



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





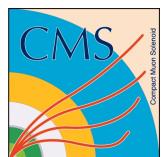


Universiteit Gent  
Faculteit Wetenschappen  
Vakgroep Fysica en Sterrenkunde

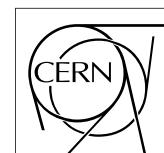
Promotoren: Dr. Michael Tytgat  
Prof. Dr. Dirk Ryckbosch

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

23

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 and the Compact Muon Solenoid . . . . .	2-1
33	2.2.1 LHC, the most powerful particle accelerator . . . . .	2-1
34	2.2.2 CMS, a multipurpose experiment . . . . .	2-2
35	2.3 Muon Phase-II Upgrade . . . . .	2-2
36	<b>3 Physics of Resistive plate chambers</b>	3-1
37	3.1 Principle . . . . .	3-1
38	3.1.1 Electron drift velocity . . . . .	3-4
39	3.2 Rate capability and time resolution of Resistive Plate Chambers . . . . .	3-4
40	3.2.1 Operation modes . . . . .	3-4
41	3.2.2 Detector designs and performance . . . . .	3-6
42	3.2.2.1 Double-gap RPC . . . . .	3-6
43	3.2.2.2 Multigap RPC (MRPC) . . . . .	3-7
44	3.2.2.3 Charge distribution and performance limitations . . . . .	3-8
45	3.3 Signal formation . . . . .	3-11
46	3.4 Gas transport parameters . . . . .	3-11
47	<b>4 Longevity studies and Consolidation of the present CMS RPC subsystem</b>	4-1
48	4.1 Resistive Plate Chambers at CMS . . . . .	4-1
49	4.1.1 Overview . . . . .	4-1
50	4.1.2 The present RPC system . . . . .	4-2
51	4.1.3 Pulse processing of CMS RPCs . . . . .	4-3
52	4.2 Testing detectors under extreme conditions . . . . .	4-3
53	4.2.1 The Gamma Irradiation Facilities . . . . .	4-6
54	4.2.1.1 GIF . . . . .	4-6
55	4.2.1.2 GIF++ . . . . .	4-8
56	4.3 Preliminary tests at GIF . . . . .	4-10
57	4.3.1 Resistive Plate Chamber test setup . . . . .	4-10
58	4.3.2 Data Acquisition . . . . .	4-12

---

59	4.3.3	Geometrical acceptance of the setup layout to cosmic muons . . . . .	4-12
60	4.3.3.1	Description of the simulation layout . . . . .	4-13
61	4.3.3.2	Simulation procedure . . . . .	4-15
62	4.3.3.3	Results . . . . .	4-16
63	4.3.4	Photon flux at GIF . . . . .	4-16
64	4.3.4.1	Expectations from simulations . . . . .	4-16
65	4.3.4.2	Dose measurements . . . . .	4-21
66	4.3.5	Results and discussions . . . . .	4-22
67	4.4	Longevity tests at GIF++ . . . . .	4-23
68	4.4.1	Description of the Data Acquisition . . . . .	4-26
69	4.4.2	RPC current, environmental and operation parameter monitoring . . . . .	4-27
70	4.4.3	Measurement procedure . . . . .	4-28
71	4.4.4	Longevity studies results . . . . .	4-28
72	<b>5</b>	<b>Investigation on high rate RPCs</b>	<b>5-1</b>
73	5.1	Rate limitations and ageing of RPCs . . . . .	5-1
74	5.1.1	Low resistivity electrodes . . . . .	5-1
75	5.1.2	Low noise front-end electronics . . . . .	5-1
76	5.2	Construction of prototypes . . . . .	5-1
77	5.3	Results and discussions . . . . .	5-1
78	<b>6</b>	<b>Conclusions and outlooks</b>	<b>6-1</b>
79	6.1	Conclusions . . . . .	6-1
80	6.2	Outlooks . . . . .	6-1
81	<b>A</b>	<b>A data acquisition software for CAEN VME TDCs</b>	<b>A-1</b>
82	A.1	GIF++ DAQ file tree . . . . .	A-1
83	A.2	Usage of the DAQ . . . . .	A-2
84	A.3	Description of the readout setup . . . . .	A-3
85	A.4	Data read-out . . . . .	A-3
86	A.4.1	V1190A TDCs . . . . .	A-4
87	A.4.2	DataReader . . . . .	A-6
88	A.4.3	Data quality flag . . . . .	A-10
89	A.5	Communications . . . . .	A-12
90	A.5.1	V1718 USB Bridge . . . . .	A-13
91	A.5.2	Configuration file . . . . .	A-13
92	A.5.3	WebDCS/DAQ intercommunication . . . . .	A-17
93	A.5.4	Example of inter-process communication cycle . . . . .	A-18
94	A.6	Software export . . . . .	A-18
95	<b>B</b>	<b>Details on the offline analysis package</b>	<b>B-1</b>
96	B.1	GIF++ Offline Analysis file tree . . . . .	B-1
97	B.2	Usage of the Offline Analysis . . . . .	B-2
98	B.2.1	Output of the offline tool . . . . .	B-3
99	B.2.1.1	ROOT file . . . . .	B-3
100	B.2.1.2	CSV files . . . . .	B-5
101	B.3	Analysis inputs and information handling . . . . .	B-6
102	B.3.1	Dimensions file and IniFile parser . . . . .	B-6
103	B.3.2	TDC to RPC link file and Mapping . . . . .	B-7
104	B.4	Description of GIF++ setup within the Offline Analysis tool . . . . .	B-9

---

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



<sup>121</sup>

<sup>122</sup>

## Nederlandse samenvatting –Summary in Dutch–

<sup>123</sup> Le resume en Neerlandais (j'aurais peut-être pu apprendre la langue juste pour ça...).



## English summary

<sup>125</sup> Le meme résume mais en Anglais (on commencera par la hein!).



# 1

## Introduction

126

127

<sup>128</sup> **1.1 A story of High Energy Physics**

<sup>129</sup> **1.2 Organisation of this study**



# 2

130

131

## Investigating the TeV scale

### 132 2.1 The Standard Model of Particle Physics

### 133 2.2 The Large Hadron Collider and the Compact Muon Solenoid

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

#### 147 2.2.1 LHC, the most powerful particle accelerator

148 Along the path of the LHC beams, 4 main experiments have been built in order to investigate new  
149 physics.

- 150 • ALICE
- 151 • ATLAS
- 152 • CMS

153 • LHCb

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

CERN's Accelerator Complex

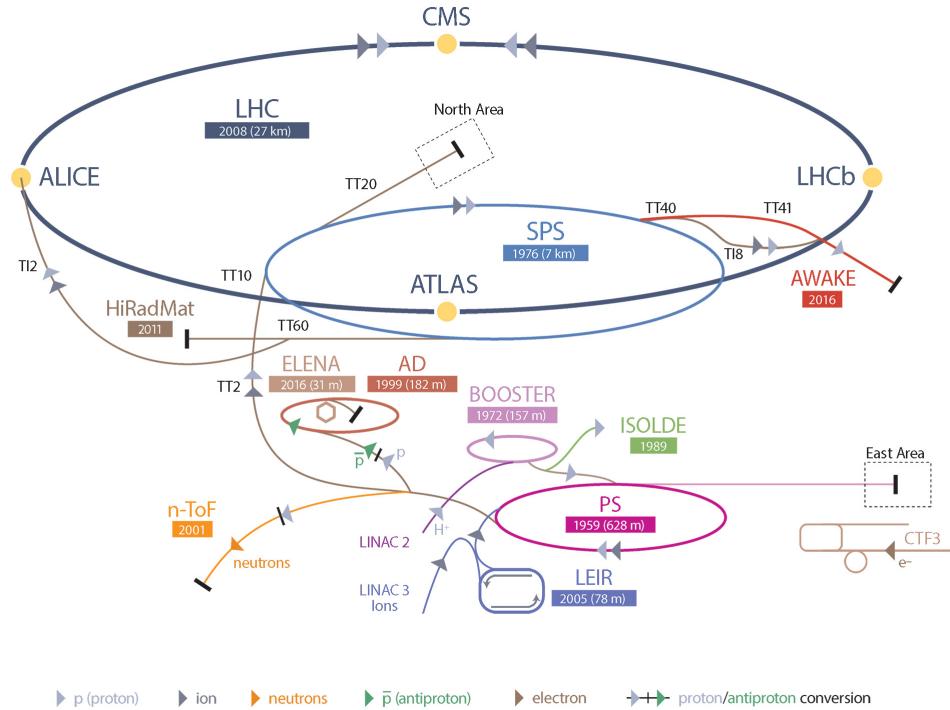


Figure 2.1: CERN accelerator complex.

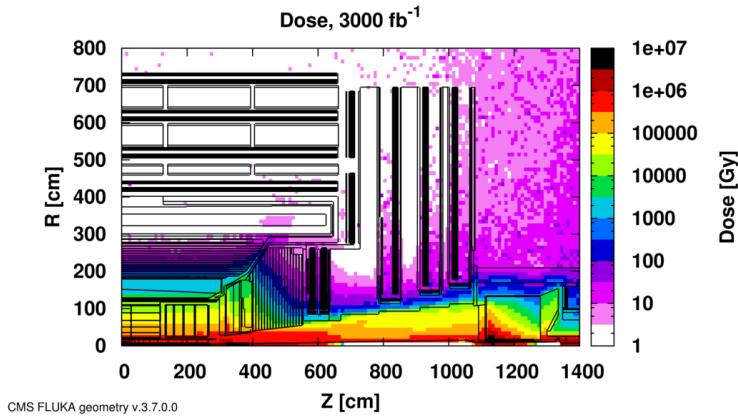
## 156 2.2.2 CMS, a multipurpose experiment

## 157 2.3 Muon Phase-II Upgrade

158 After the more than two years lasting First Long Shutdown (LS1), the Large Hadron Collider (LHC)  
 159 delivered its very first Run-II proton-proton collisions early 2015. LS1 gave the opportunity to the  
 160 LHC and to the its experiments to undergo upgrades. The accelerator is now providing collisions  
 161 at center-of-mass energy of 13 TeV and bunch crossing rate of 40 MHz, with a peak luminosity  
 162 exceeding its design value. During the first and upcoming second LHC Long Shutdown, the Compact  
 163 Muon Solenoid (CMS) detector is also undergoing a number of upgrades to maintain a high system  
 164 performance [7].

165 From the LHC Phase-2 or High Luminosity LHC (HL-LHC) period onwards, i.e. past the Third  
 166 Long Shutdown (LS3), the performance degradation due to integrated radiation as well as the average  
 167 number of inelastic collisions per bunch crossing, or pileup, will rise substantially and become a  
 168 major challenge for the LHC experiments, like CMS that are forced to address an upgrade program

for Phase-II [8]. Simulations of the expected distribution of absorbed dose in the CMS detector under HL-LHC conditions, show in figure 4.16 that detectors placed close to the beamline will have to withstand high irradiation, the radiation dose being of the order of a few tens of Gy.



*Figure 2.2: 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.*

The measurement of small production cross-section and/or decay branching ratio processes, such as the Higgs boson coupling to charge leptons or the  $B_s \rightarrow \mu^+\mu^-$  decay, is of major interest and specific upgrades in the forward regions of the detector will be required to maximize the physics acceptance on the largest possible solid angle. To ensure proper trigger performance within the present coverage, the muon system will be completed with new chambers. In figure 2.3 one can see that the existing Cathode Strip Chambers (CSCs) will be completed by Gas Electron Multipliers (GEMs) and Resistive Plate Chambers (RPCs) in the pseudo-rapidity region  $1.6 < |\eta| < 2.4$  to complete its redundancy as originally scheduled in the CMS Technical Proposal [9].

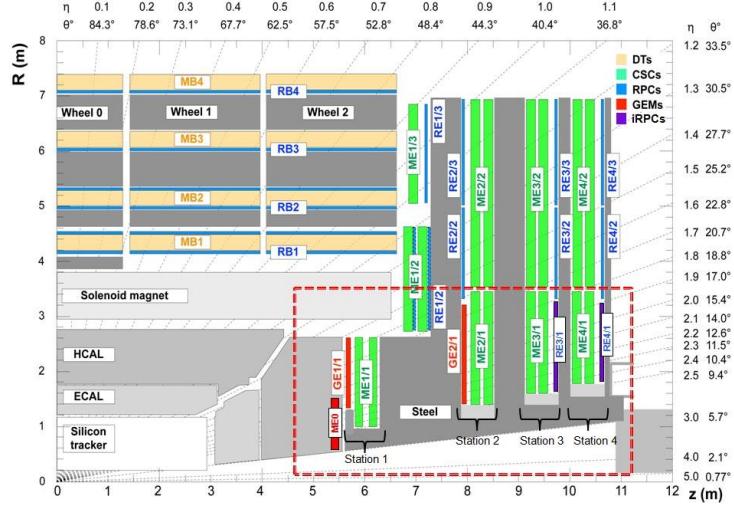
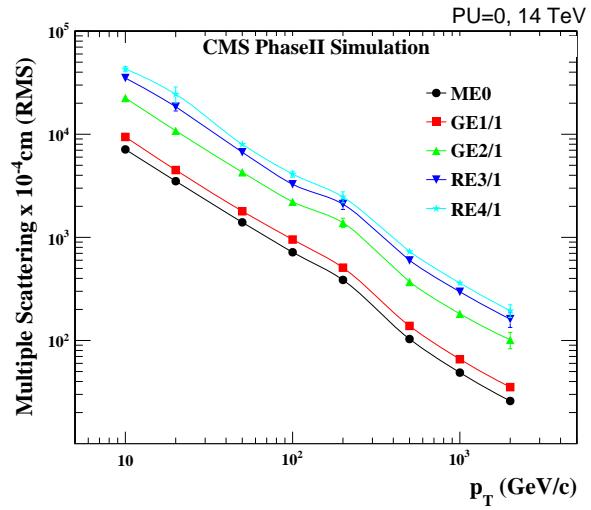


Figure 2.3: A quadrant of the muon system, showing DTs (yellow), RPCs (light blue), and CSCs (green). The locations of new forward muon detectors for Phase-II are contained within the dashed box and indicated in red for GEM stations (ME0, GE1/1, and GE2/1) and dark blue for improved RPC (iRPC) stations (RE3/1 and RE4/1).

180        RPCs are used by the CMS first level trigger for their good timing performances. Indeed, a very  
 181        good bunch crossing identification can be obtained with the present CMS RPC system, given their  
 182        fast response of the order of 1 ns. In order to contribute to the precision of muon momentum mea-  
 183        surements, muon chambers should have a spatial resolution less or comparable to the contribution  
 184        of multiple scattering [7]. Most of the plausible physics is covered only considering muons with  
 185         $p_T < 100$  GeV thus, in order to match CMS requirements, a spatial resolution of  $\mathcal{O}(\text{few mm})$  the  
 186        proposed new RPC stations, as shown by the simulation in figure 2.4. According to preliminary  
 187        designs, RE3/1 and RE4/1 readout pitch will be comprised between 3 and 6 mm and 5  $\eta$ -partitions  
 188        could be considered.



*Figure 2.4: 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

189

190

## Physics of Resistive plate chambers

191 A Resistive Plate Chamber (RPC) is a gaseous detector using the same physical processes described  
192 in Chapter 3. It has been developed in 1981 by Santonico and Cardarelli [10], under the name of  
193 *Resistive Plate Counter*, as an alternative to the local-discharge spark counters proposed in 1978  
194 by Pestov and Fedotovich [11, 12]. Working with spark chambers implied using high-pressure gas  
195 and high mechanical precision which the RPC simplified by formerly using a gas mixture of argon  
196 and butane flowed at atmospheric pressure and a constant and uniform electric field propagated  
197 in between two parallel electrode plates. Moreover, a significant increase in rate capability was  
198 introduced by the use of electrode plate material with high bulk resistivity, preventing the discharge  
199 from growing throughout the whole gas gap. Indeed, the effect of using resistive electrodes is that  
200 the constant electric field is locally canceled out by the development of the discharge, limiting its  
201 growth.

202 Through its development history, different operating modes [13–15] and new detector designs [16–  
203 18] have been discovered, leading to further improvement of the rate capability of such a detector.  
204 Moreover, the addition of  $SF_6$  into the gas mix improved the stability of operation of the RPC [19,  
205 20].

206 The low developing costs and easily achievable large detection areas offered by RPCs, as well  
207 as the wide range of possible designs, made them a natural choice to as muon chambers and/or  
208 trigger detectors in multipurpose experiments such as CMS [7] or ATLAS [21], time-of-flight detec-  
209 tors in ALICE [22], calorimeter with CALICE [23] or even detectors for volcanic muography with  
210 ToMuVol [24].

### 211 3.1 Principle

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

216 the electrodes by the electric field, as shown in Figure 3.1 [25].

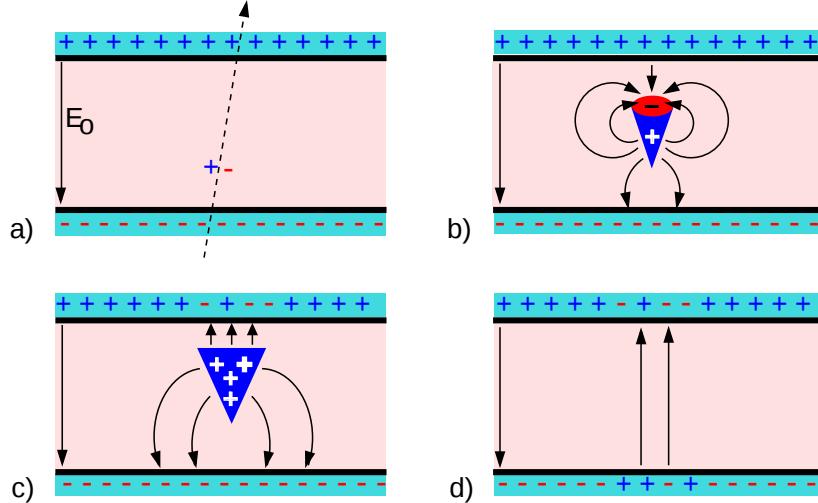


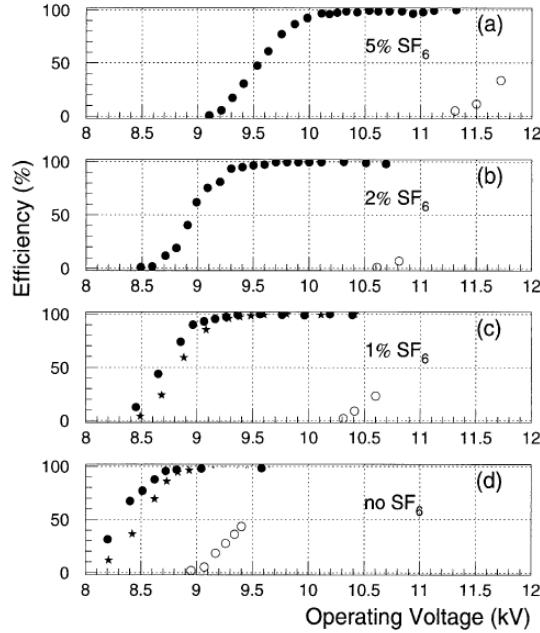
Figure 3.1: Different phases of the avalanche development in the RPC gas volume subjected to a constant electric field  $E_0$ . a) An avalanche is initiated by the primary ionisation caused by the passage of a charged particle through the gas volume. b) Due to its growing size, the avalanche starts to 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.

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

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

- 225 • Tetrafluoroethane ( $C_2F_4H_2$ ), also referred to as *Freon*, is the principal compound of the RPC  
 226 gas mixtures, with a typical fraction above 90%. It is used for it's high effective Townsend  
 227 coefficient and the great average fast charge that allows to operate the detector with a high  
 228 threshold with respect to argon, for example, that has similar effective Townsend coefficient  
 229 but suffers from a lower fast charge. To operate with similar conditions, argon would require a  
 230 higher electric field leading to a higher fraction of streamers, thus limiting the rate capability  
 231 of the detector [26].
- 232 • Isobutane (i- $C_4H_{10}$ ), only present in a few percent in the gas mixtures, is used for its UV  
 233 quenching properties [27] helping to prevent streamers due to UV photon emission during the  
 234 avalanche growth.
- 235 • Sulfur hexafluoride, ( $SF_6$ ), referred to simply as *SF6*, is used in very little quantities for its  
 236 high electronegativity. Excess of electrons are being absorbed by the compound and streamers

are suppressed [20]. 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 [19], 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  $SF_6$  [19].*

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 [28].

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

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,

253 the developing cost of such materials is quite high due to the very low demand. Thus, High-pressure  
 254 laminate (HPL) is often the choice for high rate experiments using very large RPC detection areas.  
 255 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.

### 256 3.1.1 Electron drift velocity

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

## 258 3.2 Rate capability and time resolution of Resistive Plate Cham- 259 bers

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

### 264 3.2.1 Operation modes

265 RPCs where developed early 1980s. At that time it was using an operating mode now referred to  
 266 as *streamer mode*. Streamers are large discharges that develop in between the 2 electrodes enough  
 267 to locally discharge the electrodes. If the electric field inside of the gas volume is strong enough,  
 268 with electrons being fast compared to ions, a large and dense cloud of positive ions will develop  
 269 nearby the anode and extend toward the cathode while the electrons are being collected, eventually  
 270 leading to a streamer discharge due to the increase of field seen at the cathode. the field is then strong  
 271 enough so that electrons are pulled out of the cathode. Electrodes, though they are a unique volume  
 272 of resistive material, can be assimilated to capacitors. At the moment an electric field is applied in  
 273 between their outer surfaces, the charge carriers inside of the volume will start moving leading to  
 274 a situation where there is no voltage across the electrodes and a higher density of negative charges,  
 275 i.e. electrons, on the inner surface of the cathode. Finally, when a streamer discharge occurs, these  
 276 electrons are partially released in the gas volume contributing to increase the discharge strength until  
 277 the formation of a conductive plasma, the streamer. This can be understood through Figure 3.3 [13].  
 278 Streamer signals are very convenient in terms of read-out as no amplification is required with output  
 279 pulses amplitudes of the order of a few tens to few hundreds of mV as can be seen on Figure 3.4.

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

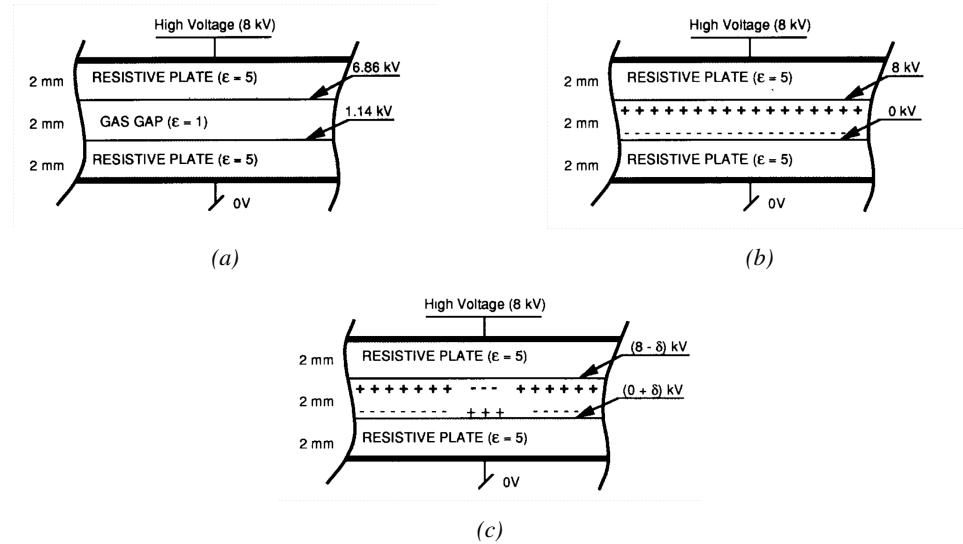


Figure 3.3: Movement of the charge carriers in an RPC. Figure 3.3a: Voltage across an RPC whose electrode have a relative permittivity of 5 at the moment the tension is applied. Figure 3.3b: After the charge carriers moved, the electrodes are charged and there is no voltage drop over the electrodes anymore. 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.

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

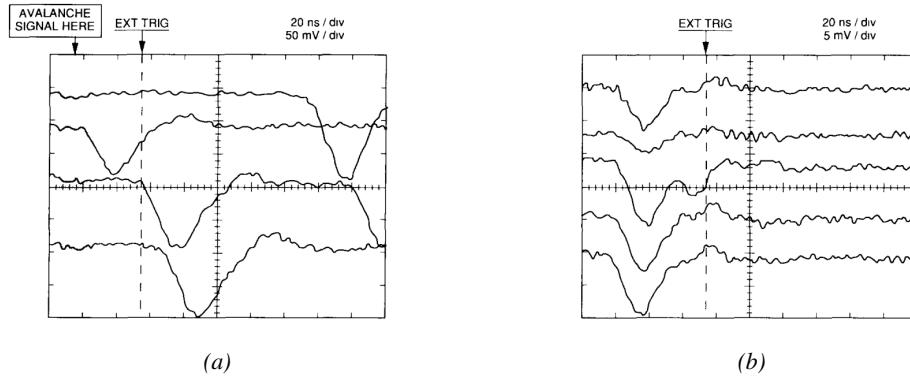


Figure 3.4: Typical oscilloscope pulses in streamer mode (Figure 3.4a) and avalanche mode (Figure 3.4b). In the case of streamer mode, the very small avalanche signal is visible.

285 This mode offers a higher rate capability by providing smaller discharges that don't affect the  
 286 electrodes charge and are more locally contained in the gas volume as was demonstrated by Crotty  
 287 with Figure 3.5 [13]. The detector only stays locally blind the time the charge carriers are recom-  
 288 bined and there is no need for electrode recharge which is a long process affecting a large portion  
 289 of the detector. Another advantage of avalanche signals over streamer is the great time consistency.  
 290 Figure 3.4 shows very clearly that avalanche signals have a very small time jitter. This is a natural  
 291 choice for high rate experiments.

