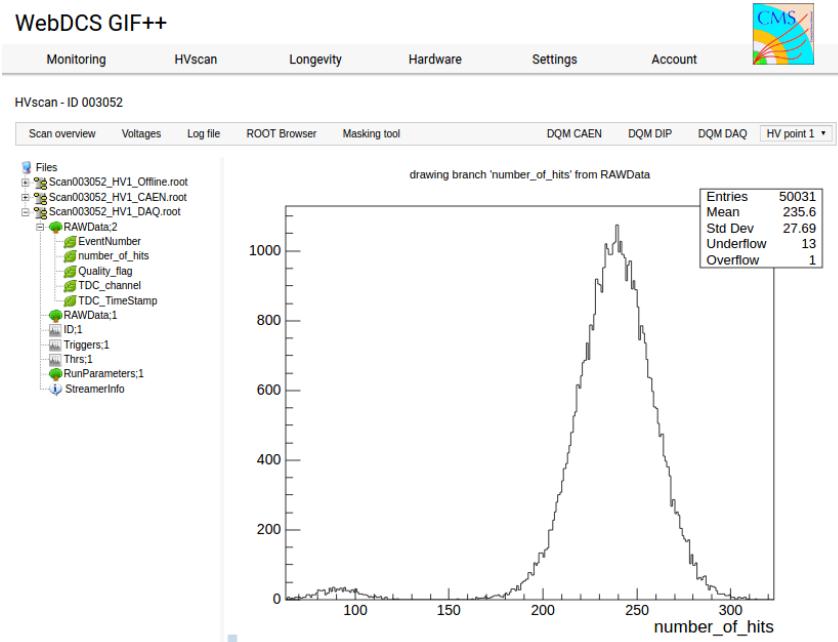
int EventCount = -9;
int nHits = -8;
int qflag = -7;
vector<int> TDCCh;
vector<float> TDCTS;

RAWDataTree->Branch("EventNumber",&EventCount, "EventNumber/I");
RAWDataTree->Branch("number_of_hits",&nHits,"number_of_hits/I");
RAWDataTree->Branch("Quality_flag",&qflag,"Quality_flag/I");
RAWDataTree->Branch("TDC_channel",&TDCCh);
RAWDataTree->Branch("TDC_TimeStamp",&TDCTS);

1285
//...
//Here read the TDC data using v1190a::Read() and place it into
//TDCData for as long as you didn't collect the requested amount
//of data.
//...

for(Uint i=0; i<TDCData.EventList->size(); i++){
    EventCount = TDCData.EventList->at(i);
    nHits = TDCData.NHitsList->at(i);
    qflag = TDCData.QFlagList->at(i);
    TDCCh = TDCData.ChannelList->at(i);
    TDCTS = TDCData.TimeStampList->at(i);
    RAWDataTree->Fill();
}
```

1286 *Source Code A.4: Highlight of the data transfer and organisation within DataReader::Run() after the data has been collected into TDCData.*



*Figure A.3: Structure of the ROOT output file generated by the DAQ. The 5 branches (EventNumber, number\_of\_hits, Quality\_flag, TDC\_channel and TDC\_TimeStamp) are visible on the left panel of the ROOT browser. On the right panel is visible the histogram corresponding to the variable nHits. In this specific example, there were approximately 50k events recorded to measure the gamma irradiation rate on the detectors. Each event is stored as a single entry in the TTree.*

#### 1287    A.4.3 Data quality flag

1288    Among the parameters that are recorded for each event, the quality flag, defined in Source Code A.5,  
 1289    is determined on the fly by checking the data recorded by every single TDC. From method `v1190a::Read()`,  
 1290    it can be understood that the content of each TDC buffer is readout one TDC at a time. Entries are  
 1291    created in the data list for the first TDC and then, when the second buffer is readout, events corre-  
 1292    sponding to entries that have already been created to store data for the previous TDC are added to  
 1293    the existing list element. On the contrary, when an event entry has not been yet created in the data  
 1294    list, a new entry is created.

```
1295
  typedef enum _QualityFlag {
    GOOD      = 1,
    CORRUPTED = 0
} QualityFlag;
```

1297    *Source Code A.5: Definition of the quality flag `enum`.*

1298    It is possible that each TDC buffer contains a different number of events. In cases where the first  
 1299    element in the buffer list is an event for corresponds to a new entry, the difference in between the  
 1300    entry from the buffer and the last entry in the data list is recorded and checked. If it is greater than 1,  
 1301    what should never be the case, the quality flag is set to CORRUPTED for this TDC and an empty entry  
 1302    is created in the place of the missing ones. Missing entries are believe to be the result of a bad hold

1303 on the TDC buffers at the moment of the readout. Indeed, the software hold is effective only on 1  
 1304 TDC at a time and no solution as been found yet to completely block the writting in the buffers when  
 1305 an IRQ is received.

1306 At the end of each BLT cycle, the ID of the last entry stored for each TDC buffer is not recorded.  
 1307 When starting the next cycle, if the first entry in the pile corresponds to an event already existing  
 1308 in the list, the readout will start from this list element and will not be able to check the difference  
 1309 in between this entry's ID and the one of the last entry that was recorded for this TDC buffer in  
 1310 the previous cycle. In the case events were missing, the flag stays at its initial value of 0, which is  
 1311 similar to CORRUPTED and it is assumed that then this TDC will not contribute to `number_of_hits`,  
 1312 `TDC_channel` or `TDC_TimeStamp`.

1313 Finally, since there will be 1 `RAWData` entry per TDC for each event (meaning `nTDCs` entries,  
 1314 referring to `DataReader` private attribute), the individual flags of each TDC will be added together.  
 1315 The final format is an integer composed `nTDCs` digits where each digit is the flag of a specific TDC.  
 1316 This is constructed using powers of 10 like follows:

```
1317 TDC 0: QFlag = 100 × _QualityFlag
1318 TDC 1: QFlag = 101 × _QualityFlag
1319 ...
1320 TDC N: QFlag = 10N × _QualityFlag
```

1321 and the final flag to be with N digits:

```
1322 QFlag = n....3210
```

1323 each digit being 1 or 0. Below is given an example with a 4 TDCs setup.

1324 If all TDCs were good : `QFlag = 1111`,

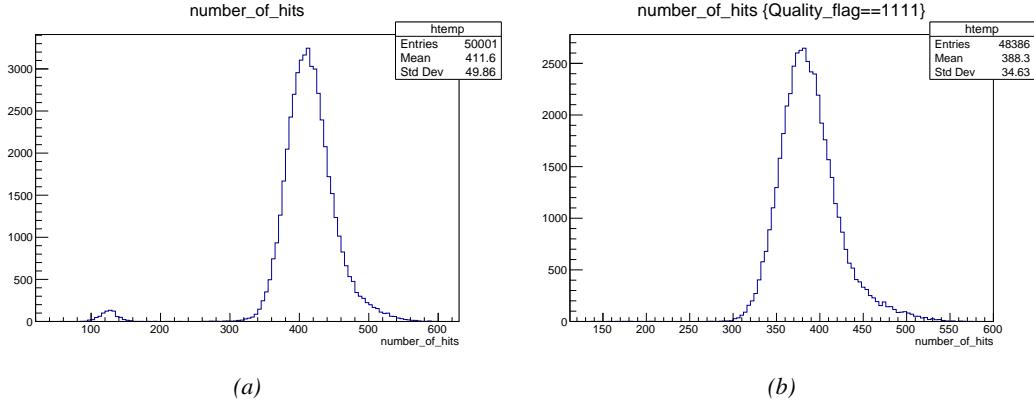
1325 but if TDC 2 was corrupted : `QFlag = 1011`.

1326 When data taking is over and the data contained in the dynamical `RAWData` structure is transferred  
 1327 to the ROOT file, all the 0s are changed into 2s by calling the method `DataReader::GetQFlag()`.  
 1328 This will help translating the flag without knowing the number of TDCs beforehand. Indeed, a flag  
 1329 111 could be due to a 3 TDC setup with 3 good individual TDC flags or to a more than 3 TDC setup  
 1330 with TDCs those ID is greater than 2 being CORRUPTED, thus giving a 0.

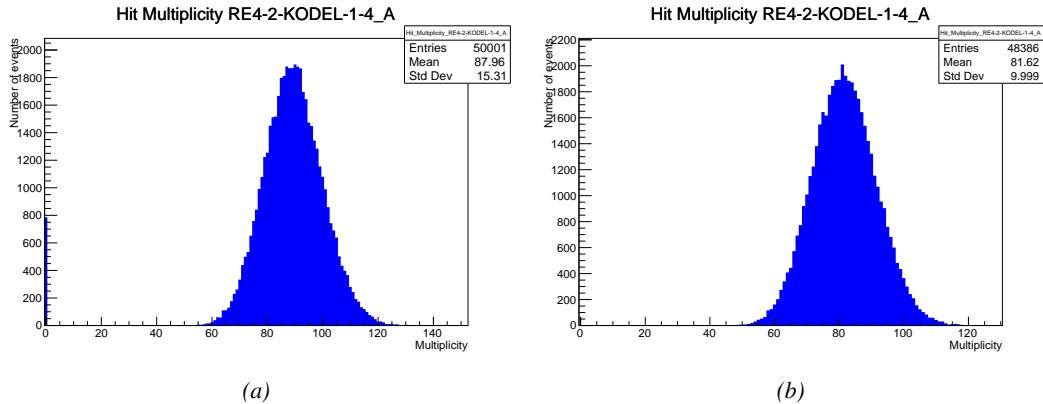
1331 The quality flag has been introduced quite late, in October 2017 only, to the list of GIFT++ DAQ  
 1332 parameters to be recorded into the output ROOT file. Before this addition, the missing data, corrupting  
 1333 the quality for the offline analysis, was contributing to artificially fill data with lower multiplicity.  
 1334 Looking at `TBranch number_of_hits` provides an information about the data of the full GIFT++  
 1335 setup. When a TDC is not able to transfer data for a specific event, the effect is a reduction of the  
 1336 total number of hits recorded in the full setup, this is what can be seen from Figure A.4. After offline  
 1337 reconstruction detector by detector, the effect of missing events can be seen in the artificially filled  
 1338 bin at multiplicity 0 shown in Figure A.5. Nonetheless, for data with high irradiation levels, as it is  
 1339 the case for Figure A.5a, discarding the fake multiplicity 0 data can be done easily during the offline  
 1340 analysis. At lower radiation, the missing events contribution becomes more problematic as the multi-  
 1341 tiplicity distribution overlaps the multiplicity 0 and that in the same time the proportion of missing

events decreases. Attempts to fit the distribution with a Poisson or skew distribution function were not conclusive and this very problem has been at the origin of the quality flag that allows to give a non ambiguous information about each event quality.

1345



*Figure A.4: The effect of the quality flag is explained by presenting the content of TBranch `number_of_hits` of a data file without `Quality_flag` in Figure A.4a and the content of the same TBranch for data corresponding to a `Quality_flag` where all TDCs were labelled as `GOOD` in Figure A.4b taken with similar conditions. It can be noted that the number of entries in Figure A.4b is slightly lower than in Figure A.4a due to the excluded events.*



*Figure A.5: Using the same data as previously showed in Figure A.4, the effect of the quality flag is explained by presenting the reconstructed hit multiplicity of a data file without `Quality_flag` in Figure A.5a and the reconstructed content of the same RPC partition for data corresponding to a `Quality_flag` where all TDCs were labelled as `GOOD` in Figure A.5b taken with similar conditions. The artificial high content of bin 0 is completely suppressed.*

## A.5 Communications

1346 To ensure data readout and dialog in between the machine and the TDCs or in between the webDCS and the DAQ, different communication solutions were used. First of all, it is important to have a

1349 module to allow the communication in between the TDCs and the computer from which the DAQ  
 1350 operates. When this communication is effective, shifters using the webDCS to control data taking  
 1351 can thus send instructions to the DAQ.

1352

### 1353 A.5.1 V1718 USB Bridge

1354 In the previous section, the data transfer has been discussed. The importance of the `v1718` object  
 1355 (Source Code A.6), used as private member of `DataReader`, was not explicated. VME master  
 1356 modules are used for communication purposes as they host the USB port that connects the pow-  
 1357 ered crate buffer to the computer where the DAQ is installed. From the source code point of view,  
 1358 this object is used to control the communication status, by reading the returned error codes with  
 1359 `v1718::CheckStatus()`, or to check for IRQs coming from the TDCs through `v1718::CheckIRQ()`.  
 1360 Finally, to ensure that triggers are blocked at the hardware level, a NIM pulse is sent out of one of the  
 1361 5 programmable outputs (`v1718::SendBUSY()`) to the VETO of the coincidence module where the  
 1362 trigger signals originate from. As long as this signal is ON, no trigger can reach the TDCs anymore.  
 1363

