



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

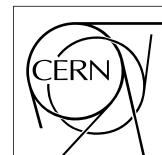
Promotoren: Dr. Michael Tytgat  
Prof. Dr. Dirk Ryckbosch

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>	xix
26	<b>English summary</b>	xxi
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	2.3 Muon Phase-II Upgrade . . . . .	2-1
34	<b>3 Amplification processes in gaseous detectors</b>	3-1
35	3.1 Signal formation . . . . .	3-1
36	3.2 Gas transport parameters . . . . .	3-1
37	<b>4 Resistive Plate Chambers</b>	4-1
38	4.1 Principle . . . . .	4-1
39	4.1.1 Electron drift velocity . . . . .	4-4
40	4.2 Rate capability and time resolution of Resistive Plate Chambers . . . . .	4-4
41	4.2.1 Operation modes . . . . .	4-4
42	4.2.2 Detector designs and performance . . . . .	4-6
43	4.2.2.1 Double-gap RPC . . . . .	4-6
44	4.2.2.2 Multigap RPC (MRPC) . . . . .	4-7
45	4.2.2.3 Charge distribution and performance limitations . . . . .	4-8
46	4.3 Resistive Plate Chambers at CMS . . . . .	4-11
47	4.3.1 Overview . . . . .	4-11
48	4.3.2 The present RPC system . . . . .	4-12
49	4.3.3 Pulse processing of CMS RPCs . . . . .	4-13
50	<b>5 Longevity studies and Consolidation of the present CMS RPC subsystem</b>	5-1
51	5.1 Testing detectors under extreme conditions . . . . .	5-1
52	5.1.1 The Gamma Irradiation Facilities . . . . .	5-3
53	5.1.1.1 GIF . . . . .	5-3
54	5.1.1.2 GIF++ . . . . .	5-5
55	5.2 Preliminary tests at GIF . . . . .	5-7
56	5.2.1 Resistive Plate Chamber test setup . . . . .	5-7
57	5.2.2 Data Acquisition . . . . .	5-9

---

58	5.2.3	Geometrical acceptance of the setup layout to cosmic muons . . . . .	5-9
59	5.2.3.1	Description of the simulation layout . . . . .	5-10
60	5.2.3.2	Simulation procedure . . . . .	5-12
61	5.2.3.3	Results . . . . .	5-13
62	5.2.4	Photon flux at GIF . . . . .	5-13
63	5.2.4.1	Expectations from simulations . . . . .	5-13
64	5.2.4.2	Dose measurements . . . . .	5-18
65	5.2.5	Results and discussions . . . . .	5-19
66	5.3	Longevity tests at GIF++ . . . . .	5-20
67	5.3.1	Description of the Data Acquisition . . . . .	5-23
68	5.3.2	RPC current, environmental and operation parameter monitoring . . . . .	5-24
69	5.3.3	Measurement procedure . . . . .	5-25
70	5.3.4	Longevity studies results . . . . .	5-25
71	<b>6</b>	<b>Investigation on high rate RPCs</b>	<b>6-1</b>
72	6.1	Rate limitations and ageing of RPCs . . . . .	6-1
73	6.1.1	Low resistivity electrodes . . . . .	6-1
74	6.1.2	Low noise front-end electronics . . . . .	6-1
75	6.2	Construction of prototypes . . . . .	6-1
76	6.3	Results and discussions . . . . .	6-1
77	<b>7</b>	<b>Conclusions and outlooks</b>	<b>7-1</b>
78	7.1	Conclusions . . . . .	7-1
79	7.2	Outlooks . . . . .	7-1
80	<b>A</b>	<b>A data acquisition software for CAEN VME TDCs</b>	<b>A-1</b>
81	A.1	GIF++ DAQ file tree . . . . .	A-1
82	A.2	Usage of the DAQ . . . . .	A-2
83	A.3	Description of the readout setup . . . . .	A-3
84	A.4	Data read-out . . . . .	A-3
85	A.4.1	V1190A TDCs . . . . .	A-4
86	A.4.2	DataReader . . . . .	A-6
87	A.4.3	Data quality flag . . . . .	A-10
88	A.5	Communications . . . . .	A-12
89	A.5.1	V1718 USB Bridge . . . . .	A-13
90	A.5.2	Configuration file . . . . .	A-13
91	A.5.3	WebDCS/DAQ intercommunication . . . . .	A-17
92	A.5.4	Example of inter-process communication cycle . . . . .	A-18
93	A.6	Software export . . . . .	A-18
94	<b>B</b>	<b>Details on the offline analysis package</b>	<b>B-1</b>
95	B.1	GIF++ Offline Analysis file tree . . . . .	B-1
96	B.2	Usage of the Offline Analysis . . . . .	B-2
97	B.2.1	Output of the offline tool . . . . .	B-3
98	B.2.1.1	ROOT file . . . . .	B-3
99	B.2.1.2	CSV files . . . . .	B-5
100	B.3	Analysis inputs and information handling . . . . .	B-6
101	B.3.1	Dimensions file and IniFile parser . . . . .	B-6
102	B.3.2	TDC to RPC link file and Mapping . . . . .	B-7
103	B.4	Description of GIF++ setup within the Offline Analysis tool . . . . .	B-9

---

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



# List of Figures

120

121	2.1	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. . . . .	2-2
122			
123			
124	2.2	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). . . . .	2-2
125			
126			
127			
128	2.3	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. . . . .	2-3
129			
130			
131	4.1	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. . . . .	4-2
132			
133			
134			
135			
136			
137			
138	4.2	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 $SF_6$ [13]. . . . .	4-3
139			
140			
141			
142	4.3	Movement of the charge carriers in an RPC. Figure 4.3a: Voltage across an RPC whose electrode have a relative permittivity of 5 at the moment the tension s applied. Figure 4.3b: 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 4.3c: The streamer discharge initiated by a charged particle transports electrons and cations towards the anode and cathode respectively. . . . .	4-5
143			
144			
145			
146			
147			
148	4.4	Typical oscilloscope pulses in streamer mode (Figure 4.4a) and avalanche mode(Figure 4.4b). In the case of streamer mode, the very small avalanche signal is visible. . . . .	4-5
149			
150	4.5	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. . . . .	4-6
151			
152			
153	4.6	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. . . . .	4-7
154			
155			
156			
157			
158			

159	4.7	Comparison of performance of CMS double and single gap RPCs using cosmic muons [24]. Figure 4.7a: Comparison of efficiency sigmoids. Figure 4.7b: Voltage distribution at 95% of maximum efficiency. Figure 4.7c: $\Delta_{10\%}^{90\%}$ distribution. . . . .	4-7
160			
161			
162	4.8	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 [25]. . . . .	4-8
163			
164	4.9	Particle identification applied to electrons in the STAR experiment. The identification is performed combining ToF and $dE/dx$ measurements [30]. . . . .	4-9
165			
166	4.10	Comparison of the detector performance of ALICE ToF MRPC [31] 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.	4-9
167			
168			
169	4.11	Ratio between total induced and drifting charge have been simulated for single gap, double-gap and multigap layouts [32]. The total induced charge for a double-gap RPC is a factor 2 higher than for a multigap. . . . .	4-10
170			
171			
172	4.12	Charge spectra have been simulated for single gap, double-gap and multigap layouts [32]. 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. . . . .	4-10
173			
174			
175			
176	4.13	The maximal theoretical efficiency is simulated for single gap, double-gap and multigap layouts [32] at a constant gap thickness of 2 mm and using an effective Townsend coefficient of 9 $\text{mm}^{-1}$ . . . . .	4-11
177			
178			
179	4.14	Signals from the RPC strips are shaped by the FEE described on Figure 4.14a. Output LVDS signals are then read-out by a TDC module connected to a computer or converted into NIM and sent to scalers. Figure 4.14b describes how these converted signals are put in coincidence with the trigger. . . . .	4-13
180			
181			
182			
183	4.15	Description of the principle of a CFD. A comparison of threshold triggering (left) and constant fraction triggering (right) is shown in Figure 4.15a. Constant fraction triggering is obtained thanks to zero-crossing technique as explained in Figure 4.15b. 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. . . . .	4-14
184			
185			
186			
187			
188			
189			
190			
191			
192	5.1	(5.1a) Extrapolation from 2016 data of single hit rate per unit area in the barrel region. (5.1b) Extrapolation from 2016 data of single hit rate per unit area in the endcap region. . . . .	5-2
193			
194			
195	5.2	Background Fluka simulation compared to 2016 Data at $L = 10^{34}\text{cm}^{-2}.\text{s}^{-1}$ in the fourth endcap disk region. A mismatch in between simulation and data can be observed. [To be understood.] . . . . .	5-3
196			
197			
198	5.3	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. . . . .	5-4
199			
200			
201			
202			
203	5.4	$^{137}\text{Cs}$ 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. . . . .	5-5
204			

---

205	5.5	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. . . . .	5-5
206	5.6	Simulated unattenuated current of photons in the xz plane (Figure 5.6a) and yz plane (Figure 5.6b) 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. . . . .	5-6
207	5.7	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 5.7a provides a side view of the setup in the xz plane while Figure 5.7b shows a top view in the yz plane. . . . .	5-7
208	5.8	RE-4-2-BARC-161 chamber is inside the tent as described in Figure 5.7. 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. . . . .	5-8
209	5.9	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. . . . .	5-9
210	5.10	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. . . . .	5-10
211	5.11	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 5.7a shows a global view of the simulated setup. Figure 5.7b shows a zommed view that allows to see the 2 scintillators as well as the full RPC plane. . . . .	5-11
212	5.12	$\gamma$ flux $F(D)$ is plot using values from table 5.1. As expected, the plot shows similar attenuation behaviours with increasing distance for each absorption factors. . . . .	5-14
213	5.13	Figure 5.13a shows the linear approximation fit done via formulae 5.7 on data from table 5.2. Figure 5.13b shows a comparison of this model with the simulated flux using a and b given in figure 5.13a in formulae 5.4 and the reference value $D_0 = 50\text{cm}$ and the associated flux for each absorption factor $F_0^{ABS}$ from table 5.1 . . . . .	5-16
214	5.14	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. . . . .	5-18
215			
216			
217			
218			
219			
220			
221			
222			
223			
224			
225			
226			
227			
228			
229			
230			
231			
232			
233			
234			
235			
236			
237			
238			
239			
240			
241			
242			
243			
244			
245			
246			
247			
248			
249			
250			
251			
252			
253			

---

254	5.15 . . . . .	5-19
255	5.16 Evolution of the maximum efficiency for RE2 (5.16a) and RE4 (5.16b) chambers	
256	with increasing extrapolated $\gamma$ rate per unit area at working point. Both irradiated	
257	(blue) and non irradiated (red) chambers are shown. . . . .	5-21
258	5.17 Evolution of the working point for RE2 (5.17a) and RE4 (5.17b) with increasing	
259	extrapolated $\gamma$ rate per unit area at working point. Both irradiated (blue) and non	
260	irradiated (red) chambers are shown. . . . .	5-21
261	5.18 Evolution of the maximum efficiency at HL-LHC conditions, i.e. a background hit	
262	rate per unit area of 300 Hz/cm <sup>2</sup> , with increasing integrated charge for RE2 (5.18a)	
263	and RE4 (5.18b) detectors. Both irradiated (blue) and non irradiated (red) chambers	
264	are shown. The integrated charge for non irradiated detectors is recorded during test	
265	beam periods and stays small with respect to the charge accumulated in irradiated	
266	chambers. . . . .	5-22
267	5.19 Comparison of the efficiency sigmoid before (triangles) and after (circles) irradiation	
268	for RE2 (5.19a) and RE4 (5.19b) detectors. Both irradiated (blue) and non irradiated	
269	(red) chambers are shown. . . . .	5-22
270	5.20 Evolution of the Bakelite resistivity for RE2 (5.20a) and RE4 (5.20b) detectors. Both	
271	irradiated (blue) and non irradiated (red) chambers are shown. . . . .	5-23
272	5.21 Evolution of the noise rate per unit area for the irradiated chamber RE2-2-BARC-9	
273	only. . . . .	5-23
274	A.1 (A.1a) View of the front panel of a V1190A TDC module [38]. (A.1b) View of the	
275	front panel of a V1718 Bridge module [39]. (A.1c) View of the front panel of a 6U	
276	6021 VME crate [40]. . . . .	A-3
277	A.2 Module V1190A <i>Trigger Matching Mode</i> timing diagram [38]. . . . .	A-4
278	A.3 Structure of the ROOT output file generated by the DAQ. The 5 branches ( <code>EventNumber</code> ,	
279	<code>number_of_hits</code> , <code>Quality_flag</code> , <code>TDC_channel</code> and <code>TDC_TimeStamp</code> ) are visible on	
280	the left panel of the ROOT browser. On the right panel is visible the histogram cor-	
281	responding to the variable <code>nHits</code> . In this specific example, there were approximately	
282	50k events recorded to measure the gamma irradiation rate on the detectors. Each	
283	event is stored as a single entry in the <code>TTree</code> . . . . .	A-10
284	A.4 The effect of the quality flag is explained by presenting the content of <code>TBranch</code>	
285	<code>number_of_hits</code> of a data file without <code>Quality_flag</code> in Figure A.4a and the con-	
286	tent of the same <code>TBranch</code> for data corresponding to a <code>Quality_flag</code> where all TDCs	
287	were labelled as <code>GOOD</code> in Figure A.4b taken with similar conditions. It can be noted	
288	that the number of entries in Figure A.4b is slightly lower then in Figure A.4a due	
289	to the excluded events. . . . .	A-12
290	A.5 Using the same data as previously showed in Figure A.4, the effect of the quality	
291	flag is explained by presenting the reconstructed hit multiplicity of a data file with-	
292	out <code>Quality_flag</code> in Figure A.5a and the reconstructed content of the same RPC	
293	partition for data corresponding to a <code>Quality_flag</code> where all TDCs were labelled as	
294	<code>GOOD</code> in Figure A.5b taken with similar conditions. The artificial high content of bin	
295	0 is completely suppressed. . . . .	A-12
296	A.6 WebDCS DAQ scan page. On this page, shifters need to choose the type of scan	
297	(Rate, Efficiency or Noise Reference scan), the gamma source configuration at the	
298	moment of data taking, the beam configuration, and the trigger mode. These in-	
299	formation will be stored in the DAQ ROOT output. Are also given the minimal	
300	measurement time and waiting time after ramping up of the detectors is over before	
301	starting the data acquisition. Then, the list of HV points to scan and the number of	
302	triggers for each run of the scan are given in the table underneath. . . . .	A-14

---

303	B.1	Example of expected hit time distributions in the cases of efficiency (Figure B.1a) and noise/gamma rate per unit area (Figure B.1b) measurements as extracted from the raw ROOT files. The unit along the x-axis corresponds to ns. The fact that "the" muon peak is not well defined in Figure B.1a is due to the contribution of all the RPCs being tested at the same time that don't necessarily have the same signal arrival time. Each individual peak can have an offset with the ones of other detectors. The inconsistancy in the first 100 ns of both time distributions is an artefact of the TDCs and are systematically rejected during the analysis. . . . .	B-16
311	B.2	The effect of the quality flag is explained by presenting the reconstructed hit multi- plicity of a data file without <code>Quality_flag</code> . The artificial high content of bin 0 is the effect of corrupted data. . . . .	B-19
314	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



# List of Tables

322

323	4.1 Properties of the most used electrode materials for RPCs. . . . .	4-4
324	5.1 Total photon flux ( $E\gamma \leq 662$ keV) with statistical error predicted considering a	
325	$^{137}\text{Cs}$ activity of 740 GBq at different values of the distance $D$ to the source along	
326	the x-axis of irradiation field [35]. . . . .	5-13
327	5.2 Correction factor $c$ is computed thanks to formulae 5.5 taking as reference $D_0 =$	
328	50 cm and the associated flux $F_0^{ABS}$ for each absorption factor available in table 5.1. .	5-15
329	5.3 The data at $D_0$ in 1997 is taken from [35]. In a second step, using Equations 5.8	
330	and 5.9, the flux at $D$ can be estimated in 1997. Then, taking into account the	
331	attenuation of the source activity, the flux at $D$ can be estimated at the time of the	
332	tests in GIF in 2014. Finally, assuming a sensitivity of the RPC to $\gamma$ $s = 2 \cdot 10^{-3}$ ,	
333	an estimation of the hit rate per unit area is obtained. . . . .	5-17
334	A.1 Inter-process communication cycles in between the webDCS and the DAQ through	
335	file string signals. . . . .	A-19



336

## List of Acronyms

337

### List of Acronyms

338

339

#### A

340

341

342 AFL Almost Full Level

343

344

#### B

345

346

347 BARC Bhabha Atomic Research Centre  
348 BLT Block Transfer  
349 BR Branching Ratio

350

351

#### C

352

353

354 CAEN Costruzioni Apparecchiature Elettroniche Nucleari S.p.A.  
355 CERN European Organization for Nuclear Research  
356 CFD Constant Fraction Discriminator  
357 CMS Compact Muon Solenoid  
358 CSC Cathode Strip Chamber

359

360

#### D

361

362

363 DAQ Data Acquisition  
364 DCS Detector Control Software  
365 DQM Data Quality Monitoring  
366 DT Drift Tube

367

368

#### F

369

370

371	FEE	Front-End Electronics
372	FEB	Front-End Board
373		
374	<b>G</b>	
375		
376		
377	GE-/-	Find a good description
378	GE1/1	Find a good description
379	GE2/1	Find a good description
380	GEANT	GEometry ANd Tracking - a series of software toolkit platforms developed by CERN
381		
382	GEM	Gas Electron Multiplier
383	GIF	Gamma Irradiation Facility
384	GIF++	new Gamma Irradiation Facility
385		
386	<b>H</b>	
387		
388		
389	HL-LHC	High Luminosity LHC
390	HPL	High-pressure laminate
391	HV	High Voltage
392		
393	<b>I</b>	
394		
395		
396	iRPC	improved RPC
397	IRQ	Interrupt Request
398		
399	<b>L</b>	
400		
401		
402	LHC	Large Hadron Collider
403	LS1	First Long Shutdown
404	LS3	Third Long Shutdown
405	LV	Low Voltage
406	LVDS	Low-Voltage Differential Signaling
407		
408	<b>M</b>	
409		
410		
411	MC	Monte Carlo
412	MCNP	Monte Carlo N-Particle
413	ME-/-	Find good description

414	ME0	Find good description
415	MRPC	Multigap RPC
416		
417	<b>N</b>	
418		
420	NIM	Nuclear Instrumentation Module logic signals
421		
422		
423	<b>P</b>	
424		
425	PMT	PhotoMultiplier Tube
426		
427		
428	<b>R</b>	
429		
430	RE-/-	Find a good description
431	RE2/2	Find a good description
432	RE3/1	Find a good description
433	RE3/2	Find a good description
434	RE4/1	Find a good description
435	RE4/2	Find a good description
436	RE4/3	Find a good description
437	RMS	Root Mean Square
438	ROOT	a framework for data processing born at CERN
439	RPC	Resistive Plate Chamber
440		
441		
442	<b>S</b>	
443		
444	SPS	Super Proton Synchrotron
445		
446		
447	<b>T</b>	
448		
449	TDC	Time-to-Digital Converter
450	ToF	Time-of-flight
451		
452		
453	<b>W</b>	
454		
455	webDCS	Web Detector Control System



457

## Nederlandse samenvatting –Summary in Dutch–

459 Le resume en Neerlandais (j'aurais peut-être de apprendre la langue juste pour ca...).

458



## English summary

<sup>461</sup> Le meme résume mais en Anglais (on commencera par la hein!).



# 1

## Introduction

462

463

<sup>464</sup> **1.1 A story of High Energy Physics**

<sup>465</sup> **1.2 Organisation of this study**



# 2

466

## Investigating the TeV scale

### 468 2.1 The Standard Model of Particle Physics

### 469 2.2 The Large Hadron Collider and the Compact Muon Solenoid

### 470 2.3 Muon Phase-II Upgrade

471 After the more than two years lasting First Long Shutdown (LS1), the Large Hadron Collider (LHC)  
472 delivered its very first Run-II proton-proton collisions early 2015. LS1 gave the opportunity to the  
473 LHC and to the its experiments to undergo upgrades. The accelerator is now providing collisions  
474 at center-of-mass energy of 13 TeV and bunch crossing rate of 40 MHz, with a peak luminosity  
475 exceeding its design value. During the first and upcoming second LHC Long Shutdown, the Compact  
476 Muon Solenoid (CMS) detector is also undergoing a number of upgrades to maintain a high system  
477 performance [1].