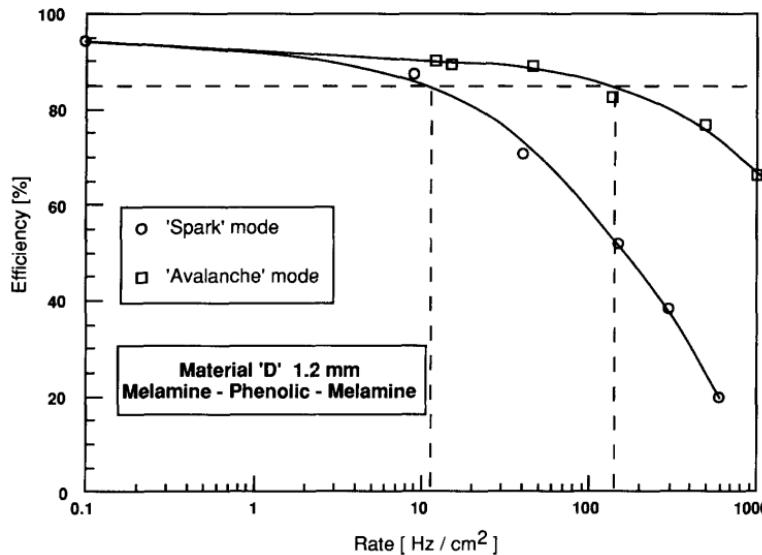


Figure 3.5: Rate capability comparison for the streamer and avalanche mode of operation. An order of magnitude in rate capability for a maximal efficiency drop of 10% is gained by using the avalanche mode over the streamer mode.

### 292 3.2.2 Detector designs and performance

293 Different RPC design have been used and each of them present its own advantages. Historically,  
 294 the first type of RPC to have been developed is what is referred now to *no narrow gap* RPC [10, 29].  
 295 After the avalanche mode has been discovered [13], it has been proven that increasing the width  
 296 of the gas gap lead to higher rate capability, due to lower charge deposition per avalanche, and  
 297 lower power dissipation [29]. Nevertheless, by increasing the gas gap width, the time resolution of  
 298 the detector decreases. This is a natural result if the increase of active gas volume in the detector is  
 299 taken into account. Indeed, for a given threshold, only the small fraction of gas closest to the cathode  
 300 will provide enough gain to have a detectable signal. In the case of a wider gas volume, the active  
 301 region is then larger and a larger time jitter is introduced with the variation of starting position of the  
 302 avalanche, as discussed in [16]. To solve improve both the time resolution and the rate capability,  
 303 different methods were used trying to get advantages of both narrow and wide gap RPCs.

#### 304 3.2.2.1 Double-gap RPC

305 Double-gap RPCs are made out of 2 narrow RPC detectors. The 2 RPC gaps are stacked on top of  
 306 each other as shown in Figure 3.6. This detector layout, popularized by the two multipurpose experi-  
 307 ments CMS [7] and ATLAS [21] at LHC, can be used as an OR system in which each individual  
 308 chamber participates in the output signal. The gain of such a detector is greatly reduced with respect  
 309 to single-gap RPCs with an efficiency plateau reached at lower voltage, as visible on Figure 3.7.

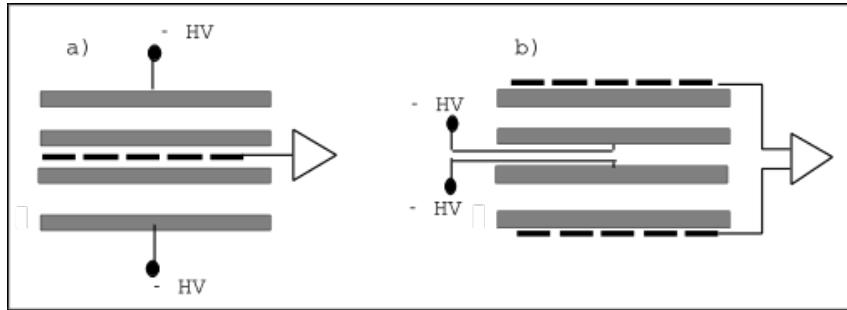


Figure 3.6: Possible double-gap RPC layouts: a) "standard" 1D double-gap RPC, as used in CMS experiment, where the anodes are facing each other and a 1D read-out plane is sandwiched in between them, b) double read-out double-gap RPC as used in ATLAS experiment, where the cathodes are facing each other 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.

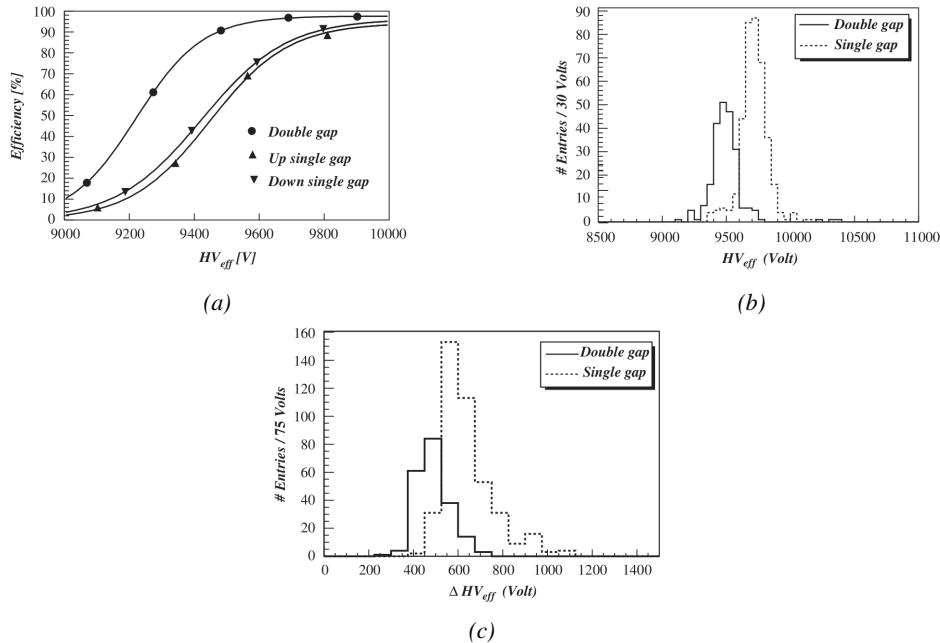
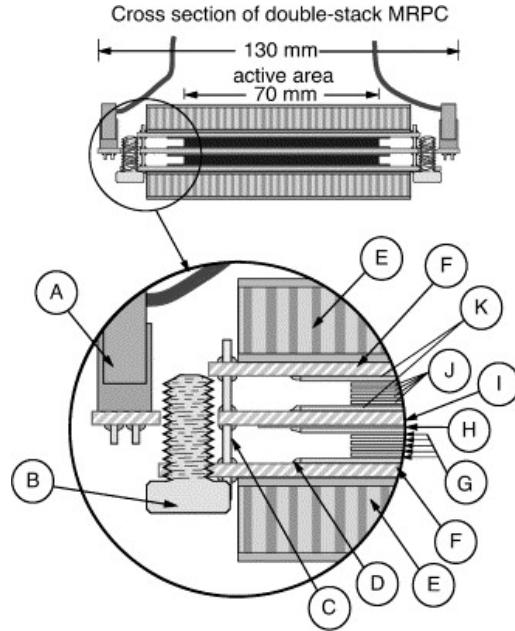


Figure 3.7: Comparison of performance of CMS double and single gap RPCs using cosmic muons [30]. Figure 3.7a: Comparison of efficiency sigmoids. Figure 3.7b: Voltage distribution at 95% of maximum efficiency. Figure 3.7c:  $\Delta_{10\%}^{90\%}$  distribution.

### 3.2.2.2 Multigap RPC (MRPC)

MRPCs are layouts in which floating sub electrode plates are placed into a wide gap RPC to divide the gas volume and create a sum of narrow gaps [16, 17]. The time resolution of such a detector can 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 representing ALICE Time-of-flight (ToF) MRPCs.

Sometimes used as a double multigap RPC, taking advantage of the OR of double gap RPCs, the MRPC is mainly used as ToF detector [31–35] 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 [31].*

317 to perform particle identification as explained by Williams in [36]. The principle of particle iden-  
 318 tification using ToF is simply the measurement of the velocity of a particle. Indeed, particles are  
 319 defined by their mass (for the parameter of interest here, their electric charge being measured using  
 320 the bending angle of the particles traveling through a magnetic field) and this mass can be calculated  
 321 by measuring the velocity  $\beta$  and momentum of the particle:

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

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

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

326 An example of particle identification is given for the case of STAR experiment in Figure 3.9.  
 327 Another benefit of using such small gas gaps is the strong reduction of the average avalanche  
 328 volume and thus of the blind spot on MRPCs leading to an improved rate capability. Multigaps can  
 329 sustain backgrounds of several kHz/cm<sup>2</sup> as demonstrated in Figure 3.10.

### 330 3.2.2.3 Charge distribution and performance limitations

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

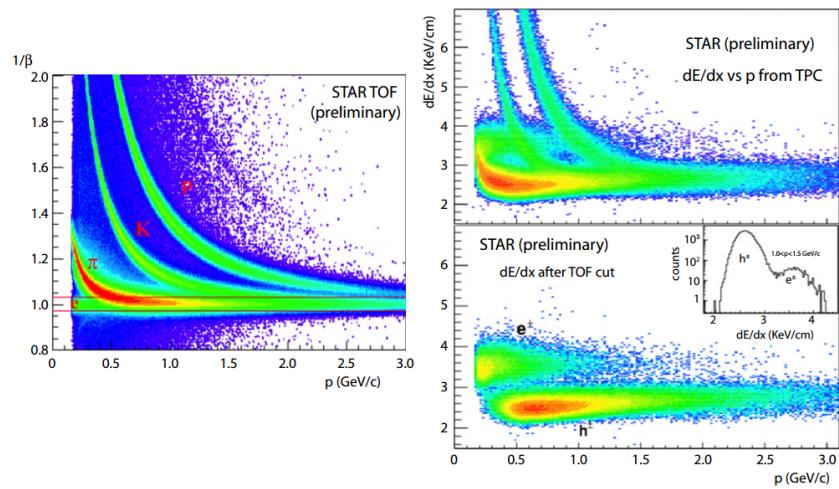


Figure 3.9: Particle identification applied to electrons in the STAR experiment. The identification is performed combining ToF and  $dE/dx$  measurements [36].

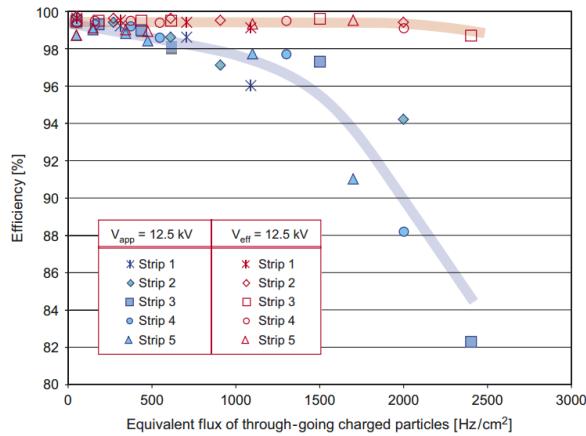


Figure 3.10: Comparison of the detector performance of ALICE ToF MRPC [37] at fixed applied voltage (in blue) and at fixed effective voltage (in red). The effective voltage is kept fixed by increasing the applied voltage accordingly to the current drawn by the detector.

333 gas volumes providing very consistent signals.

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

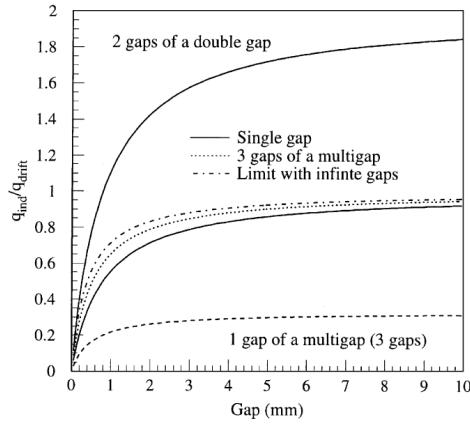


Figure 3.11: Ratio between total induced and drifting charge have been simulated for single gap, double-gap and multigap layouts [38]. The total induced charge for a double-gap RPC is a factor 2 higher than for a multigap.

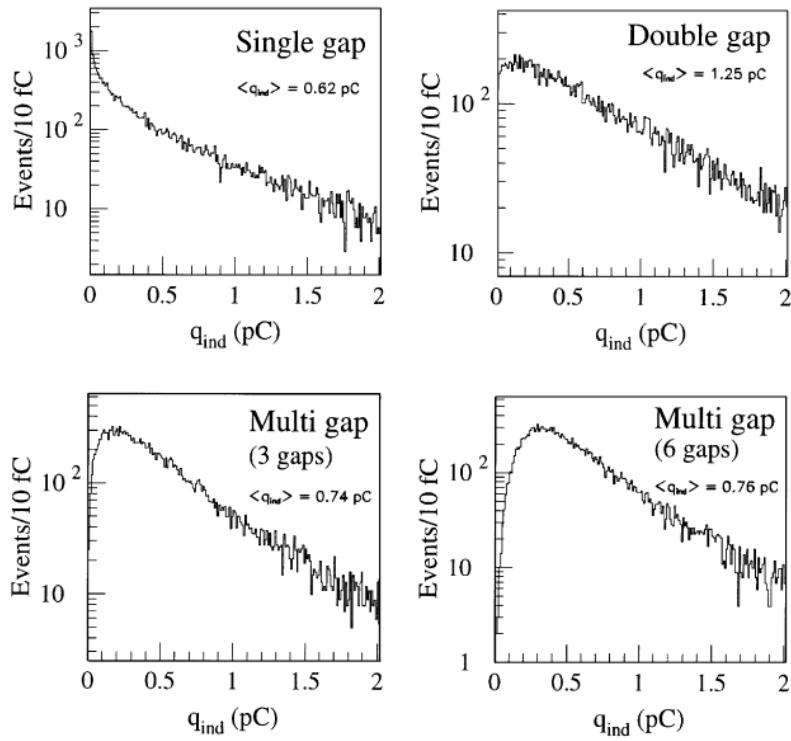
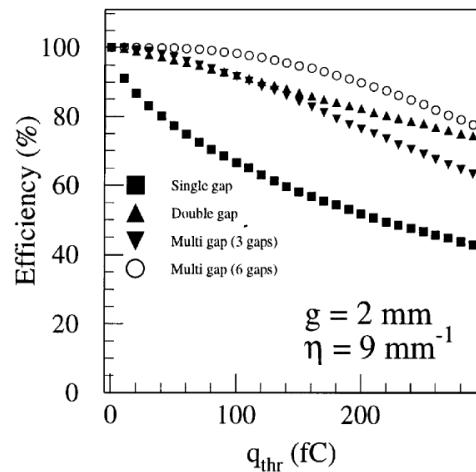


Figure 3.12: Charge spectra have been simulated for single gap, double-gap and multigap layouts [38]. It appears that when single gap shows a decreasing spectrum, double and multigap layouts exhibit a spectrum whose peak is detached from the origin. The detachment gets stronger as the number of gaps increases.



*Figure 3.13: The maximal theoretical efficiency is simulated for single gap, double-gap and multigap layouts [38] at a constant gap thickness of 2 mm and using an effective Townsend coefficient of  $9 \text{ mm}^{-1}$ .*

### 343 3.3 Signal formation

### 344 3.4 Gas transport parameters



# 4

345

346

347

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

348

### 4.1 Resistive Plate Chambers at CMS

349

#### 4.1.1 Overview

350

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

353

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

361

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

367

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

368

369

370

371

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

### 373 4.1.2 The present RPC system

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

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

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

395 The Link Board has 96 input channels (one channel corresponds to one RPC strip). The input  
 396 signals are the  $\sim 100$  ns binary pulses which are synchronous to the RPC hits, but not to the LHC  
 397 clock (which drives the entire CMS electronics). Thus the first step of the FEB signals processing  
 398 is synchronization, i.e. assignment of the signals to the BXes (25 ns periods). Then the data are  
 399 compressed with a simple zero-suppressing algorithm (the input channels are grouped into 8 bit  
 400 partitions, only the partitions with at least one nonzero bit are selected for each BX). Next, the non-  
 401 empty partitions are time-multiplexed i.e. if there are more than one such partition in a given BX,  
 402 they are sent one-by-one in consecutive BXes. The data from 3 neighbouring LBs are concentrated  
 403 by the middle LB which contains the optical transmitter for sending them to the USC over a fiber at  
 404 1.6 Gbps.

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

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

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

416 front end electronics.

### 417 4.1.3 Pulse processing of CMS RPCs

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

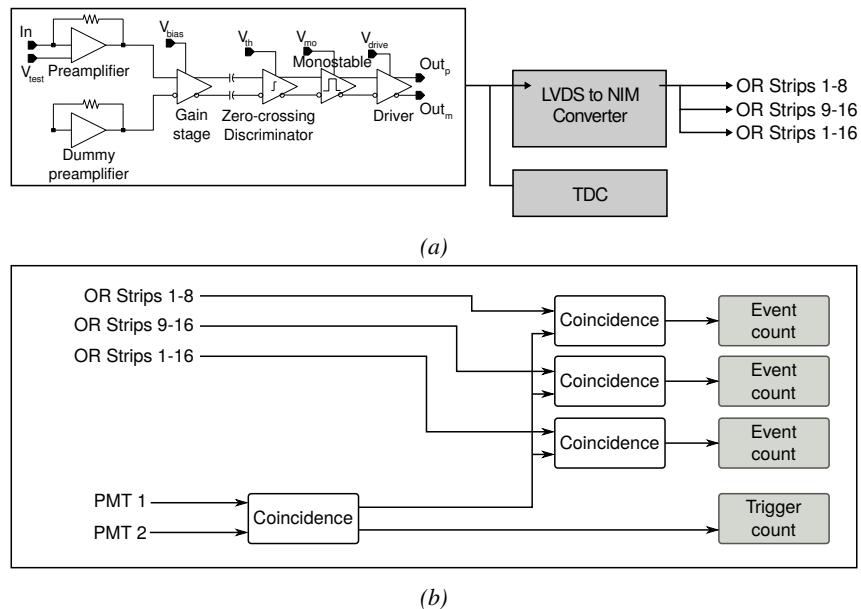
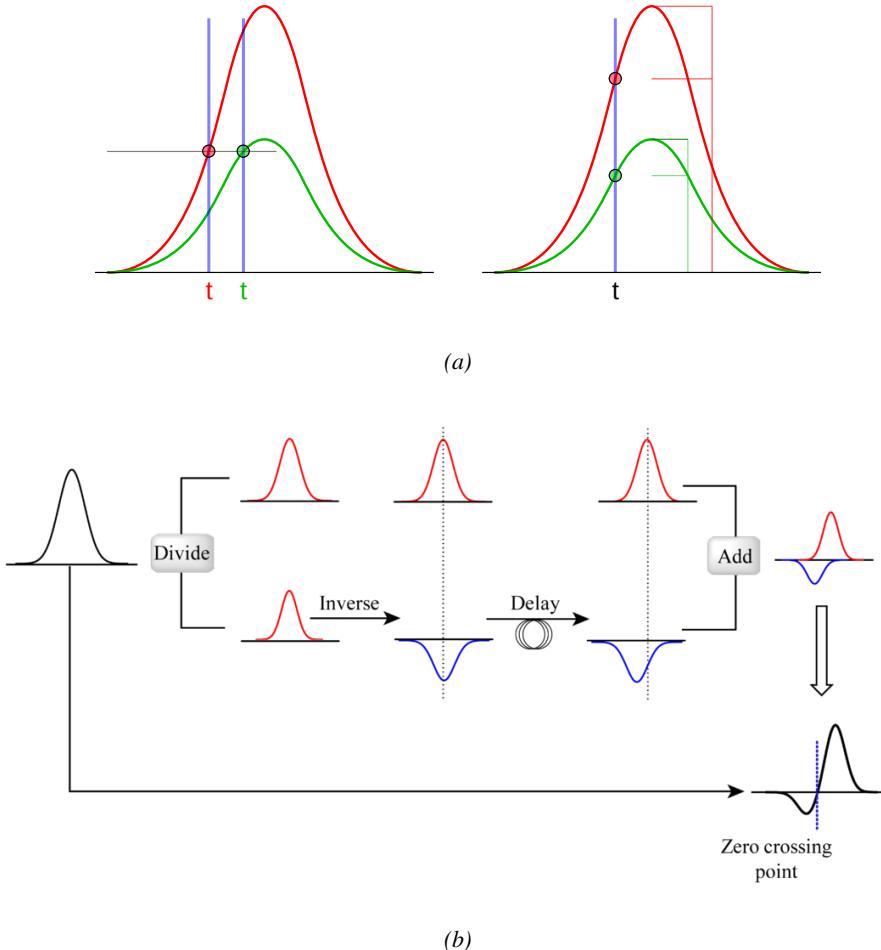


Figure 4.1: Signals from the RPC strips are shaped by the FEE described on Figure 4.1a. Output LVDS signals are then read-out by a TDC module connected to a computer or converted into NIM and sent to scalers. Figure 4.1b describes how these converted signals are put in coincidence with the trigger.

## 426 4.2 Testing detectors under extreme conditions

427 The upgrade from LHC to HL-LHC will increase the peak luminosity from  $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$  to reach  
 428  $5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ , increasing in the same way the total expected background to which the RPC  
 429 system will be subjected to. Composed of low energy gammas and neutrons from  $p$ - $p$  collisions, low  
 430 momentum primary and secondary muons, puch-through hadrons from calorimeters, and particles  
 431 produced in the interaction of the beams with collimators, the background will mostly affect the  
 432 regions of CMS that are the closest to the beam line, i.e. the RPC detectors located in the endcaps.  
 433 [To update.]

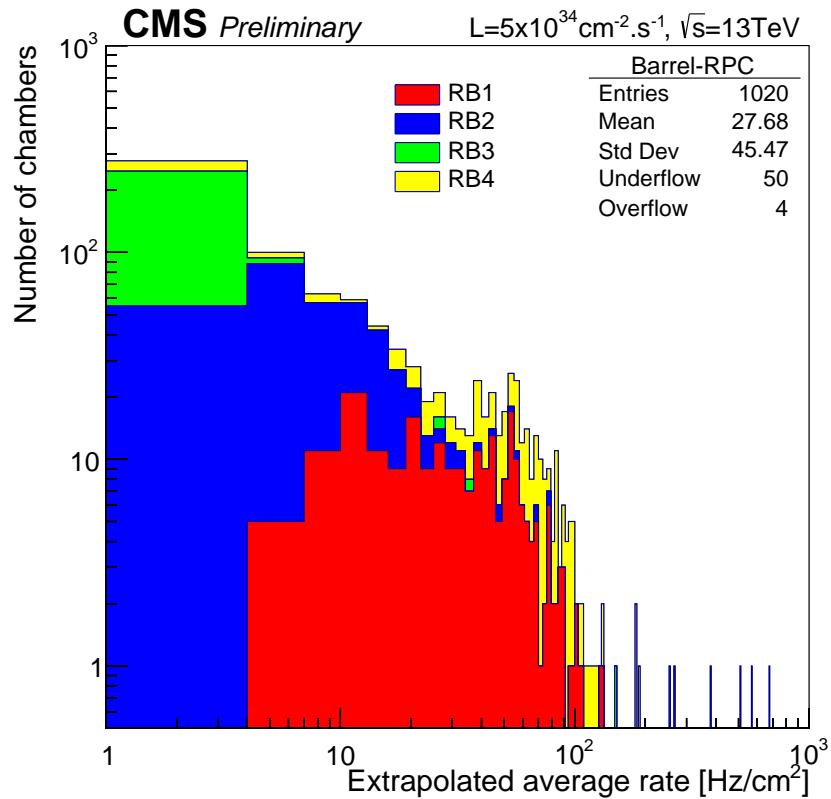
434



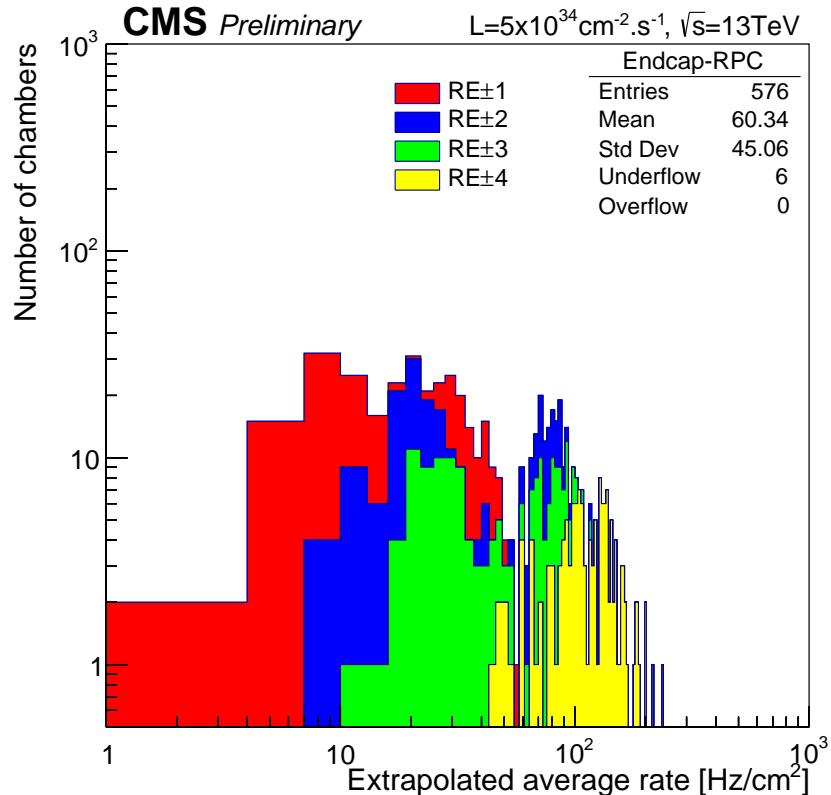
*Figure 4.2: Description of the principle of a CFD. A comparison of threshold triggering (left) and constant fraction triggering (right) is shown in Figure 4.2a. Constant fraction triggering is obtained thanks to zero-crossing technique as explained in Figure 4.2b. The signal arriving at the input of 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.*

435     The 2016 data allowed to study the values of the background rate in all RPC system. In Figure  
 436     4.3, the distribution of the chamber background hit rate per unit area is shown at a luminosity  
 437     of  $5 \times 10^{34} \text{ cm}^{-2} \cdot \text{s}^{-1}$  linearly extrapolating from data collected in 2016 [ref mentioning the linear  
 438     dependency of rate vs lumi]. The maximum rate per unit area at HL-LHC conditions is expected to  
 439     be of the order of  $600 \text{ Hz/cm}^2$  (including a safety factor 3). Nevertheless, Fluka simulations have  
 440     conducted in order to understand the background at HL-LHC conditions. The comparison to the  
 441     data has shown, in Figure 4.4, a discrepancy of a factor 2 even though the order of magnitude is  
 442     consistent. [Understand mismatch.]