```
1364
1365   class v1718{
1366     private:
1367       int           Handle;
1368       Data32        Data;          // Data
1369       CVIRQLevels  Level;        // Interrupt level
1370       CVAddressModifier AM;        // Addressing Mode
1371       CVDataWidth   DataSize;      // Data Format
1372       Data32        BaseAddress;  // Base Address
1373
1374     public:
1375       v1718(IniFile *inifile);
1376       ~v1718();
1377       long          GetHandle(void) const;
1378       int           SetData(Data16 data);
1379       Data16        GetData(void);
1380       int           SetLevel(CVIRQLevels level);
1381       CVIRQLevels  GetLevel(void);
1382       int           SetAM(CVAddressModifier am);
1383       CVAddressModifier GetAM(void);
1384       int           SetDatasize(CVDataWidth datasize);
1385       CVDataWidth   GetDataSize(void);
1386       int           SetBaseAddress(Data16 baseaddress);
1387       Data16        GetBaseAddress(void);
1388       void          CheckStatus(CVErrorCodes status) const;
1389       bool          CheckIRQ();
1390       void          SetPulsers();
1391       void          SendBUSY(BusyLevel level);
1392   };
1393
```

1365 *Source Code A.6: Description of C++ object v1718.*

### 1366 A.5.2 Configuration file

1367 The DAQ software takes as input a configuration file written using INI standard [61]. This file is  
 1368 partly filled with the information provided by the shifters when starting data acquisition using the  
 1369 webDCS, as shown by Figure A.6. This information is written in section [`General`] and will later

1370 be stored in the ROOT file that contains the DAQ data as can be seen from Figure A.3. Indeed,  
 1371 another `TTree` called `RunParameters` as well as the 2 histograms `ID`, containing the scan number,  
 1372 start and stop time stamps, and `Triggers`, containing the number of triggers requested by the shifter,  
 1373 are available in the data files. Moreover, `ScanID` and `HV` are then used to construct the file name  
 1374 thanks to the method `DataReader::GetFileName()`.

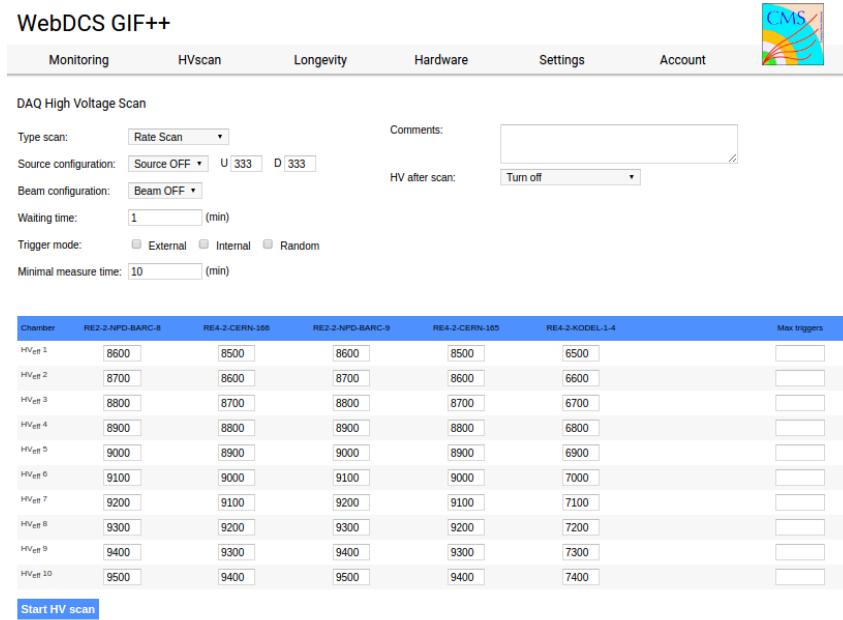


Figure A.6: WebDCS DAQ scan page. On this page, shifters need to choose the type of scan (Rate, Efficiency or Noise Reference scan), the gamma source configuration at the moment of data taking, the beam configuration, and the trigger mode. These information will be stored in the DAQ ROOT output. Are also given the minimal measurement time and waiting time after ramping up of the detectors is over before starting the data acquisition. Then, the list of HV points to scan and the number of triggers for each run of the scan are given in the table underneath.

1375 The rest of the information is written beforehand in the configuration file template, as explicated  
 1376 in Source Code A.7, and contains the hardware addresses to the different VME modules in the  
 1377 setup as well as settings for the TDCs. As the TDC settings available in the configuration file are not  
 1378 supposed to be modified, an improvement would be to remove them from the configuration file and  
 1379 to hardcode them inside of the DAQ code itself or to place them into a different INI file that would  
 1380 host only the TDC settings to lower the probability for a bad manipulation of the configuration file  
 1381 that can be modified from one of webDCS' menus.

1382

```
[General]
TdcS=4
ScanID=$scanid
HV=$HV
RunType=$runtype
MaxTriggers=$maxtriggers
Beam=$beam
[VMEInterface]
Type=V1718
BaseAddress=0xFF0000
Name=VmeInterface
[TDC0]
Type=V1190A
BaseAddress=0x00000000
Name=Tdc0
StatusA00-15=1
StatusA16-31=1
StatusB00-15=1
StatusB16-31=1
StatusC00-15=1
StatusC16-31=1
StatusD00-15=1
StatusD16-31=1
[TDC1]
Type=V1190A
BaseAddress=0x11110000
Name=Tdc1
StatusA00-15=1
StatusA16-31=1
StatusB00-15=1
StatusB16-31=1
StatusC00-15=1
StatusC16-31=1
StatusD00-15=1
StatusD16-31=1
[TDC2]
Type=V1190A
BaseAddress=0x22220000
Name=Tdc2
StatusA00-15=1
StatusA16-31=1
StatusB00-15=1
StatusB16-31=1
StatusC00-15=1
StatusC16-31=1
StatusD00-15=1
StatusD16-31=1
[TDC3]
Type=V1190A
BaseAddress=0x44440000
Name=Tdc3
StatusA00-15=1
StatusA16-31=1
StatusB00-15=1
StatusB16-31=1
StatusC00-15=1
StatusC16-31=1
StatusD00-15=1
StatusD16-31=1
[TDCSettings]
TriggerExtraSearchMargin=0
TriggerRejectMargin=0
TriggerTimeSubtraction=0b1
TdcDetectionMode=0b01
TdcResolution=0b10
TdcDeadTime=0b00
TdcHeadTrailer=0b1
TdcEventSize=0b1001
TdcTestMode=0b0
BLTMode=1
```

1383

*Source Code A.7: INI configuration file template for 4 TDCs. In section [General], the number of TDCs is explicitated and information about the ongoing run is given. Then, there are sections for each and every VME modules. There buffer addresses are given and for the TDCs, the list of channels to enable is given. Finally, in section [TDCSettings], a part of the TDC settings are given.*

1385     In order to retrieve the information of the configuration file, the object `IniFile` has been developed  
 1386     to provide an INI parser, presented in Source Code A.8. It contains private methods returning a  
 1387     boolean to check the type of line written in the file, whether a comment, a group header or a key line  
 1388     (`IniFile::CheckIfComment()`, `IniFile::CheckIfGroup()` and `IniFile::CheckIfToken()`). The  
 1389     key may sometimes be referred to as *token* in the source code. Moreover, the private element  
 1390     `FileData` is a map of `const string` to `string` that allows to store the data contained inside the  
 1391     configuration file via the public method `IniFile::GetFileData()` following the formatting (see  
 1392     method `IniFile::Read()`):

```
1393
 1394     string group, token, value;
 1395     // Get the field values for the 3 strings.
 1396     // Then concatenate group and token together as a single string
 1397     // with a dot separation.
 1398     token = group + "." + token;
 1399     FileData[token] = value;
```

1395     More methods have been written to translate the different keys into the right variable format  
 1396     when used by the DAQ. For example, to get a `float` value out of the configuration file data, knowing  
 1397     the group and the key needed, the method `IniFile::floatType()` can be used. It takes 3 arguments  
 1398     being the group name and key name (both `string`), and a default `float` value used as exception in  
 1399     the case the expected combination of group and key cannot be found in the configuration file. This  
 1400     default value is then used and the DAQ continues on working after sending an alert in the log file for  
 1401     further debugging.

```

1402 typedef map< const string, string > IniFileData;
1403
class IniFile{
    private:
        bool          CheckIfComment (string line);
        bool          CheckIfGroup(string line, string& group);
        bool          CheckIfToken(string line, string& key, string& value);
        string         FileName;
        IniFileData   FileData;
        int           Error;

    public:
        IniFile();
        IniFile(string filename);
        virtual      ~IniFile();

        // Basic file operations
        void          SetFileName(string filename);
        int           Read();
        int           Write();
        IniFileData GetFileData();

        // Data readout methods
        Data32         addressType (string groupname, string keyname, Data32
→     defaultvalue);
        long          intType     (string groupname, string keyname, long
→     defaultvalue);
        long long    longType    (string groupname, string keyname, long long
→     defaultvalue );
        string         stringType  (string groupname, string keyname, string
→     defaultvalue );
        float         floatType   (string groupname, string keyname, float
→     defaultvalue );

        // Error methods
        string         GetErrorMsg();
    };

```

1404       *Source Code A.8: Description of C++ object `IniFile` used as a parser for INI file format.*

### 1405     A.5.3 WebDCS/DAQ intercommunication

1406     When shifters send instructions to the DAQ via the configuration file, it is the webDCS itself that  
1407     gives the start command to the DAQ and then the 2 softwares use inter-process communication  
1408     through file to synchronise themselves. This communication file is represented by the variable **const**  
1409     string \_\_runstatuspath.

1410     On one side, the webDCS sends commands or status that are readout by the DAQ:

- 1411       • INIT, status sent when launching a scan and read via function `CtrlRunStatus(...)`,
- 1412       • START, command to start data taking and read via function `CheckSTART()`,
- 1413       • STOP, command to stop data taking at the end of the scan and read via function `CheckSTOP()`,  
1414        and
- 1415       • KILL, command to kill data taking sent by user and read via function `CheckKILL()`

1416 and on the other, the DAQ sends status that are controled by the webDCS:

- 1417     ● `DAQ_RDY`, sent with `SendDAQReady()` to signify that the DAQ is ready to receive commands  
1418       from the webDCS,
- 1419     ● `RUNNING`, sent with `SendDAQRunning()` to signify that the DAQ is taking data,
- 1420     ● `DAQ_ERR`, sent with `SendDAQError()` to signify that the DAQ didn't receive the expected com-  
1421       mand from the webDCS or that the launch command didn't have the right number of argu-  
1422       ments,
- 1423     ● `RD_ERR`, sent when the DAQ wasn't able to read the communication file, and
- 1424     ● `WR_ERR`, sent when the DAQ wasn't able to write into the communication file.

#### 1425     **A.5.4 Example of inter-process communication cycle**

1426 Under normal conditions, the webDCS and the DAQ processes exchange commands and status via  
1427 the file hosted at the address `__runstatuspath`, as explained in subsection A.5.3. An example of  
1428 cycle is given in Table A.1. In this example, the steps 3 to 5 are repeated as long as the webDCS tells  
1429 the DAQ to take data. A data taking cycle is the equivalent as what is called a *Scan* in GIFT++ jargon,  
1430 referring to a set a runs with several HV steps. Each repetition of steps 3 to 5 is then equivalent to a  
1431 single *Run*.

1432

1433 At any moment during the data taking, for any reason, the shifter can decide that the data taking  
1434 needs to be stopped before it reached the end of the scheduled cycle. Thus at any moment on the  
1435 cycle, the content of the inter-process communication file will be changed to `KILL` and the DAQ will  
1436 shut down right away. The DAQ checks for `KILL` signals every 5s after the TDCs configuration is  
1437 over. So far, the function `CheckKILL()` has been used only inside of the data taking loop of method  
1438 `DataReader::Run()` and thus, if the shifter decides to KILL the data taking during the TDC con-  
1439 figuration phase or the HV ramping in between 2 HV steps, the DAQ will not be stopped smoothly  
1440 and a *force kill* command will be sent to stop the DAQ process that is still awake on the computer.  
1441 Improvements can be brought on this part of the software to make sure that the DAQ can safely  
1442 shutdown at any moment.

1443

#### 1444     **A.6 Software export**

1445 In section A.2 was discussed the fact that the DAQ as written in its last version is not a standalone  
1446 software. It is possible to make it a standalone program that could be adapted to any VME setup  
1447 using V1190A and V1718 modules by creating a GUI for the software or by printing the log mes-  
1448 sages that are normally printed in the webDCS through the log file, directly into the terminal. This  
1449 method was used by the DAQ up to version 3.0 moment where the webDCS was completed. Also, it  
1450 is possible to check branches of DAQ v2.X to have example of communication through a terminal.

1451

1452     DAQ v2.X is nonetheless limited in it's possibilities and requires a lot of offline manual interven-  
1453       tions from the users. Indeed, there is no communication of the software with the detectors' power  
1454       supply system that would allow for a user a predefine a list of voltages to operate the detectors at

| step | actions of webDCS  | status of DAQ  | __runstatuspath |
|------|--|--|-----------------|
| 1    | launch DAQ<br>ramp voltages<br>ramping over<br>wait for currents stabilization | readout of IniFile<br>configuration of TDCs                                | INIT            |
| 2    |  | configuration done<br>send DAQ ready<br>wait for START signal              | DAQ_RDY         |
| 3    | waiting time over<br>send START  |  | START           |
| 4    | wait for run to end<br>monitor DAQ run status                                  | data taking ongoing<br>check for KILL signal                               | RUNNING         |
| 5    |  | run over<br>send DAQ_RDY<br>wait for next DCS signal                       | DAQ_RDY         |
| 6    | ramp voltages<br>ramping over<br>wait for currents stabilization               |  | DAQ_RDY         |
| 3    | waiting time over<br>send START  |  | START           |
| 4    | wait for run to end<br>monitor DAQ run status                                  | update IniFile information<br>data taking ongoing<br>check for KILL signal | RUNNING         |
| 5    |  | run over<br>send DAQ_RDY<br>wait for next DCS signal                       | DAQ_RDY         |
| 7    | send command STOP  | DAQ shuts down   | STOP            |

Table A.1: Inter-process communication cycles in between the webDCS and the DAQ through file string signals.

1455 and loop over to take data without any further manual intervention. In v2.X, the data is taken for a  
1456 single detector setting and at the end of each run, the softwares asks the user if he intends on taking  
1457 more runs. If so, the software invites the user to set the operating voltages accordingly to what is  
1458 necessary and to manual update the configuration file in consequence. This working mode can be a  
1459 very first approach before an evolution and has been successfully used by colleagues from different  
1460 collaborations.

1461

1462 For a more robust operation, it is recommended to develop a GUI or a web application to inter-  
1463 face the DAQ. Moreover, to limit the amount of manual interventions, and thus the probability to  
1464 make mistakes, it is also recommended to add an extra feature into the DAQ by installing the HV  
1465 Wrapper library provided by CAEN of which an example of use in a similar DAQ software devel-  
1466 opped by a master student of UGent, and called TinyDAQ, is provided on UGent's github. Then, this  
1467 HV Wrapper will help you communicating with and give instructions to a CAEN HV powered crate  
1468 and can be added into the DAQ at the same level where the communication with the user was made  
1469 in DAQ v2.X. In case you are using another kind of power system for your detectors, it is stringly  
1470 adviced to use HV modules or crates that can be remotely controled via a using C++ libraries.

1471

# B

1472

1473

## Details on the offline analysis package

1474 The data collected in GIF++ thanks to the DAQ described in Appendix A is difficult to interpret by  
1475 a human user that doesn't have a clear idea of the raw data architecture of the ROOT data files. In  
1476 order to render the data human readable, a C++ offline analysis tool was designed to provide users  
1477 with detector by detector histograms that give a clear overview of the parameters monitored during  
1478 the data acquisition [62]. In this appendix, details about this software in the context of GIF++, as of  
1479 how the software was written and how it functions will be given.

### 1480 B.1 GIF++ Offline Analysis file tree

1481 GIF++ Offline Analysis source code is fully available on github at [https://github.com/aafagot/GIF\\_OfflineAnalysis](https://github.com/aafagot/GIF_OfflineAnalysis). The software requires ROOT as non-optionnal dependency  
1482 as it takes ROOT files in input and write an output ROOT file containing histograms. To compile the  
1483 GIF++ Offline Analysis project is compiled with cmake. To compile, first a build/ directory must  
1484 be created to compile from there:

```
1486 mkdir build  
1487 cd build  
1488 cmake ..  
1489 make  
1490 make install
```

1488 To clean the directory and create a new build directory, the bash script cleandir.sh can be used:

```
1489  
1490 ./cleandir.sh
```

1491 The source code tree is provided below along with comments to give an overview of the files' con-  
1492 tent. The different objects created for this project (`Infrastructure`, `Trolley`, `RPC`, `Mapping`, `RPCHit`,  
1493 `RPCCluster` and `Inifile`) will be described in details in the following sections.

1494

```

GIF_OfflineAnalysis
├── bin
│   └── offlineanalysis ..... EXECUTABLE
├── build..... CMAKE COMPILATION DIRECTORY
└── ...
    ├── include..... LIST OF C++ HEADER FILES
    │   ├── Cluster.h..... DECLARATION OF OBJECT RPCCLUSTER
    │   ├── Current.h..... DECLARATION OF GETCURRENT ANALYSIS MACRO
    │   ├── GIFTrolley.h..... DECLARATION OF OBJECT TROLLEY
    │   ├── Infrastructure.h..... DECLARATION OF OBJECT INFRASTRUCTURE
    │   ├── IniFile.h..... DECLARATION OF OBJECT INI FILE FORINI PARSER
    │   ├── Mapping.h..... DECLARATION OF OBJECT MAPPING
    │   ├── MsgSvc.h..... DECLARATION OF OFFLINE LOG MESSAGES
    │   ├── OfflineAnalysis.h..... DECLARATION OF DATA ANALYSIS MACRO
    │   ├── RPCTracker.h..... DECLARATION OF OBJECT RPC
    │   ├── RPCHit.h..... DECLARATION OF OBJECT RPCHIT
    │   ├── types.h..... DEFINITION OF USEFUL VARIABLE TYPES
    │   └── utils.h..... DECLARATION OF USEFUL FUNCTIONS
    ├── obj..... BINARY FILES CREATED BY COMPILER
    └── ...
        ├── src..... LIST OF C++ SOURCE FILES
        │   ├── Cluster.cc..... DEFINITION OF OBJECT RPCCLUSTER
        │   ├── Current.cc .....

```

1495

## B.2 Usage of the Offline Analysis

1496

In order to use the Offline Analysis tool, it is necessary to know the Scan number and the HV Step of the run that needs to be analysed. This information needs to be written in the following format:

1498

1499

```
Scan00XXXX_HVY
```

1500

1501

where XXXX is the scan ID and Y is the high voltage step (in case of a high voltage scan, data will be taken for several HV steps). This format corresponds to the base name of data files in the database

1502 of the GIF++ webDCS. Usually, the offline analysis tool is automatically called by the webDCS at  
 1503 the end of data taking or by a user from the webDCS panel if an update of the tool was brought.  
 1504 Nonetheless, an expert can locally launch the analysis for tests on the GIF++ computer, or a user can  
 1505 get the code on its local machine from github and download data from the webDCS for its own anal-  
 1506 ysis. To launch the code, the following command can be used from the `GIF_OfflineAnalysis` folder:

1507  
 1508     `bin/offlineanalysis /path/to/Scan00XXXX_HVY`

1509 where, `/path/to/Scan00XXXX_HVY` refers to the local data files. Then, the offline tool will by itself  
 1510 take care of finding all available ROOT data files present in the folder, as listed below:

- 1511
  - 1512         ● `Scan00XXXX_HVY_DAQ.root` containing the TDC data as described in Appendix ?? (events, hit  
     and timestamp lists), and
  - 1513         ● `Scan00XXXX_HVY_CAEN.root` containing the CAEN mainframe data recorded by the monitor-  
     ing tool webDCS during data taking (HVs and currents of every HV channels). This file is  
     created independently of the DAQ.

## 1516     **B.2.1 Output of the offline tool**

### 1517     **B.2.1.1 ROOT file**

1518 The analysis gives in output ROOT datafiles that are saved into the data folder and called using the  
 1519 naming convention `Scan00XXXX_HVY_Offline.root`. Inside those, a list of `TH1` histograms can be  
 1520 found. Its size will vary as a function of the number of detectors in the setup as each set of histograms  
 1521 is produced detector by detector. For each partition of each chamber, can be found:

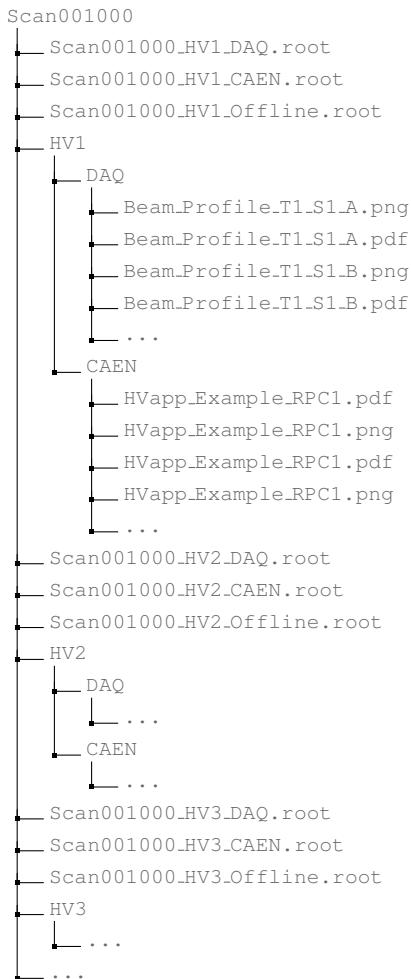
- 1522
  - 1523         ● `Time_Profile_Tt_Sc_p` shows the time profile of all recorded events (number of events per  
     time bin),
  - 1524         ● `Hit_Profile_Tt_Sc_p` shows the hit profile of all recorded events (number of events per chan-  
     nel),
  - 1526         ● `Hit_Multiplicity_Tt_Sc_p` shows the hit multiplicity (number of hits per event) of all recorded  
     events (number of occurrences per multiplicity bin),
  - 1528         ● `Strip_Mean_Noise_Tt_Sc_p` shows noise/gamma rate per unit area for each strip in a se-  
     lected time range. After filters are applied on `Time_Profile_Tt_Sc_p`, the filtered version  
     of `Hit_Profile_Tt_Sc_p` is normalised to the total integrated time and active detection area  
     of a single channel,
  - 1532         ● `Strip_Activity_Tt_Sc_p` shows noise/gamma activity for each strip (normalised version of  
     previous histogram - strip activity = strip rate / average partition rate),
  - 1534         ● `Strip_Homogeneity_Tt_Sc_p` shows the *homogeneity* of a given partition ( $\text{homogeneity} = \exp(-\text{strip rates standard deviation(strip rates in partition/average partition rate)})$ ),
  - 1536         ● `mask_Strip_Mean_Noise_Tt_Sc_p` shows noise/gamma rate per unit area for each masked  
     strip in a selected time range. Offline, the user can control the noise/gamma rate and decide to  
     mask the strips that are judged to be noisy or dead. This is done via the *Masking Tool* provided  
     by the webDCS,

- 1540     ● `mask_Strip_Activity_Tt_Sc_p` shows noise/gamma activity per unit area for each masked  
1541       strip with respect to the average rate of active strips,
- 1542     ● `NoiseCSize_H_Tt_Sc_p` shows noise/gamma cluster size, a cluster being constructed out of  
1543       adjacent strips giving a signal at the *same time* (hits within a time window of 25 ns),
- 1544     ● `NoiseCMult_H_Tt_Sc_p` shows noise/gamma cluster multiplicity (number of reconstructed  
1545       clusters per event),
- 1546     ● `Chip_Mean_Noise_Tt_Sc_p` shows the same information than `Strip_Mean_Noise_Tt_Scp` us-  
1547       ing a different binning (1 chip corresponds to 8 strips),
- 1548     ● `Chip_Activity_Tt_Sc_p` shows the same information than `Strip_Activity_Tt_Scp` using  
1549       chip binning,
- 1550     ● `Chip_Homogeneity_Tt_Sc_p` shows the homogeneity of a given partition using chip binning,
- 1551     ● `Beam_Profile_Tt_Sc_p` shows the estimated beam profile when taking efficiency scan. This  
1552       is obtained by filtering `Time_Profile_Tt_Sc_p` to only consider the muon peak where the  
1553       noise/gamma background has been subtracted. The resulting hit profile corresponds to the  
1554       beam profile on the detector channels,
- 1555     ● `L0_Efficiency_Tt_Sc_p` shows the level 0 efficiency that was estimated **without** muon track-  
1556       ing,
- 1557     ● `MuonCSize_H_Tt_Sc_p` shows the level 0 muon cluster size that was estimated **without** muon  
1558       tracking, and
- 1559     ● `MuonCMult_H_Tt_Sc_p` shows the level 0 muon cluster multiplicity that was estimated **without**  
1560       muon tracking.

1561     In the histogram labels,  $t$  stands for the trolley number (1 or 3),  $c$  for the chamber slot label in  
1562     trolley  $t$  and  $p$  for the partition label (A, B, C or D depending on the chamber layout) as explained  
1563     in Chapter ??.

1564  
1565     In the context of GIF++, an extra script called by the webDCS is called to extract the histograms  
1566     from the ROOT files. The histograms are then stored in PNG and PDF formats into the correspond-  
1567     ing folder (a single folder per HV step, so per ROOT file). the goal is to then display the histograms  
1568     on the Data Quality Monitoring (DQM) page of the webDCS in order for the users to control the  
1569     quality of the data taking at the end of data taking. An example of histogram organisation is given  
1570     below:

1571



1572        ***Here can put some screens from the webDCS to show the DQM and the plots available to users.***  
 1573

1574        **B.2.1.2 CSV files**

1575        Moreover, up to 3 CSV files can be created depending on which ones of the 3 input files were in the  
 1576        data folder:

- 1577        • `Offline-Corrupted.csv`, is used to keep track of the amount of data that was corrupted and  
 1578           removed from old data format files that don't contain any data quality flag.
- 1579        • `Offline-Current.csv`, contains the summary of the currents and voltages applied on each  
 1580           RPC HV channel.
- 1581        • `Offline-L0-EffC1.csv`, is used to write the efficiencies, cluster size and cluster multiplicity  
 1582           of efficiency runs. Note that `L0` refers here to *Level 0* and means that the results of efficiency and  
 1583           clusterization are a first approximation calculated without performing any muon tracking in

1584 between the different detectors. This offline tool provides the user with a preliminar calculation  
 1585 of the efficiency and of the muon event parameters. Another analysis software especially  
 1586 dedicated to muon tracking is called on selected data to retrieve the results of efficiency and  
 1587 muon clusterization using a tracking algorithm to discriminate noise or gamma from muons  
 1588 as muons are the only particles that pass through the full setup, leaving hits than can be used  
 1589 to reconstruct their tracks.

- 1590     ● `Offline-Rate.csv`, is used to write the noise or gamma rates measured in the detector readout  
 1591 partitions.

1592     Note that these 4 CSV files are created along with their *headers* (`Offline-[...]-Header.csv`  
 1593 containing the names of each data columns) and are automatically merged together when the offline  
 1594 analysis tool is called from the webDCS, contrary to the case where the tool is runned locally from  
 1595 the terminal as the merging bash script is then not called. Thus, the resulting files, used to make  
 1596 official plots, are:

- 1597         ● `Corrupted.csv`,  
 1598         ● `Current.csv`,  
 1599         ● `L0-EffCl.csv`.  
 1600         ● `Rate.csv`.

## 1601     **B.3 Analysis inputs and information handling**

1602     The usage of the Offline Analysis tool as well as its output have been presented in the previous section.  
 1603     It is now important to dig further and start looking at the source code and the inputs necessary  
 1604 for the tool to work. Indeed, other than the raw ROOT data files that are analysed, more information  
 1605 needs to be imported inside of the program to perform the analysis such as the description of the  
 1606 setup inside of GIFT++ at the time of data taking (number of trolleys, of RPCs, dimensions of the  
 1607 detectors, etc...) or the mapping that links the TDC channels to the coresponding RPC channels in  
 1608 order to translate the TDC information into human readable data. 2 files are used to transmit all this  
 1609 information:

- 1610  
 1611         ● `Dimensions.ini`, that provides the necessary setup and RPC information, and  
 1612         ● `ChannelsMapping.csv`, that gives the link between the TDC and RPC channels as well as the  
 1613         *mask* for each channel (masked or not?).

### 1614     **B.3.1 Dimensions file and InFile parser**

1615     This input file, present in every data folder, allows the analysis tool to know of the number of active  
 1616 trolleys, the number of active RPCs in those trolleys, and the details about each RPCs such as  
 1617 the number of RPC gaps, the number of pseudo-rapidity partitions (for CMS-like prototypes), the  
 1618 number of strips per partion or the dimensions. To do so, there are 3 types of groups in the INI file  
 1619 architecture. A first general group, appearing only once at the head of the document, gives information  
 1620 about the number of active trolleys as well as their IDs, as presented in Source Code B.1. For

1621 each active trolley, a group similar to Source Code B.2 can be found containing information about  
 1622 the number of active detectors in the trolley and their IDs. Each trolley group as a `Tt` name format,  
 1623 where `t` is the trolley ID. Finally, for each detector stored in slots of an active trolley, there is a group  
 1624 providing information about their names and dimensions, as shown in Source Code B.3. Each slot  
 1625 group as a `TtSs` name format, where `s` is the slot ID of trolley `t` where the active RPC is hosted.