478 From the LHC Phase-2 or High Luminosity LHC (HL-LHC) period onwards, i.e. past the Third  
479 Long Shutdown (LS3), the performance degradation due to integrated radiation as well as the average  
480 number of inelastic collisions per bunch crossing, or pileup, will rise substantially and become a  
481 major challenge for the LHC experiments, like CMS that are forced to address an upgrade program  
482 for Phase-II [2]. Simulations of the expected distribution of absorbed dose in the CMS detector  
483 under HL-LHC conditions, show in figure 5.14 that detectors placed close to the beamline will have  
484 to withstand high irradiation, the radiation dose being of the order of a few tens of Gy.

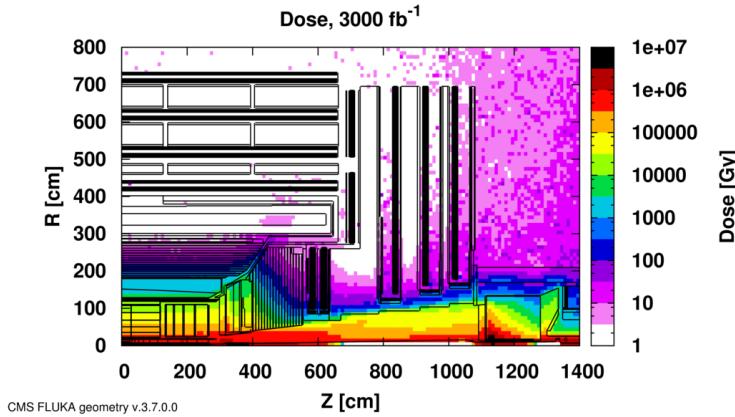


Figure 2.1: Absorbed dose in the CMS cavern after an integrated luminosity of  $3000 \text{ fb}^{-1}$ .  $R$  is the transverse distance from the beamline and  $Z$  is the distance along the beamline from the Interaction Point at  $Z=0$ .

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 2.2 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 [3].

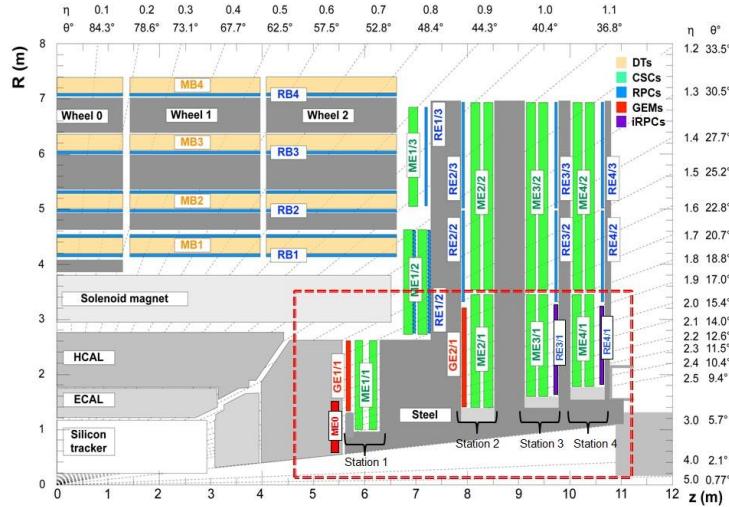


Figure 2.2: A quadrant of the muon system, showing DTs (yellow), RPCs (light blue), and CSCs (green). The locations of new forward muon detectors for Phase-II are contained within the dashed box and indicated in red for GEM stations ( $ME0$ ,  $GE1/1$ , and  $GE2/1$ ) and dark blue for improved RPC (iRPC) stations ( $RE3/1$  and  $RE4/1$ ).

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

good bunch crossing identification can be obtained with the present CMS RPC system, given their fast response of the order of 1 ns. In order to contribute to the precision of muon momentum measurements, muon chambers should have a spatial resolution less or comparable to the contribution of multiple scattering [1]. Most of the plausible physics is covered only considering muons with  $p_T < 100$  GeV thus, in order to match CMS requirements, a spatial resolution of  $\mathcal{O}(\text{few mm})$  the proposed new RPC stations, as shown by the simulation in figure 2.3. According to preliminary designs, RE3/1 and RE4/1 readout pitch will be comprised between 3 and 6 mm and 5  $\eta$ -partitions could be considered.

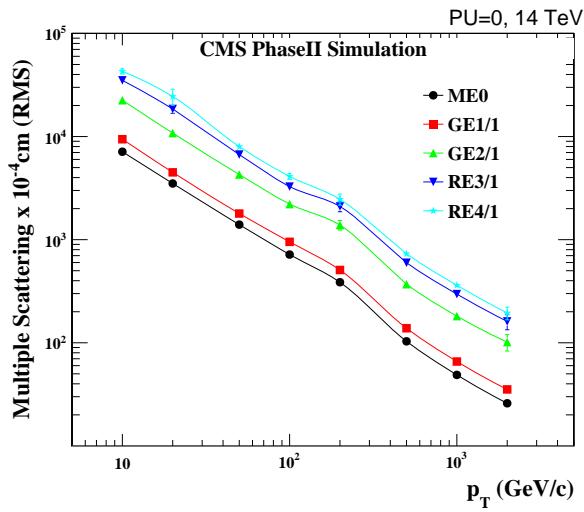


Figure 2.3: RMS of the multiple scattering displacement as a function of muon  $p_T$  for the proposed forward muon stations. All of the electromagnetic processes such as bremsstrahlung and magnetic field effect are included in the simulation.



# 3

502

503

## Amplification processes in gaseous detectors

504

### **3.1 Signal formation**

505

### **3.2 Gas transport parameters**



# 4

506

507

## Resistive Plate Chambers

508 A Resistive Plate Chamber (RPC) is a gaseous detector using the same physical processes described  
509 in Chapter 3. It has been developed in 1981 by Santonico and Cardarelli [4], under the name of  
510 *Resistive Plate Counter*, as an alternative to the local-discharge spark counters proposed in 1978  
511 by Pestov and Fedotovich [5, 6]. Working with spark chambers implied using high-pressure gas  
512 and high mechanical precision which the RPC simplified by formerly using a gas mixture of argon  
513 and butane flowed at atmospheric pressure and a constant and uniform electric field propagated  
514 in between two parallel electrode plates. Moreover, a significant increase in rate capability was  
515 introduced by the use of electrode plate material with high bulk resistivity, preventing the discharge  
516 from growing throughout the whole gas gap. Indeed, the effect of using resistive electrodes is that  
517 the constant electric field is locally canceled out by the development of the discharge, limiting its  
518 growth.

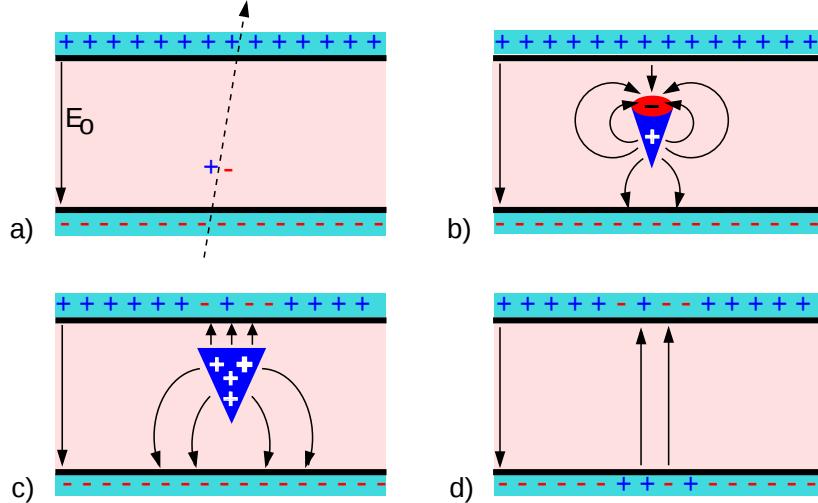
519 Through its development history, different operating modes [7–9] and new detector designs [10–  
520 12] have been discovered, leading to further improvement of the rate capability of such a detector.  
521 Moreover, the addition of  $SF_6$  into the gas mix improved the stability of operation of the RPC [13,  
522 14].

523 The low developing costs and easily achievable large detection areas offered by RPCs, as well  
524 as the wide range of possible designs, made them a natural choice to as muon chambers and/or  
525 trigger detectors in multipurpose experiments such as CMS [1] or ATLAS [15], time-of-flight detec-  
526 tors in ALICE [16], calorimeter with CALICE [17] or even detectors for volcanic muography with  
527 ToMuVol [18].

### 528 4.1 Principle

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

533 the electrodes by the electric field, as shown in Figure 4.1 [19].



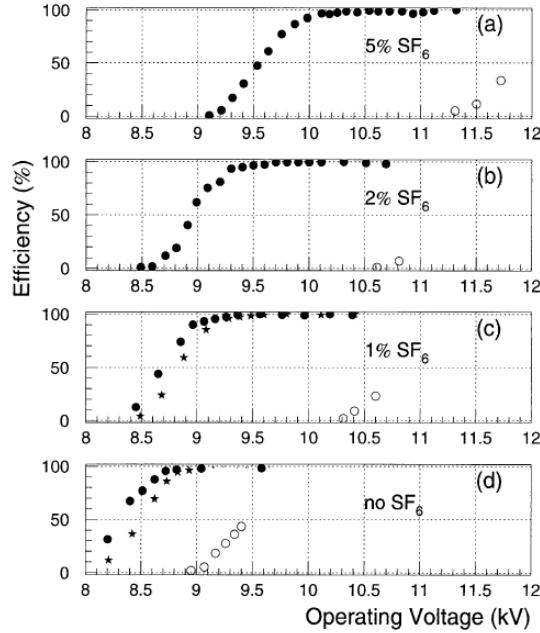
534 *Figure 4.1: Different phases of the avalanche development in the RPC gas volume subjected to a constant  
535 electric field  $E_0$ . a) An avalanche is initiated by the primary ionisation caused by the passage of a charged  
536 particle through the gas volume. b) Due to its growing size, the avalanche starts to locally influence the electric  
537 field. c) The electrons, lighter than the cations reach the anode first. d) The ions reach the cathode. While  
538 the charges have not recombined, the electric field in the small region around the avalanche stays affected and  
539 locally blind the detector.*

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

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

- 548 • Tetrafluoroethane ( $C_2F_4H_2$ ), also referred to as *Freon*, is the principal compound of the RPC  
549 gas mixtures, with a typical fraction above 90%. It is used for it's high effective Townsend  
550 coefficient and the great average fast charge that allows to operate the detector with a high  
551 threshold with respect to argon, for example, that has similar effective Townsend coefficient  
552 but suffers from a lower fast charge. To operate with similar conditions, argon would require a  
553 higher electric field leading to a higher fraction of streamers, thus limiting the rate capability  
554 of the detector [20].
- 555 • Isobutane (i- $C_4H_{10}$ ), only present in a few percent in the gas mixtures, is used for its UV  
556 quenching properties [21] helping to prevent streamers due to UV photon emission during the  
557 avalanche growth.
- 558 • Sulfur hexafluoride, ( $SF_6$ ), referred to simply as *SF6*, is used in very little quantities for its  
559 high electronegativity. Excess of electrons are being absorbed by the compound and streamers

554       are suppressed [14]. Nevertheless, a fraction of  $SF_6$  higher than 1% will not bring any extra  
 555       benefice in terms of streamer cancelation power but will lead to higher operating voltage [13],  
 556       as can be understood through Figure 4.2.



*Figure 4.2: 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  $SF_6$  [13].*

557       After an avalanche developed in the gas, a time long compared to the development of a discharge  
 558       is needed to recombine the charge carriers in the electrode material due to their resistivity. This  
 559       property has the advantage of affecting the local electric field and avoiding sparks in the detector  
 560       but, on the other hand, the rate capability is intrinsically limited by the time constant  $\tau_{RPC}$  of the  
 561       detector. Using a quasi-static approximation of Maxwell's equations for weakly conducting media,  
 562       it can be shown that the time constant  $\tau_{RPC}$  necessary to the charge recombination at the interface  
 563       in between the electrode and the gas volume is given by the Formula 4.1 [22].

$$\tau_{RPC} = \frac{\epsilon_{electrode} + \epsilon_{gas}}{\sigma_{electrode} + \sigma_{gas}} \quad (4.1)$$

564       A gas can be assimilated to vacuum, leading to  $\epsilon_{gas} = \epsilon_0$  and  $\sigma_{gas} = 0$ , and the electrodes  
 565       permittivity and conductivity can be written as  $\epsilon_{electrode} = \epsilon_r \epsilon_0$  and  $\sigma_{electrode} = 1/\rho_{electrode}$ ,  
 566       showing the strong dependence of the time constant to the electrodes resistivity in Formula 4.2.

$$\tau_{RPC} = (\epsilon_r + 1)\epsilon_0 \times \rho_{electrode} \quad (4.2)$$

567       Very few materials with a low enough resistivity exist in nature. The resistivity targeted to build  
 568       RPCs ranges from  $10^9$  to  $10^{12} \Omega \cdot \text{cm}$ . The most common RPC electrode materials are displayed in  
 569       Table 4.1. When the doped glass and ceramics can offer short time constants of the order of 1 ms,

570 the developing cost of such materials is quite high due to the very low demand. Thus, High-pressure  
 571 laminate (HPL) is often the choice for high rate experiments using very large RPC detection areas.  
 572 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 4.1: Properties of the most used electrode materials for RPCs.

### 573 4.1.1 Electron drift velocity

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

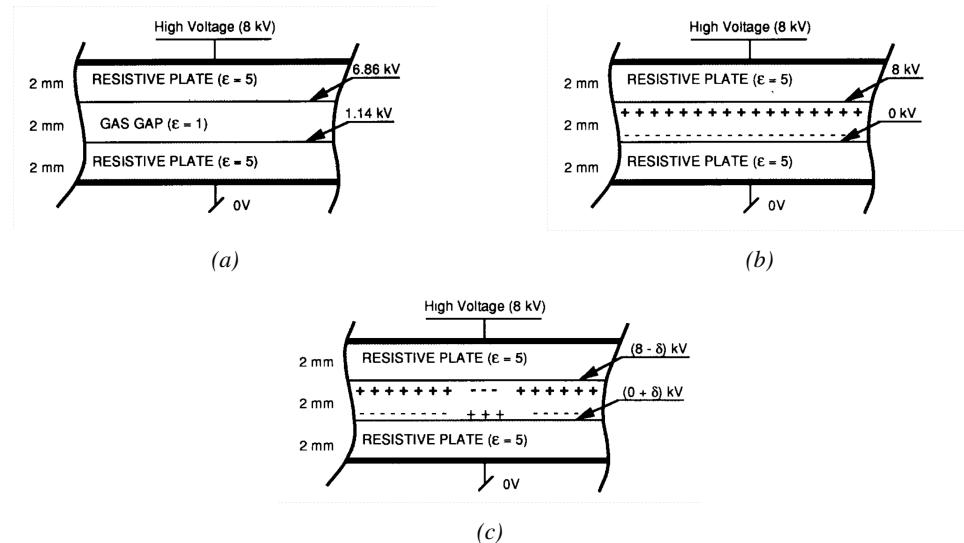
## 575 4.2 Rate capability and time resolution of Resistive Plate Cham- 576 bers

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

### 581 4.2.1 Operation modes

582 RPCs where developed early 1980s. At that time it was using an operating mode now referred to  
 583 as *streamer mode*. Streamers are large discharges that develop in between the 2 electrodes enough  
 584 to locally discharge the electrodes. If the electric field inside of the gas volume is strong enough,  
 585 with electrons being fast compared to ions, a large and dense cloud of positive ions will develop  
 586 nearby the anode and extend toward the cathode while the electrons are being collected, eventually  
 587 leading to a streamer discharge due to the increase of field seen at the cathode. the field is then strong  
 588 enough so that electrons are pulled out of the cathode. Electrodes, though they are a unique volume  
 589 of resistive material, can be assimilated to capacitors. At the moment an electric field is applied in  
 590 between their outer surfaces, the charge carriers inside of the volume will start moving leading to  
 591 a situation where there is no voltage across the electrodes and a higher density of negative charges,  
 592 i.e. electrons, on the inner surface of the cathode. Finally, when a streamer discharge occurs, these  
 593 electrons are partially released in the gas volume contributing to increase the discharge strength until  
 594 the formation of a conductive plasma, the streamer. This can be understood through Figure 4.3 [7].  
 595 Streamer signals are very convenient in terms of read-out as no amplification is required with output  
 596 pulses amplitudes of the order of a few tens to few hundreds of mV as can be seen on Figure 4.4.

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



*Figure 4.3: Movement of the charge carriers in an RPC. Figure 4.3a: Voltage across an RPC whose electrode have a relative permittivity of 5 at the moment the tension  $s$  applied. Figure 4.3b: 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 4.3c: The streamer discharge initiated by a charged particle transports electrons and cations towards the anode and cathode respectively.*

order of a few mv as shown on Figure 4.4, and requires amplification. This is the *avalanche mode* of RPC operation.

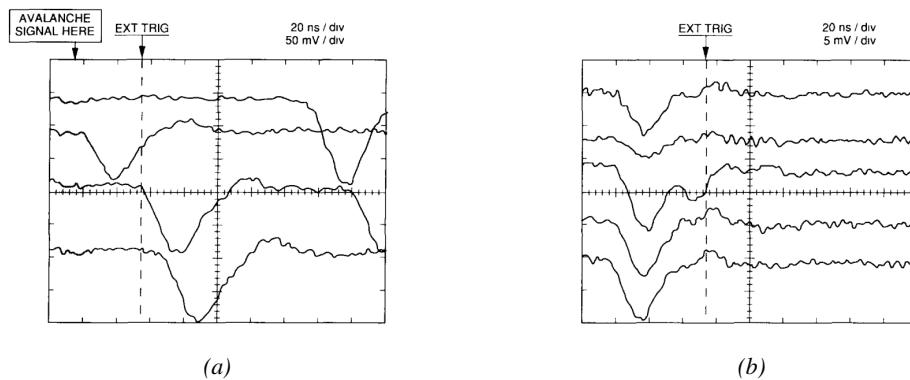


Figure 4.4: Typical oscilloscope pulses in streamer mode (Figure 4.4a) and avalanche mode (Figure 4.4b). In the case of streamer mode, the very small avalanche signal is visible.

This mode offers a higher rate capability by providing smaller discharges that don't affect the electrodes charge and are more locally contained in the gas volume as was demonstrated by Crotty with Figure 4.5 [7]. The detector only stays locally blind the time the charge carriers are recombined and there is no need for electrode recharge which is a long process affecting a large portion of the detector. Another advantage of avalanche signals over streamer is the great time consistency. Figure 4.4 shows very clearly that avalanche signals have a very small time jitter. This is a natural choice for high rate experiments.

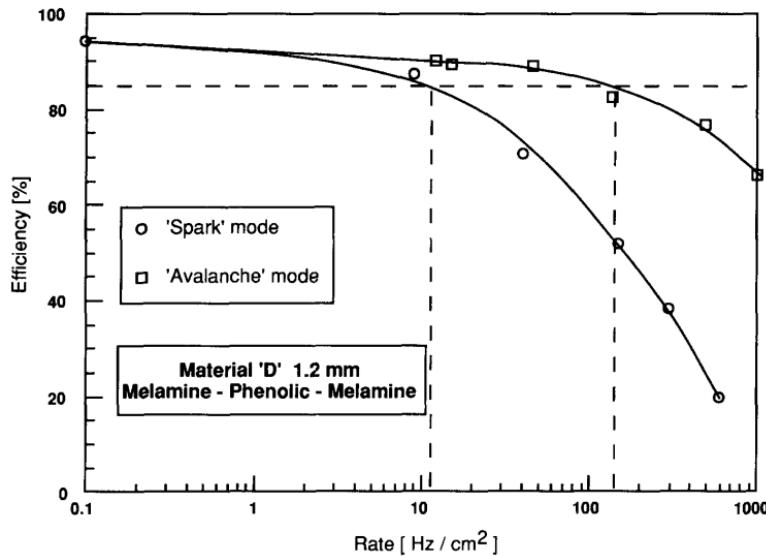


Figure 4.5: Rate capability comparison for the streamer and avalanche mode of operation. An order of magnitude in rate capability for a maximal efficiency drop of 10% is gained by using the avalanche mode over the streamer mode.

### 609 4.2.2 Detector designs and performance

610 Different RPC design have been used and each of them present its own advantages. Historically, the  
 611 first type of RPC to have been developed is what is referred now to *no narrow gap* RPC [4, 23]. After  
 612 the avalanche mode has been discovered [7], it has been proven that increasing the width of the gas  
 613 gap lead to higher rate capability, due to lower charge deposition per avalanche, and lower power  
 614 dissipation [23]. Nevertheless, by increasing the gas gap width, the time resolution of the detector  
 615 decreases. This is a natural result if the increase of active gas volume in the detector is taken into  
 616 account. Indeed, for a given threshold, only the small fraction of gas closest to the cathode will  
 617 provide enough gain to have a detectable signal. In the case of a wider gas volume, the active  
 618 region is then larger and a larger time jitter is introduced with the variation of starting position of the  
 619 avalanche, as discussed in [10]. To solve improve both the time resolution and the rate capability,  
 620 different methods were used trying to get advantages of both narrow and wide gap RPCs.