443



(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.

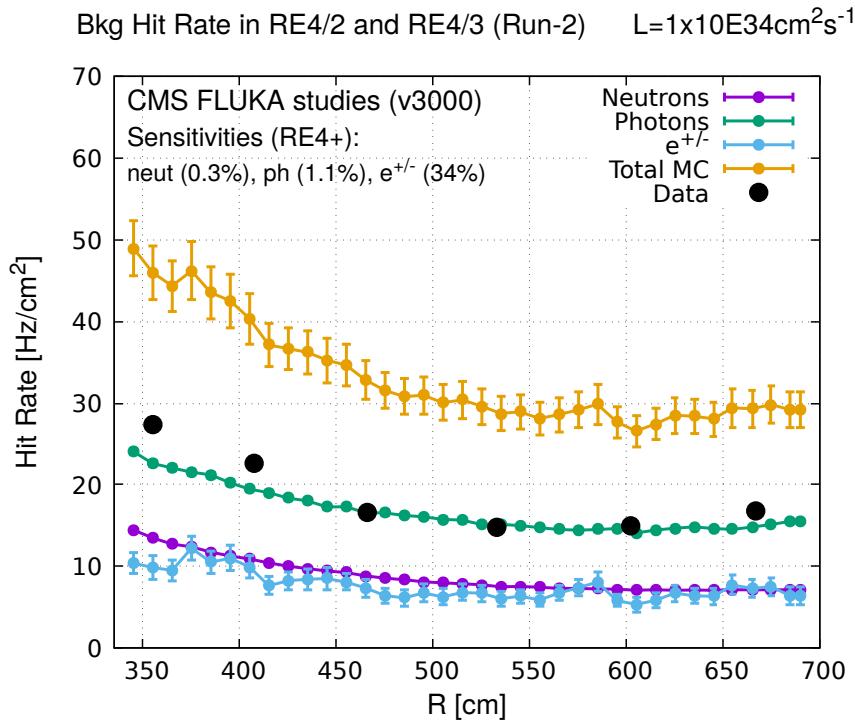


Figure 4.4: Background Fluka simulation compared to 2016 Data at  $L = 10^{34} \text{ cm}^{-2} \cdot \text{s}^{-1}$  in the fourth endcap disk region. A mismatch in between simulation and data can be observed. [\[To be understood.\]](#)

444 In the past, extensive long-term tests were carried out at several gamma and neutron facilities  
 445 certifying the detector performance. Both full size and small prototype RPCs have been irradiated  
 446 with photons up to an integrated charge of  $\sim 0.05C/\text{cm}^2$  and  $\sim 0.4C/\text{cm}^2$ , respectively [39, 40].  
 447 During Run-I, the RPC system provided stable operation and excellent performance and did not  
 448 show any aging effects for integrated charge of the order of  $0.01C/\text{cm}^2$ . Projections on currents  
 449 from 2016 Data, has allowed to determine that the total integrated charge, by the end of HL-LHC,  
 450 would be of the order of  $1C/\text{cm}^2$  (including a safety factor 3). [\[Corresponding figure needed.\]](#)

451

## 4.2.1 The Gamma Irradiation Facilities

### 4.2.1.1 GIF

452 Located in the SPS West Area at the downstream end of the X5 test beam, the Gamma Irradiation  
 453 Facility (GIF) was a test area in which particle detectors were exposed to a particle beam in presence  
 454 of an adjustable gamma background [41]. Its goal was to reproduce background conditions these  
 455 detectors would suffer in their operating environment at LHC. GIF layout is shown in Figure 4.5.  
 456 Gamma photons are produced by a strong  $^{137}\text{Cs}$  source installed in the upstream part of the zone  
 457 inside a lead container. The source container includes a collimator, designed to irradiate a  $6 \times 6 \text{ m}^2$   
 458 area at 5 m maximum to the source. A thin lens-shaped lead filter helps providing with a uniform  
 459 outgoing flux in a vertical plane, orthogonal to the beam direction. The principal collimator hole  
 460 provides a pyramidal aperture of  $74^\circ \times 74^\circ$  solid angle and provides a photon flux in a pyramidal vol-

461

462

463 ume along the beam axis. The photon rate is controled by further lead filters allowing the maximum  
 464 rate to be limited and to vary within a range of four orders of magnitude. Particle detectors under test  
 465 are then placed within the pyramidal volume in front of the source, perpendicularly to the beam line  
 466 in order to profit from the homogeneous photon flux. Adjusting the background flux of photons can  
 467 then be done by using the filters and choosing the position of the detectors with respect to the source.  
 468

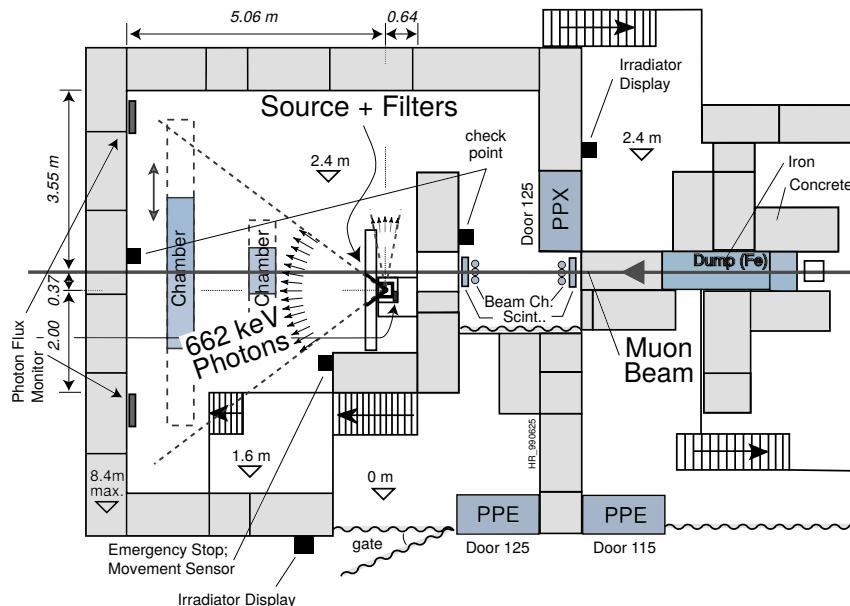


Figure 4.5: Layout of the test beam zone called X5c GIF at CERN. Photons from the radioactive source produce a sustained high rate of random hits over the whole area. The zone is surrounded by 8 m high and 80 cm thick concrete walls. Access is possible through three entry points. Two access doors for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

469 As described on Figure 4.6, the  $^{137}\text{Cs}$  source emits a 662 keV photon in 85% of the decays. An  
 470 activity of 740 GBq was measured on the 5<sup>th</sup> March 1997. To estimate the strength of the flux in  
 471 2014, it is necessary to consider the nuclear decay through time assiciated to the Cesium source  
 472 whose half-life is well known ( $t_{1/2} = (30.05 \pm 0.08)$  y). The GIF tests were done in between the  
 473 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  
 474 the activity from 740 GBq in 1997 to 494 GBq in 2014.  
 475

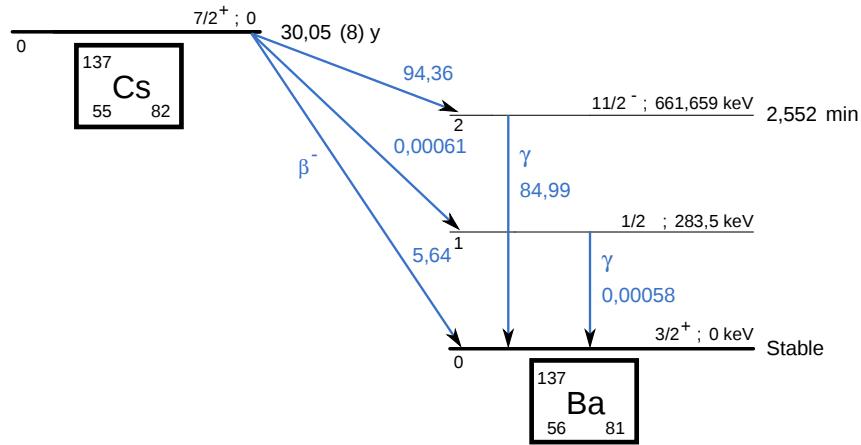


Figure 4.6:  $^{137}\text{Cs}$  decays by  $\beta^-$  emission to the ground state of  $^{137}\text{Ba}$  ( $BR = 5.64\%$ ) and via the 662 keV isomeric level of  $^{137}\text{Ba}$  ( $BR = 94.36\%$ ) whose half-life is 2.55 min.

#### 4.2.1.2 GIF++

The new Gamma Irradiation Facility (GIF++), located in the SPS North Area at the downstream end of the H4 test beam, has replaced its predecessor during LS1 and has been operational since spring 2015 [42]. Like GIF, GIF++ features a  $^{137}\text{Cs}$  source of 662 keV gamma photons, their fluence being controlled with a set of filters of various attenuation factors. The source provides two separated large irradiation areas for testing several full-size muon detectors with continuous homogeneous irradiation, as presented in Figure 4.7.

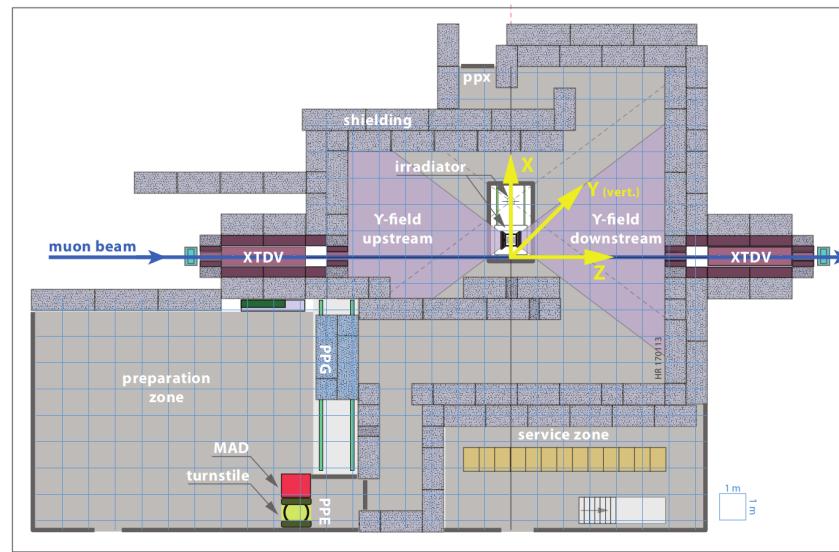


Figure 4.7: Floor plan of the GIF++ facility. When the facility downstream of the GIF++ takes electron beam, a beam pipe is installed along the beam line ( $z$ -axis). The irradiator can be displaced laterally (its center moves from  $x = 0.65 \text{ m}$  to  $2.15 \text{ m}$ ), to increase the distance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

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

487

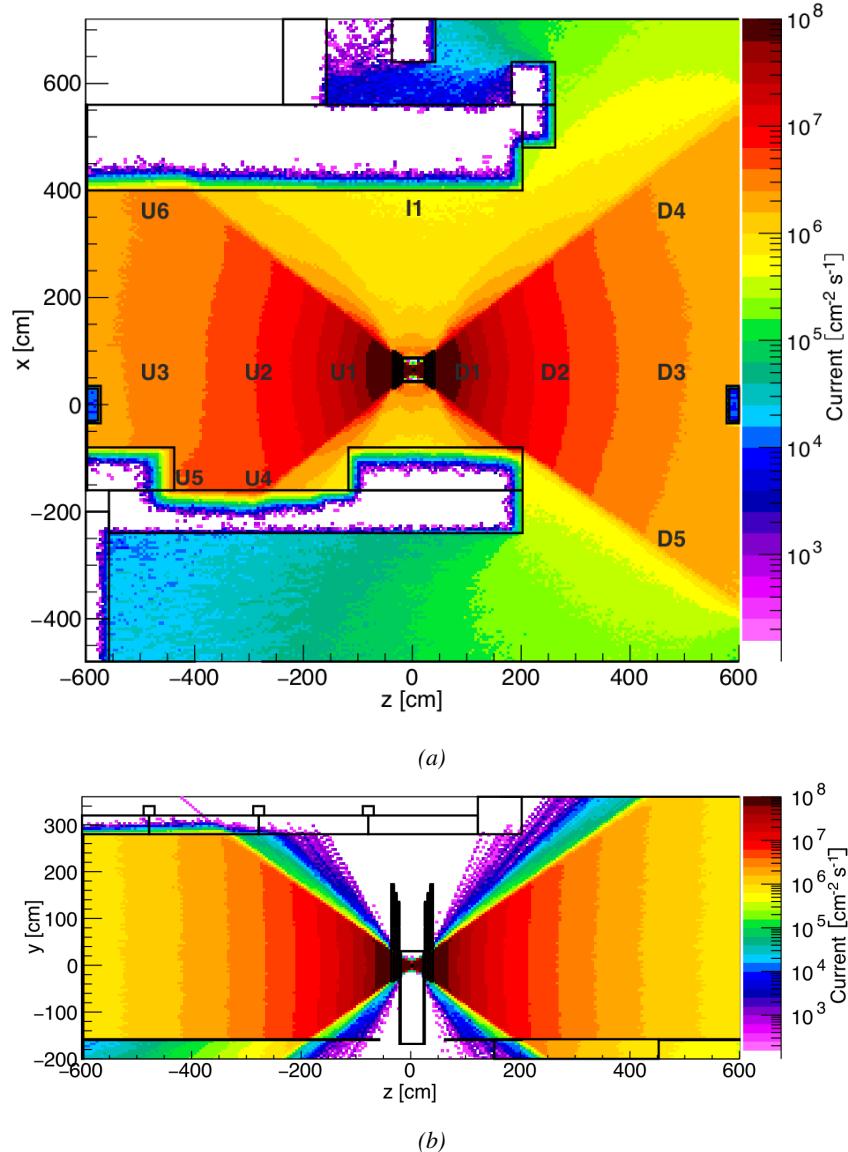


Figure 4.8: Simulated unattenuated current of photons in the xz plane (Figure 4.8a) and yz plane (Figure 4.8b) through the source at  $x = 0.65 \text{ m}$  and  $y = 0 \text{ m}$ . With angular correction filters, the current of 662 keV photons is made uniform in xy planes.

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

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

494

## 495 4.3 Preliminary tests at GIF

### 496 4.3.1 Resistive Plate Chamber test setup

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

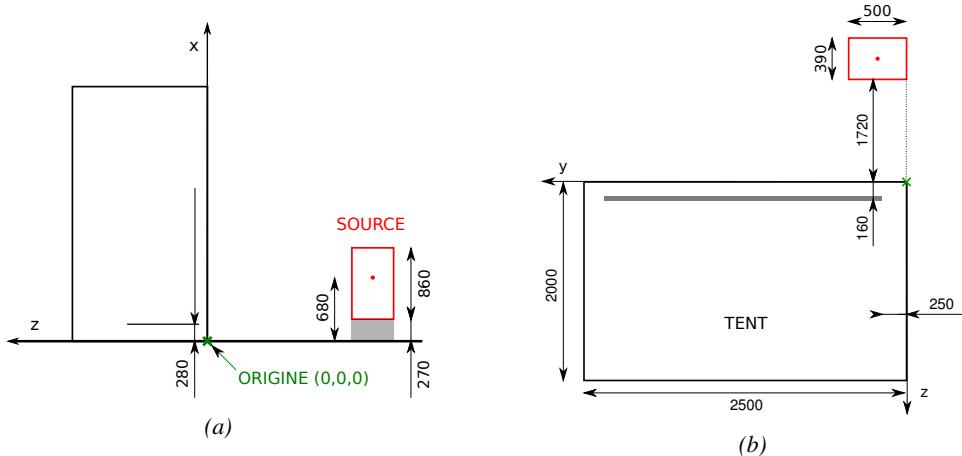
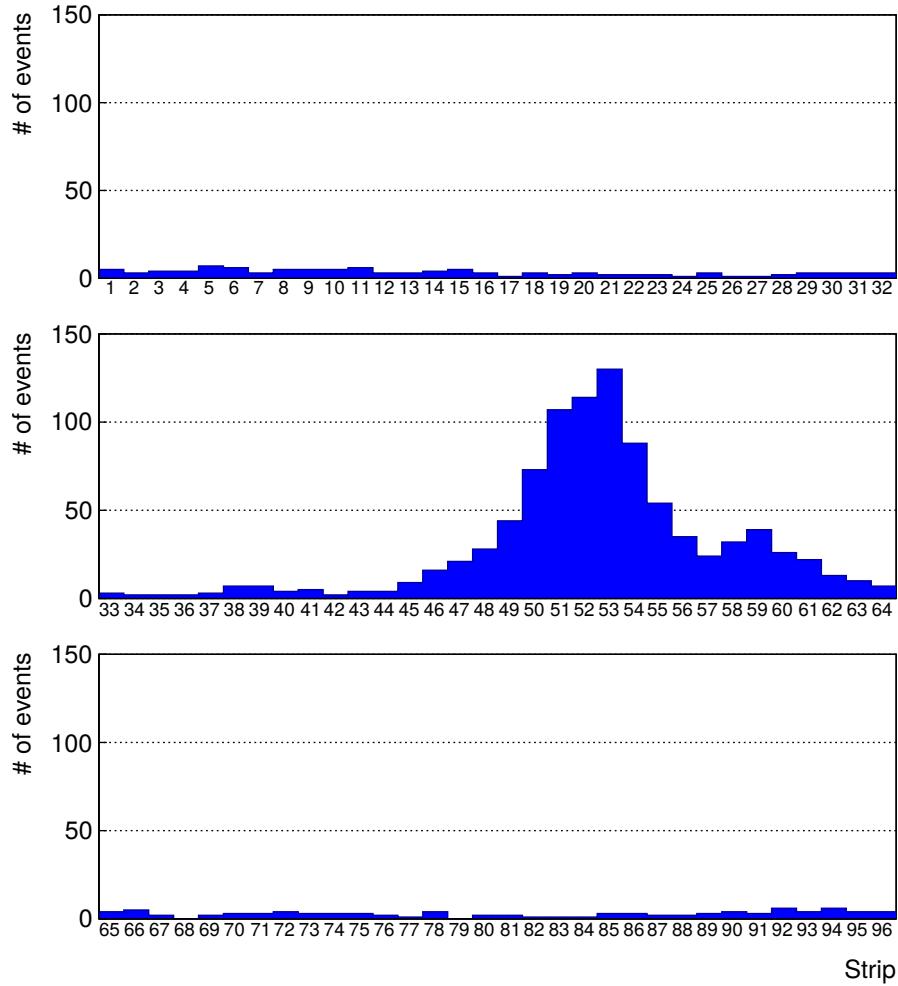


Figure 4.9: Description of the RPC setup. Dimensions are given in mm. A tent containing RPCs is placed at 1720 mm from the source container. The source is situated in the center of the container. RE-4-2-BARC-161 chamber is 160 mm inside the tent. This way, the distance between the source and the chambers 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.*

504 At the time of the tests, the beam not being operational anymore, a trigger composed of 2 plastic  
 505 scintillators has been placed in front of the setup with an inclination of 10 deg with respect to the  
 506 detector plane in order to look at cosmic muons. Using this particular trigger layout, shown on Fig-  
 507 ure 4.10, leads to a cosmic muon hit distribution into the chamber similar to the one in Figure 4.11.  
 508 Measured without gamma irradiation, two peaks can be seen on the profile of partition B, centered  
 509 on strips 52 and 59. Section ?? will help us understand that these two peaks are due respectively to  
 510 forward and backward coming cosmic particles where forward coming particles are first detected by  
 511 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.*

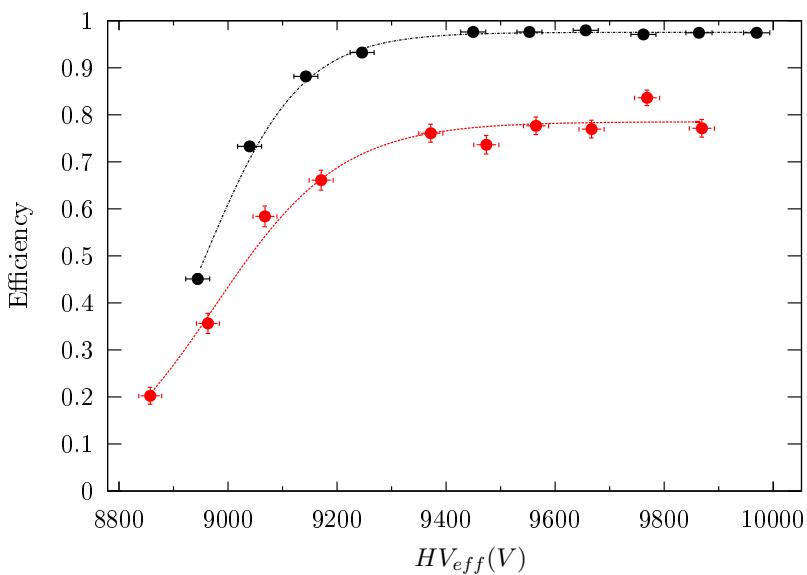
### 512 4.3.2 Data Acquisition

### 513 4.3.3 Geometrical acceptance of the setup layout to cosmic muons

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

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

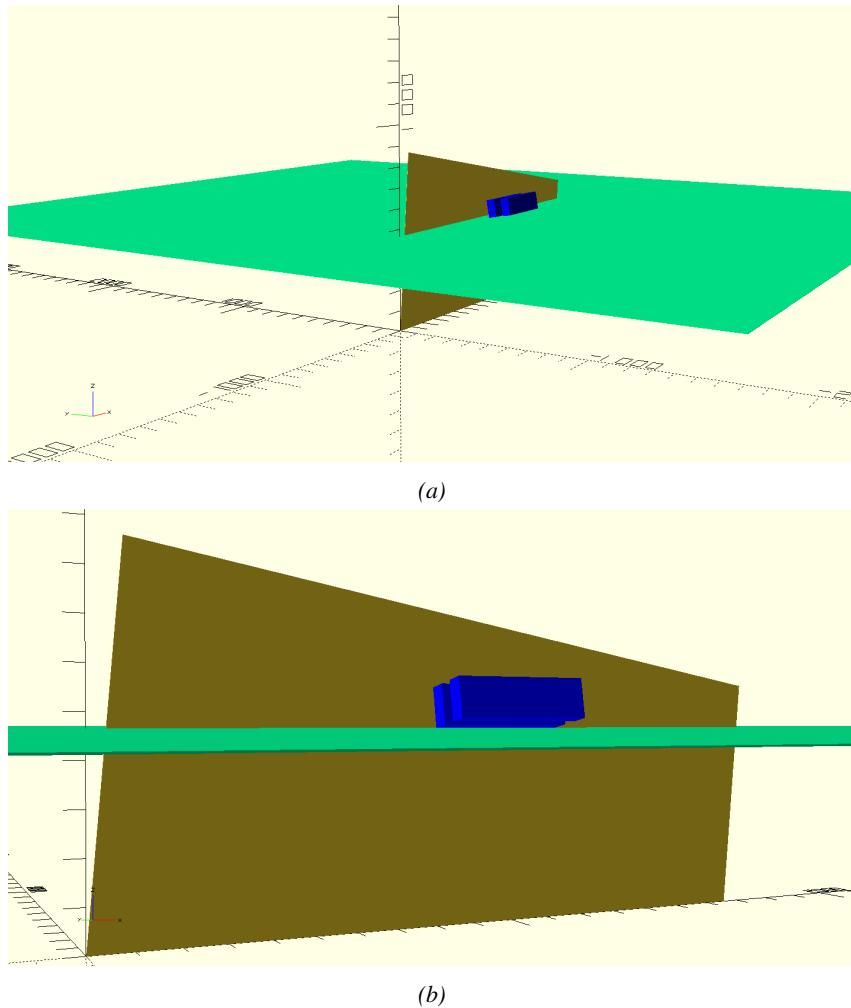
520 An inclination has been given to the cosmic telescope to maximize the muon flux. A good com-  
 521 promise had to be found between good enough muon flux and narrow enough hit distribution to  
 522 be sure to contain all the events into only one half partitions as required from the limited available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



530 *Figure 4.12: Results are derived from data taken on half-partition B2 only. On the 18<sup>th</sup> of June 2014, data  
 531 has been taken on chamber RE-2-BARC-161 at building 904 (Prevessin Site) with cosmic muons providing us a  
 532 reference efficiency plateau of  $(97.54 \pm 0.15)\%$  represented by a black curve. A similar measurement has been  
 533 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  
 534 red curve.*

#### 529 4.3.3.1 Description of the simulation layout

530 The layout of GIF setup has been reproduced and incorporated into a Monte Carlo (MC) simulation  
 531 to study the influence of the disposition of the telescope on the final distribution measured by the  
 532 RPC. A 3D view of the simulated layout is given into Figure 4.13. Muons are generated randomly  
 533 in a horizontal plane located at a height corresponding to the lowest point of the PMTs. This way,  
 534 the needed size of the plane in order to simulate events happening at very big azimuthal angles (i.e.  
 535  $\theta \approx \pi$ ) can be kept relatively small. The muon flux is designed to follow the usual  $\cos^2\theta$  distribution  
 536 for cosmic particle. The goal of the simulation is to look at muons that pass through the muon  
 537 telescope composed of the two scintillators and define their distribution onto the RPC plane. During  
 538 the reconstruction, the RPC plane is then divided into its strips and each muon track is assigned to a  
 539 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.*

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

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

554 **4.3.3.2 Simulation procedure**

555 The simulation software has been designed using C++ and the output data is saved into ROOT  
 556 histograms. Simulations start for a threshold  $T_{scint}$  varying in a range from 0 to 45 mm in steps  
 557 of 5 mm, where  $T_{scint} = 0$  mm corresponds to the case where there isn't any threshold apply on  
 558 the input signal while  $T_{scint} = 45$  mm, which is the scintillator thickness, is the case where muons  
 559 cannot arrive orthogonally onto the scintillator surface. For a given  $T_{scint}$ , a set of RPC thresholds  
 560 are considered. The RPC threshold,  $T_{RPC}$  varies from 2 mm, the thickness of the gas volume, to  
 561 3 mm in steps of 0.25 mm. For each  $(T_{scint}; T_{RPC})$  pair,  $N_\mu = 10^8$  muons are randomly generated  
 562 inside the muon plane described in the previous paragraph with an azimuthal angle  $\theta$  chosen to follow  
 563 a  $\cos^2\theta$  distribution.