```
1626 [General]
1627 nTrolleys=2
  TrolleysID=13
```

1628 *Source Code B.1: Example of `[General]` group as might be found in `Dimensions.ini`. In Gif++, only 2 trolleys are available to hold RPCs and place them inside of the bunker for irradiation. The IDs of the trolleys are written in a single string as "13" and then read character by character by the program.*

```
1629 [T1]
  nSlots=4
  SlotsID=1234
```

1630 *Source Code B.2: Example of trolley group as might be found in `Dimensions.ini`. In this example, the file tells that there are 4 detectors placed in the holding slots of the trolley `T1` and that their IDs, written as a single string variable, are 1, 2, 3 and 4.*

```
1631 [T1S1]
  Name=RE2-2-NPD-BARC-8
  Partitions=3
  Gaps=3
  Gap1=BOT
  Gap2=TN
  Gap3=TW
  AreaGap1=11694.25
  AreaGap2=6432
  AreaGap3=4582.82
  Strips=32
  ActiveArea-A=157.8
  ActiveArea-B=121.69
  ActiveArea-C=93.03
```

1632 *Source Code B.3: Example of slot group as might be found in `Dimensions.ini`. In this example, the file provides information about a detector named `RE2-2-NPD-BARC-8`, having 3 pseudo-rapidity readout partitions and stored in slot `S1` of trolley `T1`. This is a CMS RE2-2 type of detector. This information will then be used for example to compute the rate per unit area calculation.*

1633 This information is readout and stored in a C++ object called `IniFile`, that parses the information  
 1634 in the INI input file and stores it into a local buffer for later use. This INI parser is the exact same  
 1635 one that was previously developed for the Gif++ DAQ and described in Appendix A.5.2.

### 1636 B.3.2 TDC to RPC link file and Mapping

1637 The same way the INI dimension file information is stored using `map`, the channel mapping and mask  
 1638 information is stored and accessed through `map`. First of all, the mapping CSV file is organised into  
 1639 3 columns separated by tabulations (and not by commas, as expected for CSV files as it is easier using  
 1640 streams to read tab or space separated data using C++):

1641

1642       RPC\_channel           TDC\_channel           mask

1643       using as formatting for each field:

1644

|      |       |      |   |
|------|-------|------|---|
| 1645 | TSCCC | TCCC | M |
|------|-------|------|---|

1646 TSCCC is a 5-digit integer where  $\tau$  is the trolley ID,  $s$  the slot ID in which the RPC is held insite  
 1647       the trolley  $\tau$  and ccc is the RPC channel number, or *strip* number, that can take values up to  
 1648       3-digits depending on the detector,

1649 TCCC is a 4 digit integer where  $\tau$  is the TDC ID, ccc is the TDC channel number that can take values  
 1650       in between 0 and 127, and

1651 M is a 1-digit integer indicating if the channel should be considered ( $M = 1$ ) or discarded ( $M = 0$ )  
 1652       during analysis.

1653       This mapping and masking information is readout and stored thanks to the object `Mapping`, pre-  
 1654       sented in Source Code B.4. Similarly to `IniFile` objects, this class has private methods. The first  
 1655       one, `Mapping::CheckIfNewLine()` is used to find the newline character '`\n`' or return character  
 1656       '`\r`' (depending on which kind of operating system interacted with the file). This is used for the  
 1657       simple reason that the masking information has been introduced only during the year 2017 but the  
 1658       channel mapping files exist since 2015 and the very beginning of data taking at Gif++. This means  
 1659       that in the older data folders, before the upgrade, the channel mapping file only had 2 columns, the  
 1660       RPC channel and the TDC channel. For compatibility reasons, this method helps controling the  
 1661       character following the readout of the 2 first fields of a line. In case any end of line character is  
 1662       found, no mask information is present in the file and the default  $M = 1$  is used. On the contrary, if  
 1663       the next character was a tabulation or a space, the mask information is present.

1664       Once the 3 fields have been readout, the second private method `Mapping::CheckIfTDCCh()` is  
 1665       used to control that the TDC channel is an existing TDC channel. Finally, the information is stored  
 1666       into 3 different maps (`Link`, `ReverseLink` and `Mask`) thanks to the public method `Mapping::Read()`.  
 1667 `Link` allows to get the RPC channel by knowing the TDC channel while `ReverseLink` does the op-  
 1668       posite by returning the TDC channel by knowing the RPC channel. Finally, `Mask` returns the mask  
 1669       associated to a given RPC channel.

```

1670 typedef map<Uint,Uint> MappingData;
1671
1672 class Mapping {
1673     private:
1674         bool          CheckIfNewLine(char next);
1675         bool          CheckIfTDCCh(Uint channel);
1676         string        FileName;
1677         MappingData  Link;
1678         MappingData  ReverseLink;
1679         MappingData  Mask;
1680         int           Error;
1681
1682     public:
1683         Mapping();
1684         Mapping(string baseName);
1685         ~Mapping();
1686
1687         void SetFileName(const string filename);
1688         int  Read();
1689         Uint GetLink(Uint tdcchannel);
1690         Uint GetReverse(Uint rpcchannel);
1691         Uint GetMask(Uint rpcchannel);
1692     };

```

1672 *Source Code B.4: Description of C++ object Mapping used as a parser for the channel mapping and mask file.*

## 1673 B.4 Description of GIF++ setup within the Offline Analysis tool

1674 In the previous section, the tool input files have been discussed. The dimension file information is  
 1675 stored in a map hosted by the `IniFile` object. But this information is then used to create a series of  
 1676 new objects that helps defining the GIF++ infrastructure directly into the Offline Analysis. Indeed,  
 1677 from the `RPC`, to the more general `Infrastructure`, every element of the GIF++ infrastrucutre is  
 1678 recreated for each data analysis based on the information provided in input. All this information  
 1679 about the infrastructure will be used to assign each hit signal to a specific strip channel of a specific  
 1680 detector, and having a specific active area. This way, rate per unit area calculation is possible.  
 1681

### 1682 B.4.1 RPC objects

1683 `RPC` objects have been developped to represent physical active detectors in GIF++ at the moment  
 1684 of data taking. Thus, there are as many `RPC` objects created during the analysis than there were  
 1685 active `RPCs` tested during a run. Each `RPC` hosts the information present in the corresponding INI  
 1686 slot group, as shown in B.3, and organises it using a similar architecture. This can be seen from  
 1687 Source Code B.5.

1688 To make the object more compact, the lists of gap labels, of gap active areas and strip active  
 1689 areas are stored into `vector` dynamical containers. `RPC` objects are always contructed thanks to the  
 1690 dimension file information stored into the `IniFILE` and their ID, using the format `TtSs`. Using the  
 1691 `RPC` ID, the constructor calls the methods of `IniFILE` to initialise the `RPC`. The other constructors  
 1692 are not used but exist in case of need. Finally, some getters have been written to access the different  
 1693 private parameters storing the detector information.

```

1694
class RPC{
    private:
        string      name;           //RPC name as in webDCS database
        Uint        nGaps;          //Number of gaps in the RPC
        Uint        nPartitions;    //Number of partitions in the RPC
        Uint        nStrips;         //Number of strips per partition
        vector<string> gaps;       //List of gap labels (BOT, TOP, etc...)
        vector<float>  gapGeo;        //List of gap active areas
        vector<float>  stripGeo;      //List of strip active areas

    public:
        RPC();
        RPC(string ID, IniFile* geofile);
        RPC(const RPC& other);
        ~RPC();
        RPC& operator=(const RPC& other);

        string GetName();
        Uint GetNGaps();
        Uint GetNPartitions();
        Uint GetNStrips();
        string GetGap(Uint g);
        float GetGapGeo(Uint g);
        float GetStripGeo(Uint p);
};

1695

```

1696 *Source Code B.5: Description of C++ objects RPC that describe each active detectors used during data taking.*

## 1697 B.4.2 Trolley objects

1698 Trolley objects have been developped to represent physical active trolleys in GIFT++ at the moment  
 1699 of data taking. Thus, there are as many trolley objects created during the analysis than there were  
 1700 active trolleys hosting tested RPCs during a run. Each Trolley hosts the information present in the  
 1701 corresponding INI trolley group, as shown in B.2, and organises it using a similar architecture. In  
 1702 addition to the information hosted in the INI file, these object have a dynamical container of RPC  
 1703 objects, representing the active detectors the active trolley was hosting at the time of data taking.  
 1704 This can been seen from Source Code B.6.

1705 Trolley objects are always contructed thanks to the dimension file information stored into the  
 1706 IniFILE and their ID, using the format Tt. Using the Trolley ID, the constructor calls the methods  
 1707 of IniFILE to initialise the Trolley. Retrieving the information of the RPC IDs via SlotsID, a new  
 1708 RPC is constructed and added to the container RPCs for each character in the ID string. The other  
 1709 constructors are not used but exist in case of need. Finally, some getters have been written to access  
 1710 the different private parameters storing the trolley and detectors information.

```

1711
class Trolley{
    private:
        Uint          nSlots; //Number of active RPCs in the considered trolley
        string        SlotsID; //Active RPC IDs written into a string
        vector<RPC*> RPCs;   //List of active RPCs

    public:
        //Constructors, destructor and operator =
        Trolley();
        Trolley(string ID, IniFile* geofile);
        Trolley(const Trolley& other);
        ~Trolley();
        Trolley& operator=(const Trolley& other);

        //Get GIFTrolley members
        Uint  GetNSlots();
        string GetSlotsID();
        Uint   GetSlotID(Uint s);

        //Manage RPC list
        RPC*  GetRPC(Uint r);
        void  DeleteRPC(Uint r);

        //Methods to get members of RPC objects stored in RPCs
        string GetName(Uint r);
        Uint   GetNGaps(Uint r);
        Uint   GetNPartitions(Uint r);
        Uint   GetNStrips(Uint r);
        string GetGap(Uint r, Uint g);
        float  GetGapGeo(Uint r, Uint g);
        float  GetStripGeo(Uint r, Uint p);
};

1712

```

*Source Code B.6: Description of C++ objects Trolley that describe each active trolley used during data taking.*

#### 1714     **B.4.3 Infrastructure object**

1715 The `Infrastructure` object has been developed to represent the GIFT++ bunker area dedicated to  
 1716 CMS RPC experiments. With this very specific object, all the information about the CMS RPC  
 1717 setup within GIFT++ at the moment of data taking is stored. It hosts the information present in the  
 1718 corresponding INI general group, as shown in B.1, and organises it using a similar architecture. In  
 1719 addition to the information hosted in the INI file, this object have a dynamical container of `Trolley`  
 1720 objects, representing the active tolleys in GIFT++ area. This can be seen from Source Code B.7.

1721 The `Infrastructure` object is always contructed thanks to the dimension file information stored  
 1722 into the `IniFILE`. Retrieving the information of the trolley IDs via `TrolleysID`, a new `Trolley` is  
 1723 constructed and added to the container `Trolleys` for each character in the ID `string`. By extension,  
 1724 it is easy to understand that the process described in Section B.4.2 for the construction of RPCs  
 1725 takes place when a trolley is constructed. The other constructors are not used but exist in case of  
 1726 need. Finally, some getters have been written to access the different private parameters storing the  
 1727 infrastructure, tolleys and detectors information.

```

1728 class Infrastructure {
1729     private:
1730         Uint             nTrolleys;    //Number of active Trolleys in the run
1731         string          TrolleysID;   //Active trolley IDs written into a string
1732         vector<Trolley*> Trolleys;   //List of active Trolleys (struct)
1733
1734     public:
1735         //Constructors and destructor
1736         Infrastructure();
1737         Infrastructure(IniFile* geofile);
1738         Infrastructure(const Infrastructure& other);
1739         ~Infrastructure();
1740         Infrastructure& operator=(const Infrastructure& other);
1741
1742         //Get Infrastructure members
1743         Uint  GetNTrolleys();
1744         string GetTrolleysID();
1745         Uint   GetTrolleyID(Uint t);
1746
1747         //Manage Trolleys
1748         Trolley* GetTrolley(Uint t);
1749         void    DeleteTrolley(Uint t);
1750
1751         //Methods to get members of GIFTrolley objects stored in Trolleys
1752         Uint  GetNSlots(Uint t);
1753         string GetSlotsID(Uint t);
1754         Uint   GetSlotID(Uint t, Uint s);
1755         RPC*  GetRPC(Uint t, Uint r);
1756
1757         //Methods to get members of RPC objects stored in RPCs
1758         string GetName(Uint t, Uint r);
1759         Uint   GetNGaps(Uint t, Uint r);
1760         Uint   GetNPartitions(Uint t, Uint r);
1761         Uint   GetNStrips(Uint t, Uint r);
1762         string GetGap(Uint t, Uint r, Uint g);
1763         float  GetGapGeo(Uint t, Uint r, Uint g);
1764         float  GetStripGeo(Uint t, Uint r, Uint p);
1765     };

```

Source Code B.7: Description of C++ object *Infrastructure* that contains the full information about CMS RPC experiment in *GIF++*.

## 1731 B.5 Handeling of data

1732 As discussed in Appendix A.4.2, the raw data as a `TTree` architecture where every entry is related to  
1733 a trigger signal provided by a muon or a random pulse, whether the goal of the data taking was to  
1734 measure the performance of the detector or the noise/gamma background respectively. Each of these  
1735 entries, referred also as events, contain a more or less full list of hits in the TDC channels to which  
1736 the detectors are connected. To this list of hits corresponds a list of time stamps, marking the arrival  
1737 of the hits within the TDC channel.

1738 The infrastructure of the CMS RPC experiment within *GIF++* being defined, combining the  
1739 information about the raw data with the information provided by both the mapping/mask file and the  
1740 dimension file allows to build new physical objects that will help in computing efficiency or rates.

### 1741 B.5.1 RPC hits

1742 The raw data stored in the ROOT file as output of the GIFT++ DAQ, is readout by the analysis tool  
 1743 using the structure `RAWData` presented in Source Code B.9 that differs from the structure presented  
 1744 in Appendix A.4.2 as it is not meant to hold all of the data contained in the ROOT file. In this sense,  
 1745 this structure is in the case of the offline analysis tool not a dynamical object and will only be storing  
 1746 a single event contained in a single entry of the `TTree`.

```
1747
1748 class RPCHit {
    private:
        Uint Channel;      //RPC channel according to mapping (5 digits)
        Uint Trolley;     //0, 1 or 3 (1st digit of the RPC channel)
        Uint Station;     //Slot where is held the RPC in Trolley (2nd digit)
        Uint Strip;       //Physical RPC strip where the hit occurred (last 3
                           → digits)
        Uint Partition;   //Readout partition along eta segmentation
        float TimeStamp; //Time stamp of the arrival in TDC

    public:
        //Constructors, destructor & operator =
        RPCHit();
        RPCHit(Uint channel, float time, Infrastructure* Infra);
        RPCHit(const RPCHit& other);
        ~RPCHit();
        RPCHit& operator=(const RPCHit& other);

        //Get RPCHit members
        Uint GetChannel();
        Uint GetTrolley();
        Uint GetStation();
        Uint GetStrip();
        Uint GetPartition();
        float GetTime();
    };

    typedef vector<RPCHit> HitList;
    typedef struct GIFHitList { HitList rpc[NTROLLEYS][NSLOTS][NPARTITIONS]; } →
        GIFHitList;

    bool SortHitbyStrip(RPCHit h1, RPCHit h2);
    bool SortHitbyTime(RPCHit h1, RPCHit h2);
}
```

1749 *Source Code B.8: Description of C++ object `RPCHit`.*

```
1750
1751 struct RAWData{
    int iEvent;           //Event i
    int TDCNHits;        //Number of hits in event i
    int QFlag;            //Quality flag list (1 flag digit per TDC)
    vector<Uint> *TDCCh; //List of channels giving hits per event
    vector<float> *TDCTS; //List of the corresponding time stamps
};
```

1751 *Source Code B.9: Description of C++ structure `RAWData`.*

1752 Each member of the structure is then linked to the corresponding branch of the ROOT data tree,  
 1753 as shown in the example of Source Code B.10, and using the method `GetEntry(int i)` of the ROOT  
 1754 class `TTree` will update the state of the members of `RAWData`.