#### 621 4.2.2.1 Double-gap RPC

622 Double-gap RPCs are made out of 2 narrow RPC detectors. The 2 RPC gaps are stacked on top of  
 623 each other as shown in Figure 4.6. This detector layout, popularized by the two multipurpose experiments  
 624 CMS [1] and ATLAS [15] at LHC, can be used as an OR system in which each individual  
 625 chamber participates in the output signal. The gain of such a detector is greatly reduced with respect  
 626 to single-gap RPCs with an efficiency plateau reached at lower voltage, as visible on Figure 4.7.

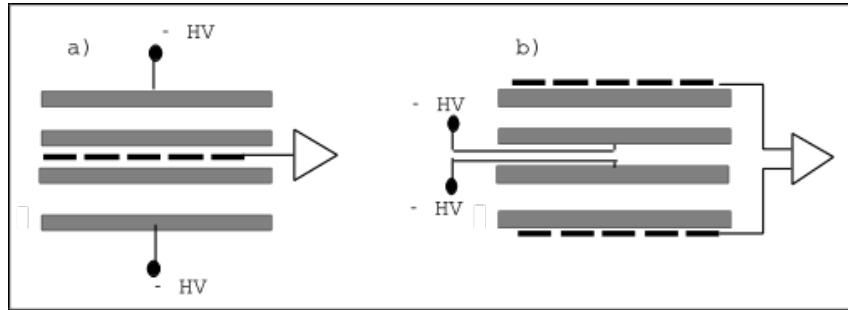


Figure 4.6: Possible double-gap RPC layouts: a) "standard" 1D double-gap RPC, as used in CMS experiment, where the anodes are facing each other and a 1D read-out plane is sandwiched between them, b) double read-out double-gap RPC as used in ATLAS experiment, where the cathodes are facing each other and 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.

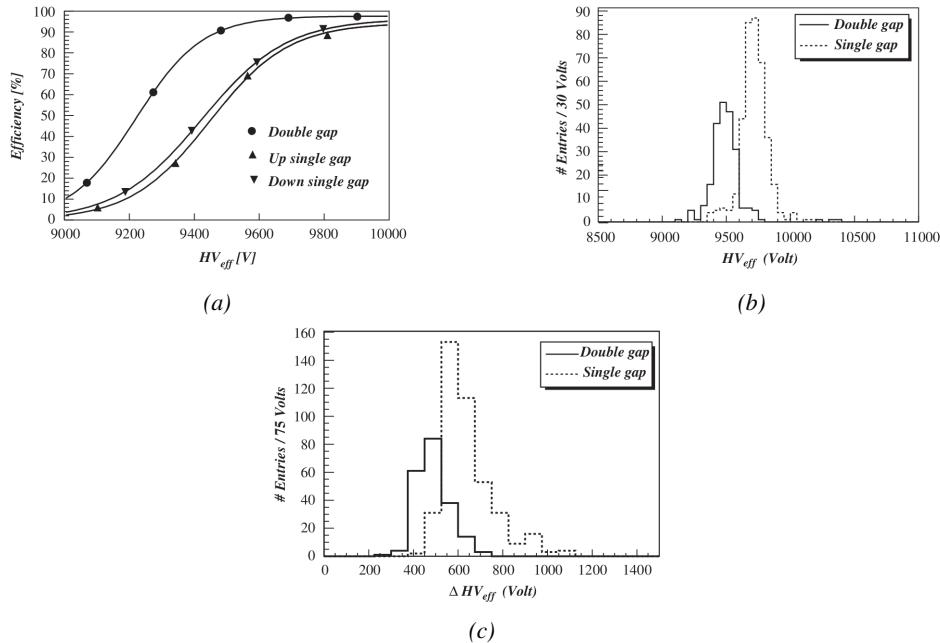
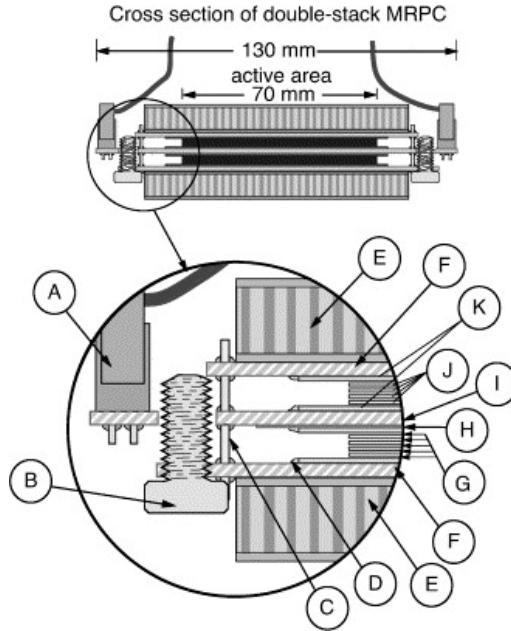


Figure 4.7: Comparison of performance of CMS double and single gap RPCs using cosmic muons [24]. Figure 4.7a: Comparison of efficiency sigmoids. Figure 4.7b: Voltage distribution at 95% of maximum efficiency. Figure 4.7c:  $\Delta_{10\%}^{90\%}$  distribution.

#### 627 4.2.2.2 Multigap RPC (MRPC)

628 MRPCs are layouts in which floating sub electrode plates are placed into a wide gap RPC to divide  
 629 the gas volume and create a sum of narrow gaps [10, 11]. The time resolution of such a detector can  
 630 reach of few tens of ps, with gas gaps of the order of a few hundred  $\mu m$  as shown in Figure 4.8  
 631 representing ALICE Time-of-flight (ToF) MRPCs.

632 Sometimes used as a double multigap RPC, taking advantage of the OR of double gap RPCs,  
 633 the MRPC is mainly used as ToF detector [25–29] due to its excellent timing properties that allow



*Figure 4.8: 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 [25].*

634 to perform particle identification as explained by Williams in [30]. The principle of particle iden-  
 635 tification using ToF is simply the measurement of the velocity of a particle. Indeed, particles are  
 636 defined by their mass (for the parameter of interest here, their electric charge being measured using  
 637 the bending angle of the particles traveling through a magnetic field) and this mass can be calculated  
 638 by measuring the velocity  $\beta$  and momentum of the particle:

$$\beta = \frac{p}{\sqrt{p^2 + m^2}} \quad (4.3)$$

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

$$T = \frac{L}{v} = \frac{L}{c \cdot \beta}, \quad \Delta T = T_1 - T_2 = \frac{L}{c} \left( \sqrt{1 + m_1^2/p^2} - \sqrt{1 + m_2^2/p^2} \right) \cong (m_1^2 - m_2^2) \frac{L}{2cp^2} \quad (4.4)$$

643 An example of particle identification is given for the case of STAR experiment in Figure 4.9.

644 Another benefit of using such small gas gaps is the strong reduction of the average avalanche  
 645 volume and thus of the blind spot on MRPCs leading to an improved rate capability. Multigaps can  
 646 sustain backgrounds of several kHz/cm<sup>2</sup> as demonstrated in Figure 4.10.

#### 647 4.2.2.3 Charge distribution and performance limitations

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

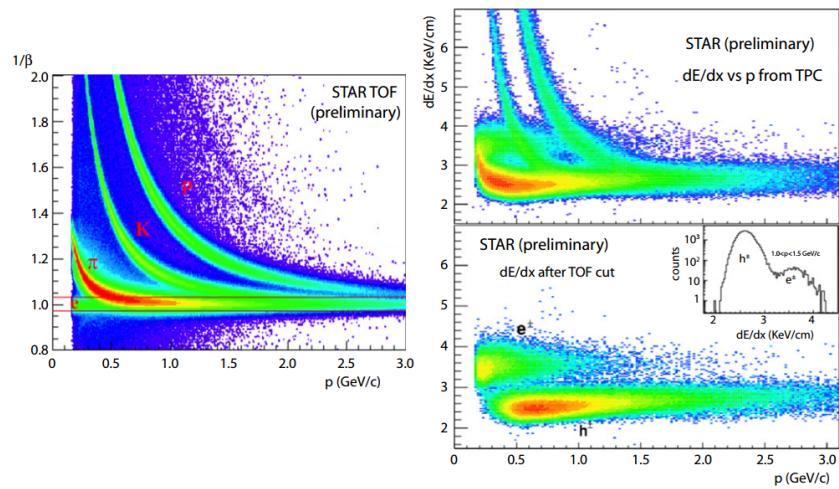


Figure 4.9: Particle identification applied to electrons in the STAR experiment. The identification is performed combining ToF and  $dE/dx$  measurements [30].

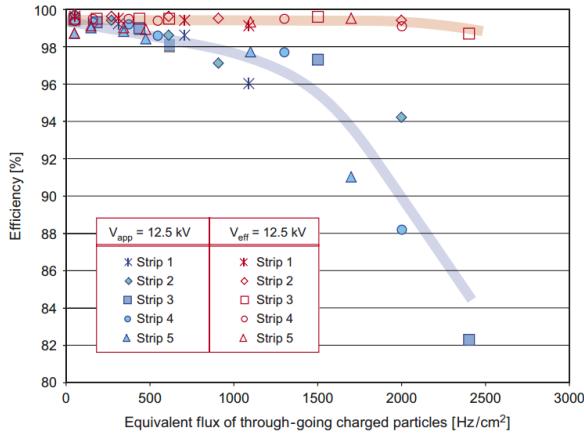


Figure 4.10: Comparison of the detector performance of ALICE ToF MRPC [31] at fixed applied voltage (in blue) and at fixed effective voltage (in red). The effective voltage is kept fixed by increasing the applied voltage accordingly to the current drawn by the detector.

gas volumes providing very consistent signals.

On the charge spectrum point of view, each layout has its own advantages. When the double-gap has the highest induced over drifting charge ratio, as seen in Figure 4.11, the multigap has a charge spectrum strongly detached from the origin, as visible in Figure 4.12. A high induced over drifting charge ratio means that the double gap can be safely operated at high threshold or that at similar threshold it can be operated with a twice smaller drifting charge, meaning a higher rate capability. On the other hand, the strong detachment of the charge spectrum from the origin in the MRPC case allows to reach a higher efficiency with increasing threshold as most of the induced charge is not low due to the convolution of several single gap spectra. The range of stable efficiency increases with the number of gap, as presented in Figure 4.13.

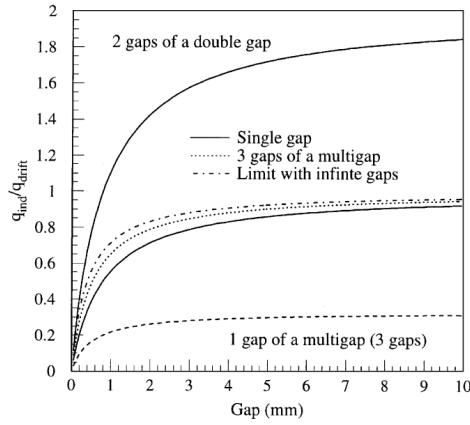


Figure 4.11: Ratio between total induced and drifting charge have been simulated for single gap, double-gap and multigap layouts [32]. The total induced charge for a double-gap RPC is a factor 2 higher than for a multigap.

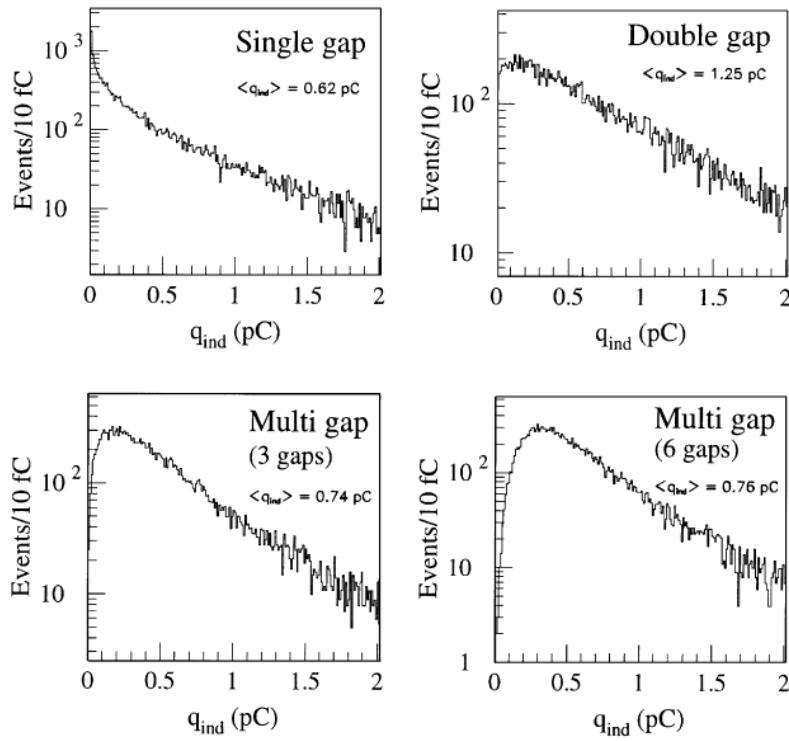


Figure 4.12: Charge spectra have been simulated for single gap, double-gap and multigap layouts [32]. It appears that when single gap shows a decreasing spectrum, double and multigap layouts exhibit a spectrum whose peak is detached from the origin. The detachment gets stronger as the number of gaps increases.

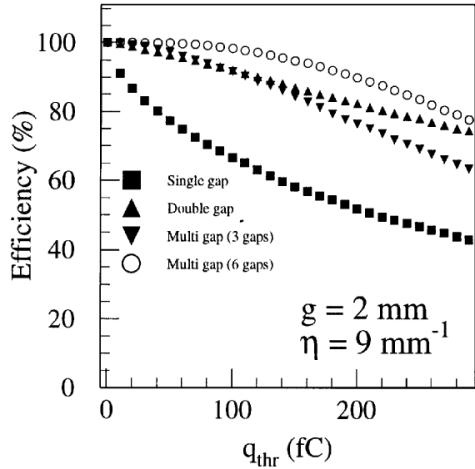


Figure 4.13: The maximal theoretical efficiency is simulated for single gap, double-gap and multigap layouts [32] at a constant gap thickness of 2 mm and using an effective Townsend coefficient of  $9 \text{ mm}^{-1}$ .

## 4.3 Resistive Plate Chambers at CMS

### 4.3.1 Overview

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].

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?]

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].

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 reduction and efficiency improvement for both trigger and offline reconstruction.

### 685 4.3.2 The present RPC system

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

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

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

707 The Link Board has 96 input channels (one channel corresponds to one RPC strip). The input  
 708 signals are the  $\sim 100$  ns binary pulses which are synchronous to the RPC hits, but not to the LHC  
 709 clock (which drives the entire CMS electronics). Thus the first step of the FEB signals processing  
 710 is synchronization, i.e. assignment of the signals to the BXes (25 ns periods). Then the data are  
 711 compressed with a simple zero-suppressing algorithm (the input channels are grouped into 8 bit  
 712 partitions, only the partitions with at least one nonzero bit are selected for each BX). Next, the non-  
 713 empty partitions are time-multiplexed i.e. if there are more than one such partition in a given BX,  
 714 they are sent one-by-one in consecutive BXes. The data from 3 neighbouring LBs are concentrated  
 715 by the middle LB which contains the optical transmitter for sending them to the USC over a fiber at  
 716 1.6 Gbps.

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

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

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

### 4.3.3 Pulse processing of CMS RPCs

Signals induced by cosmic particle in the RPC strips are shaped by standard CMS RPC Front-End Electronics (FEE) following the scheme of Figure 4.14. On a first stage, analogic signals are amplified and then sent to the Constant Fraction Discriminator (CFD) described in Figure 4.15. At the end of the chain, 100 ns long pulses are sent in the LVDS output. These output signal are sent on one side to a V1190A Time-to-Digital Converter (TDC) module from CAEN and on the other to an OR module to count the number of detected signals. Trigger and hit coincidences are monitored using scalers. The TDC is used to store the data into ROOT files. These files are thus analysed to understand the detectors performance.

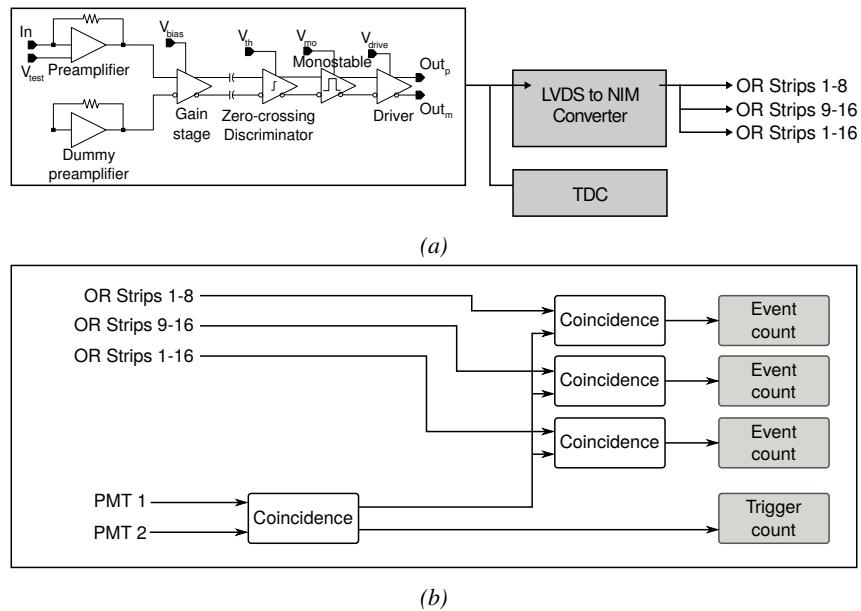


Figure 4.14: Signals from the RPC strips are shaped by the FEE described on Figure 4.14a. Output LVDS signals are then read-out by a TDC module connected to a computer or converted into NIM and sent to scalers. Figure 4.14b describes how these converted signals are put in coincidence with the trigger.

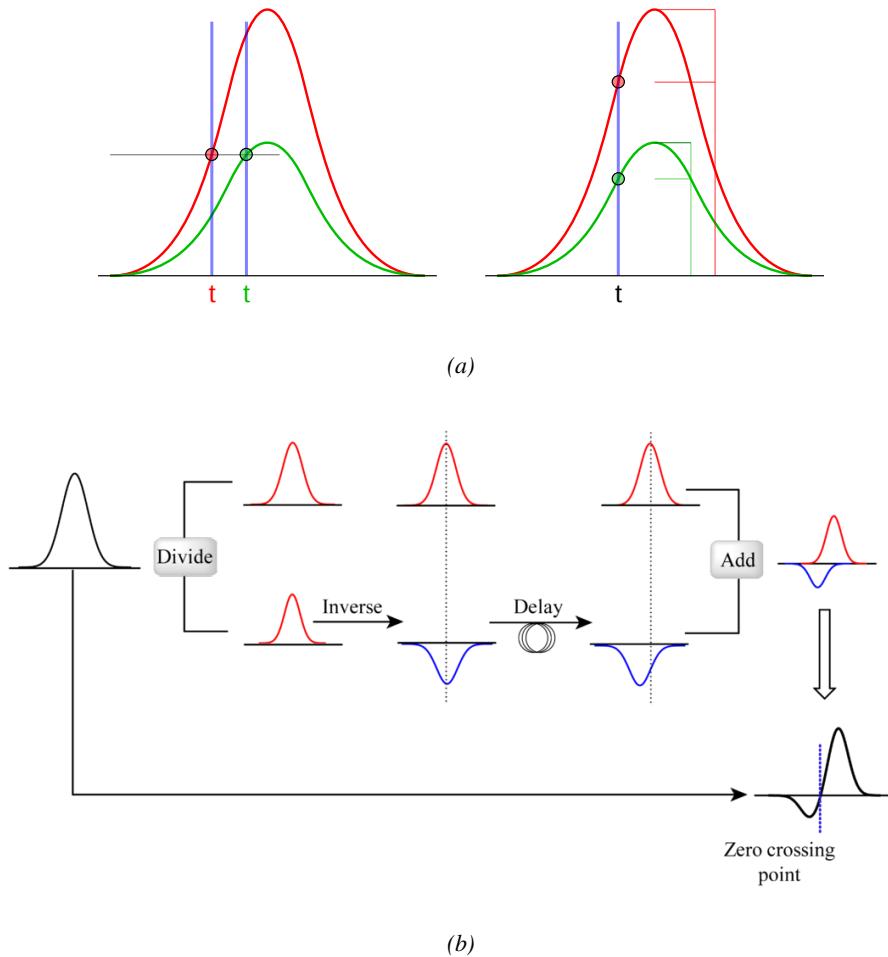


Figure 4.15: Description of the principle of a CFD. A comparison of threshold triggering (left) and constant fraction triggering (right) is shown in Figure 4.15a. Constant fraction triggering is obtained thanks to zero-crossing technique as explained in Figure 4.15b. The signal arriving at the input 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.