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

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

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

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

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

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

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

594 **4.3.3.3 Results**

595 **Influence of  $T_{scint}$  on the muon distribution**

596 **Influence of  $T_{RPC}$  on the muon distribution**

597 **Influence of the telescope inclination on the muon distribution**

598 **Comparison to data taken at GIF without irradiation**

599 **4.3.4 Photon flux at GIF**

600 **4.3.4.1 Expectations from simulations**

601 In order to understand and evaluate the  $\gamma$  flux in the GIF area, simulations had been conducted in  
 602 1999 and published by S. Agosteo et al [41]. Table 4.1 presented in this article gives us the  $\gamma$  flux  
 603 for different distances  $D$  to the source. This simulation was done using GEANT and a Monte Carlo  
 604 N-Particle (MCNP) transport code, and the flux  $F$  is given in number of  $\gamma$  per unit area and unit time  
 605 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 [41].

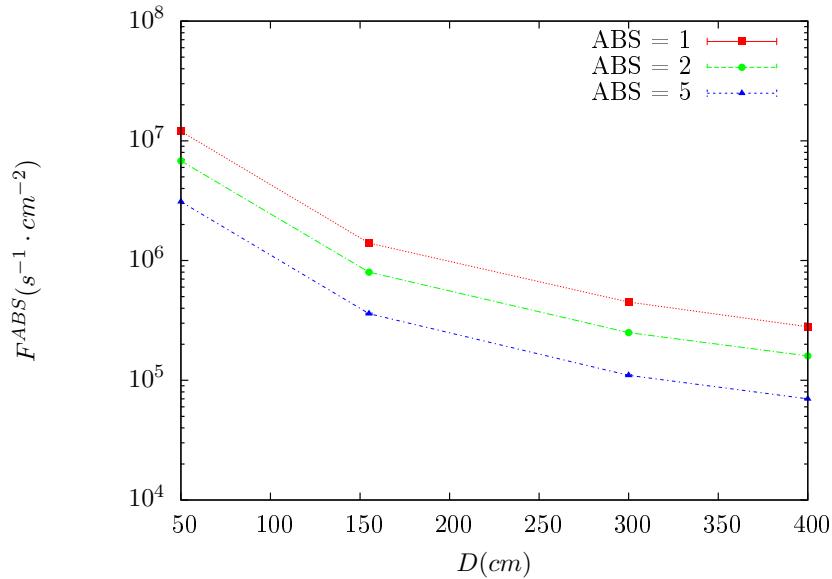


Figure 4.14:  $\gamma$  flux  $F(D)$  is plot using values from table 4.1. As expected, the plot shows similar attenuation behaviours with increasing distance for each absorption factors.

The simulation doesn't directly provides 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 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.

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

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

$$F^{ABS} = F_0^{ABS} \left( a + \frac{bD_0}{D} \right)^2 \quad (4.8)$$

$$\Delta F^{ABS} = F^{ABS} \left[ \frac{\Delta F_0^{ABS}}{F_0^{ABS}} + 2 \frac{\Delta a + \Delta b \frac{D_0}{D}}{a + \frac{bD_0}{D}} \right] \quad (4.9)$$

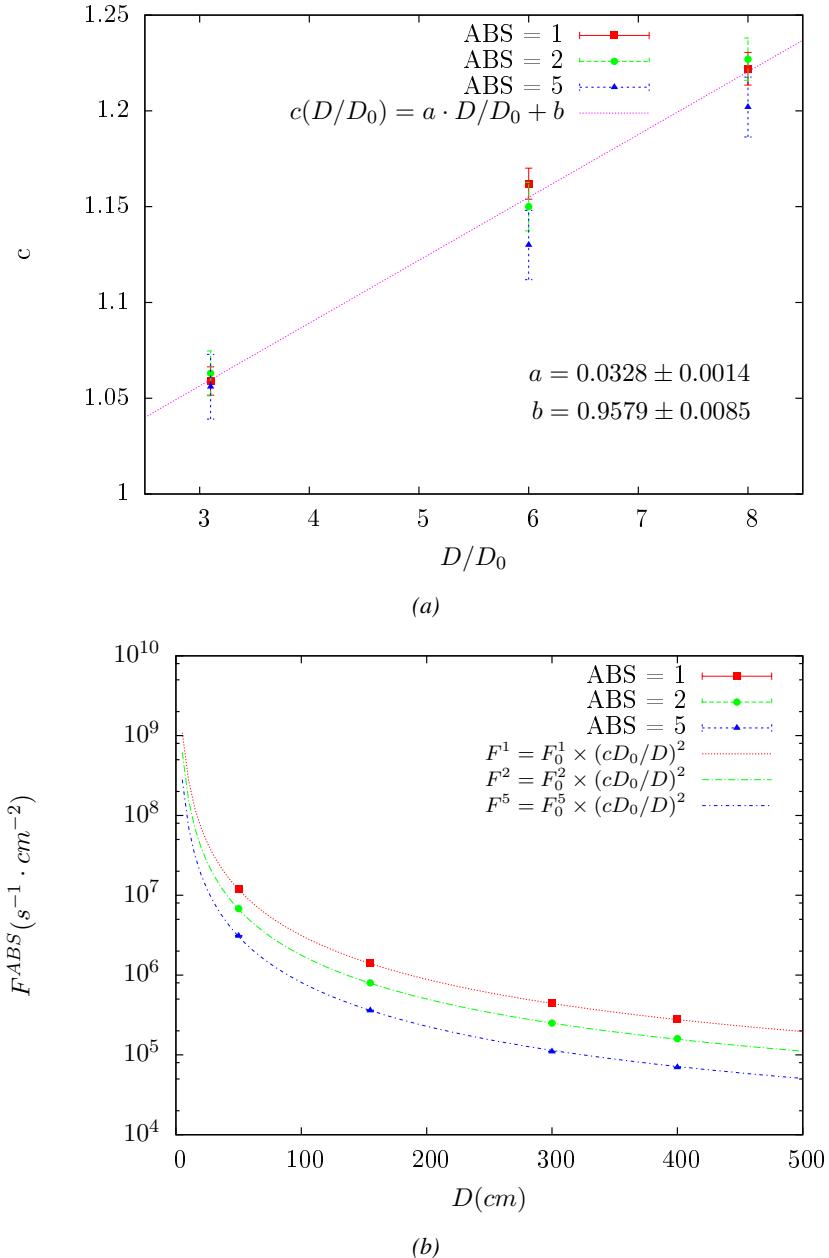


Figure 4.15: Figure 4.15a shows the linear approximation fit done via formulae 4.7 on data from table 4.2. Figure 4.15b shows a comparison of this model with the simulated flux using  $a$  and  $b$  given in figure 4.15a in formulae 4.4 and the reference value  $D_0 = 50\text{cm}$  and the associated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

625 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  
 626 resulting in an attenuation of the activity from 740 GBq in 1997 to 494 GBq in 2014. All the needed  
 627 information to extrapolate the flux through our detector in 2014 has now been assembled, leading  
 628 to the Table 4.3. It is interesting to note that for a common RPC sensitivity to  $\gamma$  of  $2 \cdot 10^{-3}$ , the  
 629 order of magnitude of the estimated hit rate per unit area is of the order of the kHz for the fully  
 630 opened source. Moreover, taking profit of the two working absorbers, it will be possible to scan  
 631 background rates at 0 Hz,  $\sim 300$  Hz as well as  $\sim 600$  Hz. Without source, a good estimate of the  
 632 intrinsic performance will be available. Then at 300 Hz, the goal will be to show that the detectors  
 633 fulfill the performance certification of CMS RPCs. Then a first idea of the performance of the  
 634 detectors at higher background will be provided with absorption factors 2 ( $\sim 600$  Hz) and 1 (no  
 635 absorption). *[Here I will also put a reference to the plot showing the estimated background rate at  
 636 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 [ $Hz cm^{-2}$ ] 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 [41]. 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>637</sup> **4.3.4.2 Dose measurements**

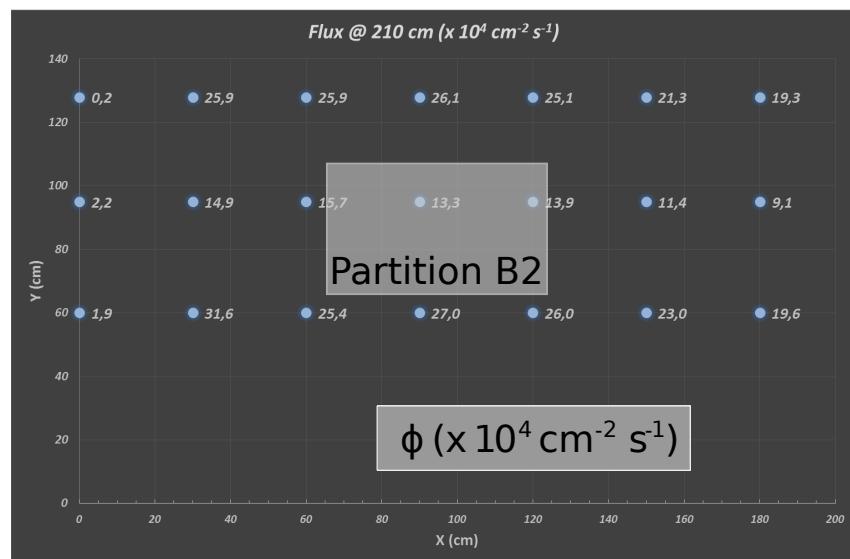


Figure 4.16: Dose measurements has been done in a plane corresponding to the tents front side. This plan is 1900 mm away from the source. As explained in the first chapter, a lens-shaped lead filter provides a uniform photon flux in the vertical plan orthogonal to the beam direction. If the second line 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>638</sup> **4.3.5 Results and discussions**

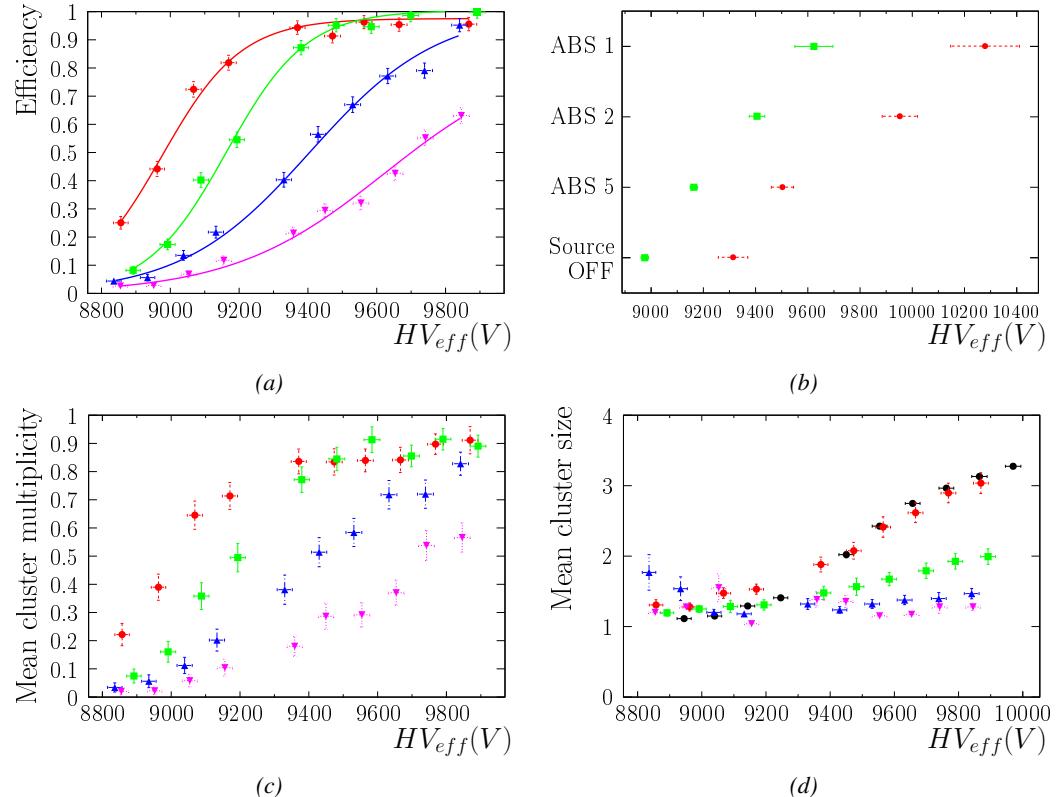


Figure 4.17

---

639    **4.4 Longevity tests at GIF++**

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

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

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

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

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

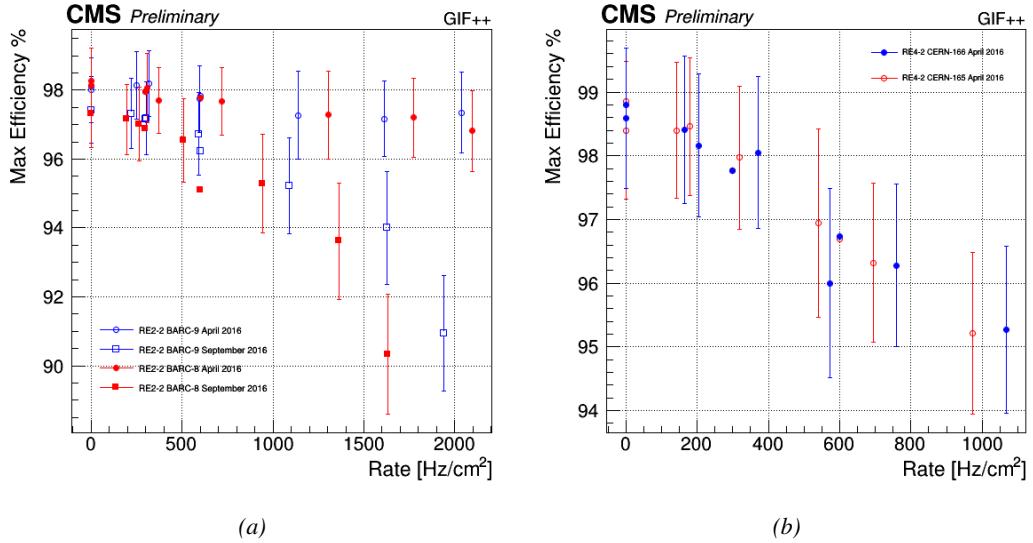


Figure 4.18: Evolution of the maximum efficiency for RE2 (4.18a) and RE4 (4.18b) chambers with increasing extrapolated  $\gamma$  rate per unit area at working point. Both irradiated (blue) and non irradiated (red) chambers are shown.

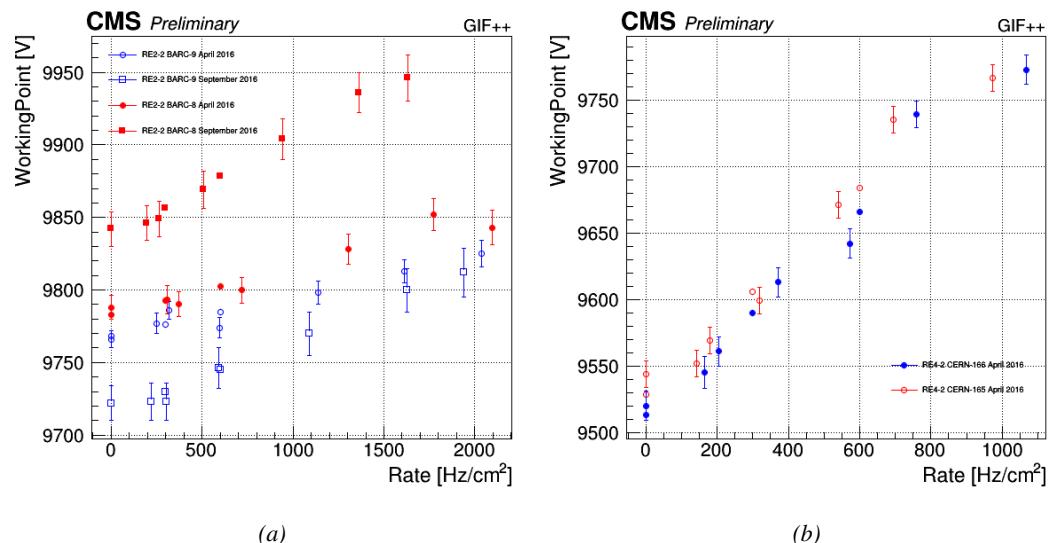
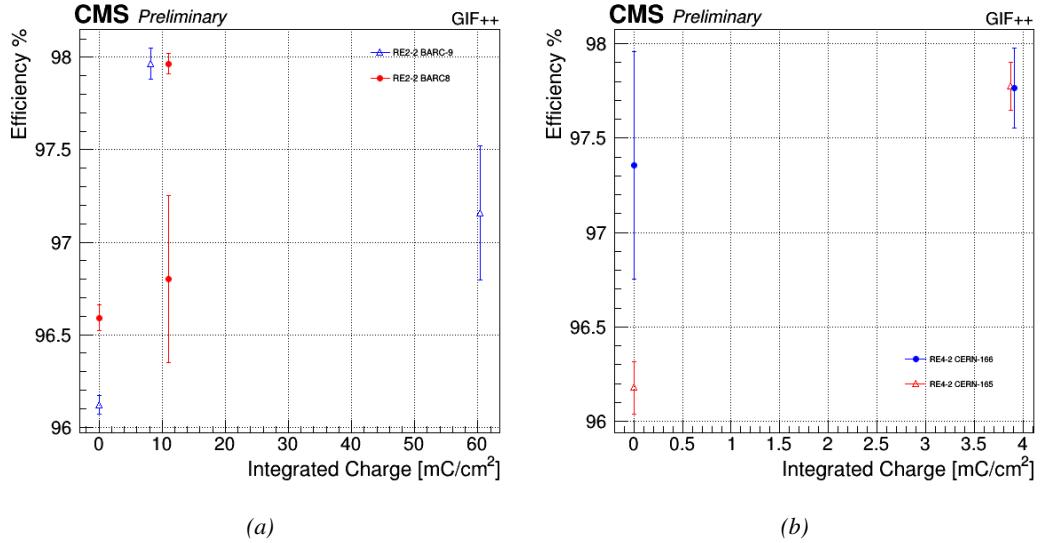
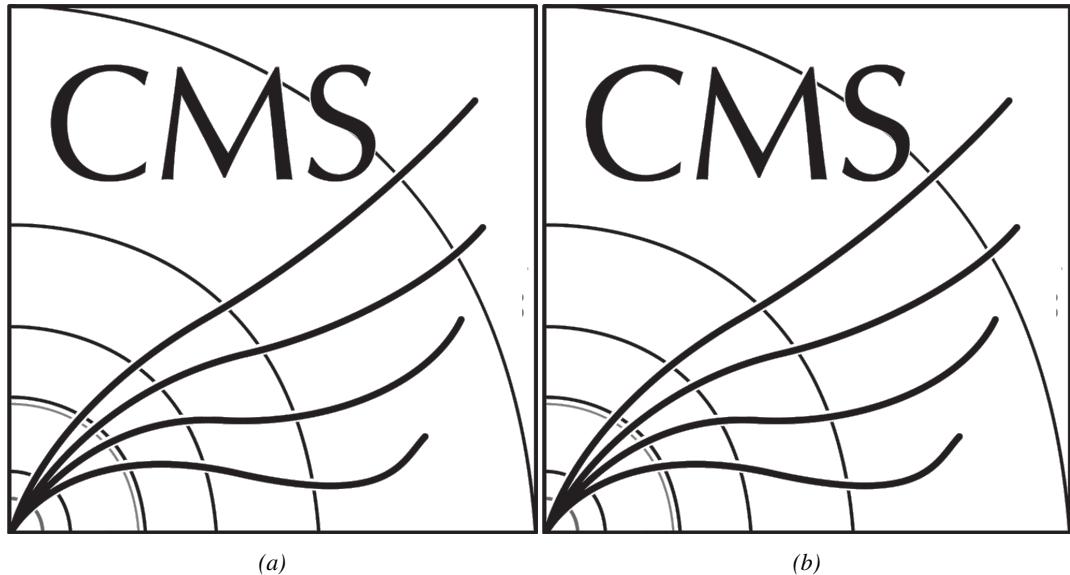


Figure 4.19: Evolution of the working point for RE2 (4.19a) and RE4 (4.19b) with increasing extrapolated  $\gamma$  rate per unit area at working point. Both irradiated (blue) and non irradiated (red) chambers are shown.



*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.*

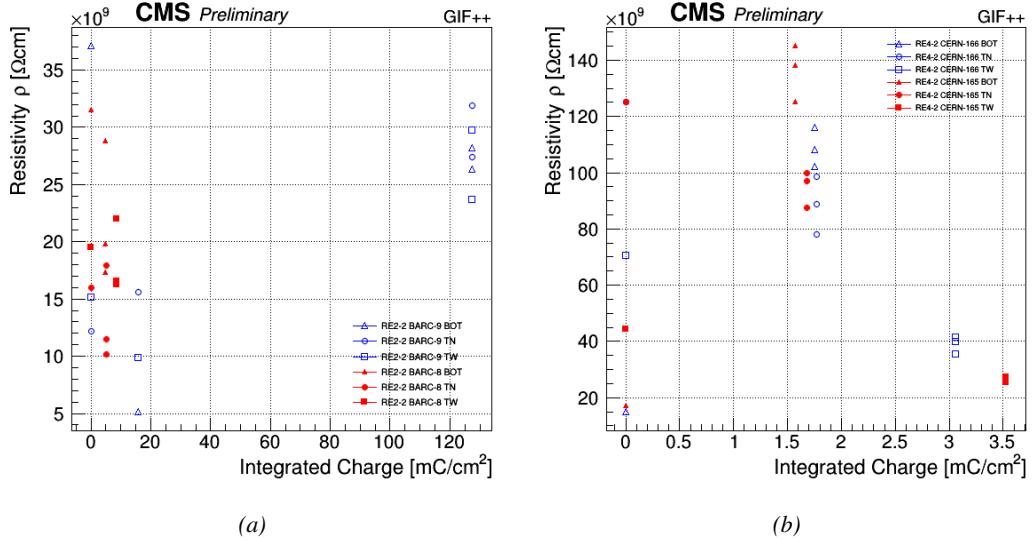


Figure 4.22: Evolution of the Bakelite resistivity for RE2 (4.22a) and RE4 (4.22b) detectors. Both irradiated (blue) and non-irradiated (red) chambers are shown.

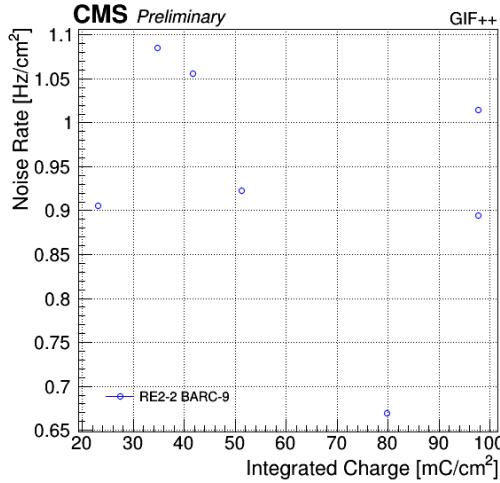


Figure 4.23: Evolution of the noise rate per unit area for the irradiated chamber RE2-2-BARC-9 only.

#### 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 GIFT++ 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

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

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

696 Two different types of tests are conducted on RPCs via the DAQ. Indeed, the performance of the  
 697 detectors is measured periodically during dedicated test beam periods using the H4 muon beam. In  
 698 between these test beam periods, when the beam is not available, the chambers are irradiated by the  
 699  $^{137}\text{Cs}$  in order to accumulate deposited charge and the gamma background is measured.

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

706 In the case of performance test, the trigger signal used for data acquisition is generated by the  
 707 coincidence of three scintillators. A first one is placed upstream outside of the bunker, a second one  
 708 is placed downstream outside of the bunker, while a third one is placed in front of the trolley, close by  
 709 the chambers. Every time a trigger is sent to the TDCs, i.e. every time a muon is detected, knowing  
 710 the time delay in between the trigger and the RPC signals, signals located in the right time window  
 711 are extracted from the buffers and saved for later analysis. Signals are taken in a time window of  
 712 400 ns centered on the muon peak (here we could show a time spectrum). On the other hand, in the  
 713 case of background rate measurement, the trigger signal needs to be "random" not to measure muons  
 714 but to look at gamma background. A trigger pulse is continuously generated at a rate of 300 Hz using  
 715 a dual timer. To integrate an as great as possible time, all signals contained within a time window of  
 716 10us prior to the random trigger signal are extracted form the buffers and saved for further analysis  
 717 (here another time spectrum to illustrate could be useful, maybe even place both spectrum together  
 718 as a single Figure).

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

#### 724 **4.4.2 RPC current, environmental and operation parameter monitoring**

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

$$727 \quad HV_{eff} = HV_{app} \times \left( 0.2 + 0.8 \cdot \frac{P_0}{P} \times \frac{T}{T_0} \right) \quad (4.10)$$

<sup>727</sup> where  $T_0$  (=293 K) and  $P_0$  (=990 mbar) are the reference values.

<sup>728</sup> **4.4.3 Measurement procedure**

<sup>729</sup> Insert a short description of the online tools (DAQ, DCS, DQM).

<sup>730</sup> Insert a short description of the offline tools : tracking and efficiency algorithm.

<sup>731</sup> Identify long term aging effects we are monitoring the rates per strip.

<sup>732</sup> **4.4.4 Longevity studies results**

# 5

733

734

## Investigation on high rate RPCs

735 **5.1 Rate limitations and ageing of RPCs**

736 **5.1.1 Low resistivity electrodes**

737 **5.1.2 Low noise front-end electronics**

738 **5.2 Construction of prototypes**

739 **5.3 Results and discussions**



# 6

740

741

## Conclusions and outlooks

<sup>742</sup> **6.1 Conclusions**

<sup>743</sup> **6.2 Outlooks**



# A

744

745

## A data acquisition software for CAEN VME TDCs

746

747 Certifying detectors in the perspective of HL-LHC required to develop tools for the GIF++ experiment.  
748 Among them was the C++ Data Acquisition (DAQ) software that allows to make the communications  
749 in between a computer and TDC modules in order to retrieve the RPC data [43]. In this  
750 appendix, details about this software, as of how the software was written, how it functions and how  
751 it can be exported to another similar setup, will be given.