```

1755     TTree* dataTree = (TTree*)dataFile.Get("RAWData");
1756     RAWData data;
1757
1758     dataTree->SetBranchAddress("EventNumber", &data.iEvent);
1759     dataTree->SetBranchAddress("number_of_hits", &data.TDCNHits);
1760     dataTree->SetBranchAddress("Quality_flag", &data.QFlag);
1761     dataTree->SetBranchAddress("TDC_channel", &data.TDCCh);
1762     dataTree->SetBranchAddress("TDC_TimeStamp", &data.TDCTS);

```

1757       *Source Code B.10: Example of link in between RAWData and TTree.*

1758       The data is then analysed entry by entry and to each element of the TDC channel list, a `RPCHit` is  
1759       constructed by linking each TDC channel to the corresponding RPC channel thanks to the `Mapping`  
1760       object. The information carried by the RPC channel format allows to easily retrieve the trolley and  
1761       slot from which the hit was recorded (see section B.3.2). Using these 2 values, the readout partition  
1762       can be found by knowing the strip channel and comparing it with the number of partitions and strips  
1763       per partition stored into the `Infrastructure` object.

1764       Thus `RPCHit` objects are then stored into 3D dynamical list called `GIFHitList` (Source Code B.9)  
1765       where the 3 dimensions refer to the 3 layers of the readout in `GIF++` : in the bunker there are *trolleys*  
1766       ( $\tau$ ) holding detectors in *slots* ( $s$ ) and each detector readout is divided into 1 or more pseudo-rapidity  
1767       *partitions* ( $p$ ). Using these 3 information allows to assign an address to each readout partition and  
1768       this address will point to a specific hit list.

1769

## 1770       B.5.2 Clusters of hits

1771       All the hits contained in the ROOT file have been sorted into the different hit lists through the  
1772       `GIFHitList`. At this point, it is possible to start looking for clusters. A cluster is a group of adjacent  
1773       strips getting hits within a time window of 25 ns. These strips are then assumed to be part of the same  
1774       physical avalanche signal generated by a muon passing through the chamber or by the interaction of  
1775       a gamma stopping into the electrodes of the RPCs.

1776       To keep the cluster information, `RPCCluster` objects have been defined as shown in Source  
1777       Code B.11. Using the information of each individual `RPCHit` taken out of the hit list, it stores  
1778       the cluster size (number of adjacent strips composing the cluster), the first and last hit, the center for  
1779       spatial reconstruction and finally the start and stop time stamps as well as te time spread in between  
1780       the first and last hit.

```

1781
class RPCCluster{
    private:
        Uint ClusterSize; //Size of cluster #ID
        Uint FirstStrip; //First strip of cluster #ID
        Uint LastStrip; //Last strip of cluster #ID
        float Center; //Center of cluster #ID ((first+last)/2)
        float StartStamp; //Time stamp of the earliest hit of cluster #ID
        float StopStamp; //Time stamp of the latest hit of cluster #ID
        float TimeSpread; //Time difference between earliest and latest hits
                           //of cluster #ID
    public:
        //Constructors, destructor & operator =
        RPCCluster();
        RPCCluster(HitList List, Uint cID, Uint cSize, Uint first, Uint firstID);
        RPCCluster(const RPCCluster& other);
        ~RPCCluster();
        RPCCluster& operator=(const RPCCluster& other);

        //Get Cluster members
        Uint GetID();
        Uint GetSize();
        Uint GetFirstStrip();
        Uint GetLastStrip();
        float GetCenter();
        float GetStart();
        float GetStop();
        float GetSpread();
    };

    typedef vector<RPCCluster> ClusterList;

    //Other functions to build cluster lists out of hit lists
    void BuildClusters(HitList &cluster, ClusterList &clusterList);
    void Clusterization(HitList &hits, TH1 *hcSize, TH1 *hcMult);
}

```

1783                         *Source Code B.11: Description of C++ object Cluster.*

1784     To investigate the hit list of a given detector partition, the function `Clusterization()` defined  
 1785     in `include/Cluster.h` needs the hits in the list to be time sorted. This is achieved by calling func-  
 1786     tion `sort()` of library `<algorithm>` using the comparator `SortHitbyTime(RPCHit h1, RPCHit h2)`  
 1787     defined in `include/RPCHit.h` that returns `true` if the time stamp of hit `h1` is lower than that of `h2`.  
 1788     A first isolation of strips is made only based on time information. All the hits within the 25 ns win-  
 1789     dow are taken separately from the rest. Then, this sub-list of hits is sorted this time by ascending  
 1790     strip number, using this time the comparator `SortHitbyStrip(RPCHit h1, RPCHit h2)`. Finally, the  
 1791     groups of adjacent strips are used to construct `RPCCluster` objects that are then stored in a temporary  
 1792     list of clusters that is at the end of the process used to know how many clusters were reconstructed  
 1793     and to fill their sizes into an histogram that will allows to know the mean size of muon or gamma  
 1794     clusters.

1795

## 1796     **B.6 DAQ data Analysis**

1797     All the ingredients to analyse GIFT++ data have been defined. This section will focus on the different  
 1798     part of the analysis performed on the data, from determining the type of data the tool is dealing with

<sup>1799</sup> to calculating the rate in each detector or reconstructing muon or gamma clusters.

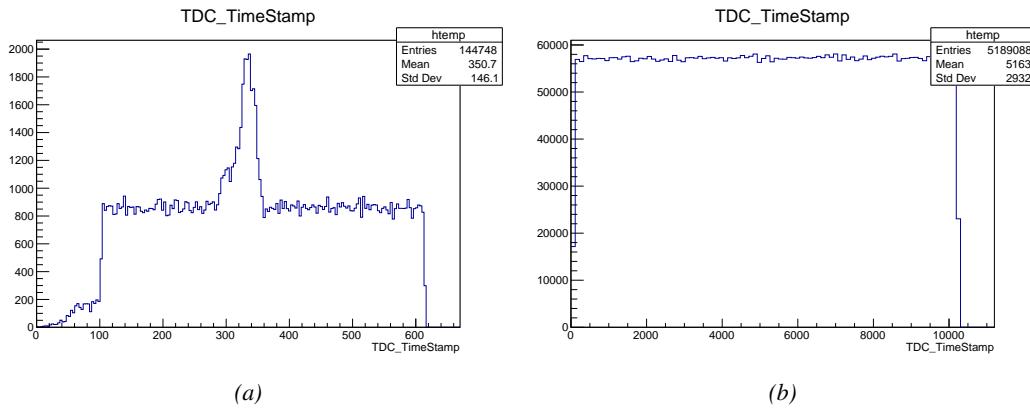
### <sup>1800</sup> B.6.1 Determination of the run type

<sup>1801</sup> In GIF++, both the performance of the detectors in detecting muons in an irradiated environment and  
<sup>1802</sup> the gamma background can be independantly measured. These corresponds to different run types  
<sup>1803</sup> and thus, to different TDC settings giving different data to look at.

<sup>1804</sup>

<sup>1805</sup> In the case of performance measurements, the trigger for data taking is provided by the coinci-  
<sup>1806</sup> dence of several scintillators when muons from the beam passing through the area are detected. Data  
<sup>1807</sup> is collected in a 600 ns wide window around the arrival of muons in the RPCs. The expected time  
<sup>1808</sup> distribution of hits is shown in Figure B.1a. The muon peak is clearly visible in the center of the  
<sup>1809</sup> distribution and is to be extracted from the gamma background that composes the flat part of the  
<sup>1810</sup> distribution.

<sup>1811</sup> On the other hand, gamma background or noise measurements are focussed on the non muon  
<sup>1812</sup> related physics and the trigger needs to be independant from the muons to give a good measurement  
<sup>1813</sup> of the gamma/noise distribution as seen by the detectors. The trigger is then provided by a pulse  
<sup>1814</sup> generator at a frequency of 300 Hz whose pulse is not likely to be on time with a muon. In order  
<sup>1815</sup> to increase the integrated time without increasing the acquisition time too much, the width of the  
<sup>1816</sup> acquisition windows are increased to 10  $\mu$ s. The time distribution of the hits is expected to be flat, as  
<sup>1817</sup> shown by Figure B.1b.



*Figure B.1: Example of expected hit time distributions in the cases of efficiency (Figure B.1a) and noise/gamma rate per unit area (Figure B.1b) measurements as extracted from the raw ROOT files. The unit along the x-axis corresponds to ns. The fact that "the" muon peak is not well defined in Figure B.1a is due to the contribution of all the RPCs being tested at the same time that don't necessarily have the same signal arrival time. Each individual peak can have an offset with the ones of other detectors. The inconsistency in the first 100 ns of both time distributions is an artefact of the TDCs and are systematically rejected during the analysis.*

<sup>1818</sup> The ROOT files include a TTree called RunParameters containing, among other things, the in-  
<sup>1819</sup> formation related to the type of run. The run type can then be accessed as described by Source  
<sup>1820</sup> Code B.12 and the function IsEfficiencyRun() is then used to determine if the run file is an effi-  
<sup>1821</sup> ciency run or, on the contrary, another type of run (noise or gamma measurement).

```

1822     TTree* RunParameters = (TTree*)dataFile.Get("RunParameters");
1823     TString* RunType = new TString();
1824     RunParameters->SetBranchAddress("RunType", &RunType);
1825     RunParameters->GetEntry(0);

```

1824       *Source Code B.12: Access to the run type contained in TTree\* RunParameters.*

1825       Finally, the data files will have a slightly different content whether it was collected before or after  
1826       October 2017 and the upgrade of the DAQ software that brought a new information into the ROOT  
1827       output. This is discussed in Appendix A.4.3 and implies that the analysis will differ a little depending  
1828       on the data format. Indeed, as no information on the data quality is stored, in older data files, the cor-  
1829       rections for missing events has to be done at the end of the analysis. The information about the type  
1830       of data format is stored in the variable **bool** `isNewFormat` by checking the list of branches contained  
1831       in the data tree via the methods `TTree::GetListOfBranches()` and `TCollection::Contains()`.

### 1832       **B.6.2 Beam time window calculation for efficiency runs**

1833       Knowing the run type is important first of all to know the width of the acquisition window to be used  
1834       for the rate calculation and finally to be able to seek for muons. Indeed, the peak that appears in the  
1835       time distribution for each detectors is then fitted to extract the most probable time window in which  
1836       the tool should look for muon hits. The data outside of this time window is then used to evaluate the  
1837       noise or gamma background the detector was subjected to during the data taking. Computing the  
1838       position of the peak is done calling the function `SetBeamWindow()` defined in file `src/RPCHit.cc` that  
1839       loops a first time on the data. The data is first sorted in a 3D array of 1D histograms (`GIFH1Array`, see  
1840       `include/types.h`). Then the location of the highest bin is determined using `TH1::GetMaximumBin()`  
1841       and is used to define a window in which a gaussian fit will be applied to compute the peak width.  
1842       This window is a 80 ns defined by Formula B.1 around the central bin.

$$t_{center}(ns) = bin \times width_{bin}(ns) \quad (\text{B.1a})$$

$$[t_{low}; t_{high}] = [t_{center} - 40; t_{center} + 40] \quad (\text{B.1b})$$

1843       Before the fit is performed, the average number of noise/gamma hits per bin is evaluated using  
1844       the data outside of the fit window. Excluding the first 100 ns, the average number of hits per bin  
1845       due to the noise or gamma is defined by Formula B.2 after extracting the amount of hits in the time  
1846       windows  $[100; t_{low}]$  and  $[t_{high}; 600]$  thanks to the method `TH1::Integral()`. This average number  
1847       of hits is then subtracted to every bin of the 1D histogram, in order to *clean* it from the noise or  
1848       gamma contribution as much as possible to improve the fit quality. Bins where  $\langle n_{hits} \rangle$  is greater  
1849       than the actual bin content are set to 0.

$$\Delta t_{noise}(ns) = 600 \overbrace{-t_{high} + t_{low}}^{-80ns} - 100 = 420ns \quad (\text{B.2a})$$

$$\langle n_{hits} \rangle = width_{bin}(ns) \times \frac{\sum_{t=100}^{t_{low}} + \sum_{t=t_{high}}^{600}}{\Delta t_{noise}(ns)} \quad (\text{B.2b})$$

1850       Finally, the fit parameters are extracted and saved for each detector in 3D arrays of **float**  
1851       (`muonPeak`, see `include/types.h`), a first one for the mean arrival time of the muons, `PeakTime`,

1852 and a second one for the width of the peak, `PeakWidth`. The width is defined as  $6\sigma$  of the gaussian  
 1853 fit. The same settings are applied to every partitions of the same detector. To determine which one  
 1854 of the detector's partitions is directly illuminated by the beam, the peak height of each partition is  
 1855 compared and the highest one is then used to define the peak settings.

### 1856 **B.6.3 Data loop and histogram filling**

1857 3D arrays of histogram are created to store the data and display it on the DQM of G4F++ webDCS  
 1858 for the use of shifters. These histograms, presented in section B.2.1.1, are filled while looping on  
 1859 the data. Before starting the analysis loop, it is necessary to control the entry quality for the new  
 1860 file formats featuring `QFlag`. If the `QFlag` value for this entry shows that 1 TDC or more have a  
 1861 `CORRUPTED` flag, then this event is discarded. The loss of statistics is low enough to be neglected.  
 1862 `QFlag` is controlled using the function `IsCorruptedEvent()` defined in `src/utils.cc`. As explained  
 1863 in Appendix A.4.3, each digit of this integer represents a TDC flag that can be 1 or 2. Each 2 is  
 1864 the sign of a `CORRUPTED` state. Then, the data is accessed entry by entry in the ROOT `TTree` using  
 1865 `RAWData` and each hit in the hit list is assigned to a detector channel and saved in the corresponding  
 1866 histograms. In the first part of the analysis, in which the loop over the ROOT file's content is  
 1867 performed, the different steps are:

1868 **1- RPC channel assignment and control:** a check is done on the RPC channel extracted thanks  
 1869 to the mapping via the method `Mapping::GetLink()`. If the channel is not initialised and is 0, or if  
 1870 the TDC channel was greater than 5127, the hit is discarded. This means there was a problem in the  
 1871 mapping. Often a mapping problem leads to the crash of the offline tool.

1872 **2- Creation of a `RPCHit` object:** to easily get the trolley, slot and partition in which the hit has  
 1873 been assigned, this object is particularly helpful.

1874 **3- General histograms are filled:** the hit is filled into the time distribution and the general hit  
 1875 distribution histograms, and if the arrival time is within the first 100 ns, it is discarded and nothing  
 1876 else happens and the loop proceeds with the next hit in the list.

1877 **4- Multiplicity counter:** the hit multiplicity counter of the corresponding detectors incremented.

1878 **5-a- Efficiency runs - Is the hit within the peak window? :** if the peak is contained in the peak  
 1879 window previously defined in section B.6.2, the hit is filled into the beam hit profile histogram of  
 1880 the corresponding chamber, added into the list of muon hits and increments the counter of *in time*  
 1881 hits. The term *in time* here refers to the hits that are likely to be muons by arriving in the expected  
 1882 time window. If the hit is outside of the peak window, it is filled into the noise profile histogram  
 1883 of the corresponding detector, added into the list of noise/gamma hits and increments the counter of  
 1884 noise/gamma hits.

1885 **5-b- Noise/gamma rate runs - Noise histograms are filled:** the hit is filled into the noise profile  
 1886 histogram of the corresponding detector, added into the list of noise/gamma hits and increments the  
 1887 counter of noise/gamma hits.

1888

1889 After the loop on the hit list of the entry is over, the next step is to clusterize the 3D lists filled  
 1890 in the previous steps. A 3D loop is then started over the active trolley, slot and RPC partitions to  
 1891 access these objects. Each `NoiseHitList` and `MuonHitList`, in case of efficiency run, are clusterized  
 1892 as described in section B.5.2. There corresponding cluster size and multiplicity histograms are filled  
 1893 at the end of the clustering process. Then, the efficiency histogram is filled in case of efficiency run.  
 1894 The selection is simply made by checking whether the RPC detected signals in the peak window  
 1895 during this event. Nevertheless, it is useful to highlight that at this level, it is not possible yet to  
 1896 discriminate in between a muon hit and noise or gamma hit. Thus, `MuonCSize_H`, `MuonCMult_H`  
 1897 and `Efficiency0_H` are subjected to noise and gamma contamination. This contamination will be  
 1898 estimated and corrected at the moment the results will be written into output CSV files. Finally, the  
 1899 loop ends on the filling of the general hit multiplicity histogram.

#### 1900 **B.6.4 Results calculation**

1901 As mentioned in section B.2.1, the analysis of DAQ data provides the user with 3 CSV files and  
 1902 a ROOT file associated to each and every ROOT data file. The fourth CSV file is provided by the  
 1903 extraction of the CEAN main frame data monitored during data taking and will be discussed later.  
 1904 After looping on the data in the previous part of the analysis macro, the output files are created and a  
 1905 3D loop on each RPC readout partitions is started to extract the histograms parameters and compute  
 1906 the final results.

1907

##### 1908 **B.6.4.1 Rate normalisation**

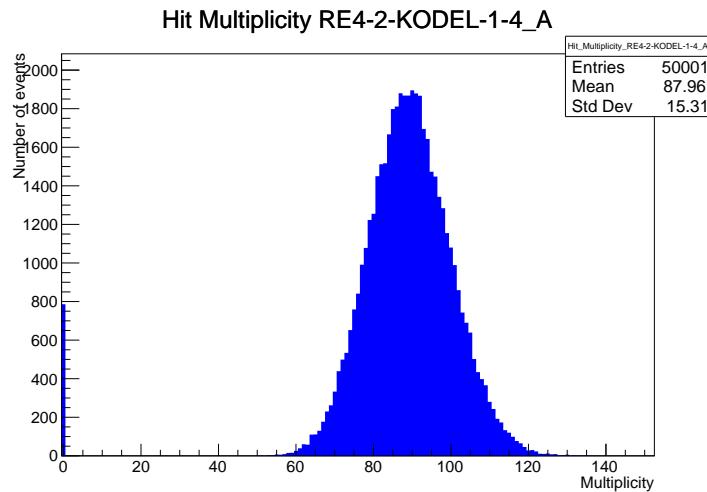


Figure B.2: The effect of the quality flag is explained by presenting the reconstructed hit multiplicity of a data file without `Quality_flag`. The artificial high content of bin 0 is the effect of corrupted data.

1909 To analyse old data format files, not containing any quality flag, it is needed to estimate the amount  
 1910 of corrupted data via a fit as the corrupted data will always fill events with a fake "0 multiplicity".  
 1911 Indeed, as no hits were stored in the DAQ ROOT files, these events artificially contribute to fill  
 1912 the bin corresponding to a null multiplicity, as shown in Figure B.2. In the case the mean of the

hit multiplicity distribution is high, the contribution of the corrupted data can easily be evaluated for later correction by comparing the level of the bin at multiplicity 0 and of a skew fit curve that should indicate a value consistent with 0. A skew fit has been chosen over a Poisson fit as it was giving better results for lower mean multiplicity values. Nevertheless, for low irradiation cases, as explained in Appendix A.4.3, the hit multiplicity distribution mean is, on the contrary, rather small and the probability to record events without hits can't be considered small anymore, leading to a difficult and non-reliable estimation of the corruption. As can be seen in Source Code B.13, conditions have been applied to prevent bad fits and wrong corruption estimation in cases where :

- The difference in between the data for multiplicity 1 and the corresponding fit value should be lower than 1% of the total amount of data :  $\frac{|n_{m=1} - sk(1)|}{N_{tot}} < 0.01$  where  $n_{m=1}$  is the number of entries with multiplicity 1,  $sk(1)$  the value of the skew fit, as defined by Formula 4.3, for multiplicity 1 and  $N_{tot}$  the total number of entries.

- The amount of data contained in the multiplicity 0 bin should not exceed 40% :  $\frac{n_{m=0}}{N_{tot}} \leq 0.4$  where  $n_{m=0}$  is the number of entries with multiplicity 0. This number has been determined to be the maximum to be able to separate the excess of data due to corruption from the hit multiplicity distribution.

Those 2 conditions need to be fulfilled to estimate the corruption of old data format files. If the fit was successful, the level of corruption is written in `Offline-Corrupted.csv` and the number of corrupted entries, refered as the integer `nEmptyEvent`, is subtracted from the total number of entries when the rate normalisation factor is computed as explicit in Source Code B.13. Note that for new data format files, the number of corrupted entries being set to 0, the definition of `rate_norm` stays valid.