# 5

738

739

740

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

741

### 5.1 Testing detectors under extreme conditions

742

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

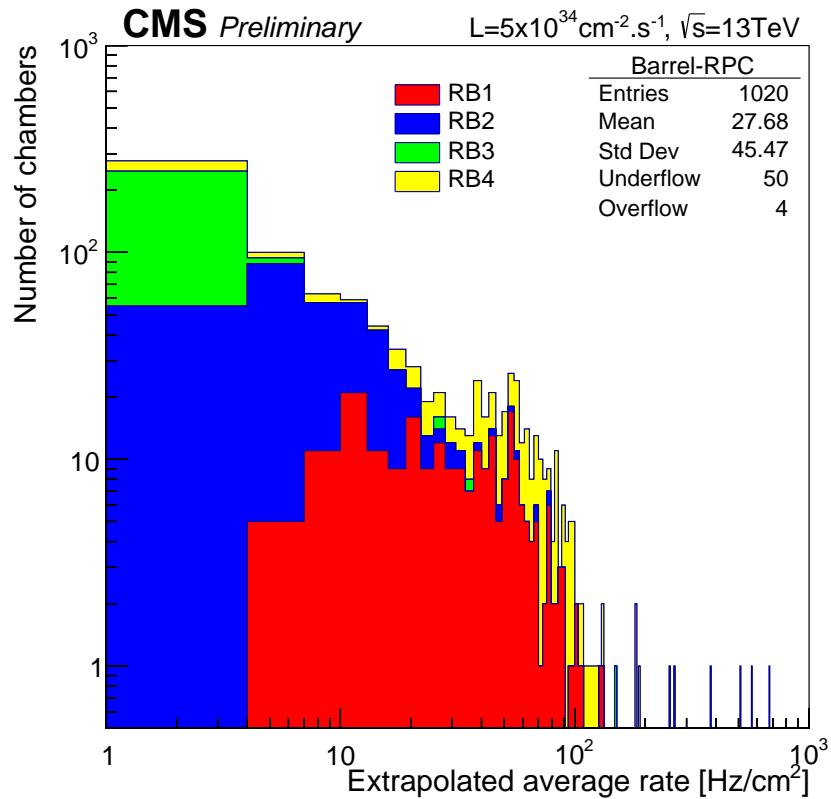
[To update.]

749

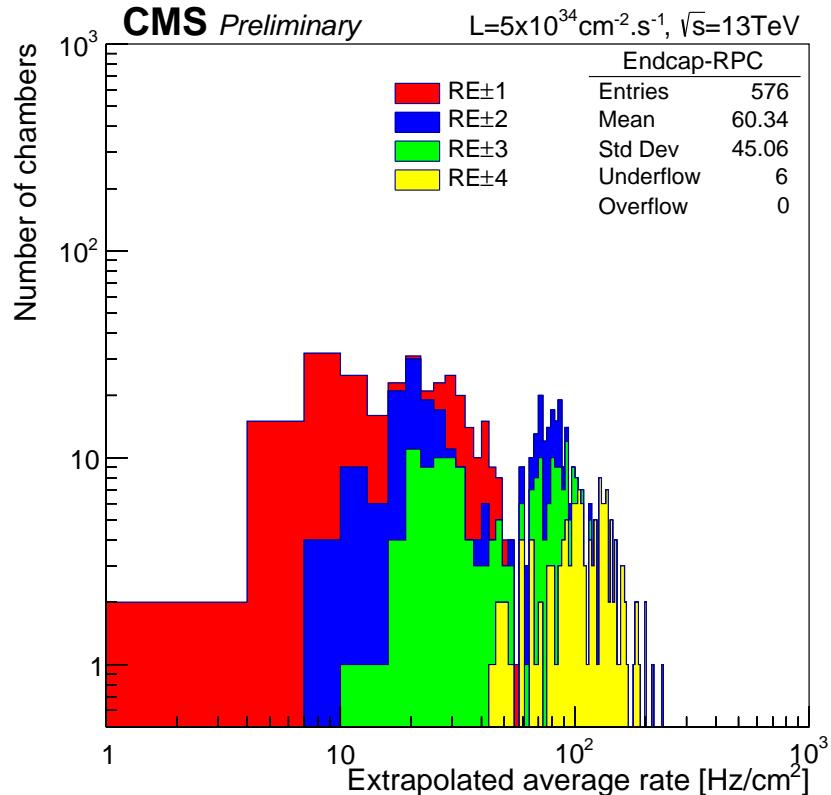
750

The 2016 data allowed to study the values of the background rate in all RPC system. In Figure 5.1, the distribution of the chamber background hit rate per unit area is shown at a luminosity of  $5 \times 10^{34} \text{ cm}^{-2} \cdot \text{s}^{-1}$  linearly extrapolating from data collected in 2016 [ref mentioning the linear dependency of rate vs lumi]. The maximum rate per unit area at HL-LHC conditions is expected to be of the order of  $600 \text{ Hz/cm}^2$  (including a safety factor 3). Nevertheless, Fluka simulations have conducted in order to understand the background at HL-LHC conditions. The comparison to the data has shown, in Figure 5.2, a discrepancy of a factor 2 even though the order of magnitude is consistent. [Understand mismatch.]

758



(a)



(b)

Figure 5.1: (5.1a) Extrapolation from 2016 data of single hit rate per unit area in the barrel region. (5.1b) Extrapolation from 2016 data of single hit rate per unit area in the endcap region.

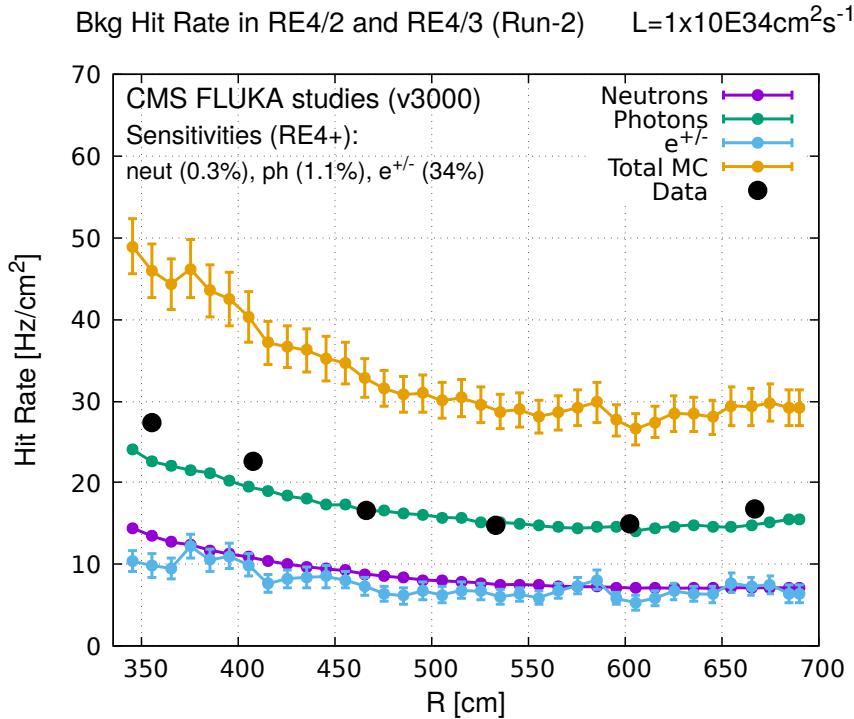


Figure 5.2: Background Fluka simulation compared to 2016 Data at  $L = 10^{34}\text{cm}^{-2}\cdot\text{s}^{-1}$  in the fourth endcap disk region. A mismatch in between simulation and data can be observed. [\[To be understood.\]](#)

In the past, extensive long-term tests were carried out at several gamma and neutron facilities certifying the detector performance. Both full size and small prototype RPCs have been irradiated with photons up to an integrated charge of  $\sim 0.05C/\text{cm}^2$  and  $\sim 0.4C/\text{cm}^2$ , respectively [33, 34]. During Run-I, the RPC system provided stable operation and excellent performance and did not show any aging effects for integrated charge of the order of  $0.01C/\text{cm}^2$ . Projections on currents from 2016 Data, has allowed to determine that the total integrated charge, by the end of HL-LHC, would be of the order of  $1C/\text{cm}^2$  (including a safety factor 3). [\[Corresponding figure needed.\]](#)

766

### 767 5.1.1 The Gamma Irradiation Facilities

768

#### 5.1.1.1 GIF

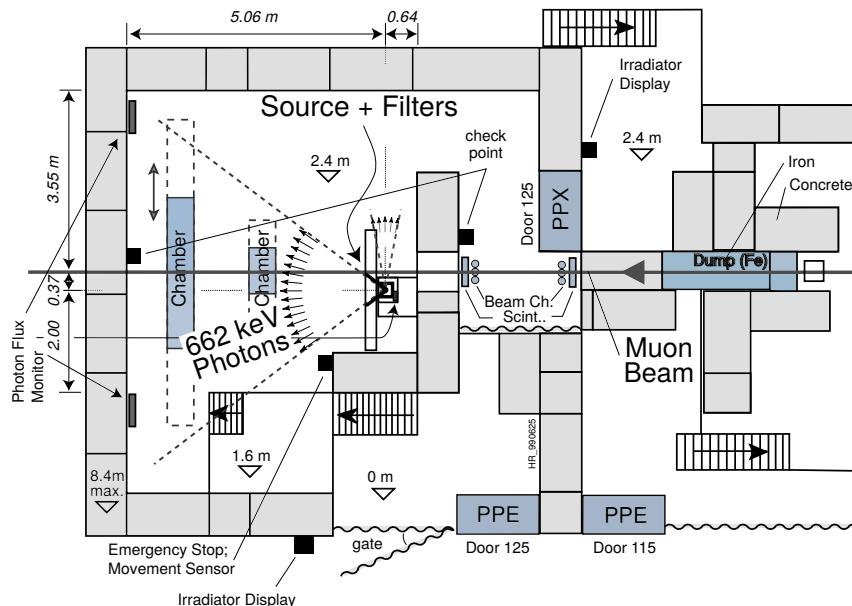
769

Located in the SPS West Area at the downstream end of the X5 test beam, the Gamma Irradiation Facility (GIF) was a test area in which particle detectors were exposed to a particle beam in presence of an adjustable gamma background [35]. Its goal was to reproduce background conditions these detectors would suffer in their operating environment at LHC. GIF layout is shown in Figure 5.3. Gamma photons are produced by a strong  $^{137}\text{Cs}$  source installed in the upstream part of the zone inside a lead container. The source container includes a collimator, designed to irradiate a  $6 \times 6\text{ m}^2$  area at 5 m maximum to the source. A thin lens-shaped lead filter helps providing with a uniform outgoing flux in a vertical plane, orthogonal to the beam direction. The principal collimator hole provides a pyramidal aperture of  $74^\circ \times 74^\circ$  solid angle and provides a photon flux in a pyramidal vol-

770

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

783



*Figure 5.3: Layout of the test beam zone called X5c GIF at CERN. Photons from the radioactive source produce a sustained high rate of random hits over the whole area. The zone is surrounded by 8 m high and 80 cm thick concrete walls. Access is possible through three entry points. Two access doors for personnel and one large gate for material. A crane allows installation of heavy equipment in the area.*

As described on Figure 5.4, the  $^{137}\text{Cs}$  source emits a 662 keV photon in 85% of the decays. An activity of 740 GBq was measured on the 5<sup>th</sup> March 1997. To estimate the strength of the flux in 2014, it is necessary to consider the nuclear decay through time assiciated to the Cesium source whose half-life is well known ( $t_{1/2} = (30.05 \pm 0.08)$  y). The GIF tests where done in between the 20<sup>th</sup> and the 31<sup>st</sup> of August 2014, i.e. at a time  $t = (17.47 \pm 0.02)$  y resulting in an attenuation of the activity from 740 GBq in 1997 to 494 GBq in 2014.

790

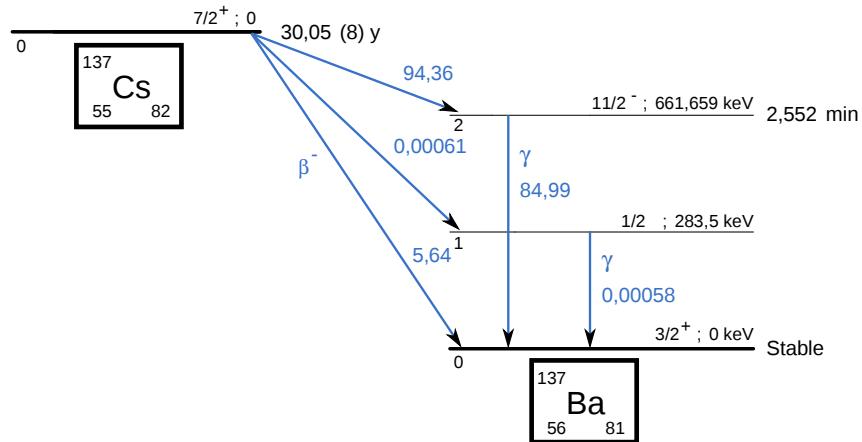


Figure 5.4:  $^{137}\text{Cs}$  decays by  $\beta^-$  emission to the ground state of  $^{137}\text{Ba}$  ( $\text{BR} = 5.64\%$ ) and via the 662 keV isomeric level of  $^{137}\text{Ba}$  ( $\text{BR} = 94.36\%$ ) whose half-life is 2.55 min.

### 5.1.1.2 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 [36]. 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 5.5.

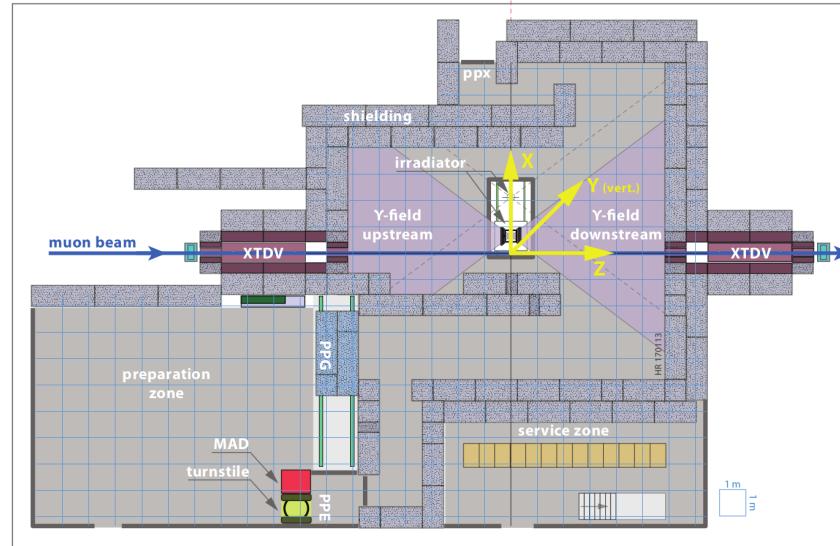


Figure 5.5: Floor plan of the GIF++ facility. When the facility downstream of the GIF++ takes electron beam, a beam pipe is installed along the beam line ( $z$ -axis). The irradiator can be displaced laterally (its center moves from  $x = 0.65 \text{ m}$  to  $2.15 \text{ m}$ ), to increase the distance to the beam pipe.

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

802

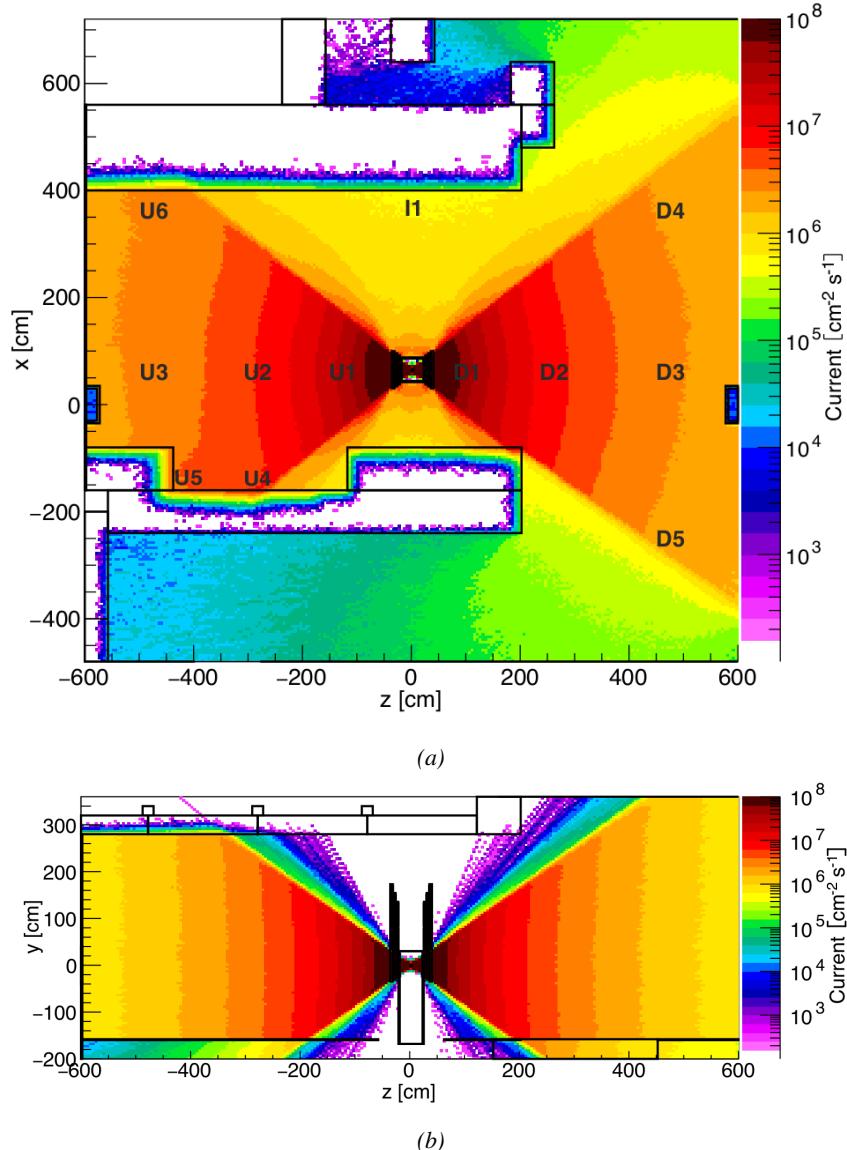


Figure 5.6: Simulated unattenuated current of photons in the xz plane (Figure 5.6a) and yz plane (Figure 5.6b) through the source at  $x = 0.65 \text{ m}$  and  $y = 0 \text{ m}$ . With angular correction filters, the current of 662 keV photons is made uniform in xy planes.

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

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

809

## 810 5.2 Preliminary tests at GIF

### 811 5.2.1 Resistive Plate Chamber test setup

812 During summer 2014, preliminary tests have been conducted in the GIF area on a newly produced  
 813 RE4/2 chamber labelled RE-4-2-BARC-161. This chamber has been placed into a trolley covered  
 814 with a tent. The position of the RPC inside the tent and of the tent related to the source is described  
 815 in Figure 5.7. To test this CMS RPC, three different absorber settings were used. First of all,  
 816 measurements were done with fully opened source. Then, to complete this preliminary study, the  
 817 gamma flux has been attenuated from a factor 2 and a factor 5. The expected gamma flux at the level  
 818 of our detector will be discussed in subsection ??.