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

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

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

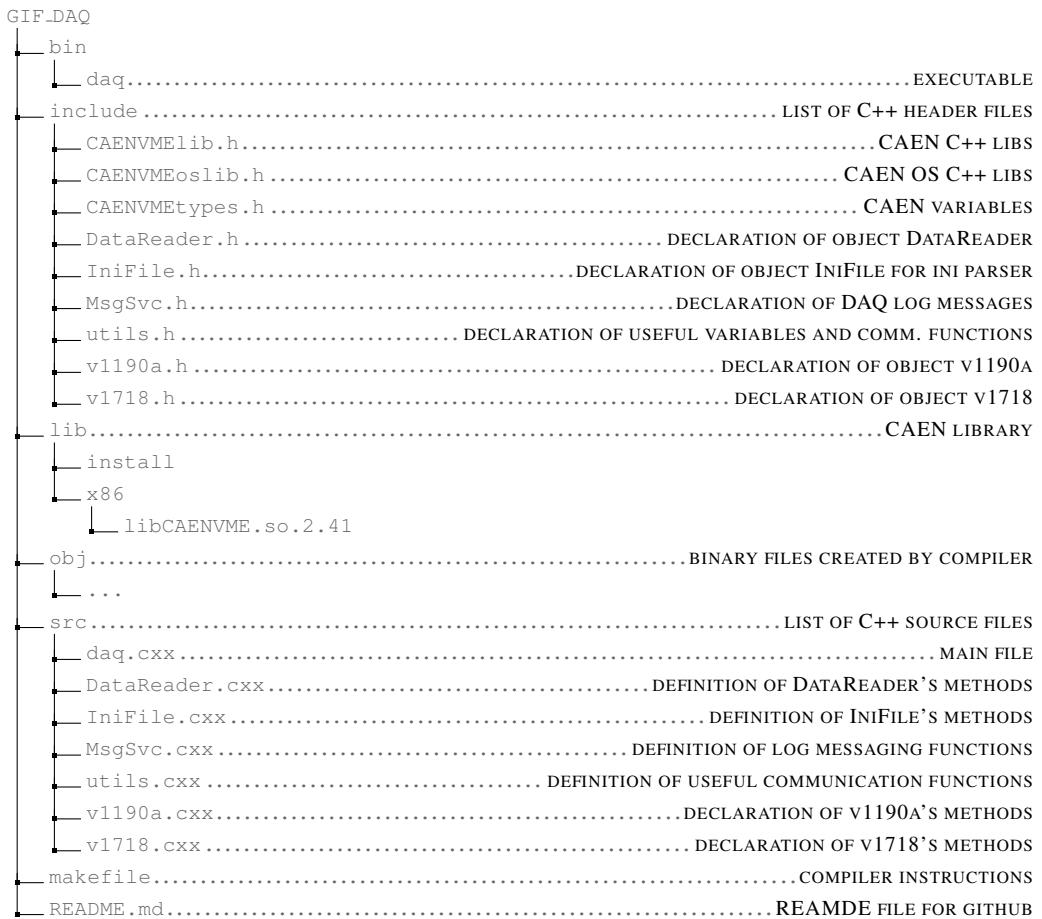
758 The CAEN VME library will not be packaged by distributions and will need to be installed man-  
759 ually. To compile the GIF++ DAQ project via a terminal, from the DAQ folder use the command:

760

761 make

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

765



## 766 A.2 Usage of the DAQ

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

773

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

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

781

### A.3 Description of the readout setup

782 The CMS RPC setup at GIF++ counts 5 V1190A Time-to-Digital Converter (TDC) manufactured  
 783 by CAEN [44]. V1190A are VME units accepting 128 independent Multi-Hit/Multi-Event TDC  
 784 channels whose signals are treated by 4 100 ps high performance TDC chips developed by CERN  
 785 / ECP-MIC Division. The communication between the computer and the TDCs to transfer data is  
 786 done via a V1718 VME master module also manufactured by CAEN and operated from a USB  
 787 port [45]. These VME modules are all hosted into a 6U VME 6021 powered crate manufactured by  
 788 W-Ie-Ne-R than can accommodate up to 21 VME bus cards [46]. These 3 components of the DAQ  
 789 setup are shown in Figure A.1.  
 790

791

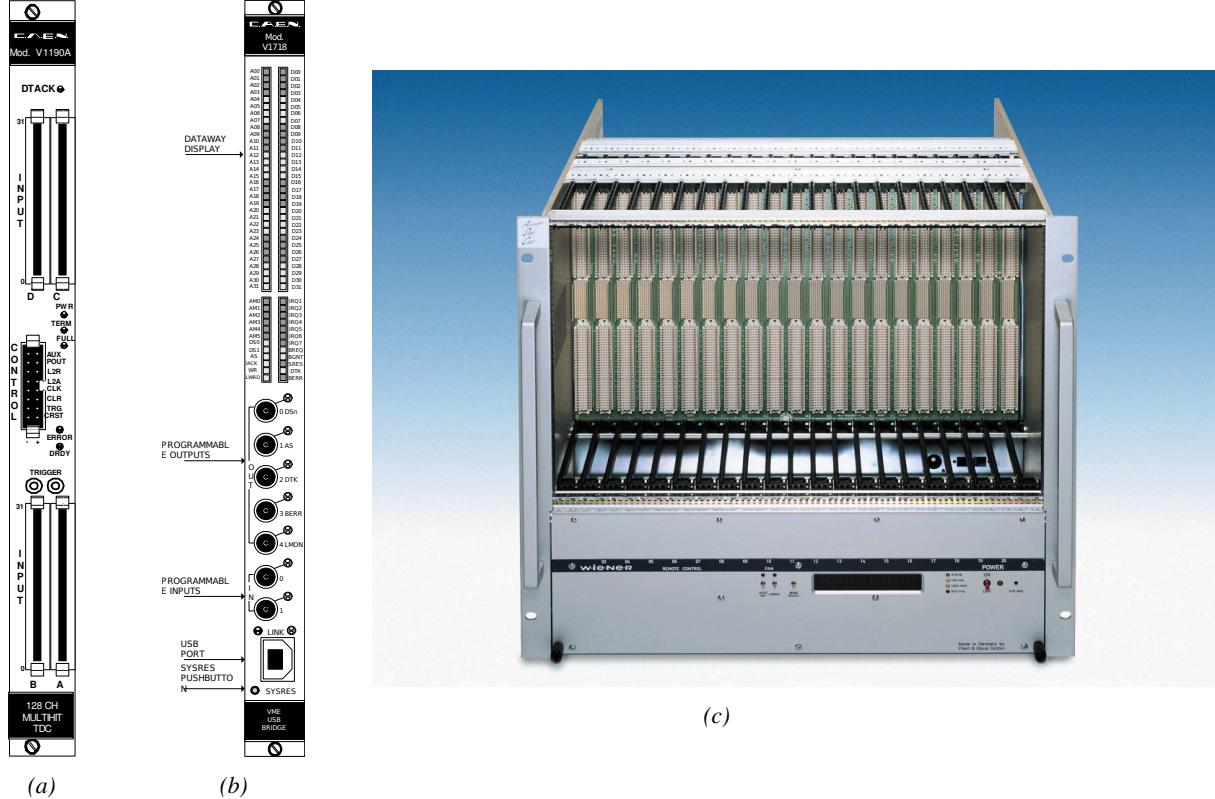


Figure A.1: (A.1a) View of the front panel of a V1190A TDC module [44]. (A.1b) View of the front panel of a V1718 Bridge module [45]. (A.1c) View of the front panel of a 6U 6021 VME crate [46].

792

### A.4 Data read-out

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

795 that comes as an input of the DAQ software.

796

#### 797 A.4.1 V1190A TDCs

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

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

```
807     void v1190a::SetTrigWindowWidth(UINT windowWidth, int ntdcs)
```

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

```
810     void v1190a::SetTrigWindowWidth(UINT windowWidth, int ntdcs)
```

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

```
813     void v1190a::SetTrigSearchMargin(UINT searchMargin, int ntdcs)
```

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

```
816     void v1190a::SetTrigRejectionMargin(UINT rejectMargin, int ntdcs)
```

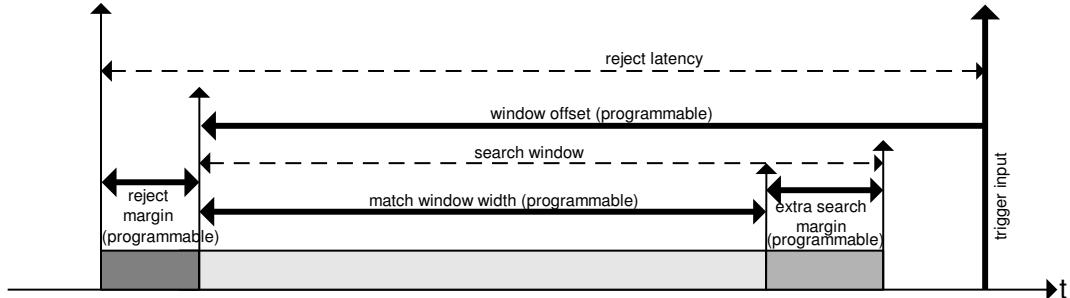


Figure A.2: Module V1190A Trigger Matching Mode timing diagram [44].

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

- 819 • **1:** the match window is entirely contained after the trigger signal,
- 820 • **2:** the match window overlaps the trigger signal, or
- 821 • **3:** the match window is entirely contained before the trigger signal as displayed on Figure A.2.

822 In both the first and second cases, the sum of the window width and of the offset can be set to  
823 a maximum of 40 clocks, which corresponds to 1  $\mu$ s. Evidently, the offset can be negative, allowing  
824 for a longer match window, with the constraint of having the window ending at most 1  $\mu$ s after the  
825 trigger signal. In the third case, the maximum negative offset allowed is of 2048 clocks (12 bit) cor-  
826 responding to 51.2  $\mu$ s, the match window being strictly smaller than the offset. In the case of GIF++,  
827 the choice has been made to use this last setting by delaying the trigger signal. During the studies  
828 performed in GIF++, both the efficiency of the RPCs, probed using a muon beam, and the noise or  
829 gamma background rate are monitored. The extra search and reject margins are left unused.  
830 To probe the efficiency of RPC detectors, the trigger time tag is provided by the coïncidence of  
831 scintillators when a bunch of muons passes through GIF++ area is used to trigger the data acquisi-  
832 tion. For this measurement, it is useful to reduce the match window width only to contain the muon  
833 information. Indeed, the delay in between a trigger signal and the detection of the corresponding  
834 muon in the RPC being very contant (typically a few tens of ns due to jitter and cable length), the  
835 muon signals are very localised in time. Thus, due to a delay of approximalety 325 ns in between  
836 the muons and the trigger, the settings where chosen to have a window width of 24 clocks (600 ns)  
837 centered on the muon peak thanks to a negative offset of 29 clocks (725 ns).  
838 On the otherhand, monitoring the rates don't require for the DAQ to look at a specific time window.  
839 It is important to integrate enough time to have a robust measurement of the rate as the number of  
840 hits per time unit. The triggerring signal is provided by a pulse generator at a frequency of 300 Hz  
841 to ensure that the data taking occurs in a random way, with respect to beam physics, to probe only  
842 the irradiation spectrum on the detectors. The match window is set to 400 clocks (10  $\mu$ s) and the  
843 negative offset to 401 clocks as it needs to exceed the value of the match window.

```

844
class v1190a
{
    private :
        long Handle;
        vector<Data32> Address;
        CVDataWidth DataWidth;
        CVAAddressModifier AddressModifier;

    public:

        v1190a(long handle, IniFile *inifile, int ntdcs);
        ~v1190a();
        Data16 write_op_reg(Data32 address, int code, string error);
        Data16 read_op_reg(Data32 address, string error);
        void Reset(int ntdcs);
        void Clear(int ntdcs);
        void TestWR(Data16 value,int ntdcs);
        void CheckTDCStatus(int ntdcs);
        void CheckCommunication(int ntdcs);
        void SetTDCTestMode(Data16 mode,int ntdcs);
        void SetTrigMatching(int ntdcs);
        void SetTrigTimeSubtraction(Data16 mode,int ntdcs);
        void SetTrigWindowWidth(Uint windowHeight,int ntdcs);
        void SetTrigWindowOffset(Uint windowOffset,int ntdcs);
        void SetTrigSearchMargin(Uint searchMargin,int ntdcs);
        void SetTrigRejectionMargin(Uint rejectMargin,int ntdcs);
        void GetTrigConfiguration(int ntdcs);
        void SetTrigConfiguration(IniFile *inifile,int ntdcs);
        void SetTDCDetectionMode(Data16 mode,int ntdcs);
        void SetTDCResolution(Data16 lsb,int ntdcs);
        void SetTDCDeadTime(Data16 time,int ntdcs);
        void SetTDCHeadTrailer(Data16 mode,int ntdcs);
        void SetTDCEventSize(Data16 size,int ntdcs);
        void SwitchChannels(IniFile *inifile,int ntdcs);
        void SetIRQ(Data32 level, Data32 count,int ntdcs);
        void SetBlockTransferMode(Data16 mode,int ntdcs);
        void Set(IniFile *inifile,int ntdcs);
        void CheckStatus(CVErrorCodes status) const;
        int ReadBlockD32(Uint tdc, const Data16 address,
                           Data32 *data, const Uint words, bool ignore_berr);
        Uint Read(RAWData *DataList,int ntdcs);
};


```

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

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

851

## 852     A.4.2 DataReader

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

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

- 861     • a **global header** providing information of the event number since the beginning of the data  
       acquisition,
- 863     • a **TDC header**,
- 864     • the **TDC data** (*if any*), 1 for each hit recorded during the event, providing the channel and the  
       time stamp associated to the hit,
- 866     • a **TDC error** providing error flags,
- 867     • a **TDC trailer**,
- 868     • a **global trigger time tag** that provides the absolute trigger time relatively to the last reset,  
       and
- 870     • a **global trailer** providing the total word count in the event.

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

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

887     Due to irradiation, an event in GIF++ can count up to 300 words per TDC. A limit of 4096 words  
 888 (12 bits) has been set to generate IRQ which represent from 14 to almost 700 events depending on  
 889 the average of hits collected per event. Then the block size has been set to 100 events with enabled  
 890 bus errors. When an AFL is reached for one of the TDCs, the VME bridge stops the acquisition by  
 892 sending a BUSY signal.

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

```
895
896   struct RAWData{
897     vector<int>           *EventList;
898     vector<int>           *NHitsList;
899     vector<int>           *QFlagList;
900     vector<vector<int>>  *Channellist;
901     vector<vector<float>> *TimeStampList;
902   };
```

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

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

```
903
904   class DataReader
905   {
906     private:
907       bool      StopFlag;
908       IniFile  *iniFile;
909       Data32    MaxTriggers;
910       v1718    *VME;
911       int      nTDCs;
912       v1190a   *TDCs;
913       RAWData  TDCData;
914
915     public:
916       DataReader();
917       virtual ~DataReader();
918       void      SetIniFile(string inifilename);
919       void      SetMaxTriggers();
920       Data32    GetMaxTriggers();
921       void      SetVME();
922       void      SetTDC();
923       int      GetQFlag(Uint it);
924       void      Init(string inifilename);
925       void      FlushBuffer();
926       void      Update();
927       string   GetFileName();
928       void      WriteRunRegistry(string filename);
929       void      Run();
930   };
```

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

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

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

```
916
RAWData TDCData;
TFile *outputFile = new TFile(outputFileName.c_str(),"recreate");
TTree *RAWDataTree = new TTree("RAWData","RAWData");

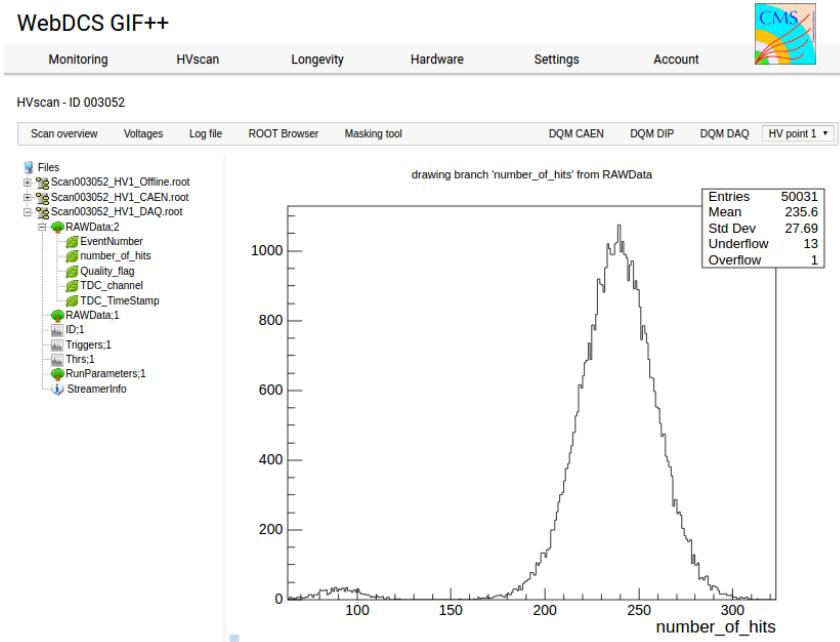
int EventCount = -9;
int nHits = -8;
int qflag = -7;
vector<int> TDCCh;
vector<float> TDCTS;

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);

917
//...
//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();
}
```

918 *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.*

#### 919    A.4.3 Data quality flag

920    Among the parameters that are recorded for each event, the quality flag, defined in Source Code A.5,  
 921    is determined on the fly by checking the data recorded by every single TDC. From method `v1190a::Read()`,  
 922    it can be understood that the content of each TDC buffer is readout one TDC at a time. Entries are  
 923    created in the data list for the first TDC and then, when the second buffer is readout, events corre-  
 924    sponding to entries that have already been created to store data for the previous TDC are added to  
 925    the existing list element. On the contrary, when an event entry has not been yet created in the data  
 926    list, a new entry is created.  
 927

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

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

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

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

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

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

```
949     TDC 0: QFlag = 100 × _QualityFlag
 950     TDC 1: QFlag = 101 × _QualityFlag
 951     ...
 952     TDC N: QFlag = 10N × _QualityFlag
```

953 and the final flag to be with N digits:

```
954     QFlag = n....3210
```

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

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

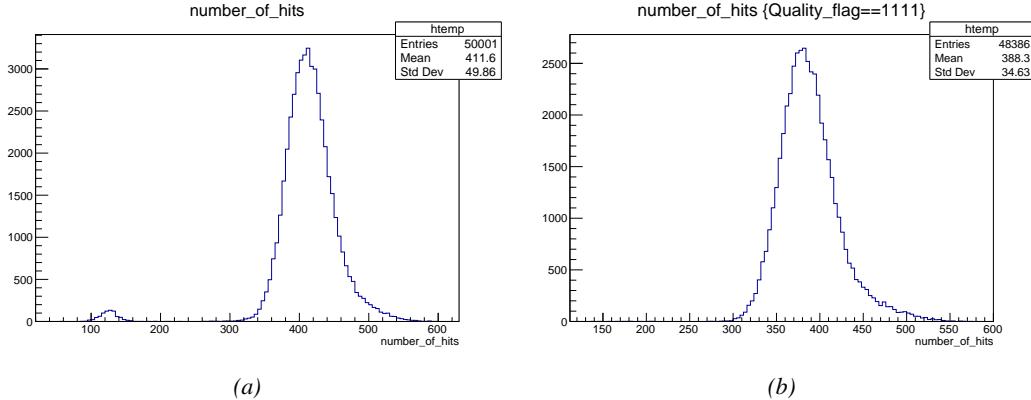
957 but if TDC 2 was corrupted : `QFlag = 1011`.

958 When data taking is over and the data contained in the dynamical `RAWData` structure is transferred  
 959 to the ROOT file, all the 0s are changed into 2s by calling the method `DataReader::GetQFlag()`.  
 960 This will help translating the flag without knowing the number of TDCs beforehand. Indeed, a flag  
 961 111 could be due to a 3 TDC setup with 3 good individual TDC flags or to a more than 3 TDC setup  
 962 with TDCs those ID is greater than 2 being CORRUPTED, thus giving a 0.

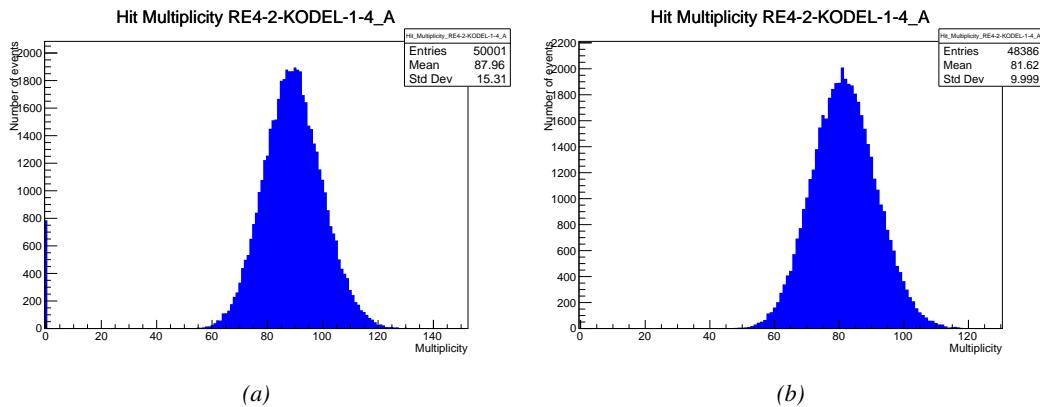
963 The quality flag has been introduced quite late, in October 2017 only, to the list of GIFT++ DAQ  
 964 parameters to be recorded into the output ROOT file. Before this addition, the missing data, corrupting  
 965 the quality for the offline analysis, was contributing to artificially fill data with lower multiplicity.  
 966 Looking at `TBranch number_of_hits` provides an information about the data of the full GIFT++  
 967 setup. When a TDC is not able to transfer data for a specific event, the effect is a reduction of the  
 968 total number of hits recorded in the full setup, this is what can be seen from Figure A.4. After offline  
 969 reconstruction detector by detector, the effect of missing events can be seen in the artificially filled  
 970 bin at multiplicity 0 shown in Figure A.5. Nonetheless, for data with high irradiation levels, as it is  
 971 the case for Figure A.5a, discarding the fake multiplicity 0 data can be done easily during the offline  
 972 analysis. At lower radiation, the missing events contribution becomes more problematic as the mul-  
 973 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.

977



*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.*

978

## A.5 Communications

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

979  
980

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

984

### 985 A.5.1 V1718 USB Bridge

986 In the previous section, the data transfer has been discussed. The importance of the `v1718` object  
 987 (Source Code A.6), used as private member of `DataReader`, was not explicated. VME master  
 988 modules are used for communication purposes as they host the USB port that connects the pow-  
 989 ered crate buffer to the computer where the DAQ is installed. From the source code point of view,  
 990 this object is used to control the communication status, by reading the returned error codes with  
 991 `v1718::CheckStatus()`, or to check for IRQs coming from the TDCs through `v1718::CheckIRQ()`.  
 992 Finally, to ensure that triggers are blocked at the hardware level, a NIM pulse is sent out of one of the  
 993 5 programmable outputs (`v1718::SendBUSY()`) to the VETO of the coincidence module where the  
 994 trigger signals originate from. As long as this signal is ON, no trigger can reach the TDCs anymore.  
 995

```
996 class v1718{
997     private:
998         int Handle;
999         Data32 Data;           // Data
1000        CVIRQLevels Level;    // Interrupt level
1001        CVAddressModifier AM;   // Addressing Mode
1002        CVDataWidth dataSize; // Data Format
1003        Data32 BaseAddress;   // Base Address
1004
1005     public:
1006         v1718(IniFile *inifile);
1007         ~v1718();
1008         long GetHandle(void) const;
1009         int GetData(Data16 data);
1010         Data16 GetData(void);
1011         int SetData(void);
1012         CVIRQLevels GetLevel(void);
1013         int SetLevel(CVIRQLevels level);
1014         CVAddressModifier GetAM(void);
1015         int SetAM(CVAddressModifier am);
1016         CVAddressModifier GetAM(void);
1017         int SetDatasize(CVDataWidth datasize);
1018         CVDataWidth GetDataSize(void);
1019         int SetBaseAddress(Data16 baseaddress);
1020         Data16 GetBaseAddress(void);
1021         void CheckStatus(CVErrorCodes status) const;
1022         bool CheckIRQ();
1023         void SetPulsers();
1024         void SendBUSY(BusyLevel level);
1025     };
1026 }
```

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

### 998 A.5.2 Configuration file

999 The DAQ software takes as input a configuration file written using INI standard [47]. This file is  
 1000 partly filled with the information provided by the shifters when starting data acquisition using the  
 1001 webDCS, as shown by Figure A.6. This information is written in section [`General`] and will later

1002 be stored in the ROOT file that contains the DAQ data as can be seen from Figure A.3. Indeed,  
 1003 another `TTree` called `RunParameters` as well as the 2 histograms `ID`, containing the scan number,  
 1004 start and stop time stamps, and `Triggers`, containing the number of triggers requested by the shifter,  
 1005 are available in the data files. Moreover, `ScanID` and `HV` are then used to construct the file name  
 1006 thanks to the method `DataReader::GetFileName()`.

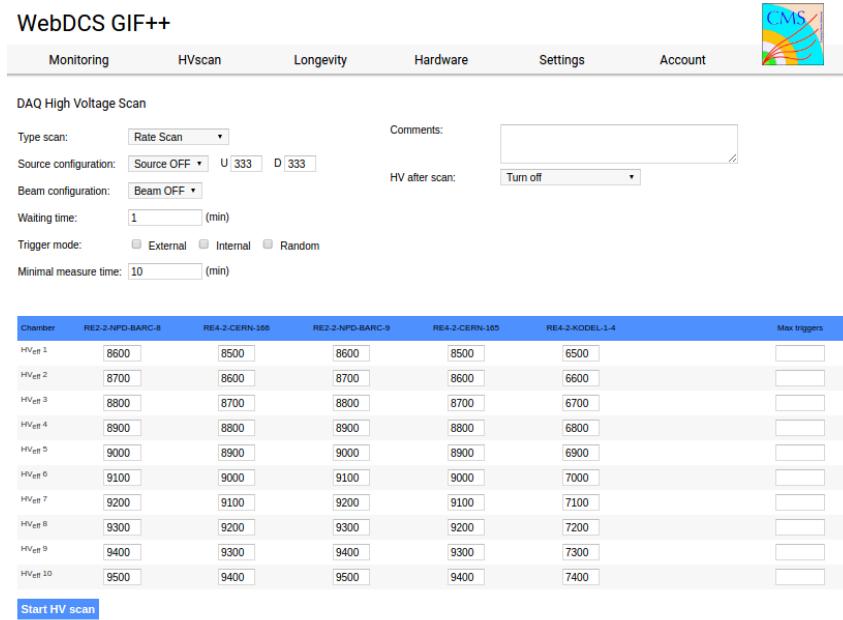


Figure A.6: WebDCS DAQ scan page. On this page, shifters need to choose the type of scan (Rate, Efficiency or Noise Reference scan), the gamma source configuration at the moment of data taking, the beam configuration, and the trigger mode. These information will be stored in the DAQ ROOT output. Are 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.