```

1935   if(!isNewFormat){
      TF1* GaussFit = new TF1("gaussfit","[0]*exp(-0.5*((x-[1])/[2])**2)",0,Xmax);
      GaussFit->SetParameter(0,100);
      GaussFit->SetParameter(1,10);
      GaussFit->SetParameter(2,1);
      HitMultiplicity_H.rpc[T][S][p]->Fit(GaussFit,"LIQR","",0.5,Xmax);

      TF1* SkewFit = new TF1("skewfit","[0]*exp(-0.5*((x-[1])/[2])**2) / (1 +
→      exp(-[3]*(x-[4])))",0,Xmax);
      SkewFit->SetParameter(0,GaussFit->GetParameter(0));
      SkewFit->SetParameter(1,GaussFit->GetParameter(1));
      SkewFit->SetParameter(2,GaussFit->GetParameter(2));
      SkewFit->SetParameter(3,1);
      SkewFit->SetParameter(4,1);
      HitMultiplicity_H.rpc[T][S][p]->Fit(SkewFit,"LIQR","",0.5,Xmax);

      double fitValue = SkewFit->Eval(1,0,0,0);
      double dataValue = (double)HitMultiplicity_H.rpc[T][S][p]->GetBinContent(2);
      double difference = TMath::Abs(dataValue - fitValue);
      double fitTOdataVSentries_ratio = difference / (double)nEntries;
      bool isFitGOOD = fitTOdataVSentries_ratio < 0.01;

1936      double nSinglehit = (double)HitMultiplicity_H.rpc[T][S][p]->GetBinContent(1);
      double lowMultRatio = nSinglehit / (double)nEntries;
      bool isMultLOW = lowMultRatio > 0.4;

      if(isFitGOOD && !isMultLOW){
          nEmptyEvent = HitMultiplicity_H.rpc[T][S][p]->GetBinContent(1);
          nPhysics = (int)SkewFit->Eval(0,0,0,0);
          if(nPhysics < nEmptyEvent)
              nEmptyEvent = nEmptyEvent-nPhysics;
      }
}

double corrupt_ratio = 100.*(double)nEmptyEvent / (double)nEntries;
outputCorrCSV << corrupt_ratio << '\t';

float rate_norm = 0.;
float stripArea = GIFIInfra->GetStripGeo(tr,sl,p);

if(IsEfficiencyRun(RunType)){
    float noiseWindow = BMTDCWINDOW - TIMEREJECT - 2*PeakWidth.rpc[T][S][p];
    rate_norm = (nEntries-nEmptyEvent)*noiseWindow*1e-9*stripArea;
} else
    rate_norm = (nEntries-nEmptyEvent)*RDMNOISEWDW*1e-9*stripArea;

```

*Source Code B.13: Definition of the rate normalisation variable. It takes into account the number of non corrupted entries and the time window used for noise calculation, to estimate the total integrated time, and the strip active area to express the result as rate per unit area.*

#### 1938    B.6.4.2 Rate and activity

1939    At this point, the strip rate histograms, StripNoiseProfile\_H.rpc[T][S][p], only contain an in-
1940    formation about the total number of noise or rate hits each channel received during the data taking.
1941    As described in Source Code B.14, a loop on the strip channels will be used to normalise the content
1942    of the rate distribution histogram for each detector partitions. The initial number of hits recorded for
1943    a given bin will be extracted and 2 values will be computed:

- 1944     ● the strip rate, defined as the number of hits recorded in the bin normalised like described in  
 1945       the previous section, using the variable `rate_norm`, and

- 1946     ● the strip activity, defined as the number of hits recorded in the bin normalised to the average  
 1947       number of hits per bin contained in the partition histogram, using the variable `averageNhit`.  
 1948       This value provides an information on the homogeneity of the detector response to the gamma  
 1949       background or of the detector noise. An activity of 1 corresponds to an average response.  
 1950       Above 1, the channel is more active than the average and bellow 1, the channel is less active.

```

int nNoise = StripNoiseProfile_H.rpc[T][S][p]->GetEntries();
float averageNhit = (nNoise>0) ? (float)(nNoise/nStripsPart) : 1.;

for(Uint st = 1; st <= nStripsPart; st++) {
    float stripRate =
        StripNoiseProfile_H.rpc[T][S][p]->GetBinContent(st)/rate_norm;
    float stripAct =
        StripNoiseProfile_H.rpc[T][S][p]->GetBinContent(st)/averageNhit;

    StripNoiseProfile_H.rpc[T][S][p]->SetBinContent(st,stripRate);
    StripActivity_H.rpc[T][S][p]->SetBinContent(st,stripAct);
}

```

1952     *Source Code B.14: Description of the loop that allows to set the content of each strip rate and strip activity  
 channel for each detector partition.*

1953     On each detector partitions, which are readout by a single FEE, all the channels are not processed  
 1954       by the same chip. Each chip can give a different noise response and thus, histograms using a chip  
 1955       binning are used to investigate chip related noise behaviours. The average values of the strip rate  
 1956       or activity grouped into a given chip are extracted using the using the function `GetChipBin()` and  
 1957       stored in dedicated histograms as described in Source Codes B.15 and B.16 respectively.

```

float GetChipBin(TH1* H, Uint chip){
    Uint start = 1 + chip*NSTRIPSCHIP;
    int nActive = NSTRIPSCHIP;
    float mean = 0.;

    for(Uint b = start; b <= (chip+1)*NSTRIPSCHIP; b++) {
        float value = H->GetBinContent(b);
        mean += value;
        if(value == 0.) nActive--;
    }

    if(nActive != 0) mean /= (float)nActive;
    else mean = 0.;

    return mean;
}

```

1960     *Source Code B.15: Function used to compute the content of a bin for an histogram using chip binning.*

```

1961   for(UInt ch = 0; ch < (nStripsPart/NSTRIPSCHIP); ch++) {
1962     ChipMeanNoiseProf_H.rpc[T][S][p]->
1963       SetBinContent(ch+1,GetChipBin(StripNoiseProfile_H.rpc[T][S][p],ch));
1964     ChipActivity_H.rpc[T][S][p]->
1965       SetBinContent(ch+1,GetChipBin(StripActivity_H.rpc[T][S][p],ch));
1966   }

```

*Source Code B.16: Description of the loop that allows to set the content of each chip rate and chip activity bins for each detector partition knowing the information contained in the corresponding strip distribution histograms.*

1963     The activity variable is used to evaluate the homogeneity of the detector response to background  
1964     or of the detector noise. The homogeneity  $h_p$  of each detector partition can be evaluated using the  
1965     formula  $h_p = \exp(-\sigma_p^R / \langle R \rangle_p)$ , where  $\langle R \rangle_p$  is the partition mean rate and  $\sigma_p^R$  is the rate standard  
1966     deviation calculated over the partition channels. The more homogeneously the rates are distributed  
1967     and the smaller will  $\sigma_p^R$  be, and the closer to 1 will  $h_p$  get. On the contrary, if the standard deviation  
1968     of the channel's rates is large,  $h_p$  will rapidly get to 0. This value is saved into histograms as shown  
1969     in Source Code B.17 and could in the future be used to monitor through time, once extracted, the  
1970     evolution of every partition homogeneity. This could be of great help to understand the apparition of  
1971     eventual hot spots due to ageing of the chambers subjected to high radiation levels. The monitored  
1972     homogeneity information could then be combined with a monitoring of the activity of each individual  
1973     channel in order to have a finer information. Monitoring tools have been suggested and need to  
1974     be developed for this purpose.