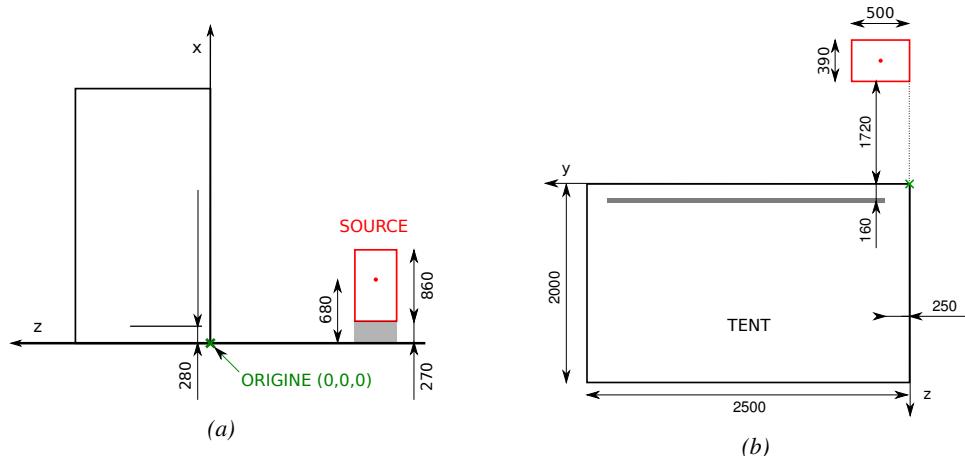
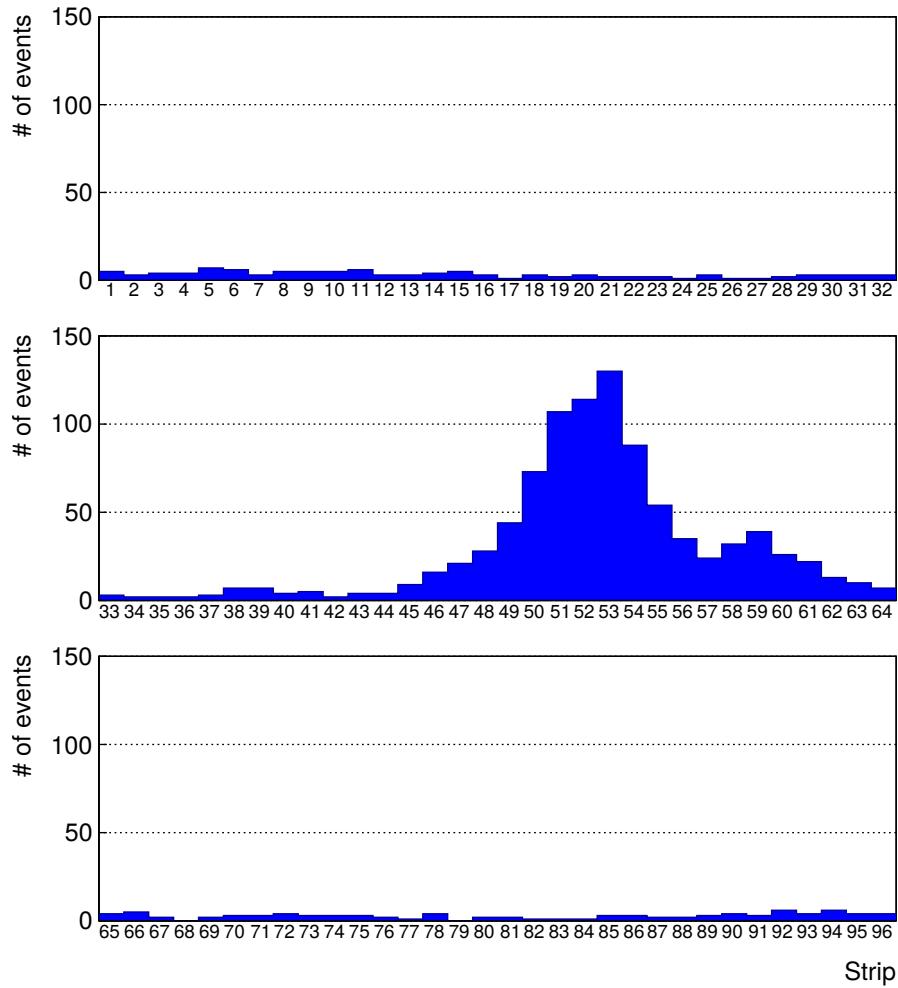


Figure 5.7: Description of the RPC setup. Dimensions are given in mm. A tent containing RPCs is placed at 1720 mm from the source container. The source is situated in the center of the container. RE-4-2-BARC-161 chamber is 160 mm inside the tent. This way, the distance between the source and the chambers plan is 2060 mm. Figure 5.7a provides a side view of the setup in the xz plane while Figure 5.7b shows a top view in the yz plane.



*Figure 5.8: RE-4-2-BARC-161 chamber is inside the tent as described in Figure 5.7. 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.*

At the time of the tests, the beam not being operational anymore, a trigger composed of 2 plastic scintillators has been placed in front of the setup with an inclination of 10 deg with respect to the detector plane in order to look at cosmic muons. Using this particular trigger layout, shown on Figure 5.8, leads to a cosmic muon hit distribution into the chamber similar to the one in Figure 5.9. Measured without gamma irradiation, two peaks can be seen on the profil of partition B, centered on strips 52 and 59. Section ?? will help us understand that these two peaks are due respectively to forward and backward coming cosmic particles where forward coming particles are first detected by the scintillators and then the RPC while the backward coming muons are first detected in the RPC.



*Figure 5.9: 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.*

### 5.2.2 Data Acquisition

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

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

protect the photomultipliers from gammas as can be seen from Figure 5.8.

An inclination has been given to the cosmic telescope to maximize the muon flux. A good compromise had to be found between good enough muon flux and narrow enough hit distribution to be sure to contain all the events into only one half partitions as required from the limited available readout hardware. Nevertheless, a consequence of the misplaced trigger, that can be seen as a loss of events in half-partition B1 in Figure 5.9, is an inefficiency. Nevertheless, the inefficiency of approximately 20 % highlighted in Figure 5.10 by comparing the performance of chamber BARC-161 in 904 and at GIF without irradiation seems too important to be explained only by the geometrical acceptance of the setup itself. Simulations have been conducted to show how the setup brings inefficiency.

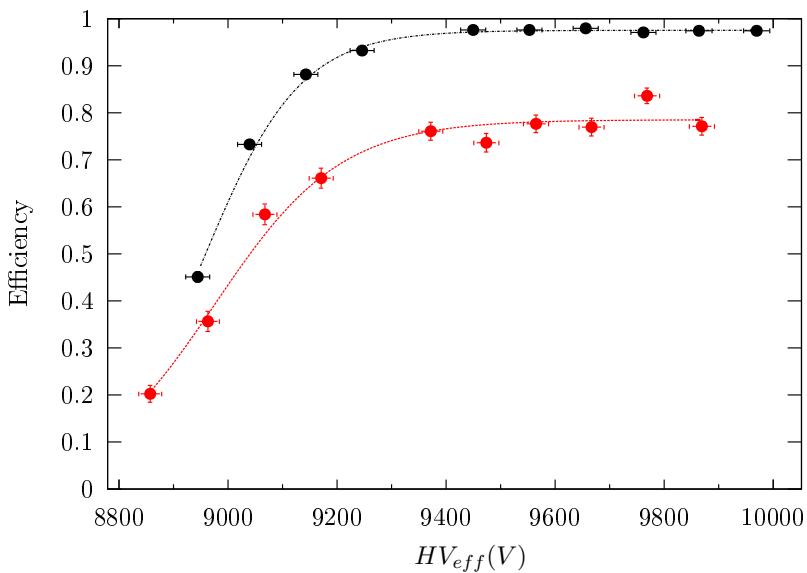
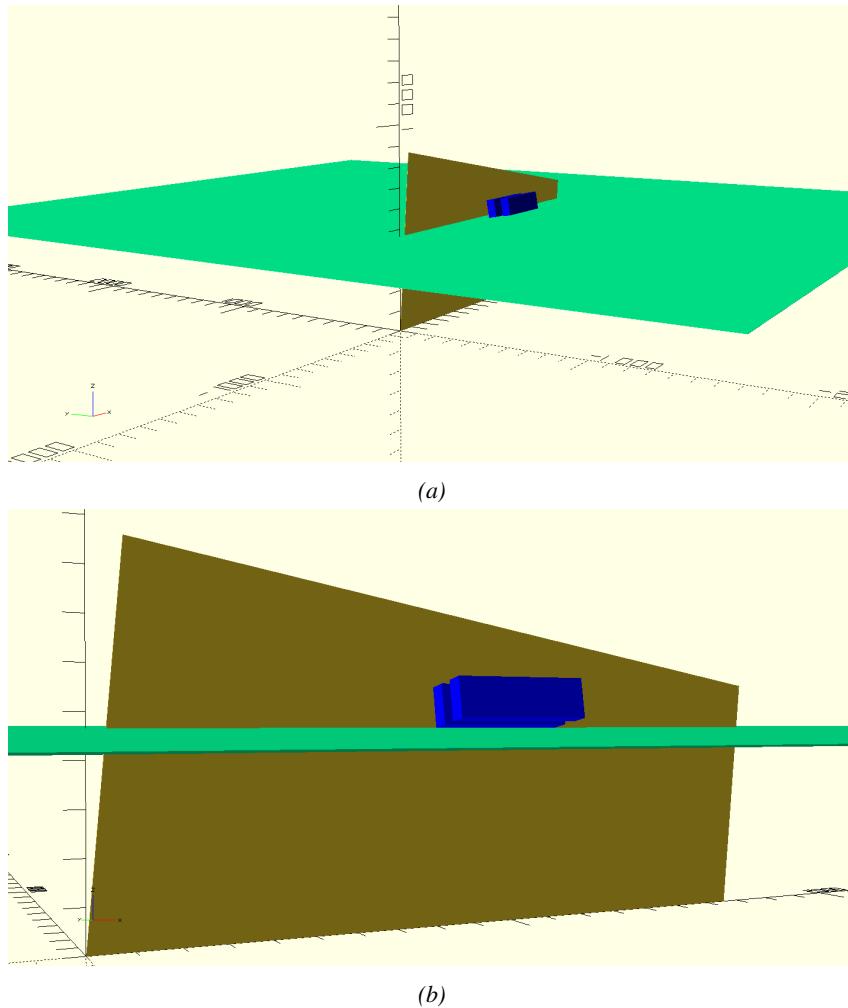


Figure 5.10: Results are derived from data taken on half-partition B2 only. On the 18<sup>th</sup> of June 2014, data has been taken on chamber RE-2-BARC-161 at building 904 (Prevessin Site) with cosmic muons providing us a reference efficiency plateau of  $(97.54 \pm 0.15)\%$  represented by a black 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.

#### 5.2.3.1 Description of the simulation layout

The layout of GIF setup has been reproduced and incorporated into a Monte Carlo (MC) simulation to study the influence of the disposition of the telescope on the final distribution measured by the RPC. A 3D view of the simulated layout is given into Figure 5.11. Muons are generated randomly in a horizontal plane located at a height corresponding to the lowest point of the PMTs. This way, the needed size of the plane in order to simulate events happening at very big azimuthal angles (i.e.  $\theta \approx \pi$ ) can be kept relatively small. The muon flux is designed to follow the usual  $\cos^2\theta$  distribution for cosmic particle. The goal of the simulation is to look at muons that pass through the muon telescope composed of the two scintillators and define their distribution onto the RPC plane. During the reconstruction, the RPC plane is then divided into its strips and each muon track is assigned to a strip.



*Figure 5.11: 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 5.7a shows a global view of the simulated setup. Figure 5.7b shows a zoomed view that allows to see the 2 scintillators as well as the full RPC plane.*

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

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

869 **5.2.3.2 Simulation procedure**

870 The simulation software has been designed using C++ and the output data is saved into ROOT  
 871 histograms. Simulations start for a threshold  $T_{scint}$  varying in a range from 0 to 45 mm in steps  
 872 of 5 mm, where  $T_{scint} = 0$  mm corresponds to the case where there isn't any threshold apply on  
 873 the input signal while  $T_{scint} = 45$  mm, which is the scintillator thickness, is the case where muons  
 874 cannot arrive orthogonally onto the scintillator surface. For a given  $T_{scint}$ , a set of RPC thresholds  
 875 are considered. The RPC threshold,  $T_{RPC}$  varies from 2 mm, the thickness of the gas volume, to  
 876 3 mm in steps of 0.25 mm. For each  $(T_{scint}; T_{RPC})$  pair,  $N_\mu = 10^8$  muons are randomly generated  
 877 inside the muon plane described in the previous paragraph with an azimuthal angle  $\theta$  chosen to follow  
 878 a  $\cos^2\theta$  distribution.

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

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

897 This simulation is then repeated for different telescope inclinations ranging in between 4 and  $20^\circ$   
 898 and varying in steps of  $2^\circ$ . Due to this inclination and to the vertical position of the detector under  
 899 test, the muon distribution reconstructed in the detector plane is asymmetrical. The choice as been  
 900 made to chose a skew distribution formula to fit the data built as the multiplication of gaussian and  
 901 sigmoidal curves together. A typical gaussian formula is given as 5.1 and has three free parameters  
 902 as  $A_g$ , its amplitude,  $\bar{x}$ , its mean value and  $\sigma$ , its root mean square. Sigmoidal curves as given by  
 903 formula 5.2 are functions converging to 0 and  $A_s$  as  $x$  diverges. The inflection point is given as  $x_i$   
 904 and  $\lambda$  is proportional to the slope at  $x = x_i$ . In the limit where  $\lambda \rightarrow \infty$ , the sigmoid becomes a  
 905 step function.

$$g(x) = A_g e^{-\frac{(x-\bar{x})^2}{2\sigma^2}} \quad (5.1)$$

$$s(x) = \frac{A_s}{1 + e^{-\lambda(x-x_i)}} \quad (5.2)$$

Finally, a possible representation of a skew distribution is given by formula 5.3 and is the product of 5.1 and 5.2. Naturally, here  $A_{sk} = A_g \times A_s$  and represents the theoretical maximum in the limit where the skew tends to a gaussian function.

$$sk(x) = g(x) \times s(x) = A_{sk} \frac{e^{\frac{-(x-\bar{x})^2}{2\sigma^2}}}{1 + e^{-\lambda(x-x_i)}} \quad (5.3)$$

### 5.2.3.3 Results

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

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

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

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

### 5.2.4 Photon flux at GIF

#### 5.2.4.1 Expectations from simulations

In order to understand and evaluate the  $\gamma$  flux in the GIF area, simulations had been conducted in 1999 and published by S. Agosteo et al [35]. Table 5.1 presented in this article gives us the  $\gamma$  flux for different distances  $D$  to the source. This simulation was done using GEANT and a Monte Carlo N-Particle (MCNP) transport code, and the flux  $F$  is given in number of  $\gamma$  per unit area and unit time along with the estimated error from these packages expressed in %.

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

Table 5.1: 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 [35].

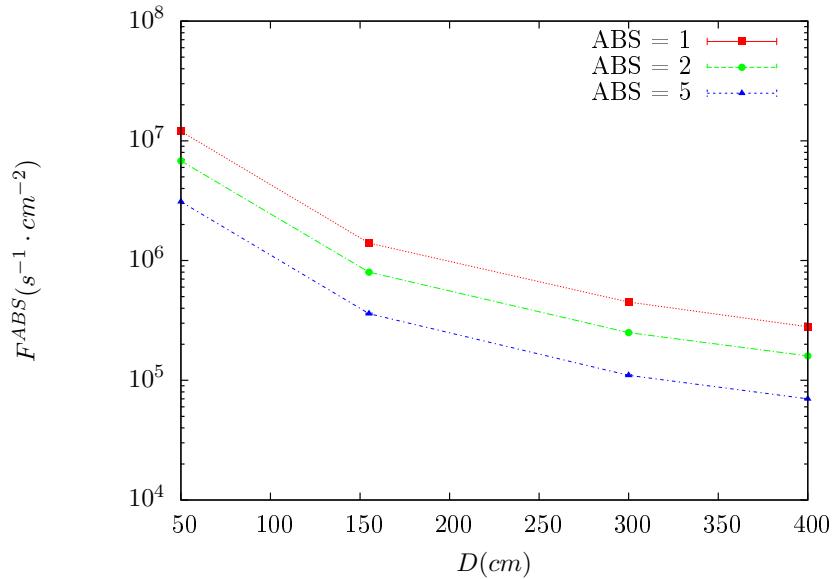


Figure 5.12:  $\gamma$  flux  $F(D)$  is plot using values from table 5.1. As expected, the plot shows similar attenuation behaviours with increasing distance for each absorption factors.

The simulation doesn't directly provide us with an estimated flux at the level of our RPC. First of all, it is needed to extract the value of the flux from the available data contained in the original paper and then to estimate the flux in 2014 at the time the experimentation took place. Figure 5.12 that contains the data from Table 5.1. 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 5.4, assuming that the flux decreases as  $1/D^2$ , where  $c$  is a fitting factor.

$$F^{ABS} = F_0^{ABS} \times \left( \frac{cD_0}{D} \right)^2 \quad (5.4)$$

By rewriting Equation 5.4, it comes that :

$$c = \frac{D}{D_0} \sqrt{\frac{F^{ABS}}{F_0^{ABS}}} \quad (5.5)$$

$$\Delta c = \frac{c}{2} \left( \frac{\Delta F^{ABS}}{F^{ABS}} + \frac{\Delta F_0^{ABS}}{F_0^{ABS}} \right) \quad (5.6)$$

Finally, using Equation 5.5 and the data in Table 5.1 with  $D_0 = 50$  cm as reference point, we can build Table 5.2. 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 5.2: Correction factor  $c$  is computed thanks to formulae 5.5 taking as reference  $D_0 = 50$  cm and the associated flux  $F_0^{ABS}$  for each absorption factor available in table 5.1.

932 For the range of  $D/D_0$  values available, it is possible to use a simple linear fit to get the evolution  
 933 of  $c$ . The linear fit will then use only 2 free parameters,  $a$  and  $b$ , as written in Equation 5.7. This gives  
 934 us the results showed in Figure 5.13. Figure 5.13b confirms that using only a linear fit to extract  $c$  is  
 935 enough as the evolution of the rate that can be obtained superimposes well on the simulation points.

$$c \left( \frac{D}{D_0} \right) = a \frac{D}{D_0} + b \quad (5.7)$$

$$F^{ABS} = F_0^{ABS} \left( a + \frac{bD_0}{D} \right)^2 \quad (5.8)$$

$$\Delta F^{ABS} = F^{ABS} \left[ \frac{\Delta F_0^{ABS}}{F_0^{ABS}} + 2 \frac{\Delta a + \Delta b \frac{D_0}{D}}{a + \frac{bD_0}{D}} \right] \quad (5.9)$$

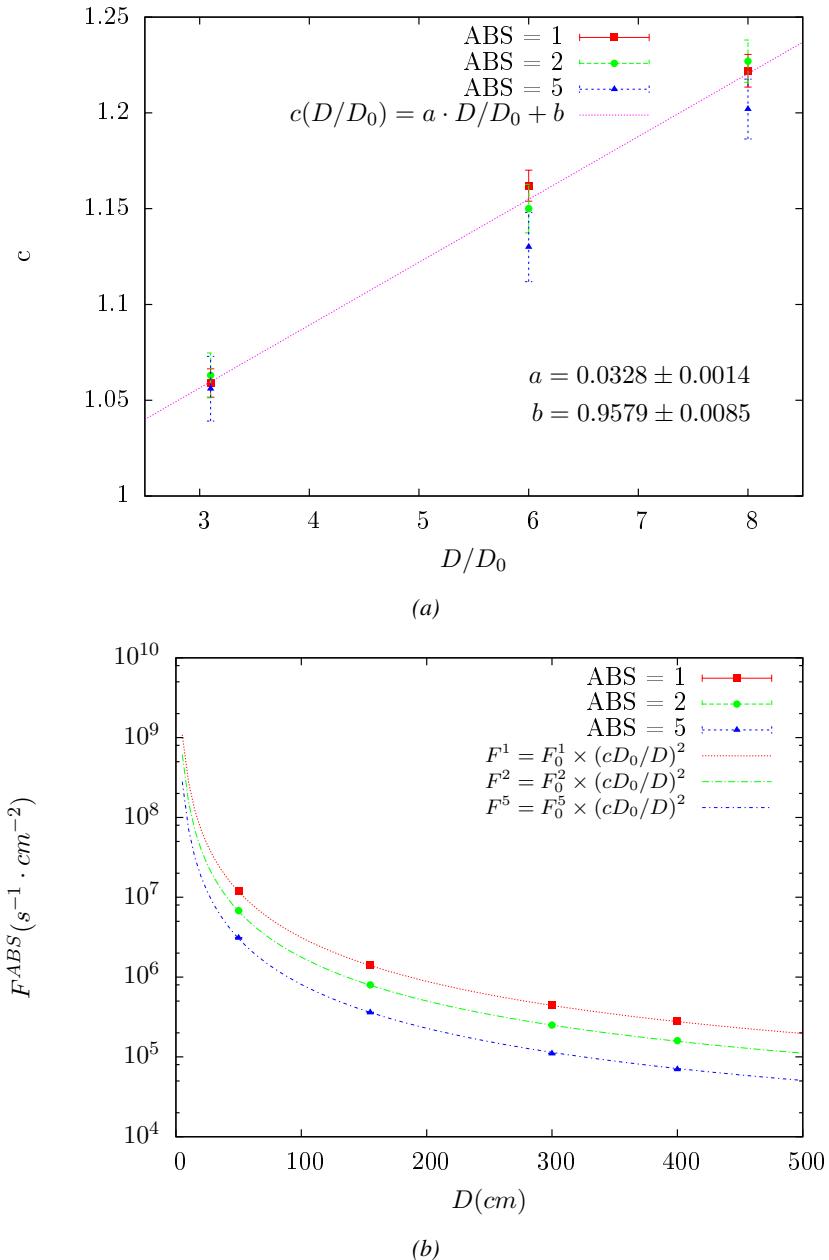


Figure 5.13: Figure 5.13a shows the linear approximation fit done via formulae 5.7 on data from table 5.2. Figure 5.13b shows a comparison of this model with the simulated flux using  $a$  and  $b$  given in figure 5.13a in formulae 5.4 and the reference value  $D_0 = 50\text{cm}$  and the associated flux for each absorption factor  $F_0^{ABS}$  from table 5.1

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

936 In the case of the 2014 GIF tests, the RPC plane is located at a distance  $D = 206\text{ cm}$  to the source.

937 Moreover, to estimate the strength of the flux in 2014, it is necessary to consider the nuclear decay

938 through time associated to the Cesium source whose half-life is well known ( $t_{1/2} = (30.05 \pm 0.08)\text{ y}$ ).