1007 The rest of the information is written beforehand in the configuration file template, as explicated  
 1008 in Source Code A.7, and contains the hardware addresses to the different VME modules in the  
 1009 setup as well as settings for the TDCs. As the TDC settings available in the configuration file are not  
 1010 supposed to be modified, an improvement would be to remove them from the configuration file and  
 1011 to hardcode them inside of the DAQ code itself or to place them into a different INI file that would  
 1012 host only the TDC settings to lower the probability for a bad manipulation of the configuration file  
 1013 that can be modified from one of webDCS' menus.

```
1014
[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
1015
[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
```

*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.*

1017     In order to retrieve the information of the configuration file, the object `IniFile` has been developed  
 1018     to provide an INI parser, presented in Source Code A.8. It contains private methods returning a  
 1019     boolean to check the type of line written in the file, whether a comment, a group header or a key line  
 1020     (`IniFile::CheckIfComment()`, `IniFile::CheckIfGroup()` and `IniFile::CheckIfToken()`). The  
 1021     key may sometimes be referred to as *token* in the source code. Moreover, the private element  
 1022     `FileData` is a map of `const string` to `string` that allows to store the data contained inside the  
 1023     configuration file via the public method `IniFile::GetFileData()` following the formatting (see  
 1024     method `IniFile::Read()`):

```
1025
  1026     string group, token, value;
  // Get the field values for the 3 strings.
  // Then concatenate group and token together as a single string
  // with a dot separation.
  token = group + "." + token;
  FileData[token] = value;
```

1027     More methods have been written to translate the different keys into the right variable format  
 1028     when used by the DAQ. For example, to get a `float` value out of the configuration file data, knowing  
 1029     the group and the key needed, the method `IniFile::floatType()` can be used. It takes 3 arguments  
 1030     being the group name and key name (both `string`), and a default `float` value used as exception in  
 1031     the case the expected combination of group and key cannot be found in the configuration file. This  
 1032     default value is then used and the DAQ continues on working after sending an alert in the log file for  
 1033     further debugging.

```

1034 typedef map< const string, string > IniFileData;
1035
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();
    };

```

1036       *Source Code A.8: Description of C++ object `IniFile` used as a parser for INI file format.*

### 1037     A.5.3 WebDCS/DAQ intercommunication

1038     When shifters send instructions to the DAQ via the configuration file, it is the webDCS itself that  
1039     gives the start command to the DAQ and then the 2 softwares use inter-process communication  
1040     through file to synchronise themselves. This communication file is represented by the variable **const**  
1041     string \_\_runstatuspath.

1042     On one side, the webDCS sends commands or status that are readout by the DAQ:

- 1043       • INIT, status sent when launching a scan and read via function `CtrlRunStatus(...)`,
- 1044       • START, command to start data taking and read via function `CheckSTART()`,
- 1045       • STOP, command to stop data taking at the end of the scan and read via function `CheckSTOP()`,  
1046        and
- 1047       • KILL, command to kill data taking sent by user and read via function `CheckKILL()`

1048 and on the other, the DAQ sends status that are controled by the webDCS:

- 1049     ● `DAQ_RDY`, sent with `SendDAQReady()` to signify that the DAQ is ready to receive commands  
1050       from the webDCS,
- 1051     ● `RUNNING`, sent with `SendDAQRunning()` to signify that the DAQ is taking data,
- 1052     ● `DAQ_ERR`, sent with `SendDAQError()` to signify that the DAQ didn't receive the expected com-  
1053       mand from the webDCS or that the launch command didn't have the right number of argu-  
1054       ments,
- 1055     ● `RD_ERR`, sent when the DAQ wasn't able to read the communication file, and
- 1056     ● `WR_ERR`, sent when the DAQ wasn't able to write into the communication file.

#### 1057     **A.5.4 Example of inter-process communication cycle**

1058 Under normal conditions, the webDCS and the DAQ processes exchange commands and status via  
1059 the file hosted at the address `__runstatuspath`, as explained in subsection A.5.3. An example of  
1060 cycle is given in Table A.1. In this example, the steps 3 to 5 are repeated as long as the webDCS tells  
1061 the DAQ to take data. A data taking cycle is the equivalent as what is called a *Scan* in GIFT++ jargon,  
1062 referring to a set a runs with several HV steps. Each repetition of steps 3 to 5 is then equivalent to a  
1063 single *Run*.

1064

1065 At any moment during the data taking, for any reason, the shifter can decide that the data taking  
1066 needs to be stopped before it reached the end of the scheduled cycle. Thus at any moment on the  
1067 cycle, the content of the inter-process communication file will be changed to `KILL` and the DAQ will  
1068 shut down right away. The DAQ checks for `KILL` signals every 5s after the TDCs configuration is  
1069 over. So far, the function `CheckKILL()` has been used only inside of the data taking loop of method  
1070 `DataReader::Run()` and thus, if the shifter decides to KILL the data taking during the TDC con-  
1071 figuration phase or the HV ramping in between 2 HV steps, the DAQ will not be stopped smoothly  
1072 and a *force kill* command will be sent to stop the DAQ process that is still awake on the computer.  
1073 Improvements can be brought on this part of the software to make sure that the DAQ can safely  
1074 shutdown at any moment.

1075

## 1076     **A.6 Software export**

1077 In section A.2 was discussed the fact that the DAQ as written in its last version is not a standalone  
1078 software. It is possible to make it a standalone program that could be adapted to any VME setup  
1079 using V1190A and V1718 modules by creating a GUI for the software or by printing the log mes-  
1080 sages that are normally printed in the webDCS through the log file, directly into the terminal. This  
1081 method was used by the DAQ up to version 3.0 moment where the webDCS was completed. Also, it  
1082 is possible to check branches of DAQ v2.X to have example of communication through a terminal.

1083

1084 DAQ v2.X is nonetheless limited in it's possibilities and requires a lot of offline manual interven-  
1085 tions from the users. Indeed, there is no communication of the software with the detectors' power  
1086 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	<code>__runstatuspath</code>
1	launch DAQ ramp voltages ramping over wait for currents stabilization	readout of IniFile configuration of TDCs	INIT
2		configuration done send DAQ ready wait for START signal	DAQ_RDY
3	waiting time over send START		START
4	wait for run to end monitor DAQ run status	data taking ongoing check for KILL signal	RUNNING
5		run over send DAQ_RDY wait for next DCS signal	DAQ_RDY
6	ramp voltages ramping over wait for currents stabilization		DAQ_RDY
3	waiting time over send START		START
4	wait for run to end monitor DAQ run status	update IniFile information data taking ongoing check for KILL signal	RUNNING
5		run over send DAQ_RDY 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.

1087 and loop over to take data without any further manual intervention. In v2.X, the data is taken for a  
1088 single detector setting and at the end of each run, the softwares asks the user if he intends on taking  
1089 more runs. If so, the software invites the user to set the operating voltages accordingly to what is  
1090 necessary and to manual update the configuration file in consequence. This working mode can be a  
1091 very first approach before an evolution and has been successfully used by colleagues from different  
1092 collaborations.

1093  
1094 For a more robust operation, it is recommended to develop a GUI or a web application to inter-  
1095 face the DAQ. Moreover, to limit the amount of manual interventions, and thus the probability to  
1096 make mistakes, it is also recommended to add an extra feature into the DAQ by installing the HV  
1097 Wrapper library provided by CAEN of which an example of use in a similar DAQ software devel-  
1098 opped by a master student of UGent, and called TinyDAQ, is provided on UGent's github. Then, this  
1099 HV Wrapper will help you communicating with and give instructions to a CAEN HV powered crate  
1100 and can be added into the DAQ at the same level where the communication with the user was made  
1101 in DAQ v2.X. In case you are using another kind of power system for your detectors, it is stringly  
1102 adviced to use HV modules or crates that can be remotely controled via a using C++ libraries.

1103

# B

1104

1105

## Details on the offline analysis package

1106 The data collected in GIF++ thanks to the DAQ described in Appendix A is difficult to interpret by  
1107 a human user that doesn't have a clear idea of the raw data architecture of the ROOT data files. In  
1108 order to render the data human readable, a C++ offline analysis tool was designed to provide users  
1109 with detector by detector histograms that give a clear overview of the parameters monitored during  
1110 the data acquisition [48]. In this appendix, details about this software in the context of GIF++, as of  
1111 how the software was written and how it functions will be given.

### 1112 B.1 GIF++ Offline Analysis file tree

1113 GIF++ Offline Analysis source code is fully available on github at [https://github.com/afagot/GIF\\_OfflineAnalysis](https://github.com/afagot/GIF_OfflineAnalysis). The software requires ROOT as non-optionnal dependency  
1114 as it takes ROOT files in input and write an output ROOT file containing histograms. To compile the  
1115 GIF++ Offline Analysis project is compiled with cmake. To compile, first a build/ directory must  
1116 be created to compile from there:

```
1118 mkdir build
1119 cd build
1120 cmake ..
1121 make
1122 make install
```

1120 To clean the directory and create a new build directory, the bash script cleandir.sh can be used:

```
1121
1122 ./cleandir.sh
```

1123 The source code tree is provided below along with comments to give an overview of the files' con-  
1124 tent. The different objects created for this project (`Infrastructure`, `Trolley`, `RPC`, `Mapping`, `RPCHit`,  
1125 `RPCCluster` and `Inifile`) will be described in details in the following sections.

1126

```

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..... DEFINITION OF GETCURRENT ANALYSIS MACRO
        │   ├── GIFTrolley.cc..... DEFINITION OF OBJECT TROLLEY
        │   ├── Infrastructure.cc..... DEFINITION OF OBJECT INFRASTRUCTURE
        │   ├── IniFile.cc..... DEFINITION OF OBJECT INI FILE FORINI PARSER
        │   ├── main.cc..... MAIN FILE
        │   ├── Mapping.cc..... DEFINITION OF OBJECT MAPPING
        │   ├── MsgSvc.cc..... DECLARATION OF OFFLINE LOG MESSAGES
        │   ├── OfflineAnalysis.cc..... DECLARATION OF DATA ANALYSIS MACRO
        │   ├── RPCTracker.cc..... DECLARATION OF OBJECT RPC
        │   ├── RPCHit.cc..... DECLARATION OF OBJECT RPCHIT
        │   └── utils.cc..... DEFINITION OF USEFUL FUNCTIONS
        ├── cleandir.sh..... BASH SCRIPT TO CLEAN BUILD DIRECTORY
        ├── CMakeLists.txt..... SET OF INSTRUCTIONS FOR CMAKE
        ├── config.h.in..... DEFINITION OF VERSION NUMBER
        └── README.md..... README FILE FOR GITHUB

```

1127

## B.2 Usage of the Offline Analysis

1128

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:

1130

1131

```
Scan00XXXX_HVY
```

1132

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

1133

1134 of the GIF++ webDCS. Usually, the offline analysis tool is automatically called by the webDCS at  
 1135 the end of data taking or by a user from the webDCS panel if an update of the tool was brought.  
 1136 Nonetheless, an expert can locally launch the analysis for tests on the GIF++ computer, or a user can  
 1137 get the code on its local machine from github and download data from the webDCS for its own anal-  
 1138 ysis. To launch the code, the following command can be used from the `GIF_OfflineAnalysis` folder:

1139  
 1140     `bin/offlineanalysis /path/to/Scan00XXXX_HVY`

1141 where, `/path/to/Scan00XXXX_HVY` refers to the local data files. Then, the offline tool will by itself  
 1142 take care of finding all available ROOT data files present in the folder, as listed below:

- 1143
  - 1144         ● `Scan00XXXX_HVY_DAQ.root` containing the TDC data as described in Appendix ?? (events, hit  
     and timestamp lists), and
  - 1145         ● `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.

1148 **B.2.1 Output of the offline tool**

1149 **B.2.1.1 ROOT file**

1150 The analysis gives in output ROOT datafiles that are saved into the data folder and called using the  
 1151 naming convention `Scan00XXXX_HVY_Offline.root`. Inside those, a list of `TH1` histograms can be  
 1152 found. Its size will vary as a function of the number of detectors in the setup as each set of histograms  
 1153 is produced detector by detector. For each partition of each chamber, can be found:

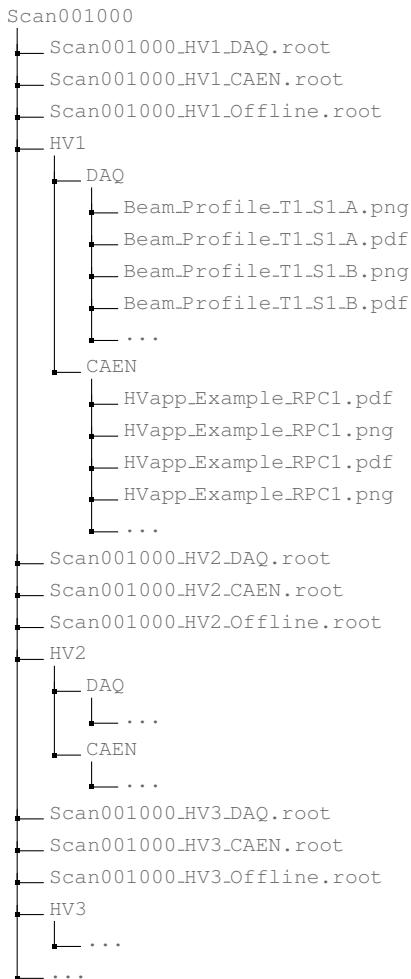
- 1154
  - 1155         ● `Time_Profile_Tt_Sc_p` shows the time profile of all recorded events (number of events per  
     time bin),
  - 1156         ● `Hit_Profile_Tt_Sc_p` shows the hit profile of all recorded events (number of events per chan-  
     nel),
  - 1158         ● `Hit_Multiplicity_Tt_Sc_p` shows the hit multiplicity (number of hits per event) of all recorded  
     events (number of occurrences per multiplicity bin),
  - 1160         ● `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,
  - 1164         ● `Strip_Activity_Tt_Sc_p` shows noise/gamma activity for each strip (normalised version of  
     previous histogram - strip activity = strip rate / average partition rate),
  - 1166         ● `Strip_Homogeneity_Tt_Sc_p` shows the *homogeneity* of a given partition ( $\text{homogeneity} = \exp(-\text{strip rates standard deviation(strip rates in partition/average partition rate)})$ ),
  - 1168         ● `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,

- 1172     ● `mask_Strip_Activity_Tt_Sc_p` shows noise/gamma activity per unit area for each masked  
1173       strip with respect to the average rate of active strips,
- 1174     ● `NoiseCSize_H_Tt_Sc_p` shows noise/gamma cluster size, a cluster being constructed out of  
1175       adjacent strips giving a signal at the *same time* (hits within a time window of 25 ns),
- 1176     ● `NoiseCMult_H_Tt_Sc_p` shows noise/gamma cluster multiplicity (number of reconstructed  
1177       clusters per event),
- 1178     ● `Chip_Mean_Noise_Tt_Sc_p` shows the same information than `Strip_Mean_Noise_Tt_Scp` us-  
1179       ing a different binning (1 chip corresponds to 8 strips),
- 1180     ● `Chip_Activity_Tt_Sc_p` shows the same information than `Strip_Activity_Tt_Scp` using  
1181       chip binning,
- 1182     ● `Chip_Homogeneity_Tt_Sc_p` shows the homogeneity of a given partition using chip binning,
- 1183     ● `Beam_Profile_Tt_Sc_p` shows the estimated beam profile when taking efficiency scan. This  
1184       is obtained by filtering `Time_Profile_Tt_Sc_p` to only consider the muon peak where the  
1185       noise/gamma background has been subtracted. The resulting hit profile corresponds to the  
1186       beam profile on the detector channels,
- 1187     ● `L0_Efficiency_Tt_Sc_p` shows the level 0 efficiency that was estimated **without** muon track-  
1188       ing,
- 1189     ● `MuonCSize_H_Tt_Sc_p` shows the level 0 muon cluster size that was estimated **without** muon  
1190       tracking, and
- 1191     ● `MuonCMult_H_Tt_Sc_p` shows the level 0 muon cluster multiplicity that was estimated **without**  
1192       muon tracking.

1193     In the histogram labels,  $t$  stands for the trolley number (1 or 3),  $c$  for the chamber slot label in  
1194     trolley  $t$  and  $p$  for the partition label (A, B, C or D depending on the chamber layout) as explained  
1195     in Chapter ??.

1196     In the context of GIF++, an extra script called by the webDCS is called to extract the histograms  
1197     from the ROOT files. The histograms are then stored in PNG and PDF formats into the correspond-  
1198     ing folder (a single folder per HV step, so per ROOT file). the goal is to then display the histograms  
1199     on the Data Quality Monitoring (DQM) page of the webDCS in order for the users to control the  
1200     quality of the data taking at the end of data taking. An example of histogram organisation is given  
1201     below:

1203



**1204      Here can put some screens from the webDCS to show the DQM and the plots available to users.**

**1205**

### **1206      B.2.1.2 CSV files**

**1207** Moreover, up to 3 CSV files can be created depending on which ones of the 3 input files were in the  
**1208** data folder:

- 1209**     • `Offline-Corrupted.csv`, is used to keep track of the amount of data that was corrupted and  
**1210**       removed from old data format files that don't contain any data quality flag.
- 1211**     • `Offline-Current.csv`, contains the summary of the currents and voltages applied on each  
**1212**       RPC HV channel.
- 1213**     • `Offline-L0-EffC1.csv`, is used to write the efficiencies, cluster size and cluster multiplicity  
**1214**       of efficiency runs. Note that `L0` refers here to *Level 0* and means that the results of efficiency and  
**1215**       clusterization are a first approximation calculated without performing any muon tracking in

1216 between the different detectors. This offline tool provides the user with a preliminar calculation  
 1217 of the efficiency and of the muon event parameters. Another analysis software especially  
 1218 dedicated to muon tracking is called on selected data to retrieve the results of efficiency and  
 1219 muon clusterization using a tracking algorithm to discriminate noise or gamma from muons  
 1220 as muons are the only particles that pass through the full setup, leaving hits than can be used  
 1221 to reconstruct their tracks.

- 1222 • `Offline-Rate.csv`, is used to write the noise or gamma rates measured in the detector readout  
 1223 partitions.

1224 Note that these 4 CSV files are created along with their *headers* (`Offline-[...]-Header.csv`  
 1225 containing the names of each data columns) and are automatically merged together when the offline  
 1226 analysis tool is called from the webDCS, contrary to the case where the tool is runned locally from  
 1227 the terminal as the merging bash script is then not called. Thus, the resulting files, used to make  
 1228 official plots, are:

- 1229 • `Corrupted.csv`,  
 1230 • `Current.csv`,  
 1231 • `L0-EffCl.csv`.  
 1232 • `Rate.csv`.

### 1233 B.3 Analysis inputs and information handling

1234 The usage of the Offline Analysis tool as well as its output have been presented in the previous section.  
 1235 It is now important to dig further and start looking at the source code and the inputs necessary  
 1236 for the tool to work. Indeed, other than the raw ROOT data files that are analysed, more information  
 1237 needs to be imported inside of the program to perform the analysis such as the description of the  
 1238 setup inside of GIFT++ at the time of data taking (number of trolleys, of RPCs, dimensions of the  
 1239 detectors, etc...) or the mapping that links the TDC channels to the coresponding RPC channels in  
 1240 order to translate the TDC information into human readable data. 2 files are used to transmit all this  
 1241 information:

- 1242  
 1243 • `Dimensions.ini`, that provides the necessary setup and RPC information, and  
 1244 • `ChannelsMapping.csv`, that gives the link between the TDC and RPC channels as well as the  
 1245 *mask* for each channel (masked or not?).

#### 1246 B.3.1 Dimensions file and InFile parser

1247 This input file, present in every data folder, allows the analysis tool to know of the number of active  
 1248 trolleys, the number of active RPCs in those trolleys, and the details about each RPCs such as  
 1249 the number of RPC gaps, the number of pseudo-rapidity partitions (for CMS-like prototypes), the  
 1250 number of strips per partion or the dimensions. To do so, there are 3 types of groups in the INI file  
 1251 architecture. A first general group, appearing only once at the head of the document, gives information  
 1252 about the number of active trolleys as well as their IDs, as presented in Source Code B.1. For

1253 each active trolley, a group similar to Source Code B.2 can be found containing information about  
 1254 the number of active detectors in the trolley and their IDs. Each trolley group as a `Tt` name format,  
 1255 where `t` is the trolley ID. Finally, for each detector stored in slots of an active trolley, there is a group  
 1256 providing information about their names and dimensions, as shown in Source Code B.3. Each slot  
 1257 group as a `TtSs` name format, where `s` is the slot ID of trolley `t` where the active RPC is hosted.

```
1258 [General]
1259 nTrolleys=2
  TrolleysID=13
```

1260 *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.*

```
1261 [T1]
  nSlots=4
  SlotsID=1234
```

1262 *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.*

```
1263 [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
```

1264 *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.*

1265 This information is readout and stored in a C++ object called `IniFile`, that parses the information  
 1266 in the INI input file and stores it into a local buffer for later use. This INI parser is the exact same  
 1267 one that was previously developed for the Gif++ DAQ and described in Appendix A.5.2.

### 1268 B.3.2 TDC to RPC link file and Mapping

1269 The same way the INI dimension file information is stored using `map`, the channel mapping and mask  
 1270 information is stored and accessed through `map`. First of all, the mapping CSV file is organised into  
 1271 3 columns separated by tabulations (and not by commas, as expected for CSV files as it is easier using  
 1272 streams to read tab or space separated data using C++):

1273

1274       RPC\_channel           TDC\_channel           mask

1275       using as formatting for each field:

1276       TSCCC           TCCC           M

1278       TSCCC is a 5-digit integer where  $T$  is the trolley ID,  $s$  the slot ID in which the RPC is held insite  
 1279       the trolley  $T$  and  $ccc$  is the RPC channel number, or *strip* number, that can take values up to  
 1280       3-digits depending on the detector,

1281       TCCC is a 4 digit integer where  $T$  is the TDC ID,  $ccc$  is the TDC channel number that can take values  
 1282       in between 0 and 127, and

1283       M is a 1-digit integer indicating if the channel should be considered ( $M = 1$ ) or discarded ( $M = 0$ )  
 1284       during analysis.

1285       This mapping and masking information is readout and stored thanks to the object `Mapping`, pre-  
 1286       sented in Source Code B.4. Similarly to `IniFile` objects, this class has private methods. The first  
 1287       one, `Mapping::CheckIfNewLine()` is used to find the newline character '`\n`' or return character  
 1288       '`\r`' (depending on which kind of operating system interacted with the file). This is used for the  
 1289       simple reason that the masking information has been introduced only during the year 2017 but the  
 1290       channel mapping files exist since 2015 and the very beginning of data taking at GIF++. This means  
 1291       that in the older data folders, before the upgrade, the channel mapping file only had 2 columns, the  
 1292       RPC channel and the TDC channel. For compatibility reasons, this method helps controling the  
 1293       character following the readout of the 2 first fields of a line. In case any end of line character is  
 1294       found, no mask information is present in the file and the default  $M = 1$  is used. On the contrary, if  
 1295       the next character was a tabulation or a space, the mask information is present.

1296       Once the 3 fields have been readout, the second private method `Mapping::CheckIfTDCCh()` is  
 1297       used to control that the TDC channel is an existing TDC channel. Finally, the information is stored  
 1298       into 3 different maps (`Link`, `ReverseLink` and `Mask`) thanks to the public method `Mapping::Read()`.  
 1299       `Link` allows to get the RPC channel by knowing the TDC channel while `ReverseLink` does the op-  
 1300       posite by returning the TDC channel by knowing the RPC channel. Finally, `Mask` returns the mask  
 1301       associated to a given RPC channel.

```

1302 typedef map<Uint,Uint> MappingData;
1303
1304 class Mapping {
1305     private:
1306         bool          CheckIfNewLine(char next);
1307         bool          CheckIfTDCCh(Uint channel);
1308         string        FileName;
1309         MappingData  Link;
1310         MappingData  ReverseLink;
1311         MappingData  Mask;
1312         int           Error;
1313
1314     public:
1315         Mapping();
1316         Mapping(string baseName);
1317         ~Mapping();
1318
1319         void SetFileName(const string filename);
1320         int  Read();
1321         Uint GetLink(Uint tdcchannel);
1322         Uint GetReverse(Uint rpcchannel);
1323         Uint GetMask(Uint rpcchannel);
1324     };

```

1304 *Source Code B.4: Description of C++ object Mapping used as a parser for the channel mapping and mask file.*

## 1305 B.4 Description of GIF++ setup within the Offline Analysis tool

1306 In the previous section, the tool input files have been discussed. The dimension file information is  
 1307 stored in a map hosted by the `IniFile` object. But this information is then used to create a series of  
 1308 new objects that helps defining the GIF++ infrastructure directly into the Offline Analysis. Indeed,  
 1309 from the `RPC`, to the more general `Infrastructure`, every element of the GIF++ infrastrucutre is  
 1310 recreated for each data analysis based on the information provided in input. All this information  
 1311 about the infrastructure will be used to assign each hit signal to a specific strip channel of a specific  
 1312 detector, and having a specific active area. This way, rate per unit area calculation is possible.  
 1313

### 1314 B.4.1 RPC objects

1315 `RPC` objects have been developped to represent physical active detectors in GIF++ at the moment  
 1316 of data taking. Thus, there are as many `RPC` objects created during the analysis than there were  
 1317 active `RPCs` tested during a run. Each `RPC` hosts the information present in the corresponding INI  
 1318 slot group, as shown in B.3, and organises it using a similar architecture. This can be seen from  
 1319 Source B.5.