```

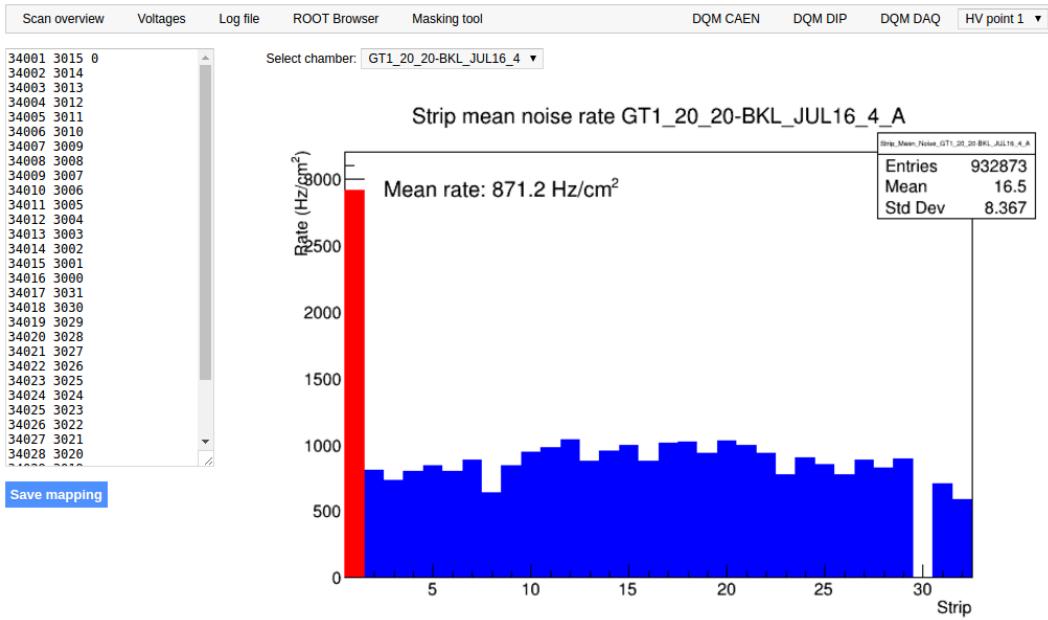
1975   float MeanPartSDev = GetTH1StdDev(StripNoiseProfile_H.rpc[T][S][p]);
1976   float strip_homog = (MeanPartRate==0)
1977     ? 0.
1978     : exp(-MeanPartSDev/MeanPartRate);
1979   StripHomogeneity_H.rpc[T][S][p]->Fill("exp -#left(#frac{\#sigma_{Strip}
1980     \rightarrow Rate}{\#mu_{Strip Rate}}\#right)",strip_homog);
1981   StripHomogeneity_H.rpc[T][S][p]->GetYaxis()->SetRangeUser(0.,1.);
1982
1983   float ChipStDevMean = GetTH1StdDev(ChipMeanNoiseProf_H.rpc[T][S][p]);
1984   float chip_homog = (MeanPartRate==0)
1985     ? 0.
1986     : exp(-ChipStDevMean/MeanPartRate);
1987   ChipHomogeneity_H.rpc[T][S][p]->Fill("exp -#left(#frac{\#sigma_{Chip}
1988     \rightarrow Rate}{\#mu_{Chip Rate}}\#right)",chip_homog);
1989   ChipHomogeneity_H.rpc[T][S][p]->GetYaxis()->SetRangeUser(0.,1.);

```

1977     *Source Code B.17: Storage of the homogeneity into dedicated histograms.*

#### 1978     **B.6.4.3 Strip masking tool**

1979     The offline tool is automatically called at the end of each data taking to analyse the data and offer  
1980     the shifter DQM histograms to control the data quality. After the histograms have been published  
1981     online in the DQM page, the shifter can decide to mask noisy or dead channels that will contribute  
1982     to bias the final rate calculation by editing the mask column of `ChannelsMapping.csv` as can be seen  
1983     in Figure B.3.



*Figure B.3: Display of the masking tool page on the webDCS. The window on the left allows the shifter to edit ChannelsMapping.csv. To mask a channel, it only is needed to set the 3rd field corresponding to the strip to mask to 0. It is not necessary for older mapping file formats to add a 1 for each strip that is not masked as the code is versatile and the default behaviour is to consider missing mask fields as active strips. The effect of the mask is directly visible for noisy channels as the corresponding bin turns red. The global effect of masking strips will be an update of the rate value showed on the histogram that will take into consideration the rejected channels.*

1984        From the code point of view, the function `GetTH1Mean()` is used to retrieve the mean rate par-  
 1985        tition by partition after the rates have been calculated strip by strip and filled into the histograms  
 1986        `StripNoiseProfile_H.rpc[T][S][p]`, as described through Source Code B.18.

1987        Once the mask for each rejected channel has been updated, the shifter can manually run the of-  
 1988        fline tool again to update the DQM plots, now including the masked strips, as well the rate results  
 1989        written in the output CSV file `Offline-Rate.csv`. If not done during the shifts, the strip masking  
 1990        procedure needs to be carefully done by the person in charge of data analysis on the scans that were  
 1991        selected to produce the final results.

```

1992   float GetTH1Mean(TH1* H){
1993     int nBins = H->GetNbinsX();
1994     int nActive = nBins;
1995     float mean = 0.;
1996
1997     for(int b = 1; b <= nBins; b++){
1998       float value = H->GetBinContent(b);
1999       mean += value;
2000       if(value == 0.) nActive--;
2001     }
2002
2003     if(nActive != 0) mean /= (float)nActive;
2004     else mean = 0.;
2005
2006     return mean;
2007   }

```

*Source Code B.18: The function `GetTH1Mean()` is used to return the mean along the y-axis of `TH1` histograms containing rate information. In order to take into account masked strips whose rate is set to 0, the function looks for masked channels and decrement the number of active channels for each null value found.*

#### 1995    B.6.4.4 Output CSV files filling

1996 All the histograms have been filled. Parameters will then be extracted from them to compute the  
 1997 final results that will later be used to produce plots. Once the results have been computed, the very  
 1998 last step of the offline macro is to write these values into the corresponding CSV outputs. Aside of  
 1999 the file `Offline-Corrupted.csv`, 2 CSV files are being written by the macro `offlineAnalysis()`,  
 2000 `Offline-Rates.csv` and `Offline-L0-EffCl.csv` that respectively contain information about noise  
 2001 or gamma rates, cluster size and multiplicity, and about level 0 reconstruction of the detector effi-  
 2002 ciency, muon cluster size and multiplicity. Details on the computation and file writing are respec-  
 2003 tively given in Sources Codes B.19 and B.20.

2004 **Noise/gamma background variables** are computed and written in the output file for each detector  
 2005 partitions. A detector average of the hit and cluster rate is also provided, as shown through Sources  
 2006 Code B.19. The variables that are written for each partition are:

- 2007    • The mean partition hit rate per unit area, `MeanPartRate`, that is extracted from the histogram  
     2008    `StripNoiseProfile_H` as the mean value along the y-axis, as described in section B.6.4.3. No  
     2009    error is recorded for the hit rate as this is considered a single measurement. No statistical error  
     2010    can be associated to it and the systematics are unknown.
- 2011    • The mean cluster size, `cSizePart`, is extracted from the histogram `NoiseCSize_H` and it's  
     2012    statistical error, `cSizePartErr`, is taken to be  $2\sigma$  of the total distribution.
- 2013    • The mean cluster multiplicity per trigger, `cMultPart`, is extracted from the histogram `NoiseCMult_H`  
     2014    and it's statistical error, `cMultPartErr`, is taken to be  $2\sigma$  of the total distribution. It is impor-  
     2015    tant to point to the fact that this variable gives an information that is dependent on the buffer  
     2016    window width used for each trigger for the calculation.
- 2017    • The mean cluster rate per unit area, `ClustPartRate`, is defined as the mean hit rate normalised

2018 to the mean cluster size and it's statistical error, `ClustPartRateErr`, is then obtained using the  
2019 relative statistical error on the mean cluster size.

```

for (UInt tr = 0; tr < GIFInfra->GetNTrolleys(); tr++){
    UInt T = GIFInfra->GetTrolleyID(tr);

    for (UInt sl = 0; sl < GIFInfra->GetNSlots(tr); sl++){
        UInt S = GIFInfra->GetSlotID(tr,sl) - 1;

        float MeanNoiseRate = 0.;
        float ClusterRate = 0.;
        float ClusterSDev = 0.;

        for (UInt p = 0; p < GIFInfra->GetNPartitions(tr,sl); p++){
            float MeanPartRate = GetTH1Mean(StripNoiseProfile_H.rpc[T][S][p]);
            float cSizePart = NoiseCSIZE_H.rpc[T][S][p]->GetMean();
            float cSizePartErr = (NoiseCSIZE_H.rpc[T][S][p]->GetEntries()==0)
                ? 0.
                : 2*NoiseCSIZE_H.rpc[T][S][p]->GetStdDev() /
                    sqrt(NoiseCSIZE_H.rpc[T][S][p]->GetEntries());
            float cMultPart = NoiseCMult_H.rpc[T][S][p]->GetMean();
            float cMultPartErr = (NoiseCMult_H.rpc[T][S][p]->GetEntries()==0)
                ? 0.
                : 2*NoiseCMult_H.rpc[T][S][p]->GetStdDev() /
                    sqrt(NoiseCMult_H.rpc[T][S][p]->GetEntries());
            float ClustPartRate = (cSizePart==0) ? 0.
                : MeanPartRate/cSizePart;
            float ClustPartRateErr = (cSizePart==0) ? 0.
                : ClustPartRate * cSizePartErr/cSizePart;

            outputRateCSV << MeanPartRate << '\t'
                << cSizePart << '\t' << cSizePartErr << '\t'
                << cMultPart << '\t' << cMultPartErr << '\t'
                << ClustPartRate << '\t' << ClustPartRateErr << '\t';

            RPCarea += stripArea * nStripsPart;
            MeanNoiseRate += MeanPartRate * stripArea * nStripsPart;
            ClusterRate += ClustPartRate * stripArea * nStripsPart;
            ClusterSDev += (cSizePart==0)
                ? 0.
                : ClusterRate*cSizePartErr/cSizePart;
        }

        MeanNoiseRate /= RPCarea;
        ClusterRate /= RPCarea;
        ClusterSDev /= RPCarea;

        outputRateCSV << MeanNoiseRate << '\t'
            << ClusterRate << '\t' << ClusterSDev << '\t';
    }
}

```

2021 *Source Code B.19: Description of rate result calculation and writing into the CSV output Offline-Rate.csv. Are saved into the file for each detector, the mean partition rate, cluster size and cluster mutiplicity, along with their errors, for each partition and as well as a detector average.*

2022 **Muon performance variables** are computed and written in the output file for each detector partitions as shown through Sources Code B.20. The variables that are written for each partition are:

2023

- 2024     • The muon efficiency, `eff`, extracted from the histogram `Efficiency0_H`. It is reminded that  
2025     this offline tool doesn't include any tracking algorithm to identify muons from the beam and  
2026     only relies on the hits arriving in the time window corresponding to the beam time. The con-  
2027     tent of the efficiency histogram is thus biased by the noise/gamma background contribution  
2028     into this window and is thus corrected by estimating the muon data content in the peak re-  
2029     gion knowing the noise/gamma content in the rate calculation region. Both time windows  
2030     being different, the choice was made to normalise the noise/gamma background calculation  
2031     window to it's equivalent beam window in order to have comparable values using the variable  
2032     `windowRatio`. Finally, to estimate the data ratio in the peak region, the variable `DataRatio`  
2033     is defined as the ratio in between the estimated mean cluster multiplicity of the muons in the  
2034     peak region, `MuonCM`, and of the total mean cluster multiplicity in the peak region, `PeakCM`.  
2035     `MuonCM` is itself defined as the difference in between the total mean cluster multiplicity in the  
2036     peak region and the normalised mean noise/gamma cluster multiplicity calculated outside of  
2037     the peak region. The statistical error related to the efficiency, `eff_err`, is computed using a  
2038     binomial distribution, as the efficiency measure the probability of "success" and "failure" to  
2039     detect muons.
- 2040     • The mean muon cluster size, `MuonCS`, is calculated using the total mean cluster size and multi-  
2041     plicity in the peak region, respectively extracted from histograms `MuonCSize_H` and `MuonCMult_H`,  
2042     the noise/gamma background mean cluster size and normalised multiplicity, extracted from  
2043     `NoiseCSize_H` and `NoiseCMult_H`, and of the estimated muon cluster multiplicity `MuonCM` pre-  
2044     viously explicated. The associated statistical error, `MuonCM_err`, is calculated using the propa-  
2045     gation of errors of the mentioned variables.
- 2046     • The mean muon cluster multiplicity in the peak region, `MuonCM`, explicated above whose sta-  
2047     tistical error, `MuonCM_err`, is the sum of statistical error associated to the total mean clus-  
2048     ter multiplicity in the peak reagion, `PeakCM_err`, and of the mean noise/gamma cluster size,  
2049     `NoiseCM_err`.

2050        In addition to these 2 CSV files, the histograms are saved in ROOT file `Scan00XXXX_HVY_Offline.root`  
2051        as explained in section B.2.1.1.

2052

```

for (UInt tr = 0; tr < GIFInfra->GetNTrolleys(); tr++) {
    UInt T = GIFInfra->GetTrolleyID(tr);
    for (UInt sl = 0; sl < GIFInfra->GetNSlots(tr); sl++) {
        UInt S = GIFInfra->GetSlotID(tr,sl) - 1;
        for (UInt p = 0; p < GIFInfra->GetNPartitions(tr,sl); p++) {
            float noiseWindow =
                BMTDCWINDOW - TIMEREJECT - 2*PeakWidth.rpc[T][S][p];
            float peakWindow = 2*PeakWidth.rpc[T][S][p];
            float windowRatio = peakWindow/noiseWindow;

            float PeakCM = MuonCMult_H.rpc[T][S][p]->GetMean();
            float PeakCS = MuonCSize_H.rpc[T][S][p]->GetMean();
            float NoiseCM = NoiseCMult_H.rpc[T][S][p]->GetMean() *windowRatio;
            float NoiseCS = NoiseCSize_H.rpc[T][S][p]->GetMean();
            float MuonCM = (PeakCM<NoiseCM) ? 0. : PeakCM-NoiseCM;
            float MuonCS = (MuonCM==0 || PeakCM*PeakCS<NoiseCM*NoiseCS)
                ? 0.
                : (PeakCM*PeakCS-NoiseCM*NoiseCS) / MuonCM;
            float PeakCM_err = (MuonCMult_H.rpc[T][S][p]->GetEntries()==0.)
                ? 0.
                : 2*MuonCMult_H.rpc[T][S][p]->GetStdDev() /
                    sqrt(MuonCMult_H.rpc[T][S][p]->GetEntries());
            float PeakCS_err = (MuonCSize_H.rpc[T][S][p]->GetEntries()==0.)
                ? 0.
                : 2*MuonCSize_H.rpc[T][S][p]->GetStdDev() /
                    sqrt(MuonCSize_H.rpc[T][S][p]->GetEntries());
            float NoiseCM_err = (NoiseCMult_H.rpc[T][S][p]->GetEntries()==0.)
                ? 0.
                : windowRatio*2*NoiseCMult_H.rpc[T][S][p]->GetStdDev() /
                    sqrt(NoiseCMult_H.rpc[T][S][p]->GetEntries());
            float NoiseCS_err = (NoiseCSize_H.rpc[T][S][p]->GetEntries()==0.)
                ? 0.
                : 2*NoiseCSize_H.rpc[T][S][p]->GetStdDev() /
                    sqrt(NoiseCSize_H.rpc[T][S][p]->GetEntries());
            float MuonCM_err = (MuonCM==0) ? 0. : PeakCM_err+NoiseCM_err;
            float MuonCS_err = (MuonCS==0 || MuonCM==0) ? 0.
                : (PeakCS*PeakCM_err + PeakCM*PeakCS_err +
                    NoiseCS*NoiseCM_err + NoiseCM*NoiseCS_err +
                    MuonCS*MuonCM_err) / MuonCM;