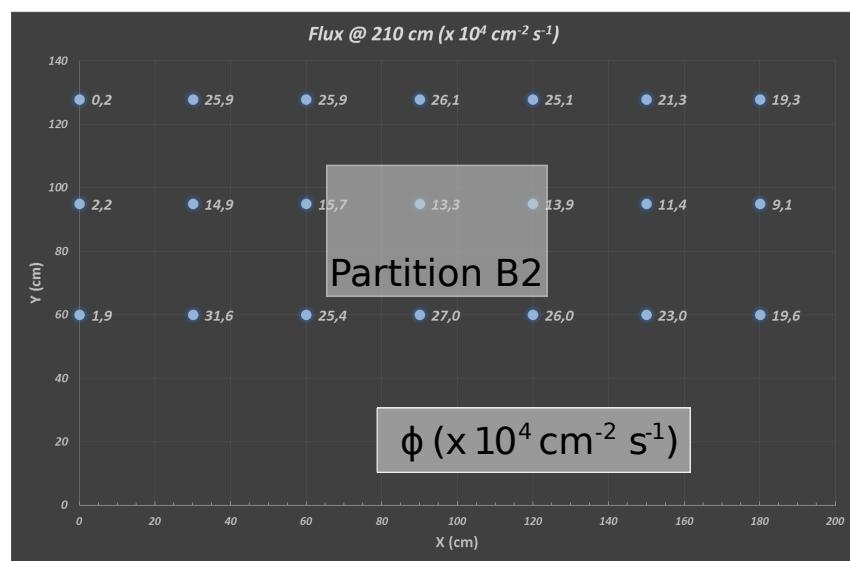
939 The very first source activity measurement has been done on the 5<sup>th</sup> of March 1997 while the GIF

940 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  
 941 resulting in an attenuation of the activity from 740 GBq in 1997 to 494 GBq in 2014. All the needed  
 942 information to extrapolate the flux through our detector in 2014 has now been assembled, leading  
 943 to the Table 5.3. It is interesting to note that for a common RPC sensitivity to  $\gamma$  of  $2 \cdot 10^{-3}$ , the  
 944 order of magnitude of the estimated hit rate per unit area is of the order of the kHz for the fully  
 945 opened source. Moreover, taking profit of the two working absorbers, it will be possible to scan  
 946 background rates at 0 Hz,  $\sim 300$  Hz as well as  $\sim 600$  Hz. Without source, a good estimate of the  
 947 intrinsic performance will be available. Then at 300 Hz, the goal will be to show that the detectors  
 948 fulfill the performance certification of CMS RPCs. Then a first idea of the performance of the  
 949 detectors at higher background will be provided with absorption factors 2 ( $\sim 600$  Hz) and 1 (no  
 950 absorption). *[Here I will also put a reference to the plot showing the estimated background rate at  
 951 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 5.3: The data at  $D_0$  in 1997 is taken from [35]. In a second step, using Equations 5.8 and 5.9, 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.

952 **5.2.4.2 Dose measurements**



*Figure 5.14: 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.*

953 **5.2.5 Results and discussions**

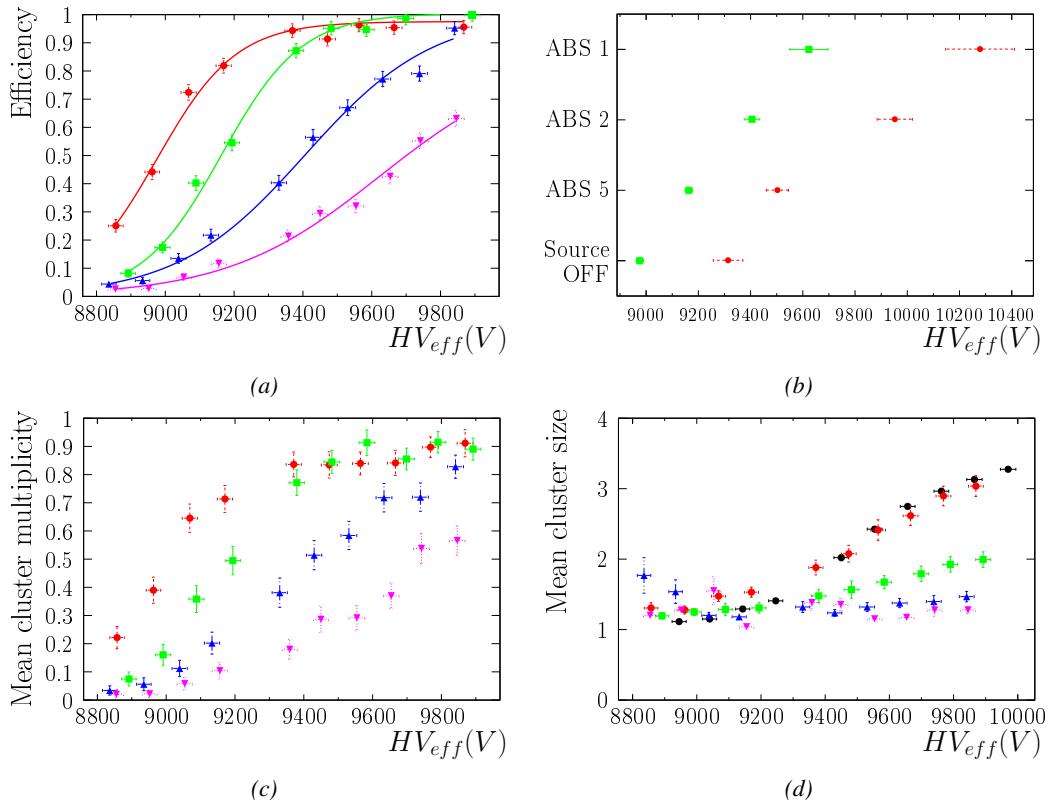


Figure 5.15

### 954 5.3 Longevity tests at GIF++

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

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

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

976 Figures 5.16 and 5.17 give us for different test beam periods, and thus for increasing integrated  
 977 charge through time, a comparison of the maximum efficiency, obtained using a sigmoid-like func-  
 978 tion, and of the working point of both irradiated and non irradiated chambers [**SIGMOID2005**]. No  
 979 aging is yet to see from this data, the shifts in  $\gamma$  rate per unit area in between irradiated and non  
 980 irradiated detectors and RE2 and RE4 types being easily explained by a difference of sensitivity due  
 981 to the various Bakelite resistivities of the HPL electrodes used for the electrode production.

982 Collecting performance data at each test beam period allows us to extrapolate the maximum effi-  
 983 ciency for a background hit rate of  $300\text{ Hz}/\text{cm}^2$  corresponding to the expected HL-LHC conditions.  
 984 Aging effects could emerge from a loss of efficiency with increasing integrated charge over time,  
 985 thus Figure 5.18 helps us understand such degradation of the performance of irradiated detectors in  
 986 comparison with non irradiated ones. The final answer for an eventual loss of efficiency is given in  
 987 Figure 5.19 by comparing for both irradiated and non irradiated detectors the efficiency sigmoids  
 988 before and after the longevity study. Moreover, to complete the performance information, the Bake-  
 989 lite resistivity is regularly measured thanks to  $\text{Ag}$  scans (Figure 5.20) and the noise rate is monitored  
 990 weekly during irradiation periods (Figure 5.21). At the end of 2016, no signs of aging were observed  
 991 and further investigation is needed to get closer to the final integrated charge requirements proposed  
 992 for the longevity study of the present CMS RPC sub-system.

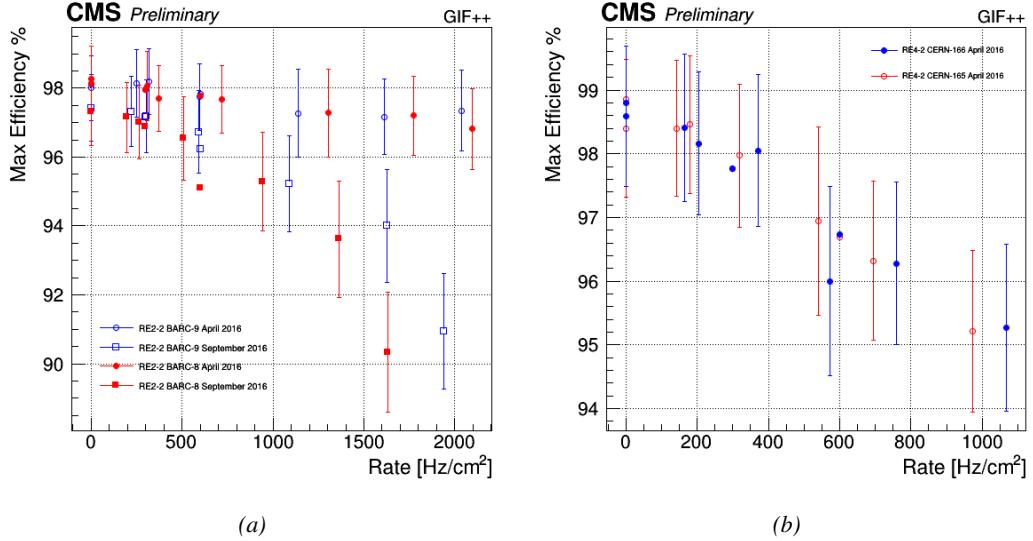


Figure 5.16: Evolution of the maximum efficiency for RE2 (5.16a) and RE4 (5.16b) chambers with increasing extrapolated  $\gamma$  rate per unit area at working point. Both irradiated (blue) and non irradiated (red) chambers are shown.

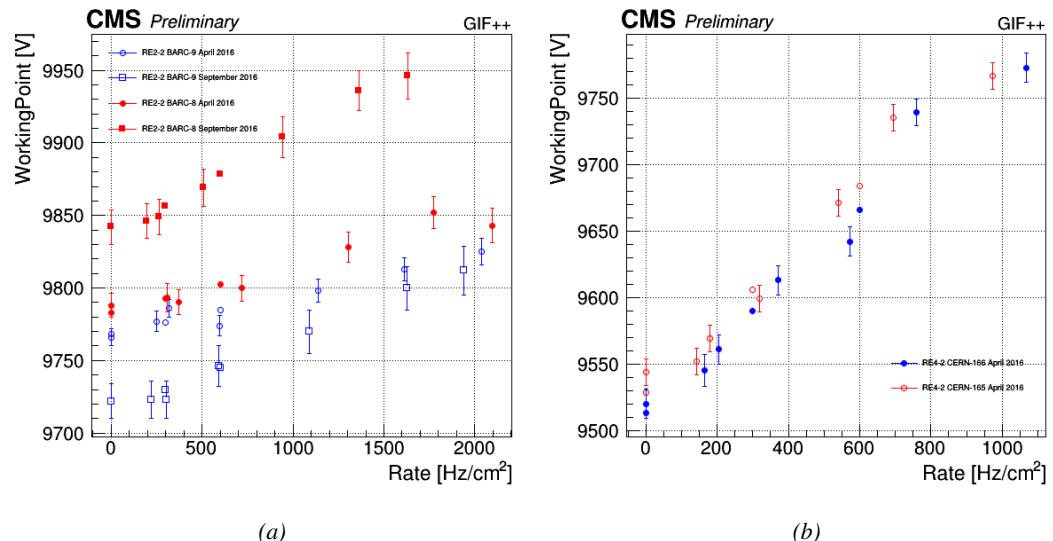


Figure 5.17: Evolution of the working point for RE2 (5.17a) and RE4 (5.17b) with increasing extrapolated  $\gamma$  rate per unit area at working point. Both irradiated (blue) and non irradiated (red) chambers are shown.

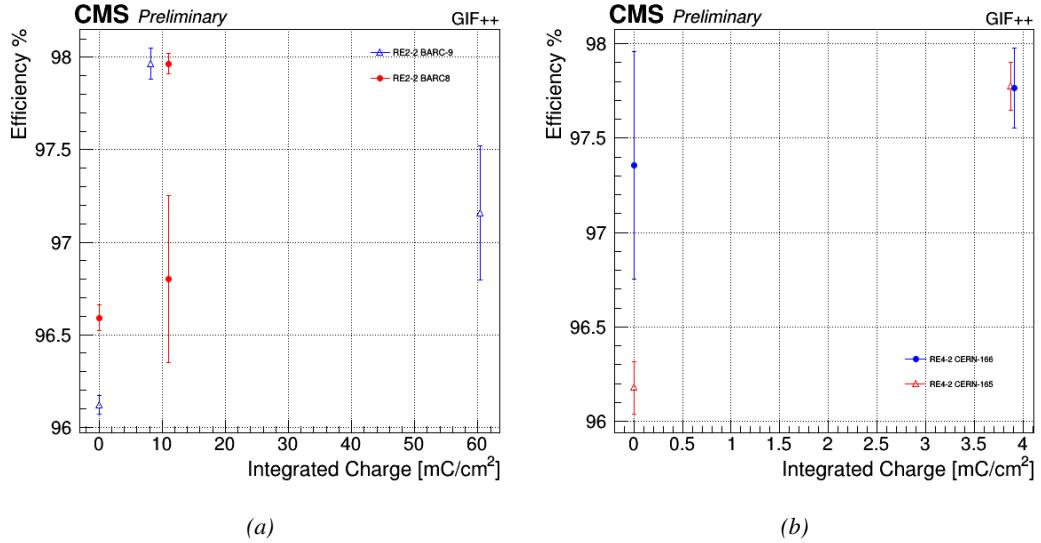


Figure 5.18: Evolution of the maximum efficiency at HL-LHC conditions, i.e. a background hit rate per unit area of  $300 \text{ Hz}/\text{cm}^2$ , with increasing integrated charge for RE2 (5.18a) and RE4 (5.18b) detectors. Both irradiated (blue) and non irradiated (red) chambers are shown. The integrated charge for non irradiated detectors is recorded during test beam periods and stays small with respect to the charge accumulated in irradiated chambers.

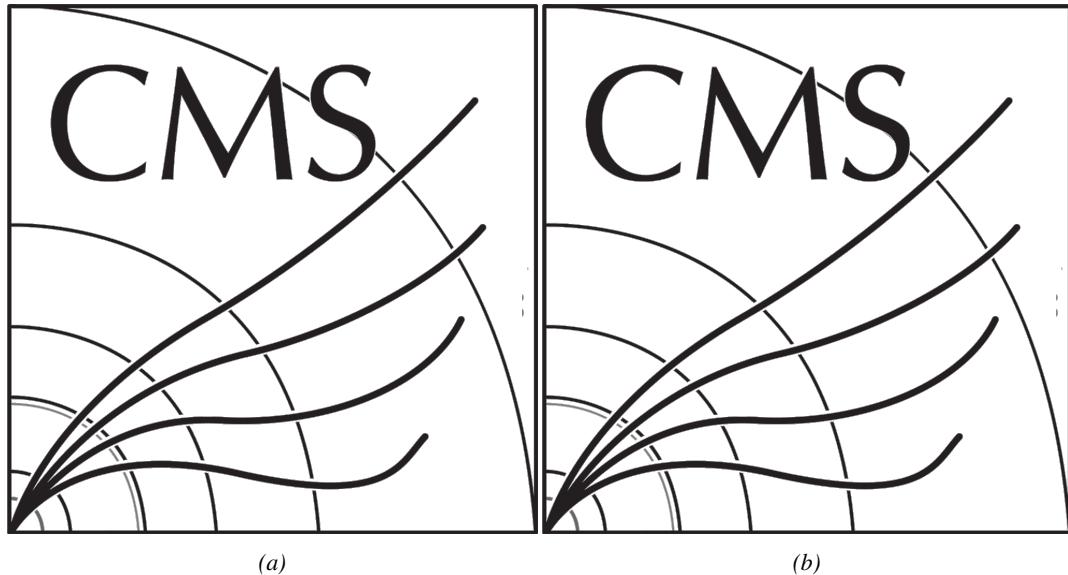


Figure 5.19: Comparison of the efficiency sigmoid before (triangles) and after (circles) irradiation for RE2 (5.19a) and RE4 (5.19b) detectors. Both irradiated (blue) and non irradiated (red) chambers are shown.

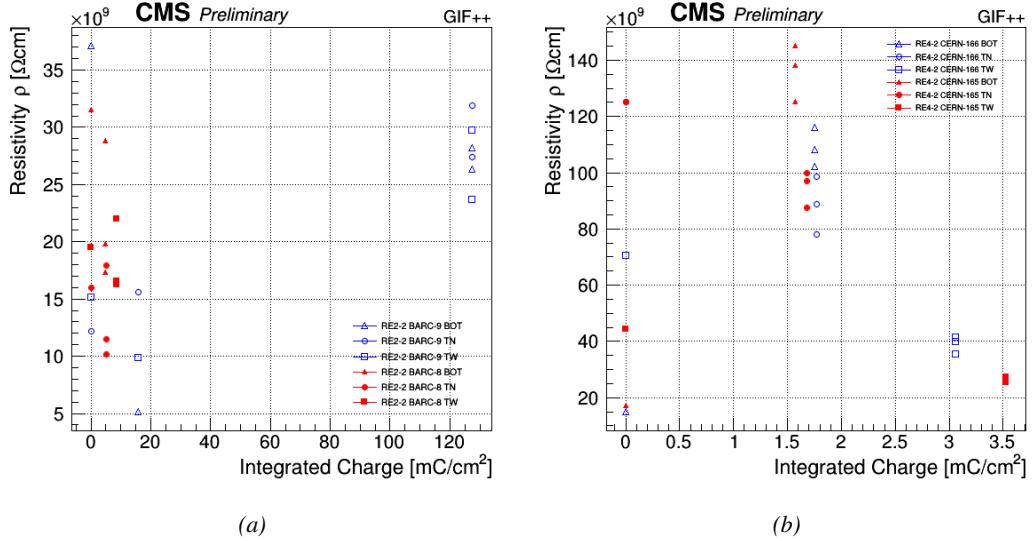


Figure 5.20: Evolution of the Bakelite resistivity for RE2 (5.20a) and RE4 (5.20b) detectors. Both irradiated (blue) and non-irradiated (red) chambers are shown.

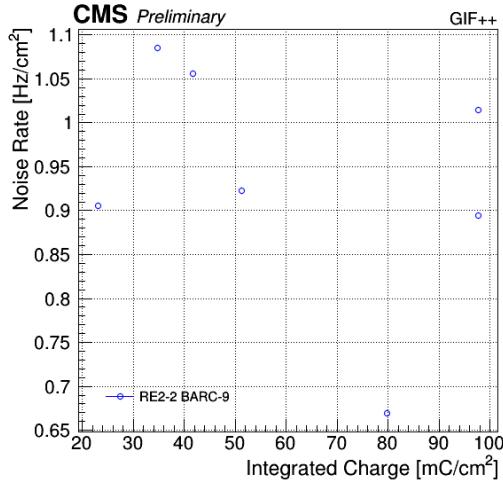


Figure 5.21: Evolution of the noise rate per unit area for the irradiated chamber RE2-2-BARC-9 only.

### 5.3.1 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

1001 moved outside the beam line and is placed in a distance of 5.2 m to the source, which irradiates the  
 1002 bunker using an attenuation filter of 2.2 which corresponds to a fluence of  $10^7 \text{ gamma/cm}^2$ .

1003 During GIF++ operation, the data collected can be divided into different categories as several  
 1004 parameters are monitored in addition to the usual RPC performance data. On one hand, to know  
 1005 the performance of a chamber, it is need to measure its efficiency and to know the background  
 1006 conditions in which it is operated. To do this, the hit signals from the chamber are recorded and  
 1007 stored in a ROOT file via a Data Acquisition (DAQ) software. On the other hand, it is also very  
 1008 important to monitor parameters such as environmental pressure and temperature, gas temperature  
 1009 and humidity, RPC HV, LV, and currents, or even source and beam status. This is done through the  
 1010 GIF++ web Detector Control Software (DCS) that stores this information in a database.