1320 To make the object more compact, the lists of gap labels, of gap active areas and strip active  
 1321 areas are stored into `vector` dynamical containers. `RPC` objects are always contructed thanks to the  
 1322 dimension file information stored into the `IniFILE` and their ID, using the format `TtSs`. Using the  
 1323 `RPC` ID, the constructor calls the methods of `IniFILE` to initialise the `RPC`. The other constructors  
 1324 are not used but exist in case of need. Finally, some getters have been written to access the different  
 1325 private parameters storing the detector information.

```

1326 class RPC{
1327     private:
1328         string      name;           //RPC name as in webDCS database
1329         Uint        nGaps;          //Number of gaps in the RPC
1330         Uint        nPartitions;    //Number of partitions in the RPC
1331         Uint        nStrips;        //Number of strips per partition
1332         vector<string> gaps;       //List of gap labels (BOT, TOP, etc...)
1333         vector<float>  gapGeo;         //List of gap active areas
1334         vector<float>  stripGeo;       //List of strip active areas
1335
1336     public:
1337         RPC();
1338         RPC(string ID, IniFile* geofile);
1339         RPC(const RPC& other);
1340         ~RPC();
1341         RPC& operator=(const RPC& other);
1342
1343         string GetName();
1344         Uint GetNGaps();
1345         Uint GetNPartitions();
1346         Uint GetNStrips();
1347         string GetGap(Uint g);
1348         float GetGapGeo(Uint g);
1349         float GetStripGeo(Uint p);
1350     };

```

1328 *Source Code B.5: Description of C++ objects RPC that describe each active detectors used during data taking.*

## 1329 B.4.2 Trolley objects

1330 Trolley objects have been developped to represent physical active trolleys in GIFT++ at the moment  
 1331 of data taking. Thus, there are as many trolley objects created during the analysis than there were  
 1332 active trolleys hosting tested RPCs during a run. Each Trolley hosts the information present in the  
 1333 corresponding INI trolley group, as shown in B.2, and organises it using a similar architecture. In  
 1334 addition to the information hosted in the INI file, these object have a dynamical container of RPC  
 1335 objects, representing the active detectors the active trolley was hosting at the time of data taking.  
 1336 This can been seen from Source Code B.6.

1337 Trolley objects are always contructed thanks to the dimension file information stored into the  
 1338 IniFILE and their ID, using the format Tt. Using the Trolley ID, the constructor calls the methods  
 1339 of IniFILE to initialise the Trolley. Retrieving the information of the RPC IDs via SlotsID, a new  
 1340 RPC is constructed and added to the container RPCs for each character in the ID string. The other  
 1341 constructors are not used but exist in case of need. Finally, some getters have been written to access  
 1342 the different private parameters storing the trolley and detectors information.

```

1343 class Trolley{
1344     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
1345
    public:
        //Constructors, destructor and operator =
        Trolley();
        Trolley(string ID, IniFile* geofile);
        Trolley(const Trolley& other);
        ~Trolley();
        Trolley& operator=(const Trolley& other);
1346
        //Get GIFTrolley members
        Uint  GetNSlots();
        string GetSlotsID();
        Uint   GetSlotID(Uint s);
1347
        //Manage RPC list
        RPC*  GetRPC(Uint r);
        void  DeleteRPC(Uint r);
1348
        //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);
    };

```

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

### 1346 B.4.3 Infrastructure object

1347 The `Infrastructure` object has been developped to represent the GIFT++ bunker area dedicated to  
 1348 CMS RPC experiments. With this very specific object, all the information about the CMS RPC  
 1349 setup within GIFT++ at the moment of data taking is stored. It hosts the information present in the  
 1350 corresponding INI general group, as shown in B.1, and organises it using a similar architecture. In  
 1351 addition to the information hosted in the INI file, this object have a dynamical container of `Trolley`  
 1352 objects, representing the active tolleys in GIFT++ area. This can be seen from Source Code B.7.

1353 The `Infrastructure` object is always contructed thanks to the dimension file information stored  
 1354 into the `IniFILE`. Retrieving the information of the trolley IDs via `TrolleysID`, a new `Trolley` is  
 1355 constructed and added to the container `Trolleys` for each character in the ID `string`. By extension,  
 1356 it is easy to understand that the process described in Section B.4.2 for the construction of RPCs  
 1357 takes place when a trolley is constructed. The other constructors are not used but exist in case of  
 1358 need. Finally, some getters have been written to access the different private parameters storing the  
 1359 infrastructure, tolleys and detectors information.

```

1360
class Infrastructure {
    private:
        Uint             nTrolleys;   //Number of active Trolleys in the run
        string          TrolleysID; //Active trolley IDs written into a string
        vector<Trolley*> Trolleys; //List of active Trolleys (struct)

    public:
        //Constructors and destructor
        Infrastructure();
        Infrastructure(IniFile* geofile);
        Infrastructure(const Infrastructure& other);
        ~Infrastructure();
        Infrastructure& operator=(const Infrastructure& other);

        //Get Infrastructure members
        Uint  GetNTrolleys();
        string GetTrolleysID();
        Uint  GetTrolleyID(Uint t);

1361
        //Manage Trolleys
        Trolley* GetTrolley(Uint t);
        void     DeleteTrolley(Uint t);

        //Methods to get members of GIFTrolley objects stored in Trolleys
        Uint  GetNSlots(Uint t);
        string GetSlotsID(Uint t);
        Uint  GetSlotID(Uint t, Uint s);
        RPC*  GetRPC(Uint t, Uint r);

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

```

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

## 1363 B.5 Handeling of data

1364 As discussed in Appendix A.4.2, the raw data as a `TTree` architecture where every entry is related to  
 1365 a trigger signal provided by a muon or a random pulse, whether the goal of the data taking was to  
 1366 measure the performance of the detector or the noise/gamma background respectively. Each of these  
 1367 entries, referred also as events, contain a more or less full list of hits in the TDC channels to which  
 1368 the detectors are connected. To this list of hits corresponds a list of time stamps, marking the arrival  
 1369 of the hits within the TDC channel.

1370 The infrastructure of the CMS RPC experiment within GIF++ being defined, combining the  
 1371 information about the raw data with the information provided by both the mapping/mask file and the  
 1372 dimension file allows to build new physical objects that will help in computing efficiency or rates.

### 1373 B.5.1 RPC hits

1374 The raw data stored in the ROOT file as output of the GIFT++ DAQ, is readout by the analysis tool  
 1375 using the structure `RAWData` presented in Source Code B.9 that differs from the structure presented  
 1376 in Appendix A.4.2 as it is not meant to hold all of the data contained in the ROOT file. In this sense,  
 1377 this structure is in the case of the offline analysis tool not a dynamical object and will only be storing  
 1378 a single event contained in a single entry of the `TTree`.

```
1379
  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);
  }
```

1381 *Source Code B.8: Description of C++ object `RPCHit`.*

```
1382
  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
  };
  
```

1383 *Source Code B.9: Description of C++ structure `RAWData`.*

1384 Each member of the structure is then linked to the corresponding branch of the ROOT data tree,  
 1385 as shown in the example of Source Code B.10, and using the method `GetEntry(int i)` of the ROOT  
 1386 class `TTree` will update the state of the members of `RAWData`.

```

1387
1388   TTree* dataTree = (TTree*)dataFile.Get("RAWData");
      RAWData data;

1389   dataTree->SetBranchAddress("EventNumber", &data.iEvent);
   dataTree->SetBranchAddress("number_of_hits", &data.TDCNHits);
   dataTree->SetBranchAddress("Quality_flag", &data.QFlag);
   dataTree->SetBranchAddress("TDC_channel", &data.TDCCh);
   dataTree->SetBranchAddress("TDC_TimeStamp", &data.TDCTS);

```

1389       *Source Code B.10: Example of link in between RAWData and TTree.*

1390       The data is then analysed entry by entry and to each element of the TDC channel list, a `RPCHit` is  
 1391       constructed by linking each TDC channel to the corresponding RPC channel thanks to the `Mapping`  
 1392       object. The information carried by the RPC channel format allows to easily retrieve the trolley and  
 1393       slot from which the hit was recorded (see section B.3.2). Using these 2 values, the readout partition  
 1394       can be found by knowing the strip channel and comparing it with the number of partitions and strips  
 1395       per partition stored into the `Infrastructure` object.

1396       Thus `RPCHit` objects are then stored into 3D dynamical list called `GIFHitList` (Source Code B.9)  
 1397       where the 3 dimensions refer to the 3 layers of the readout in `GIF++` : in the bunker there are *trolleys*  
 1398       ( $\tau$ ) holding detectors in *slots* ( $s$ ) and each detector readout is divided into 1 or more pseudo-rapidity  
 1399       *partitions* ( $p$ ). Using these 3 information allows to assign an address to each readout partition and  
 1400       this address will point to a specific hit list.

1401

## 1402     **B.5.2 Clusters of hits**

1403       All the hits contained in the ROOT file have been sorted into the different hit lists through the  
 1404       `GIFHitList`. At this point, it is possible to start looking for clusters. A cluster is a group of adjacent  
 1405       strips getting hits within a time window of 25 ns. These strips are then assumed to be part of the same  
 1406       physical avalanche signal generated by a muon passing through the chamber or by the interaction of  
 1407       a gamma stopping into the electrodes of the RPCs.

1408       To keep the cluster information, `RPCCluster` objects have been defined as shown in Source  
 1409       Code B.11. Using the information of each individual `RPCHit` taken out of the hit list, it stores  
 1410       the cluster size (number of adjacent strips composing the cluster), the first and last hit, the center for  
 1411       spatial reconstruction and finally the start and stop time stamps as well as te time spread in between  
 1412       the first and last hit.

```

1413
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);

```

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

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

1428 B.6 DAQ data Analysis

1429 All the ingredients to analyse GIF++ data have been defined. This section will focus on the different  
1430 part of the analysis performed on the data, from determining the type of data the tool is dealing with

1431 to calculating the rate in each detector or reconstructing muon or gamma clusters.

### 1432 B.6.1 Determination of the run type

1433 In GIF++, both the performance of the detectors in detecting muons in an irradiated environment and  
 1434 the gamma background can be independantly measured. These corresponds to different run types  
 1435 and thus, to different TDC settings giving different data to look at.

1436  
 1437 In the case of performance measurements, the trigger for data taking is provided by the coïncidence  
 1438 of several scintillators when muons from the beam passing through the area are detected. Data  
 1439 is collected in a 600 ns wide window around the arrival of muons in the RPCs. The expected time  
 1440 distribution of hits is shown in Figure B.1a. The muon peak is clearly visible in the center of the  
 1441 distribution and is to be extracted from the gamma background that composes the flat part of the  
 1442 distribution.

1443 On the other hand, gamma background or noise measurements are focussed on the non muon  
 1444 related physics and the trigger needs to be independant from the muons to give a good measurement  
 1445 of the gamma/noise distribution as seen by the detectors. The trigger is then provided by a pulse  
 1446 generator at a frequency of 300 Hz whose pulse is not likely to be on time with a muon. In order  
 1447 to increase the integrated time without increasing the acquisition time too much, the width of the  
 1448 acquisition windows are increased to 10  $\mu$ s. The time distribution of the hits is expected to be flat, as  
 1449 shown by Figure B.1b.

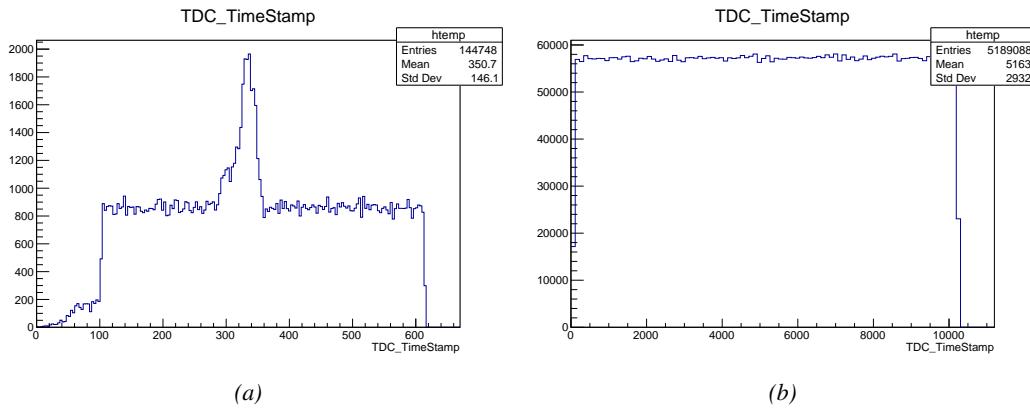


Figure B.1: Example of expected hit time distributions in the cases of efficiency (Figure B.1a) and noise/gamma rate per unit area (Figure B.1b) measurements as extracted from the raw ROOT files. The unit along the x-axis corresponds to ns. The fact that "the" muon peak is not well defined in Figure 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.

1450 The ROOT files include a TTree called RunParameters containing, among other things, the in-  
 1451 formation related to the type of run. The run type can then be accessed as described by Source  
 1452 Code B.12 and the function IsEfficiencyRun() is then used to determine if the run file is an effi-  
 1453 ciency run or, on the contrary, another type of run (noise or gamma measurement).

```

1454     TTree* RunParameters = (TTree*)dataFile.Get("RunParameters");
1455     TString* RunType = new TString();
1456     RunParameters->SetBranchAddress("RunType", &RunType);
1457     RunParameters->GetEntry(0);

```

1456       *Source Code B.12: Access to the run type contained in TTree\* RunParameters.*

1457       Finally, the data files will have a slightly different content whether it was collected before or after  
 1458       October 2017 and the upgrade of the DAQ software that brought a new information into the ROOT  
 1459       output. This is discussed in Appendix A.4.3 and implies that the analysis will differ a little depending  
 1460       on the data format. Indeed, as no information on the data quality is stored, in older data files, the cor-  
 1461       rections for missing events has to be done at the end of the analysis. The information about the type  
 1462       of data format is stored in the variable **bool** `isNewFormat` by checking the list of branches contained  
 1463       in the data tree via the methods `TTree::GetListOfBranches()` and `TCollection::Contains()`.

## 1464       **B.6.2 Beam time window calculation for efficiency runs**

1465       Knowing the run type is important first of all to know the width of the acquisition window to be used  
 1466       for the rate calculation and finally to be able to seek for muons. Indeed, the peak that appears in the  
 1467       time distribution for each detectors is then fitted to extract the most probable time window in which  
 1468       the tool should look for muon hits. The data outside of this time window is then used to evaluate the  
 1469       noise or gamma background the detector was subjected to during the data taking. Computing the  
 1470       position of the peak is done calling the function `SetBeamWindow()` defined in file `src/RPCHit.cc` that  
 1471       loops a first time on the data. The data is first sorted in a 3D array of 1D histograms (`GIFH1Array`, see  
 1472       `include/types.h`). Then the location of the highest bin is determined using `TH1::GetMaximumBin()`  
 1473       and is used to define a window in which a gaussian fit will be applied to compute the peak width.  
 1474       This window is a 80 ns defined by Formula B.1 around the central bin.

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

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

1475       Before the fit is performed, the average number of noise/gamma hits per bin is evaluated using  
 1476       the data outside of the fit window. Excluding the first 100 ns, the average number of hits per bin  
 1477       due to the noise or gamma is defined by Formula B.2 after extracting the amount of hits in the time  
 1478       windows  $[100; t_{low}]$  and  $[t_{high}; 600]$  thanks to the method `TH1::Integral()`. This average number  
 1479       of hits is then subtracted to every bin of the 1D histogram, in order to *clean* it from the noise or  
 1480       gamma contribution as much as possible to improve the fit quality. Bins where  $\langle n_{hits} \rangle$  is greater  
 1481       than the actual bin content are set to 0.

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

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

1482       Finally, the fit parameters are extracted and saved for each detector in 3D arrays of **float**  
 1483       (`muonPeak`, see `include/types.h`), a first one for the mean arrival time of the muons, `PeakTime`,

1484 and a second one for the width of the peak, `PeakWidth`. The width is defined as  $6\sigma$  of the gaussian  
 1485 fit. The same settings are applied to every partitions of the same detector. To determine which one  
 1486 of the detector's partitions is directly illuminated by the beam, the peak height of each partition is  
 1487 compared and the highest one is then used to define the peak settings.

### 1488 B.6.3 Data loop and histogram filling

1489 3D arrays of histogram are created to store the data and display it on the DQM of G4F++ webDCS  
 1490 for the use of shifters. These histograms, presented in section B.2.1.1, are filled while looping on  
 1491 the data. Before starting the analysis loop, it is necessary to control the entry quality for the new  
 1492 file formats featuring `QFlag`. If the `QFlag` value for this entry shows that 1 TDC or more have a  
 1493 `CORRUPTED` flag, then this event is discarded. The loss of statistics is low enough to be neglected.  
 1494 `QFlag` is controlled using the function `IsCorruptedEvent()` defined in `src/utils.cc`. As explained  
 1495 in Appendix A.4.3, each digit of this integer represents a TDC flag that can be 1 or 2. Each 2 is  
 1496 the sign of a `CORRUPTED` state. Then, the data is accessed entry by entry in the ROOT `TTree` using  
 1497 `RAWData` and each hit in the hit list is assigned to a detector channel and saved in the corresponding  
 1498 histograms. In the first part of the analysis, in which the loop over the ROOT file's content is  
 1499 performed, the different steps are:

1500 **1- RPC channel assignment and control:** a check is done on the RPC channel extracted thanks  
 1501 to the mapping via the method `Mapping::GetLink()`. If the channel is not initialised and is 0, or if  
 1502 the TDC channel was greater than 5127, the hit is discarded. This means there was a problem in the  
 1503 mapping. Often a mapping problem leads to the crash of the offline tool.

1504 **2- Creation of a `RPCHit` object:** to easily get the trolley, slot and partition in which the hit has  
 1505 been assigned, this object is particularly helpful.

1506 **3- General histograms are filled:** the hit is filled into the time distribution and the general hit  
 1507 distribution histograms, and if the arrival time is within the first 100 ns, it is discarded and nothing  
 1508 else happens and the loop proceeds with the next hit in the list.

1509 **4- Multiplicity counter:** the hit multiplicity counter of the corresponding detectors incremented.

1510 **5-a- Efficiency runs - Is the hit within the peak window? :** if the peak is contained in the peak  
 1511 window previously defined in section B.6.2, the hit is filled into the beam hit profile histogram of  
 1512 the corresponding chamber, added into the list of muon hits and increments the counter of *in time*  
 1513 hits. The term *in time* here refers to the hits that are likely to be muons by arriving in the expected  
 1514 time window. If the hit is outside of the peak window, it is filled into the noise profile histogram  
 1515 of the corresponding detector, added into the list of noise/gamma hits and increments the counter of  
 1516 noise/gamma hits.

1517 **5-b- Noise/gamma rate runs - Noise histograms are filled:** the hit is filled into the noise profile  
 1518 histogram of the corresponding detector, added into the list of noise/gamma hits and increments the  
 1519 counter of noise/gamma hits.

1521 After the loop on the hit list of the entry is over, the next step is to clusterize the 3D lists filled  
 1522 in the previous steps. A 3D loop is then started over the active trolley, slot and RPC partitions to  
 1523 access these objects. Each `NoiseHitList` and `MuonHitList`, in case of efficiency run, are clusterized  
 1524 as described in section B.5.2. There corresponding cluster size and multiplicity histograms are filled  
 1525 at the end of the clustering process. Then, the efficiency histogram is filled in case of efficiency run.  
 1526 The selection is simply made by checking whether the RPC detected signals in the peak window  
 1527 during this event. Nevertheless, it is useful to highlight that at this level, it is not possible yet to  
 1528 discriminate in between a muon hit and noise or gamma hit. Thus, `MuonCSize_H`, `MuonCMult_H`  
 1529 and `Efficiency0_H` are subjected to noise and gamma contamination. This contamination will be  
 1530 estimated and corrected at the moment the results will be written into output CSV files. Finally, the  
 1531 loop ends on the filling of the general hit multiplicity histogram.

#### 1532 **B.6.4 Results calculation**

1533 As mentioned in section B.2.1, the analysis of DAQ data provides the user with 3 CSV files and  
 1534 a ROOT file associated to each and every ROOT data file. The fourth CSV file is provided by the  
 1535 extraction of the CEAN main frame data monitored during data taking and will be discussed later.  
 1536 After looping on the data in the previous part of the analysis macro, the output files are created and a  
 1537 3D loop on each RPC readout partitions is started to extract the histograms parameters and compute  
 1538 the final results.

1539

##### 1540 **B.6.4.1 Rate normalisation**

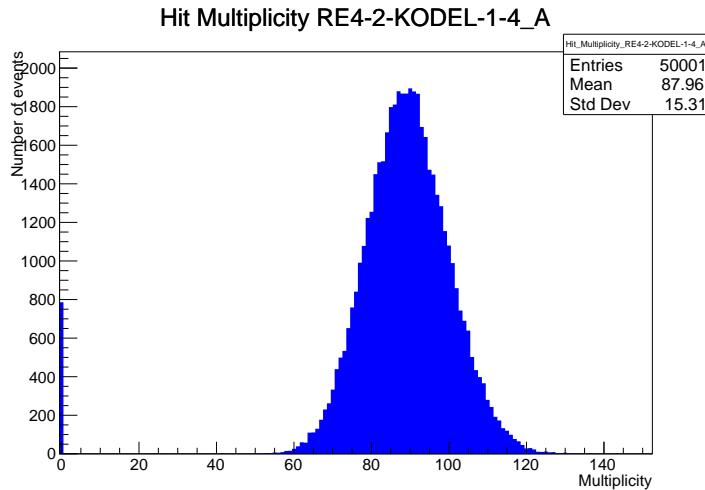


Figure B.2: The effect of the quality flag is explained by presenting the reconstructed hit multiplicity of a data file without `Quality_flag`. The artificial high content of bin 0 is the effect of corrupted data.

1541 To analyse old data format files, not containing any quality flag, it is needed to estimate the amount  
 1542 of corrupted data via a fit as the corrupted data will always fill events with a fake "0 multiplicity".  
 1543 Indeed, as no hits were stored in the DAQ ROOT files, these events artificially contribute to fill  
 1544 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.

```

1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2397
2398
2399
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2489
2490
2491
2492
2493
2494
2495
2496
2497
2497
2498
2499
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2589
2590
2591
2592
2593
2594
2595
2596
2597
2597
2598
2599
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2697
2698
2699
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2797
2798
2799
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2897
2898
2899
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2997
2998
2999
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3097
3098
3099
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3197
3198
3199
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3297
3298
3299
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3497
3498
3499
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3597
3598
3599
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3619
3620
3621

```

- 1576     ● the strip rate, defined as the number of hits recorded in the bin normalised like described in  
 1577           the previous section, using the variable `rate_norm`, and

- 1578     ● the strip activity, defined as the number of hits recorded in the bin normalised to the average  
 1579           number of hits per bin contained in the partition histogram, using the variable `averageNhit`.  
 1580           This value provides an information on the homogeneity of the detector response to the gamma  
 1581           background or of the detector noise. An activity of 1 corresponds to an average response.  
 1582           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);
}

```

1584       *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.*

1585       On each detector partitions, which are readout by a single FEE, all the channels are not processed  
 1586           by the same chip. Each chip can give a different noise response and thus, histograms using a chip  
 1587           binning are used to investigate chip related noise behaviours. The average values of the strip rate  
 1588           or activity grouped into a given chip are extracted using the using the function `GetChipBin()` and  
 1589           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;
}

```

1592       *Source Code B.15: Function used to compute the content of a bin for an histogram using chip binning.*

```

1593   for(UInt ch = 0; ch < (nStripsPart/NSTRIPSCHIP); ch++) {
1594     ChipMeanNoiseProf_H.rpc[T][S][p]->
1595       SetBinContent(ch+1,GetChipBin(StripNoiseProfile_H.rpc[T][S][p],ch));
1596     ChipActivity_H.rpc[T][S][p]->
1597       SetBinContent(ch+1,GetChipBin(StripActivity_H.rpc[T][S][p],ch));
1598   }

```

*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.*

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

```

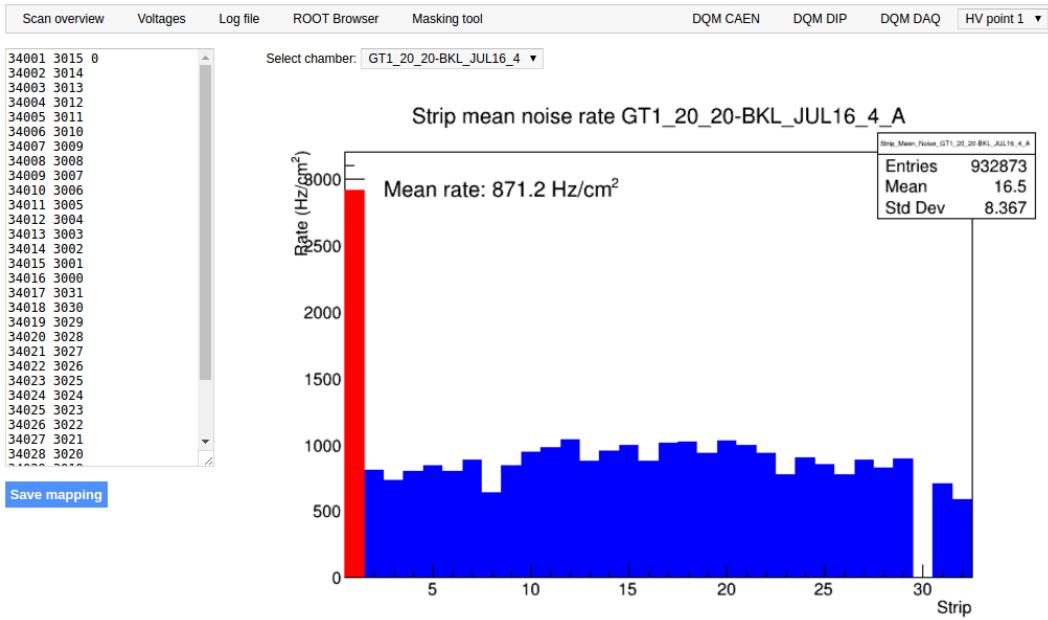
1607   float MeanPartSDev = GetTH1StdDev(StripNoiseProfile_H.rpc[T][S][p]);
1608   float strip_homog = (MeanPartRate==0)
1609     ? 0.
1610     : exp(-MeanPartSDev/MeanPartRate);
1611   StripHomogeneity_H.rpc[T][S][p]->Fill("exp -#left(#frac{\#sigma_{Strip}
1612     \rightarrow Rate}{\#mu_{Strip Rate}}\#right)",strip_homog);
1613   StripHomogeneity_H.rpc[T][S][p]->GetYaxis()->SetRangeUser(0.,1.);
1614
1615   float ChipStDevMean = GetTH1StdDev(ChipMeanNoiseProf_H.rpc[T][S][p]);
1616   float chip_homog = (MeanPartRate==0)
1617     ? 0.
1618     : exp(-ChipStDevMean/MeanPartRate);
1619   ChipHomogeneity_H.rpc[T][S][p]->Fill("exp -#left(#frac{\#sigma_{Chip}
1620     \rightarrow Rate}{\#mu_{Chip Rate}}\#right)",chip_homog);
1621   ChipHomogeneity_H.rpc[T][S][p]->GetYaxis()->SetRangeUser(0.,1.);

```

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

### 1610 B.6.4.3 Strip masking tool

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

1616 From the code point of view, the function `GetTH1Mean()` is used to retrieve the mean rate par-  
 1617 tition by partition after the rates have been calculated strip by strip and filled into the histograms  
 1618 `StripNoiseProfile_H.rpc[T][S][p]`, as described through Source Code B.18.