            float DataRatio = MuonCM/PeakCM;
            float DataRatio_err = (MuonCM==0) ? 0.
                : DataRatio*(MuonCM_err/MuonCM + PeakCM_err/PeakCM);
            float eff = DataRatio*Efficiency0_H.rpc[T][S][p]->GetMean();
            float eff_err = DataRatio*2*Efficiency0_H.rpc[T][S][p]->GetStdDev() /
                sqrt(Efficiency0_H.rpc[T][S][p]->GetEntries()) +
                Efficiency0_H.rpc[T][S][p]->GetMean()*DataRatio_err;

            outputEffCSV << eff << '\t' << eff_err << '\t'
                << MuonCS << '\t' << MuonCS_err << '\t'
                << MuonCM << '\t' << MuonCM_err << '\t';
        }
    }
}

```

2053

2054

*Source Code B.20: Description of efficiency result calculation and writing into the CSV output Offline-L0-EffCl.csv. Are saved into the file for each detector, the efficiency, corrected taking into account the background in the peak window of the time profile, muon cluster size and muon cluster multiplicity, along with their errors, for each partition and as well as a detector average.*

## 2055 B.7 Current data Analysis

2056 Detectors under test at GIF++ are connected both to a CAEN HV power supply and to a CAEN  
2057 ADC that reads the currents inside of the RPC gaps bypassing the supply cable. During data tak-  
2058 ing, the webDCS records into a ROOT file called `Scan00XXXX_HVY_CAEN.root` histograms with the  
2059 monitored parameters of both CAEN devices. Are recorded for each RPC channels (in most cases,  
2060 a channel corresponds to an RPC gap):

- 2061 • the effective voltage,  $HV_{eff}$ , set by the webDCS using the PT correction on the CAEN power  
2062 supply,
- 2063 • the applied voltage,  $HV_{app}$ , monitored by the CAEN power supply, and the statistical error  
2064 related to the variations of this value through time to follow the variation of the environmental  
2065 parameters defined as the RMS of the histogram divided by the square root of the number of  
2066 recorded points,
- 2067 • the monitored current,  $I_{mon}$ , monitored by the CAEN power supply, and the statistical error  
2068 related to the variations of this value through time to follow the variation of the environmental  
2069 parameters defined as the RMS of the histogram divided by the square root of the number of  
2070 recorded points,
- 2071 • the corresponding current density,  $J_{mon}$ , defined as the monitored current per unit area,  
2072  $J_{mon} = I_{mon}/A$ , where  $A$  is the active area of the corresponding gap,
- 2073 • the ADC current,  $I_{ADC}$ , recorded through the CAEN ADC module that monitors the dark  
2074 current in the gap itself. First of all, the resolution of such a module is better than that of  
2075 CAEN power supplies and moreover, the current is not read-out through the HV supply line  
2076 but directly at the chamber level giving the real current inside of the detector. The statistical  
2077 error is defined as the RMS of the histogram distribution divided by the square root of the  
2078 number of recorded points.

2079 Once extracted through a loop over the element of GIF++ infrastructure via the C++ macro  
2080 `GetCurrent()`, these parameters, organised in 9 columns per detector HV supply line, are written in  
2081 the output CSV file `Offline-Current.csv`. The macro can be found in the file `Current.cc`.

## References

- 2083 [1] T. Massam et al. “Experimental observation of antideuteron production”. In: *Il Nuovo Cimento A* 63 (1965), pp. 10–14.
- 2084
- 2085 [2] UA1 Collaboration. “Experimental observation of isolated large transverse energy electrons with associated missing energy at  $s = 540 \text{ GeV}$ ”. In: *Physics Letters B* 122 (1983), pp. 103–116.
- 2086
- 2087
- 2088 [3] UA2 Collaboration. “Observation of single isolated electrons of high transverse momentum in events with missing transverse energy at the CERN pp collider”. In: *Physics Letters B* 122 (1983), pp. 476–485.
- 2089
- 2090
- 2091 [4] UA1 Collaboration. “Experimental observation of lepton pairs of invariant mass around  $95 \text{ GeV}/c^2$  at the CERN SPS collider”. In: *Physics Letters B* 126 (1983), pp. 398–410.
- 2092
- 2093 [5] UA2 Collaboration. “Evidence for  $Z_0 \rightarrow e^+e^-$  at the CERN pp collider”. In: *Physics Letters B* 129 (1983), pp. 130–140.
- 2094
- 2095 [6] ALEPH Collaboration. “Determination of the number of light neutrino species”. In: *Physics Letters B* 231 (1989), pp. 519–529.
- 2096
- 2097 [7] CERN, ed. (1985).
- 2098 [8] CERN, ed. (1986).
- 2099 [9] CERN, ed. (1994).
- 2100 [10] CERN, ed. (1998).
- 2101 [11] CERN, ed. (1999).
- 2102 [12] CERN. Geneva. LHC Experiments Committee. *Letter of Intent for A Large Ion Collider Experiment [ALICE]*, note = CERN-LHCC-93-016. Tech. rep. ALICE Collaboration, 1993.
- 2103
- 2104 [13] CERN. Geneva. LHC Experiments Committee. *ATLAS : technical proposal for a general-purpose pp experiment at the Large Hadron Collider at CERN*, note = CERN-LHCC-94-43. Tech. rep. ATLAS Collaboration, 1994.
- 2105
- 2106
- 2107 [14] CERN. Geneva. LHC Experiments Committee. *CMS : letter of intent by the CMS Collaboration for a general purpose detector at LHC*, note = CERN-LHCC-92-003. Tech. rep. CMS Collaboration, 1992.
- 2108
- 2109
- 2110 [15] CERN. Geneva. LHC Experiments Committee. *LHCb : letter of intent*. Tech. rep. CERN-LHCC-95-5. LHCb Collaboration, 1995.
- 2111
- 2112 [16] L. Evans and P. Bryant. “LHC Machine”. In: *JINST* 3 (2008). S08001.
- 2113
- 2114 [17] CMS Collaboration ATLAS Collaboration. “Combined Measurement of the Higgs Boson Mass in  $pp$  Collisions at  $\sqrt{s} = 7$  and  $8 \text{ TeV}$  with the ATLAS and CMS Experiments”. In: *Physical Review Letters* 114 (2015). 191803.
- 2115
- 2116 [18] LHCb Collaboration. “Observation of  $J/\psi p$  Resonances Consistent with Pentaquark States in  $\Lambda_b^0 \rightarrow J/\psi K^- p$  Decays”. In: *Physical Review Letters* 115 (2015). 072001.
- 2117

- 2118 [19] LHCb Collaboration. “Observation of  $J/\psi\phi$  Structures Consistent with Exotic States from  
2119 Amplitude Analysis of  $B^+ \rightarrow J/\psi\phi K^+$  Decays”. In: *Physical Review Letters* 118 (2017).  
2120 022003.
- 2121 [20] CERN. Geneva. *High-Luminosity Large Hadron Collider (HL-LHC) Technical Design Re-*  
2122 *port V. 0.1*. Tech. rep. CERN-2017-007-M. 2017.
- 2123 [21] CERN. Geneva. LHC Experiments Committee. *The CMS muon project : Technical Design*  
2124 *Report*. Tech. rep. CERN-LHCC-97-032. CMS Collaboration, 1997.
- 2125 [22] CERN. Geneva. LHC Experiments Committee. *Technical Proposal for the Phase-II Upgrade*  
2126 *of the CMS Detector*. Tech. rep. CERN-LHCC-2015-010. CMS Collaboration, 2015.
- 2127 [23] CERN. Geneva. LHC Experiments Committee. *CMS, the Compact Muon Solenoid : technical*  
2128 *proposal*. Tech. rep. CERN-LHCC-94-38. CMS Collaboration, 1994.
- 2129 [24] R. Santonico and R. Cardarelli. “Development of resistive plate counters”. In: *Nucl. Instr.*  
2130 *Meth. Phys. Res.* 187 (1981), pp. 377–380.
- 2131 [25] Yu.N. Pestov and G.V. Fedotovich. *A picosecond time-of-flight spectrometer for the VEPP-2M*  
2132 *based on local-discharge spark counter*. Tech. rep. SLAC-TRANS-0184. SLAC, 1978.
- 2133 [26] W.W. Ash, ed. *Spark Counter With A Localized Discharge*. Vol. SLAC-R-250. 1982, pp. 127–  
2134 131.
- 2135 [27] I. Crotty et al. “The non-spark mode and high rate operation of resistive parallel plate cham-  
2136 bers”. In: *NIMA* 337 (1993), pp. 370–381.
- 2137 [28] I. Crotty et al. “Further studies of avalanche mode operation of resistive parallel plate cham-  
2138 bers”. In: *NIMA* 346 (1994), pp. 107–113.
- 2139 [29] R. Cardarelli et al. “Avalanche and streamer mode operation of resistive plate chambers”. In:  
2140 *NIMA* 382 (1996), pp. 470–474.
- 2141 [30] E. Cerron Zeballos et al. “A new type of resistive plate chamber: The multigap RPC”. In:  
2142 *NIMA* 374 (1996), pp. 132–135.
- 2143 [31] M.C.S. Williams. “The development of the multigap resistive plate chamber”. In: *Nucl. Phys.*  
2144 *B* 61 (1998), pp. 250–257.
- 2145 [32] H. Czyrkowski et al. “New developments on resistive plate chambers for high rate operation”.  
2146 In: *NIMA* 419 (1998), pp. 490–496.
- 2147 [33] P. Camarri et al. “Streamer suppression with SF6 in RPCs operated in avalanche mode”. In:  
2148 *NIMA* 414 (1998), pp. 317–324.
- 2149 [34] E. Cerron Zeballos et al. “Effect of adding SF6 to the gas mixture in a multigap resistive plate  
2150 chamber”. In: *NIMA* 419 (1998), pp. 475–478.
- 2151 [35] CERN. Geneva. LHC Experiments Committee. *ATLAS muon spectrometer: Technical design*  
2152 *report*. Tech. rep. CERN-LHCC-97-22. ATLAS Collaboration, 1997.
- 2153 [36] CERN. Geneva. LHC Experiments Committee. *ALICE Time-Of-Flight system (TOF) : Tech-*  
2154 *nical Design Report*. Tech. rep. CERN-LHCC-2000-012. ALICE Collaboration, 2000.
- 2155 [37] The CALICE collaboration. “First results of the CALICE SDHCAL technological proto-  
2156 type”. In: *JINST* 11 (2016).
- 2157 [38] PoS, ed. *Density Imaging of Volcanoes with Atmospheric Muons using GRPCs*. International  
2158 Europhysics Conference on High Energy Physics - HEP 2011. 2011.
- 2159 [39] C. Lippmann. “Detector Physics of Resistive Plate Chambers”. PhD thesis. Johann Wolfgang  
2160 Goethe-Universität, 2003.

## BIBLIOGRAPHY

---

- 2161 [40] M. Abbrescia et al. “Properties of C<sub>2</sub>H<sub>2</sub>F<sub>4</sub>-based gas mixture for avalanche mode operation  
2162 of resistive plate chambers”. In: *NIMA* 398 (1997), pp. 173–179.
- 2163 [41] G.Battistoni et al. “Sensitivity of streamer mode to single ionization electrons”. In: *NIMA*  
2164 235 (1985), pp. 91–97.
- 2165 [42] W. Riegler. “Induced signals in resistive plate chambers”. In: *NIMA* 491 (2002), pp. 258–271.
- 2166 [43] E. Cerron Zeballos et al. “A comparison of the wide gap and narrow gap resistive plate  
2167 chamber”. In: *NIMA* 373 (1996), pp. 35–42.
- 2168 [44] M. Abbrescia et al. “Cosmic ray tests of double-gap resistive plate chambers for the CMS  
2169 experiment”. In: *NIMA* 550 (2005), pp. 116–126.
- 2170 [45] ALICE Collaboration. “A study of the multigap RPC at the gamma irradiation facility at  
2171 CERN”. In: *NIMA* 490 (2002), pp. 58–70.
- 2172 [46] B. Bonner et al. “A multigap resistive plate chamber prototype for time-of-flight for the STAR  
2173 experiment at RHIC”. In: *NIMA* 478 (2002), pp. 176–179.
- 2174 [47] S. Yang et al. “Test of high time resolution MRPC with different readout modes for the  
2175 BESIII upgrade”. In: *NIMA* 763 (2014), pp. 190–196.
- 2176 [48] A. Akindinov et al. “RPC with low-resistive phosphate glass electrodes as a candidate for  
2177 the CBM TOF”. In: *NIMA* 572 (2007), pp. 676–681.
- 2178 [49] JINST, ed. *Development of the MRPC for the TOF system of the MultiPurpose Detector.*  
2179 RPC2016: XII Workshop on Resistive Plate Chambers and Related Detectors. 2016.
- 2180 [50] M.C.S. Williams. “Particle identification using time of flight”. In: *Journal of Physics G* 39  
2181 (2012).
- 2182 [51] A. Alici et al. “Aging and rate effects of the Multigap RPC studied at the Gamma Irradiation  
2183 Facility at CERN”. In: *NIMA* 579 (2007), pp. 979–988.
- 2184 [52] M. Abbrescia et al. “The simulation of resistive plate chambers in avalanche mode: charge  
2185 spectra and efficiency”. In: *NIMA* 431 (1999), pp. 413–427.
- 2186 [53] M. Abbrescia et al. “Study of long-term performance of CMS RPC under irradiation at the  
2187 CERN GIF”. In: *NIMA* 533 (2004), pp. 102–106.
- 2188 [54] H.C. Kim et al. “Quantitative aging study with intense irradiation tests for the CMS forward  
2189 RPCs”. In: *NIMA* 602 (2009), pp. 771–774.
- 2190 [55] S. Agosteo et al. “A facility for the test of large-area muon chambers at high rates”. In: *NIMA*  
2191 452 (2000), pp. 94–104.
- 2192 [56] PoS, ed. *CERN GIF ++ : A new irradiation facility to test large-area particle detectors for*  
2193 *the high-luminosity LHC program.* Vol. TIPP2014. 2014, pp. 102–109.
- 2194 [57] A. Fagot. *GIF++ DAQ v4.0*. 2017. URL: [https://github.com/afagot/GIF\\_DAQ](https://github.com/afagot/GIF_DAQ).
- 2195 [58] CAEN. *Mod. V1190-VX1190 A/B, 128/64 Ch Multihit TDC*. 14th ed. 2016.
- 2196 [59] CAEN. *Mod. V1718 VME USB Bridge*. 9th ed. 2009.
- 2197 [60] W-Ie-Ne-R. *VME 6021-23 VXI*. 5th ed. 2016.
- 2198 [61] Wikipedia. *INI file*. 2017. URL: [https://en.wikipedia.org/wiki/INI\\_file](https://en.wikipedia.org/wiki/INI_file).
- 2199 [62] S. Carrillo A. Fagot. *GIF++ Offline Analysis v6*. 2017. URL: [https://github.com/afagot/GIF\\_OfflineAnalysis](https://github.com/afagot/GIF_OfflineAnalysis).