1011 Two different types of tests are conducted on RPCs via the DAQ. Indeed, the performance of the  
 1012 detectors is measured periodically during dedicated test beam periods using the H4 muon beam. In  
 1013 between these test beam periods, when the beam is not available, the chambers are irradiated by the  
 1014  $^{137}\text{Cs}$  in order to accumulate deposited charge and the gamma background is measured.

1015 RPCs under test are connected through LVDS cables to V1190A Time-to-Digital Converter  
 1016 (TDC) modules manufactured by CAEN. These modules, located in the rack area outside of the  
 1017 bunker, get the logic signals sent by the chambers and save them into their buffers. Due to the  
 1018 limited size of the buffers, the collected data is regularly erased and replaced. A trigger signal is  
 1019 needed for the TDC modules to send the useful data to the DAQ computer via a V1718 CAEN USB  
 1020 communication module.

1021 In the case of performance test, the trigger signal used for data acquisition is generated by the  
 1022 coincidence of three scintillators. A first one is placed upstream outside of the bunker, a second one  
 1023 is placed downstream outside of the bunker, while a third one is placed in front of the trolley, close by  
 1024 the chambers. Every time a trigger is sent to the TDCs, i.e. every time a muon is detected, knowing  
 1025 the time delay in between the trigger and the RPC signals, signals located in the right time window  
 1026 are extracted from the buffers and saved for later analysis. Signals are taken in a time window of  
 1027 400 ns centered on the muon peak (here we could show a time spectrum). On the other hand, in the  
 1028 case of background rate measurement, the trigger signal needs to be "random" not to measure muons  
 1029 but to look at gamma background. A trigger pulse is continuously generated at a rate of 300 Hz using  
 1030 a dual timer. To integrate an as great as possible time, all signals contained within a time window of  
 1031 10us prior to the random trigger signal are extracted form the buffers and saved for further analysis  
 1032 (here another time spectrum to illustrate could be useful, maybe even place both spectrum together  
 1033 as a single Figure).

1034 The signals sent to the TDCs correspond to hit collections in the RPCs. When a particle hits  
 1035 a RPC, it induce a signal in the pickup strips of the RPC readout. If this signal is higher than the  
 1036 detection threshold, a LVDS signal is sent to the TDCs. The data is then organised into 4 branches  
 1037 keeping track of the event number, the hit multiplicity for the whole setup, and the time and channel  
 1038 profile of the hits in the TDCs.

### 1039 5.3.2 RPC current, environmental and operation parameter monitoring

1040 In order to take into account the variation of pressure and temperature between different data taking  
 1041 periods the applied voltage is corrected following the relationship :

$$1042 HV_{eff} = HV_{app} \times \left( 0.2 + 0.8 \cdot \frac{P_0}{P} \times \frac{T}{T_0} \right) \quad (5.10)$$

1042 where  $T_0$  (=293 K) and  $P_0$  (=990 mbar) are the reference values.

1043 **5.3.3 Measurement procedure**

1044 Insert a short description of the online tools (DAQ, DCS, DQM).

1045 Insert a short description of the offline tools : tracking and efficiency algorithm.

1046 Identify long term aging effects we are monitoring the rates per strip.

1047 **5.3.4 Longevity studies results**



# 6

1048

1049

## Investigation on high rate RPCs

1050 **6.1 Rate limitations and ageing of RPCs**

1051 **6.1.1 Low resistivity electrodes**

1052 **6.1.2 Low noise front-end electronics**

1053 **6.2 Construction of prototypes**

1054 **6.3 Results and discussions**



# 7

1055

1056

## Conclusions and outlooks

<sup>1057</sup> **7.1 Conclusions**

<sup>1058</sup> **7.2 Outlooks**



## References

- [1] CERN. Geneva. LHC Experiments Committee. *The CMS muon project : Technical Design Report*. Tech. rep. CERN-LHCC-97-032. CMS Collaboration, 1997.
- [2] 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.
- [3] CERN. Geneva. LHC Experiments Committee. *CMS, the Compact Muon Solenoid : technical proposal*. Tech. rep. CERN-LHCC-94-38. CMS Collaboration, 1994.
- [4] R. Santonico and R. Cardarelli. “Development of resistive plate counters”. In: *Nucl. Instr. Meth. Phys. Res.* 187 (1981), pp. 377–380.
- [5] 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.
- [6] W.W. Ash, ed. *Spark Counter With A Localized Discharge*. Vol. SLAC-R-250. 1982, pp. 127–131.
- [7] I. Crotty et al. “The non-spark mode and high rate operation of resistive parallel plate chambers”. In: *NIMA* 337 (1993), pp. 370–381.
- [8] I. Crotty et al. “Further studies of avalanche mode operation of resistive parallel plate chambers”. In: *NIMA* 346 (1994), pp. 107–113.
- [9] R. Cardarelli et al. “Avalanche and streamer mode operation of resistive plate chambers”. In: *NIMA* 382 (1996), pp. 470–474.
- [10] E. Cerron Zeballos et al. “A new type of resistive plate chamber: The multigap RPC”. In: *NIMA* 374 (1996), pp. 132–135.
- [11] M.C.S. Williams. “The development of the multigap resistive plate chamber”. In: *Nucl. Phys. B* 61 (1998), pp. 250–257.
- [12] H. Czyrkowski et al. “New developments on resistive plate chambers for high rate operation”. In: *NIMA* 419 (1998), pp. 490–496.
- [13] P. Camarri et al. “Streamer suppression with SF<sub>6</sub> in RPCs operated in avalanche mode”. In: *NIMA* 414 (1998), pp. 317–324.
- [14] E. Cerron Zeballos et al. “Effect of adding SF<sub>6</sub> to the gas mixture in a multigap resistive plate chamber”. In: *NIMA* 419 (1998), pp. 475–478.
- [15] CERN. Geneva. LHC Experiments Committee. *ATLAS muon spectrometer: Technical design report*. Tech. rep. CERN-LHCC-97-22. ATLAS Collaboration, 1997.
- [16] CERN. Geneva. LHC Experiments Committee. *ALICE Time-Of-Flight system (TOF) : Technical Design Report*. Tech. rep. CERN-LHCC-2000-012. ALICE Collaboration, 2000.
- [17] The CALICE collaboration. “First results of the CALICE SDHCAL technological prototype”. In: *JINST* 11 (2016).

- 1094 [18] PoS, ed. *Density Imaging of Volcanoes with Atmospheric Muons using GRPCs*. International  
1095 Europhysics Conference on High Energy Physics - HEP 2011. 2011.
- 1096 [19] C. Lippmann. "Detector Physics of Resistive Plate Chambers". PhD thesis. Johann Wolfgang  
1097 Goethe-Universität, 2003.
- 1098 [20] 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  
1099 of resistive plate chambers". In: *NIMA* 398 (1997), pp. 173–179.
- 1100 [21] G. Battistoni et al. "Sensitivity of streamer mode to single ionization electrons". In: *NIMA*  
1101 235 (1985), pp. 91–97.
- 1102 [22] W. Riegler. "Induced signals in resistive plate chambers". In: *NIMA* 491 (2002), pp. 258–271.
- 1103 [23] E. Cerron Zeballos et al. "A comparison of the wide gap and narrow gap resistive plate  
1104 chamber". In: *NIMA* 373 (1996), pp. 35–42.
- 1105 [24] M. Abbrescia et al. "Cosmic ray tests of double-gap resistive plate chambers for the CMS  
1106 experiment". In: *NIMA* 550 (2005), pp. 116–126.
- 1107 [25] ALICE Collaboration. "A study of the multigap RPC at the gamma irradiation facility at  
1108 CERN". In: *NIMA* 490 (2002), pp. 58–70.
- 1109 [26] B. Bonner et al. "A multigap resistive plate chamber prototype for time-of-flight for the STAR  
1110 experiment at RHIC". In: *NIMA* 478 (2002), pp. 176–179.
- 1111 [27] S. Yang et al. "Test of high time resolution MRPC with different readout modes for the  
1112 BESIII upgrade". In: *NIMA* 763 (2014), pp. 190–196.
- 1113 [28] A. Akindinov et al. "RPC with low-resistive phosphate glass electrodes as a candidate for  
1114 the CBM TOF". In: *NIMA* 572 (2007), pp. 676–681.
- 1115 [29] JINST, ed. *Development of the MRPC for the TOF system of the MultiPurpose Detector*.  
1116 RPC2016: XII Workshop on Resistive Plate Chambers and Related Detectors. 2016.
- 1117 [30] M.C.S. Williams. "Particle identification using time of flight". In: *Journal of Physics G* 39  
1118 (2012).
- 1119 [31] A. Alici et al. "Aging and rate effects of the Multigap RPC studied at the Gamma Irradiation  
1120 Facility at CERN". In: *NIMA* 579 (2007), pp. 979–988.
- 1121 [32] M. Abbrescia et al. "The simulation of resistive plate chambers in avalanche mode: charge  
1122 spectra and efficiency". In: *NIMA* 431 (1999), pp. 413–427.
- 1123 [33] M. Abbrescia et al. "Study of long-term performance of CMS RPC under irradiation at the  
1124 CERN GIF". In: *NIMA* 533 (2004), pp. 102–106.
- 1125 [34] H.C. Kim et al. "Quantitative aging study with intense irradiation tests for the CMS forward  
1126 RPCs". In: *NIMA* 602 (2009), pp. 771–774.
- 1127 [35] S. Agosteo et al. "A facility for the test of large-area muon chambers at high rates". In: *NIMA*  
1128 452 (2000), pp. 94–104.
- 1129 [36] PoS, ed. *CERN GIF ++ : A new irradiation facility to test large-area particle detectors for*  
1130 *the high-luminosity LHC program*. Vol. TIPP2014. 2014, pp. 102–109.
- 1131 [37] A. Fagot. *GIF++ DAQ v4.0*. 2017. URL: [https://github.com/afagot/GIF\\_DAQ](https://github.com/afagot/GIF_DAQ).
- 1132 [38] CAEN. *Mod. V1190-VX1190 A/B, 128/64 Ch Multihit TDC*. 14th ed. 2016.
- 1133 [39] CAEN. *Mod. V1718 VME USB Bridge*. 9th ed. 2009.
- 1134 [40] W-Ie-Ne-R. *VME 6021-23 VXI*. 5th ed. 2016.
- 1135 [41] Wikipedia. *INI file*. 2017. URL: [https://en.wikipedia.org/wiki/INI\\_file](https://en.wikipedia.org/wiki/INI_file).

- <sup>1136</sup> [42] S. Carrillo A. Fagot. *GIF++ Offline Analysis v6*. 2017. URL: [https://github.com/afagot/GIF\\_OfflineAnalysis](https://github.com/afagot/GIF_OfflineAnalysis).
- <sup>1137</sup>



# A

1138

1139

1140

## A data acquisition software for CAEN VME TDCs

1141 Certifying detectors in the perspective of HL-LHC required to develop tools for the GIF++ expe-  
1142 riment. Among them was the C++ Data Acquisition (DAQ) software that allows to make the commu-  
1143 nications in between a computer and TDC modules in order to retrieve the RPC data [37]. In this  
1144 appendix, details about this software, as of how the software was written, how it functions and how  
1145 it can be exported to another similar setup, will be given.

### 1146 A.1 GIF++ DAQ file tree

1147 GIF++ DAQ source code is fully available on github at [https://github.com/afagot/GIF\\_](https://github.com/afagot/GIF_DAQ)  
1148 DAQ. The software requires 3 non-optional dependencies:

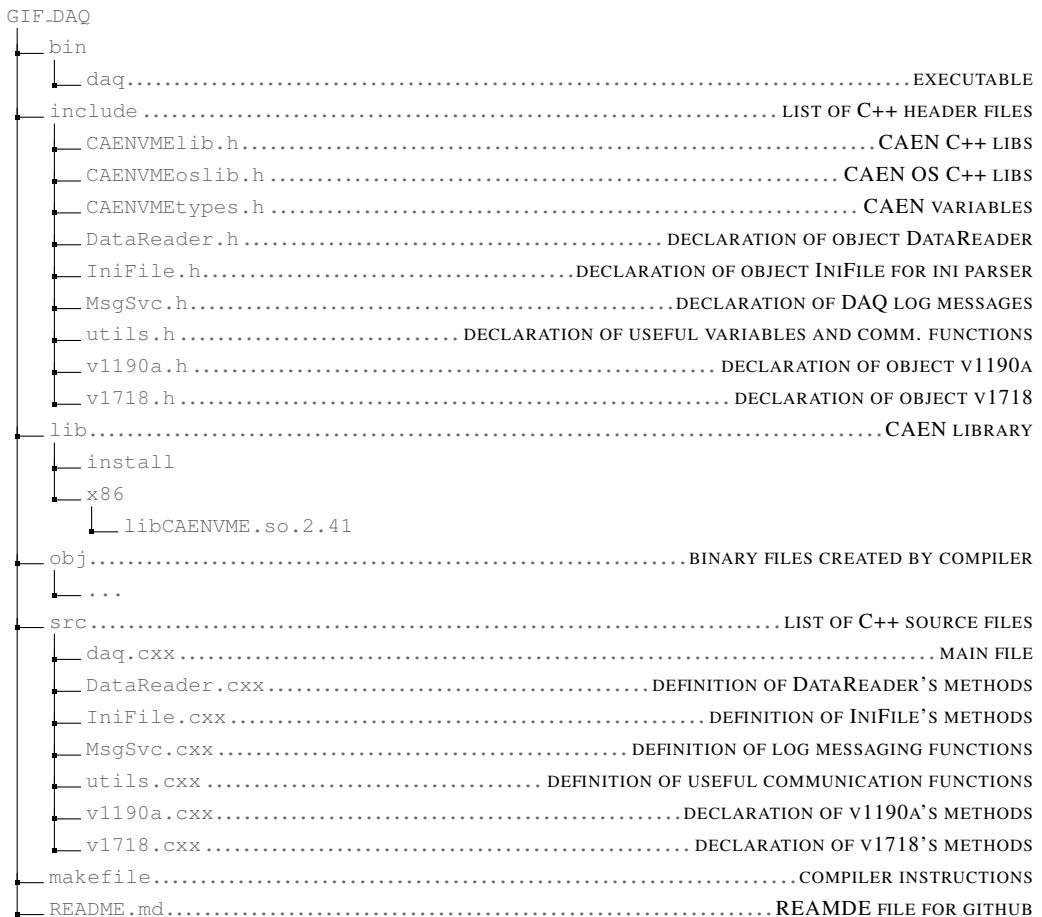
- 1149 • CAEN USB Driver, to mount the VME hardware,
- 1150 • CAEN VME Library, to communicate with the VME hardware, and
- 1151 • ROOT, to organize the collected data into a TTree.

1152 The CAEN VME library will not be packaged by distributions and will need to be installed man-  
1153 ually. To compile the GIF++ DAQ project via a terminal, from the DAQ folder use the command:

1154      make

1156 The source code tree is provided below along with comments to give an overview of the files' con-  
1157 tent. The different objects created for this project (`v1718`, `v1190a`, `IniFile` & `DataReader`) will be  
1158 described in details in the following sections.

1159



## 1160 A.2 Usage of the DAQ

1161 GIF++ DAQ, as used in GIF++, is not a standalone software. Indeed, the system being more complex,  
 1162 the DAQ only is a sub-layer of the software architecture developed to control and monitor  
 1163 the RPCs that are placed into the bunker for performance study in an irradiated environment. The top  
 1164 layer of GIF++ is a Web Detector Control System (webDCS) application. The DAQ is only called  
 1165 by the webDCS when data needs to be acquired. The webDCS operates the DAQ through command  
 1166 line. To start the DAQ, the webDCS calls:

1167  
 1168   bin/daq /path/to/the/log/file/in/the/output/data/folder

1169 where /path/to/the/log/file/in/the/output/data/folder is the only argument required. This  
 1170 log file is important for the webDCS as this file contains all the content of the communication of the  
 1171 webDCS and the different systems monitored by the webDCS. Its content is constantly displayed  
 1172 during data taking for the users to be able to follow the operations. The communication messages  
 1173 are normally sent to the webDCS log file via the functions declared in file MsgSvc.h, typically  
 1174 MSG\_INFO(string message).

1175

### 1176 A.3 Description of the readout setup

1177 The CMS RPC setup at GIF++ counts 5 V1190A Time-to-Digital Converter (TDC) manufactured  
 1178 by CAEN [38]. V1190A are VME units accepting 128 independent Multi-Hit/Multi-Event TDC  
 1179 channels whose signals are treated by 4 100 ps high performance TDC chips developed by CERN  
 1180 / ECP-MIC Division. The communication between the computer and the TDCs to transfer data is  
 1181 done via a V1718 VME master module also manufactured by CAEN and operated from a USB  
 1182 port [39]. These VME modules are all hosted into a 6U VME 6021 powered crate manufactured by  
 1183 W-Ie-Ne-R than can accommodate up to 21 VME bus cards [40]. These 3 components of the DAQ  
 1184 setup are shown in Figure A.1.

1185

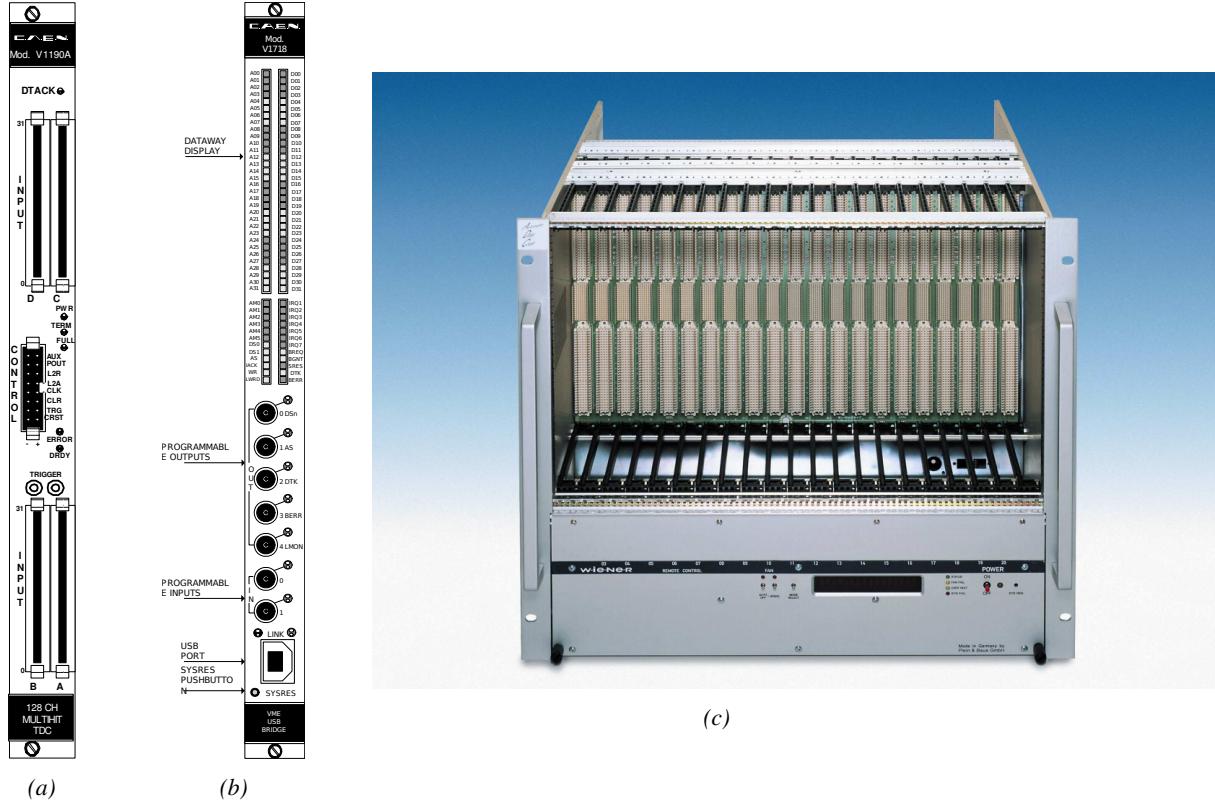


Figure A.1: (A.1a) View of the front panel of a V1190A TDC module [38]. (A.1b) View of the front panel of a V1718 Bridge module [39]. (A.1c) View of the front panel of a 6U 6021 VME crate [40].

1186

### A.4 Data read-out

1187 To efficiently perform a data readout algorithm, C++ objects to handle the VME modules (TDCs  
 1188 and VME bridge) have been created along with objects to store data and read the configuration file

1189 that comes as an input of the DAQ software.