1619 Once the mask for each rejected channel has been updated, the shifter can manually run the of-  
 1620 fline tool again to update the DQM plots, now including the masked strips, as well the rate results  
 1621 written in the output CSV file `Offline-Rate.csv`. If not done during the shifts, the strip masking  
 1622 procedure needs to be carefully done by the person in charge of data analysis on the scans that were  
 1623 selected to produce the final results.

```

1624
1625     float GetTH1Mean(TH1* H) {
1626         int nBins = H->GetNbinsX();
1627         int nActive = nBins;
1628         float mean = 0.;

1629         for(int b = 1; b <= nBins; b++) {
1630             float value = H->GetBinContent(b);
1631             mean += value;
1632             if(value == 0.) nActive--;
1633         }

1634         if(nActive != 0) mean /= (float)nActive;
1635         else mean = 0.;

1636         return mean;
1637     }

```

*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.

#### 1627 B.6.4.4 Output CSV files filling

1628 All the histograms have been filled. Parameters will then be extracted from them to compute the  
 1629 final results that will later be used to produce plots. Once the results have been computed, the very  
 1630 last step of the offline macro is to write these values into the corresponding CSV outputs. Aside of  
 1631 the file `Offline-Corrupted.csv`, 2 CSV files are being written by the macro `offlineAnalysis()`,  
 1632 `Offline-Rates.csv` and `Offline-L0-EffCl.csv` that respectively contain information about noise  
 1633 or gamma rates, cluster size and multiplicity, and about level 0 reconstruction of the detector effi-  
 1634 ciency, muon cluster size and multiplicity. Details on the computation and file writing are respec-  
 1635 tively given in Sources Codes B.19 and B.20.

1636 **Noise/gamma background variables** are computed and written in the output file for each detector  
 1637 partitions. A detector average of the hit and cluster rate is also provided, as shown through Sources  
 1638 Code B.19. The variables that are written for each partition are:

- 1639 • The mean partition hit rate per unit area, `MeanPartRate`, that is extracted from the histogram  
 1640 `StripNoiseProfile_H` as the mean value along the y-axis, as described in section B.6.4.3. No  
 1641 error is recorded for the hit rate as this is considered a single measurement. No statistical error  
 1642 can be associated to it and the systematics are unknown.
- 1643 • The mean cluster size, `cSizePart`, is extracted from the histogram `NoiseCSize_H` and it's  
 1644 statistical error, `cSizePartErr`, is taken to be  $2\sigma$  of the total distribution.
- 1645 • The mean cluster multiplicity per trigger, `cMultPart`, is extracted from the histogram `NoiseCMult_H`  
 1646 and it's statistical error, `cMultPartErr`, is taken to be  $2\sigma$  of the total distribution. It is impor-  
 1647 tant to point to the fact that this variable gives an information that is dependent on the buffer  
 1648 window width used for each trigger for the calculation.
- 1649 • The mean cluster rate per unit area, `ClustPartRate`, is defined as the mean hit rate normalised

1650 to the mean cluster size and it's statistical error, `ClustPartRateErr`, is then obtained using the  
 1651 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';
  }
}

```

1653 *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.*

1654 **Muon performance variables** are computed and written in the output file for each detector parti-  
 1655 tions as shown through Sources Code B.20. The variables that are written for each partition are:

- 1656     ● The muon efficiency, `eff`, extracted from the histogram `Efficiency0_H`. It is reminded that  
1657       this offline tool doesn't include any tracking algorithm to identify muons from the beam and  
1658       only relies on the hits arriving in the time window corresponding to the beam time. The con-  
1659       tent of the efficiency histogram is thus biased by the noise/gamma background contribution  
1660       into this window and is thus corrected by estimating the muon data content in the peak re-  
1661       gion knowing the noise/gamma content in the rate calculation region. Both time windows  
1662       being different, the choice was made to normalise the noise/gamma background calculation  
1663       window to it's equivalent beam window in order to have comparable values using the variable  
1664       `windowRatio`. Finally, to estimate the data ratio in the peak region, the variable `DataRatio`  
1665       is defined as the ratio in between the estimated mean cluster multiplicity of the muons in the  
1666       peak region, `MuonCM`, and of the total mean cluster multiplicity in the peak region, `PeakCM`.  
1667       `MuonCM` is itself defined as the difference in between the total mean cluster multiplicity in the  
1668       peak region and the normalised mean noise/gamma cluster multiplicity calculated outside of  
1669       the peak region. The statistical error related to the efficiency, `eff_err`, is computed using a  
1670       binomial distribution, as the efficiency measure the probability of "success" and "failure" to  
1671       detect muons.
- 1672     ● The mean muon cluster size, `MuonCS`, is calculated using the total mean cluster size and multi-  
1673       plicity in the peak region, respectively extracted from histograms `MuonCSize_H` and `MuonCMult_H`,  
1674       the noise/gamma background mean cluster size and normalised multiplicity, extracted from  
1675       `NoiseCSize_H` and `NoiseCMult_H`, and of the estimated muon cluster multiplicity `MuonCM` pre-  
1676       viously explicated. The associated statistical error, `MuonCM_err`, is calculated using the propa-  
1677       gation of errors of the mentioned variables.
- 1678     ● The mean muon cluster multiplicity in the peak region, `MuonCM`, explicated above whose sta-  
1679       tistical error, `MuonCM_err`, is the sum of statistical error associated to the total mean clus-  
1680       ter multiplicity in the peak reagion, `PeakCM_err`, and of the mean noise/gamma cluster size,  
1681       `NoiseCM_err`.

1682       In addition to these 2 CSV files, the histograms are saved in ROOT file `Scan00XXXX_HVY_Offline.root`  
1683       as explained in section B.2.1.1.

1684

```

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';
        }
    }
}

```

1685

*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.*

1686

## 1687    B.7 Current data Analysis

1688 Detectors under test at GIF++ are connected both to a CAEN HV power supply and to a CAEN  
1689 ADC that reads the currents inside of the RPC gaps bypassing the supply cable. During data tak-  
1690 ing, the webDCS records into a ROOT file called `Scan00XXXX_HVY_CAEN.root` histograms with the  
1691 monitored parameters of both CAEN devices. Are recorded for each RPC channels (in most cases,  
1692 a channel corresponds to an RPC gap):

- 1693     • the effective voltage,  $HV_{eff}$ , set by the webDCS using the PT correction on the CAEN power  
1694       supply,
- 1695     • the applied voltage,  $HV_{app}$ , monitored by the CAEN power supply, and the statistical error  
1696       related to the variations of this value through time to follow the variation of the environmental  
1697       parameters defined as the RMS of the histogram divided by the square root of the number of  
1698       recorded points,
- 1699     • the monitored current,  $I_{mon}$ , monitored by the CAEN power supply, and the statistical error  
1700       related to the variations of this value through time to follow the variation of the environmental  
1701       parameters defined as the RMS of the histogram divided by the square root of the number of  
1702       recorded points,
- 1703     • the corresponding current density,  $J_{mon}$ , defined as the monitored current per unit area,  
1704        $J_{mon} = I_{mon}/A$ , where  $A$  is the active area of the corresponding gap,
- 1705     • the ADC current,  $I_{ADC}$ , recorded through the CAEN ADC module that monitors the dark  
1706       current in the gap itself. First of all, the resolution of such a module is better than that of  
1707       CAEN power supplies and moreover, the current is not read-out through the HV supply line  
1708       but directly at the chamber level giving the real current inside of the detector. The statistical  
1709       error is defined as the RMS of the histogram distribution divided by the square root of the  
1710       number of recorded points.

1711     Once extracted through a loop over the element of GIF++ infrastructure via the C++ macro  
1712     `GetCurrent()`, these parameters, organised in 9 columns per detector HV supply line, are written in  
1713     the output CSV file `Offline-Current.csv`. The macro can be found in the file `Current.cc`.

# List of Figures

1714

1715	2.1	CERN accelerator complex. . . . .	2-2
1716	2.2	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. . . . .	2-3
1717			
1718	2.3	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). . . . .	2-4
1719			
1720	2.4	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. . . . .	2-5
1721			
1722			
1723	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
1724			
1725	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 $SF_6$ [19]. . . . .	3-3
1726			
1727	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
1728			
1729	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
1730			
1731	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
1732			
1733			
1734			
1735			
1736			
1737			
1738			
1739			
1740			
1741			
1742			
1743			
1744			
1745			
1746			
1747			

1748	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
1749		
1750		
1751		
1752		
1753		
1754	3.7 Comparison of performance of CMS double and single gap RPCs using cosmic muons [30]. Figure 3.7a: Comparison of efficiency sigmoid. Figure 3.7b: Voltage distribution at 95% of maximum efficiency. Figure 3.7c: $\Delta_{10\%}^{90\%}$ distribution. . . . .	3-7
1755		
1756		
1757		
1758	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 [31]. . . . .	3-8
1759		
1760		
1761	3.9 Particle identification applied to electrons in the STAR experiment. The identification is performed combining ToF and $dE/dx$ measurements [36]. . . . .	3-9
1762		
1763		
1764	3.10 Comparison of the detector performance of ALICE ToF MRPC [37] 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
1765		
1766		
1767	3.11 Ratio between total induced and drifting charge have been simulated for single gap, double-gap and multigap layouts [38]. The total induced charge for a double-gap RPC is a factor 2 higher than for a multigap. . . . .	3-10
1768		
1769		
1770	3.12 Charge spectra have been simulated for single gap, double-gap and multigap layouts [38]. 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
1771		
1772		
1773	3.13 The maximal theoretical efficiency is simulated for single gap, double-gap and multigap layouts [38] at a constant gap thickness of 2 mm and using an effective Townsend coefficient of 9 $\text{mm}^{-1}$ . . . . .	3-11
1774		
1775		
1776		
1777		
1778	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
1779		
1780		
1781		
1782		
1783		
1784		
1785		
1786	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
1787		
1788		
1789		
1790	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
1791		
1792	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

---

LIST OF FIGURES

1793	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.	4-7
1794		A crane allows installation of heavy equipment in the area. . . . .	
1795	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
1796			
1797	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
1798			
1799	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
1800			
1801	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
1802			
1803	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
1804			
1805	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
1806			
1807	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
1808			
1809	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
1810			
1811	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
1812			
1813			
1814			
1815			
1816			
1817			
1818			
1819			
1820			
1821			
1822			
1823			
1824			
1825			
1826			
1827			
1828			
1829			
1830			
1831			
1832			
1833			
1834			
1835			
1836			
1837			
1838			

1839	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 . . . . .	4-19
1840			
1841			
1842			
1843	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. . . . .	4-21
1844			
1845			
1846			
1847			
1848			
1849	4.17	. . . . .	4-22
1850	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. . . . .	4-24
1851			
1852			
1853	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. . . . .	4-24
1854			
1855			
1856	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. . . . .	4-25
1857			
1858			
1859			
1860			
1861			
1862	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. . . . .	4-25
1863			
1864			
1865	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. . . . .	4-26
1866			
1867	4.23	Evolution of the noise rate per unit area for the irradiated chamber RE2-2-BARC-9 only. . . . .	4-26
1868			
1869	A.1	(A.1a) View of the front panel of a V1190A TDC module [44]. (A.1b) View of the front panel of a V1718 Bridge module [45]. (A.1c) View of the front panel of a 6U 6021 VME crate [46]. . . . .	A-3
1870			
1871			
1872	A.2	Module V1190A <i>Trigger Matching Mode</i> timing diagram [44]. . . . .	A-4
1873	A.3	Structure of the ROOT output file generated by the DAQ. The 5 branches ( <code>EventNumber</code> , <code>number_of_hits</code> , <code>Quality_flag</code> , <code>TDC_channel</code> and <code>TDC_TimeStamp</code> ) are visible on the left panel of the ROOT browser. On the right panel is visible the histogram corresponding to the variable <code>nHits</code> . 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 <code>TTree</code> . . . . .	A-10
1874			
1875			
1876			
1877			
1878			

---

LIST OF FIGURES

1879	A.4	The effect of the quality flag is explained by presenting the content of <code>TBranch number_of_hits</code> of a data file without <code>Quality_flag</code> in Figure A.4a and the content of the same <code>TBranch</code> for data corresponding to a <code>Quality_flag</code> where all TDCs were labelled as <code>GOOD</code> 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.	A-12
1880	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
1881	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
1882	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.	B-16
1883	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
1884	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
1885			
1886			
1887			
1888			
1889			
1890			
1891			
1892			
1893			
1894			
1895			
1896			
1897			
1898			
1899			
1900			
1901			
1902			
1903			
1904			
1905			
1906			
1907			
1908			
1909			
1910			
1911			
1912			
1913			
1914			
1915			
1916			



# List of Tables

1917

1918	3.1 Properties of the most used electrode materials for RPCs. . . . .	3-4
1919	4.1 Total photon flux ( $E\gamma \leq 662$ keV) with statistical error predicted considering a	
1920	$^{137}\text{Cs}$ activity of 740 GBq at different values of the distance $D$ to the source along	
1921	the x-axis of irradiation field [41]. . . . .	4-16
1922	4.2 Correction factor $c$ is computed thanks to formulae 4.5 taking as reference $D_0 =$	
1923	50 cm and the associated flux $F_0^{ABS}$ for each absorption factor available in table 4.1. .	4-18
1924	4.3 The data at $D_0$ in 1997 is taken from [41]. In a second step, using Equations 4.8	
1925	and 4.9, the flux at $D$ can be estimated in 1997. Then, taking into account the	
1926	attenuation of the source activity, the flux at $D$ can be estimated at the time of the	
1927	tests in GIF in 2014. Finally, assuming a sensitivity of the RPC to $\gamma$ $s = 2 \cdot 10^{-3}$ ,	
1928	an estimation of the hit rate per unit area is obtained. . . . .	4-20
1929	A.1 Inter-process communication cycles in between the webDCS and the DAQ through	
1930	file string signals. . . . .	A-19



1931

# List of Acronyms

1932

## List of Acronyms

1933

1934

1935

### A

1936

1937 AFL

Almost Full Level

1938

1939

1940

### B

1941

1942 BARC

Bhabha Atomic Research Centre

1943 BLT

Block Transfer

1944 BR

Branching Ratio

1945

1946

### C

1948

1949 CAEN

Costruzioni Apparecchiature Elettroniche Nucleari S.p.A.

1950 CERN

European Organization for Nuclear Research

1951 CFD

Constant Fraction Discriminator

1952 CMS

Compact Muon Solenoid

1953 CSC

Cathode Strip Chamber

1954

1955

### D

1956

1958 DAQ

Data Acquisition

1959 DCS

Detector Control Software

1960 DQM

Data Quality Monitoring

1961 DT

Drift Tube

1962

1963

### F

1964

1965

---

LIST OF ACRONYMS

1966	FEE	Front-End Electronics
1967	FEB	Front-End Board
1968		
1969		
1970	<b>G</b>	
1971		
1972	GE-/-	Find a good description
1973	GE1/1	Find a good description
1974	GE2/1	Find a good description
1975	GEANT	GEometry ANd Tracking - a series of software toolkit platforms developed by CERN
1976		
1977	GEM	Gas Electron Multiplier
1978	GIF	Gamma Irradiation Facility
1979	GIF++	new Gamma Irradiation Facility
1980		
1981		
1982	<b>H</b>	
1983		
1984	HL-LHC	High Luminosity LHC
1985	HPL	High-pressure laminate
1986	HV	High Voltage
1987		
1988		
1989	<b>I</b>	
1990		
1991	iRPC	improved RPC
1992	IRQ	Interrupt Request
1993	ISR	Intersecting Storage Rings
1994		
1995		
1996	<b>L</b>	
1997		
1998	LEP	Large Electron-Positron
1999	LHC	Large Hadron Collider
2000	LS1	First Long Shutdown
2001	LS3	Third Long Shutdown
2002	LV	Low Voltage
2003	LVDS	Low-Voltage Differential Signaling
2004		
2005		
2006	<b>M</b>	
2007		

---

## LIST OF ACRONYMS

---

2008	<b>MC</b>	Monte Carlo
2009	<b>MCNP</b>	Monte Carlo N-Particle
2010	<b>ME-/-</b>	Find good description
2011	<b>ME0</b>	Find good description
2012	<b>MRPC</b>	Multigap RPC
2013		
2014		
2015	<b>N</b>	
2016		
2017	<b>NIM</b>	Nuclear Instrumentation Module logic signals
2018		
2019		
2020	<b>P</b>	
2021		
2022	<b>PS</b>	Proton Synchrotron
2023	<b>PMT</b>	PhotoMultiplier Tube
2024		
2025		
2026	<b>R</b>	
2027		
2028	<b>RE-/-</b>	Find a good description
2029	<b>RE2/2</b>	Find a good description
2030	<b>RE3/1</b>	Find a good description
2031	<b>RE3/2</b>	Find a good description
2032	<b>RE4/1</b>	Find a good description
2033	<b>RE4/2</b>	Find a good description
2034	<b>RE4/3</b>	Find a good description
2035	<b>RMS</b>	Root Mean Square
2036	<b>ROOT</b>	a framework for data processing born at CERN
2037	<b>RPC</b>	Resistive Plate Chamber
2038		
2039		
2040	<b>S</b>	
2041		
2042	<b>SC</b>	Synchrocyclotron
2043	<b>SPS</b>	Super Proton Synchrotron
2044		
2045		
2046	<b>T</b>	
2047		
2048	<b>TDC</b>	Time-to-Digital Converter

---

LIST OF ACRONYMS

2049	ToF	Time-of-flight
2050		
2051	<b>W</b>	
2052		
2053		
2054	webDCS	Web Detector Control System

## References

- [1] T. Massam et al. “Experimental observation of antideuteron production”. In: *Il Nuovo Cimento A* 63 (1965), pp. 10–14.
- [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.
- [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.
- [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.
- [5] UA2 Collaboration. “Evidence for  $Z_0 \rightarrow e^+e^-$  at the CERN pp collider”. In: *Physics Letters B* 129 (1983), pp. 130–140.
- [6] ALEPH Collaboration. “Determination of the number of light neutrino species”. In: *Physics Letters B* 231 (1989), pp. 519–529.
- [7] CERN. Geneva. LHC Experiments Committee. *The CMS muon project : Technical Design Report*. Tech. rep. CERN-LHCC-97-032. CMS Collaboration, 1997.
- [8] CERN. Geneva. LHC Experiments Committee. *Technical Proposal for the Phase-II Upgrade of the CMS Detector*. Tech. rep. CERN-LHCC-2015-010. CMS Collaboration, 2015.
- [9] CERN. Geneva. LHC Experiments Committee. *CMS, the Compact Muon Solenoid : technical proposal*. Tech. rep. CERN-LHCC-94-38. CMS Collaboration, 1994.
- [10] R. Santonico and R. Cardarelli. “Development of resistive plate counters”. In: *Nucl. Instr. Meth. Phys. Res.* 187 (1981), pp. 377–380.
- [11] Yu.N. Pestov and G.V. Fedotovich. *A picosecond time-of-flight spectrometer for the VEPP-2M based on local-discharge spark counter*. Tech. rep. SLAC-TRANS-0184. SLAC, 1978.
- [12] W.W. Ash, ed. *Spark Counter With A Localized Discharge*. Vol. SLAC-R-250. 1982, pp. 127–131.
- [13] I. Crotty et al. “The non-spark mode and high rate operation of resistive parallel plate chambers”. In: *NIMA* 337 (1993), pp. 370–381.
- [14] I. Crotty et al. “Further studies of avalanche mode operation of resistive parallel plate chambers”. In: *NIMA* 346 (1994), pp. 107–113.
- [15] R. Cardarelli et al. “Avalanche and streamer mode operation of resistive plate chambers”. In: *NIMA* 382 (1996), pp. 470–474.
- [16] E. Cerron Zeballos et al. “A new type of resistive plate chamber: The multigap RPC”. In: *NIMA* 374 (1996), pp. 132–135.
- [17] M.C.S. Williams. “The development of the multigap resistive plate chamber”. In: *Nucl. Phys. B* 61 (1998), pp. 250–257.

- 2092 [18] H. Czyrkowski et al. “New developments on resistive plate chambers for high rate operation”.  
2093 In: *NIMA* 419 (1998), pp. 490–496.
- 2094 [19] P. Camarri et al. “Streamer suppression with SF<sub>6</sub> in RPCs operated in avalanche mode”. In: *NIMA* 414 (1998), pp. 317–324.
- 2095 [20] E. Cerron Zeballos et al. “Effect of adding SF<sub>6</sub> to the gas mixture in a multigap resistive plate  
2096 chamber”. In: *NIMA* 419 (1998), pp. 475–478.
- 2098 [21] CERN. Geneva. LHC Experiments Committee. *ATLAS muon spectrometer: Technical design  
2099 report*. Tech. rep. CERN-LHCC-97-22. ATLAS Collaboration, 1997.
- 2100 [22] CERN. Geneva. LHC Experiments Committee. *ALICE Time-Of-Flight system (TOF) : Tech-  
2101 nical Design Report*. Tech. rep. CERN-LHCC-2000-012. ALICE Collaboration, 2000.
- 2102 [23] The CALICE collaboration. “First results of the CALICE SDHCAL technological proto-  
2103 type”. In: *JINST* 11 (2016).
- 2104 [24] PoS, ed. *Density Imaging of Volcanoes with Atmospheric Muons using GRPCs*. International  
2105 Europhysics Conference on High Energy Physics - HEP 2011. 2011.
- 2106 [25] C. Lippmann. “Detector Physics of Resistive Plate Chambers”. PhD thesis. Johann Wolfgang  
2107 Goethe-Universität, 2003.
- 2108 [26] 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  
2109 of resistive plate chambers”. In: *NIMA* 398 (1997), pp. 173–179.
- 2110 [27] G. Battistoni et al. “Sensitivity of streamer mode to single ionization electrons”. In: *NIMA*  
2111 235 (1985), pp. 91–97.
- 2112 [28] W. Riegler. “Induced signals in resistive plate chambers”. In: *NIMA* 491 (2002), pp. 258–271.
- 2113 [29] E. Cerron Zeballos et al. “A comparison of the wide gap and narrow gap resistive plate  
2114 chamber”. In: *NIMA* 373 (1996), pp. 35–42.
- 2115 [30] M. Abbrescia et al. “Cosmic ray tests of double-gap resistive plate chambers for the CMS  
2116 experiment”. In: *NIMA* 550 (2005), pp. 116–126.
- 2117 [31] ALICE Collaboration. “A study of the multigap RPC at the gamma irradiation facility at  
2118 CERN”. In: *NIMA* 490 (2002), pp. 58–70.
- 2119 [32] B. Bonner et al. “A multigap resistive plate chamber prototype for time-of-flight for the STAR  
2120 experiment at RHIC”. In: *NIMA* 478 (2002), pp. 176–179.
- 2121 [33] S. Yang et al. “Test of high time resolution MRPC with different readout modes for the  
2122 BESIII upgrade”. In: *NIMA* 763 (2014), pp. 190–196.
- 2123 [34] A. Akindinov et al. “RPC with low-resistive phosphate glass electrodes as a candidate for  
2124 the CBM TOF”. In: *NIMA* 572 (2007), pp. 676–681.
- 2125 [35] JINST, ed. *Development of the MRPC for the TOF system of the MultiPurpose Detector*.  
2126 RPC2016: XII Workshop on Resistive Plate Chambers and Related Detectors. 2016.
- 2127 [36] M.C.S. Williams. “Particle identification using time of flight”. In: *Journal of Physics G* 39  
2128 (2012).
- 2129 [37] A. Alici et al. “Aging and rate effects of the Multigap RPC studied at the Gamma Irradiation  
2130 Facility at CERN”. In: *NIMA* 579 (2007), pp. 979–988.
- 2131 [38] M. Abbrescia et al. “The simulation of resistive plate chambers in avalanche mode: charge  
2132 spectra and efficiency”. In: *NIMA* 431 (1999), pp. 413–427.
- 2133 [39] M. Abbrescia et al. “Study of long-term performance of CMS RPC under irradiation at the  
2134 CERN GIF”. In: *NIMA* 533 (2004), pp. 102–106.

## BIBLIOGRAPHY

---

- 2135 [40] H.C. Kim et al. “Quantitative aging study with intense irradiation tests for the CMS forward  
2136 RPCs”. In: *NIMA* 602 (2009), pp. 771–774.
- 2137 [41] S. Agosteo et al. “A facility for the test of large-area muon chambers at high rates”. In: *NIMA*  
2138 452 (2000), pp. 94–104.
- 2139 [42] PoS, ed. *CERN GIF ++ : A new irradiation facility to test large-area particle detectors for*  
2140 *the high-luminosity LHC program*. Vol. TIPP2014. 2014, pp. 102–109.
- 2141 [43] A. Fagot. *GIF++ DAQ v4.0*. 2017. URL: [https://github.com/afagot/GIF\\_DAQ](https://github.com/afagot/GIF_DAQ).
- 2142 [44] CAEN. *Mod. V1190-VX1190 A/B, 128/64 Ch Multihit TDC*. 14th ed. 2016.
- 2143 [45] CAEN. *Mod. V1718 VME USB Bridge*. 9th ed. 2009.
- 2144 [46] W-Ie-Ne-R. *VME 6021-23 VXI*. 5th ed. 2016.
- 2145 [47] Wikipedia. *INI file*. 2017. URL: [https://en.wikipedia.org/wiki/INI\\_file](https://en.wikipedia.org/wiki/INI_file).
- 2146 [48] S. Carrillo A. Fagot. *GIF++ Offline Analysis v6*. 2017. URL: [https://github.com/afagot/GIF\\_OfflineAnalysis](https://github.com/afagot/GIF_OfflineAnalysis).
- 2147