1190

#### 1191 A.4.1 V1190A TDCs

1192 The DAQ used at GIF takes profit of the *Trigger Matching Mode* offered by V1190A modules.  
 1193 This setting is enabled through the method `v1190a::SetTrigMatching (int ntdcs)` where `ntdcs`  
 1194 is the total number of TDCs in the setup this setting needs to be enabled for (Source Code A.1). A  
 1195 trigger matching is performed in between a trigger time tag, a trigger signal sent into the TRIGGER  
 1196 input of the TDC visible on Figure A.1a, and the channel time measurements, signals recorded from  
 1197 the detectors under test in our case. Control over this data acquisition mode, explained through  
 1198 Figure A.2, is offered via 4 programmable parameters:

- 1199 • **match window:** the matching between a trigger and a hit is done within a programmable time  
 1200 window. This is set via the method

1201 `void v1190a::SetTrigWindowWidth(Uint windowHeight, int ntdcs)`

- 1202 • **window offset:** temporal distance between the trigger tag and the start of the trigger matching  
 1203 window. This is set via the method

1204 `void v1190a::SetTrigWindowWidth(Uint windowHeight, int ntdcs)`

- 1205 • **extra search margin:** an extended time window is used to ensure that all matching hits are  
 1206 found. This is set via the method

1207 `void v1190a::SetTrigSearchMargin(Uint searchMargin, int ntdcs)`

- 1208 • **reject margin:** older hits are automatically rejected to prevent buffer overflows and to speed  
 1209 up the search time. This is set via the method

1210 `void v1190a::SetTrigRejectionMargin(Uint rejectMargin, int ntdcs)`

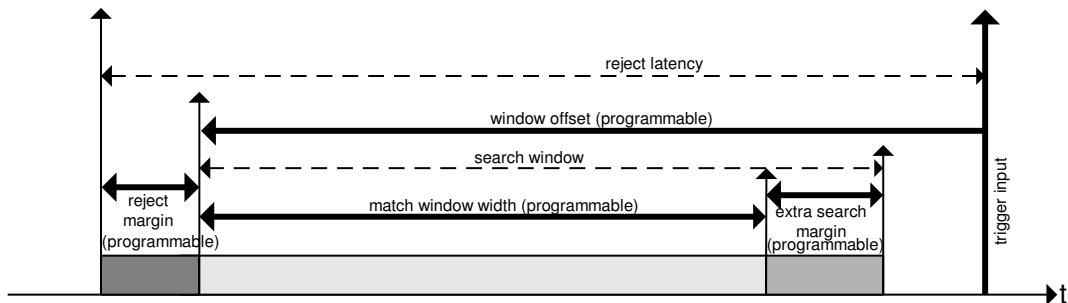


Figure A.2: Module V1190A Trigger Matching Mode timing diagram [38].

1211 Each of these 4 parameters are given in number of clocks, 1 clock being 25 ns long. It is easy to  
 1212 understand at this level that there are 3 possible functioning settings:

- 1213 • **1:** the match window is entirely contained after the trigger signal,

- 1214 • **2:** the match window overlaps the trigger signal, or

- 1215 • **3:** the match window is entirely contained before the trigger signal as displayed on Figure A.2.

1216 In both the first and second cases, the sum of the window width and of the offset can be set to  
1217 a maximum of 40 clocks, which corresponds to 1  $\mu$ s. Evidently, the offset can be negative, allowing  
1218 for a longer match window, with the constraint of having the window ending at most 1  $\mu$ s after the  
1219 trigger signal. In the third case, the maximum negative offset allowed is of 2048 clocks (12 bit) cor-  
1220 responding to 51.2  $\mu$ s, the match window being strictly smaller than the offset. In the case of GIF++,  
1221 the choice has been made to use this last setting by delaying the trigger signal. During the studies  
1222 performed in GIF++, both the efficiency of the RPCs, probed using a muon beam, and the noise or  
1223 gamma background rate are monitored. The extra search and reject margins are left unused.  
1224 To probe the efficiency of RPC detectors, the trigger time tag is provided by the coïncidence of  
1225 scintillators when a bunch of muons passes through GIF++ area is used to trigger the data acquisi-  
1226 tion. For this measurement, it is useful to reduce the match window width only to contain the muon  
1227 information. Indeed, the delay in between a trigger signal and the detection of the corresponding  
1228 muon in the RPC being very contant (typically a few tens of ns due to jitter and cable length), the  
1229 muon signals are very localised in time. Thus, due to a delay of approximalety 325 ns in between  
1230 the muons and the trigger, the settings where chosen to have a window width of 24 clocks (600 ns)  
1231 centered on the muon peak thanks to a negative offset of 29 clocks (725 ns).  
1232 On the otherhand, monitoring the rates don't require for the DAQ to look at a specific time window.  
1233 It is important to integrate enough time to have a robust measurement of the rate as the number of  
1234 hits per time unit. The triggerring signal is provided by a pulse generator at a frequency of 300 Hz  
1235 to ensure that the data taking occurs in a random way, with respect to beam physics, to probe only  
1236 the irradiation spectrum on the detectors. The match window is set to 400 clocks (10  $\mu$ s) and the  
1237 negative offset to 401 clocks as it needs to exceed the value of the match window.

```

1238
class v1190a
{
    private :
        long             Handle;
        vector<Data32>   Address;
        CVDataWidth      DataWidth;
        CVAddressModifier 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);
};

1239

```

1240       *Source Code A.1: Description of C++ object v1190a.*

1241       The v1190a object, defined in the DAQ software as in Source Code A.1, offers the possibility to  
 1242       concatenate all TDCs in the readout setup into a single object containing a list of hardware addresses  
 1243       (addresses to access the TDCs' buffer through the VME crate) and each constructor and method acts  
 1244       on the list of TDCs.  
 1245

#### 1246      A.4.2 DataReader

1247      Enabled thanks to v1190a::SetBlockTransferMode(Data16 mode, int ntdcs), the data transfer  
 1248      is done via Block Transfer (BLT). Using BLT allows to tranfer a fixed number of events called a  
 1249      *block*. This is used together with an Almost Full Level (AFL) of the TDCs' output buffers, defined

1250 through `v1190a::SetIRQ(Data32 level, Data32 count, int ntdcs)`. This AFL gives the maximum amount of 32735 words (16 bits, corresponding to the depth of a TDC output buffer) that can  
 1251 written in a buffer before an Interrupt Request (IRQ) is generated and seen by the VME Bridge,  
 1252 stopping the data acquisition to transfer the content of each TDC buffers before resuming. For each  
 1253 trigger, 6 words or more are written into the TDC buffer:

- 1255     • a **global header** providing information of the event number since the beginning of the data  
       acquisition,
- 1257     • a **TDC header**,
- 1258     • the **TDC data** (*if any*), 1 for each hit recorded during the event, providing the channel and the  
       time stamp associated to the hit,
- 1260     • a **TDC error** providing error flags,
- 1261     • a **TDC trailer**,
- 1262     • a **global trigger time tag** that provides the absolute trigger time relatively to the last reset,  
       and
- 1264     • a **global trailer** providing the total word count in the event.

1265     As previously described in Section 4.3.3, CMS RPC FEEs provide us with 100 ns long LVDS  
 1266 output signals that are injected into the TDCs' input. Any avalanche signal that gives a signal above  
 1267 the FEEs threshold is thus recorded by the TDCs as a hit within the match window. Each hit is  
 1268 assigned to a specific TDC channel with a time stamp, with a precision of 100 ps. The reference  
 1269 time,  $t_0 = 0$ , is provided by the beginning of the match window. Thus for each trigger, coming from  
 1270 a scintillator coïncidence or the pulse generator, a list of hits is stored into the TDCs' buffers and  
 1271 will then be transferred into a ROOT Tree.

1272     When the BLT is used, it is easy to understand that the maximum number of words that have  
 1273 been set as ALF will not be a finite number of events or, at least, the number of events that would  
 1274 be recorded into the TDC buffers will not be a multiple of the block size. In the last BLT cycle to  
 1275 transfer data, the number of events to transfer will most probably be lower than the block size. In that  
 1276 case, the TDC can add fillers at the end of the block but this option requires to send more data to the  
 1277 computer and is thus a little slower. Another solution is to finish the transfer after the last event by  
 1278 sending a bus error that states that the BLT reached the last event in the pile. This method has been  
 1279 chosen in GIF++.

1281     Due to irradiation, an event in GIF++ can count up to 300 words per TDC. A limit of 4096 words  
 1282 (12 bits) has been set to generate IRQ which represent from 14 to almost 700 events depending on  
 1283 the average of hits collected per event. Then the block size has been set to 100 events with enabled  
 1284 bus errors. When an AFL is reached for one of the TDCs, the VME bridge stops the acquisition by  
 1285 sending a BUSY signal.

1287

1288     The data is then transferred one TDC at a time into a structure called `RAWData` (Source Code A.2).

```
1289
1290 struct RAWData{
1291     vector<int>           *EventList;
1292     vector<int>           *NHitsList;
1293     vector<int>           *QFlagList;
1294     vector<vector<int>>   *Channellist;
1295     vector<vector<float>>  *TimeStampList;
1296 };
1297 
```

1291                 *Source Code A.2: Description of data holding C++ structure `RAWData`.*

1292     In order to organize the data transfer and the data storage, an object called `DataReader` was  
1293     created (Source Code A.3). On one hand, it has `v1718` and `v1190a` objects as private members for  
1294     communication purposes, such as VME modules settings via the configuration file `*iniFile` or data  
1295     read-out through `v1190a::Read()` and on the other hand, it contains the structure `RAWData` that allows  
1296     to organise the data in vectors reproducing the tree structure of a ROOT file.

```
1297
1298 class DataReader
1299 {
1300     private:
1301     bool      StopFlag;
1302     IniFile  *iniFile;
1303     Data32    MaxTriggers;
1304     v1718    *VME;
1305     int       nTDCs;
1306     v1190a   *TDCs;
1307     RAWData  TDCData;
1308
1309     public:
1310     DataReader();
1311     virtual ~DataReader();
1312     void      SetIniFile(string inifilename);
1313     void      SetMaxTriggers();
1314     Data32    GetMaxTriggers();
1315     void      SetVME();
1316     void      SetTDC();
1317     int       GetQFlag(Uint it);
1318     void      Init(string inifilename);
1319     void      FlushBuffer();
1320     void      Update();
1321     string   GetFileName();
1322     void      WriteRunRegistry(string filename);
1323     void      Run();
1324 };
1325 
```

1299                 *Source Code A.3: Description of C++ object `DataReader`.*

1300     Each event is transferred from `TDCData` and saved into branches of a ROOT `TTree` as 3 integers  
1301     that represent the event ID (`EventCount`), the number of hits read from the TDCs (`nHits`), and the  
1302     quality flag that provides information for any problem in the data transfer (`qflag`), and 2 lists of  
1303     `nHits` elements containing the fired TDC channels (`TDCCh`) and their respective time stamps (`TDCTS`),  
1304     as presented in Source Code A.4. The ROOT file file is named using information contained into  
1305     the configuration file, presented in section A.5.2. The needed information is extracted using method  
1306     `DataReader::GetFileName()` and allow to build the output filename format `ScanXXXXXX_HVX_DAQ.root`

1307 where ScanXXXXXX is a 6 digit number representing the scan number into GIFT++ database and HVX  
 1308 the HV step within the scan that can be more than a single digit. An example of ROOT data file is  
 1309 provided with Figure A.3.

```
1310
RAWData TDCData;
TFile *outputFile = new TFile(outputFileName.c_str(),"recreate");
TTree *RAWDataTree = new TTree("RAWData","RAWData");

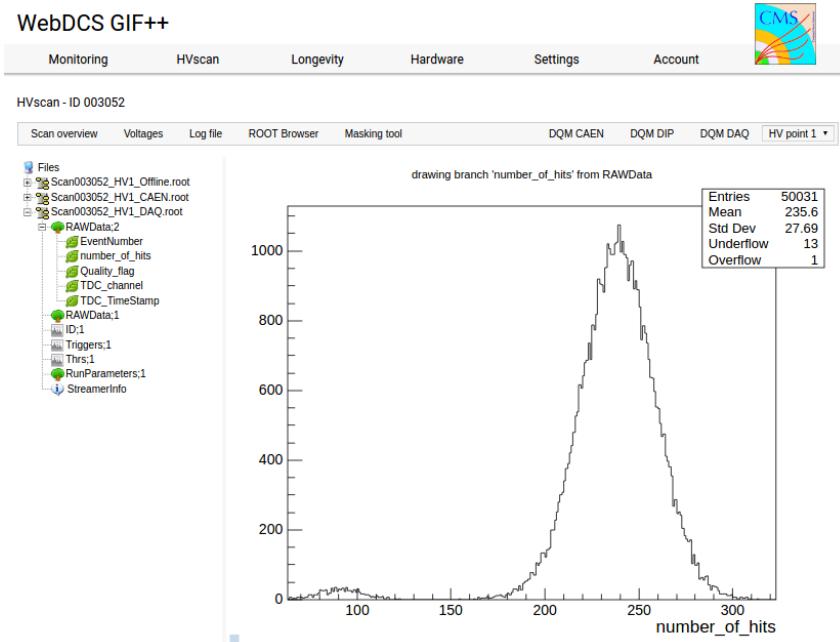
int EventCount = -9;
int nHits = -8;
int qflag = -7;
vector<int> TDCCh;
vector<float> TDCTS;

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);

1311
//...
//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();
}
```

1312 *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.*

#### 1313    A.4.3 Data quality flag

1314    Among the parameters that are recorded for each event, the quality flag, defined in Source Code A.5,  
 1315    is determined on the fly by checking the data recorded by every single TDC. From method `v1190a::Read()`,  
 1316    it can be understood that the content of each TDC buffer is readout one TDC at a time. Entries are  
 1317    created in the data list for the first TDC and then, when the second buffer is readout, events corre-  
 1318    sponding to entries that have already been created to store data for the previous TDC are added to  
 1319    the existing list element. On the contrary, when an event entry has not been yet created in the data  
 1320    list, a new entry is created.

```
1321
 1322 typedef enum _QualityFlag {
 1323     GOOD      = 1,
 1324     CORRUPTED = 0
 1325 } QualityFlag;
```

1323    *Source Code A.5: Definition of the quality flag `enum`.*

1324    It is possible that each TDC buffer contains a different number of events. In cases where the first  
 1325    element in the buffer list is an event for corresponds to a new entry, the difference in between the  
 1326    entry from the buffer and the last entry in the data list is recorded and checked. If it is greater than 1,  
 1327    what should never be the case, the quality flag is set to CORRUPTED for this TDC and an empty entry  
 1328    is created in the place of the missing ones. Missing entries are believe to be the result of a bad hold

1329 on the TDC buffers at the moment of the readout. Indeed, the software hold is effective only on 1  
 1330 TDC at a time and no solution as been found yet to completely block the writting in the buffers when  
 1331 an IRQ is received.

1332 At the end of each BLT cycle, the ID of the last entry stored for each TDC buffer is not recorded.  
 1333 When starting the next cycle, if the first entry in the pile corresponds to an event already existing  
 1334 in the list, the readout will start from this list element and will not be able to check the difference  
 1335 in between this entry's ID and the one of the last entry that was recorded for this TDC buffer in  
 1336 the previous cycle. In the case events were missing, the flag stays at its initial value of 0, which is  
 1337 similar to CORRUPTED and it is assumed that then this TDC will not contribute to `number_of_hits`,  
 1338 `TDC_channel` or `TDC_TimeStamp`.

1339 Finally, since there will be 1 `RAWData` entry per TDC for each event (meaning `nTDCs` entries,  
 1340 referring to `DataReader` private attribute), the individual flags of each TDC will be added together.  
 1341 The final format is an integer composed `nTDCs` digits where each digit is the flag of a specific TDC.  
 1342 This is constructed using powers of 10 like follows:

```
1343 TDC 0: QFlag = 100 × _QualityFlag
1344 TDC 1: QFlag = 101 × _QualityFlag
1345 ...
1346 TDC N: QFlag = 10N × _QualityFlag
```

1347 and the final flag to be with N digits:

```
1348 QFlag = n....3210
```

1349 each digit being 1 or 0. Below is given an example with a 4 TDCs setup.

1350 If all TDCs were good : `QFlag = 1111`,

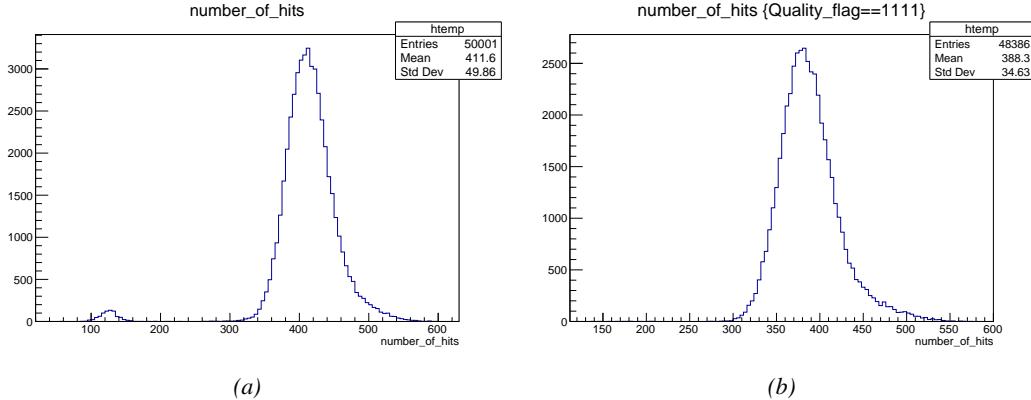
1351 but if TDC 2 was corrupted : `QFlag = 1011`.

1352 When data taking is over and the data contained in the dynamical `RAWData` structure is transferred  
 1353 to the ROOT file, all the 0s are changed into 2s by calling the method `DataReader::GetQFlag()`.  
 1354 This will help translating the flag without knowing the number of TDCs beforehand. Indeed, a flag  
 1355 111 could be due to a 3 TDC setup with 3 good individual TDC flags or to a more than 3 TDC setup  
 1356 with TDCs those ID is greater than 2 being CORRUPTED, thus giving a 0.

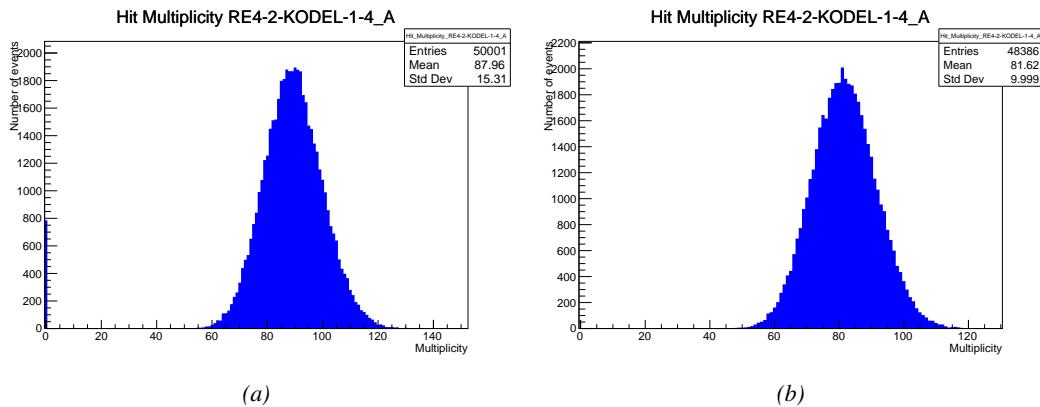
1357 The quality flag has been introduced quite late, in October 2017 only, to the list of GIFT++ DAQ  
 1358 parameters to be recorded into the output ROOT file. Before this addition, the missing data, corrupting  
 1359 the quality for the offline analysis, was contributing to artificially fill data with lower multiplicity.  
 1360 Looking at `TBranch number_of_hits` provides an information about the data of the full GIFT++  
 1361 setup. When a TDC is not able to transfer data for a specific event, the effect is a reduction of the  
 1362 total number of hits recorded in the full setup, this is what can be seen from Figure A.4. After offline  
 1363 reconstruction detector by detector, the effect of missing events can be seen in the artificially filled  
 1364 bin at multiplicity 0 shown in Figure A.5. Nonetheless, for data with high irradiation levels, as it is  
 1365 the case for Figure A.5a, discarding the fake multiplicity 0 data can be done easily during the offline  
 1366 analysis. At lower radiation, the missing events contribution becomes more problematic as the mul-  
 1367 tiplicity distribution overlaps the multiplicity 0 and that in the same time the proportion of missing

1368 events decreases. Attempts to fit the distribution with a Poisson or skew distribution function were  
 1369 not conclusive and this very problem has been at the origin of the quality flag that allows to give a  
 1370 non ambiguous information about each event quality.

1371



*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 then 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.*

1372

## A.5 Communications

1373 To ensure data readout and dialog in between the machine and the TDCs or in between the webDCS  
 1374 and the DAQ, different communication solutions were used. First of all, it is important to have a

1375 module to allow the communication in between the TDCs and the computer from which the DAQ  
1376 operates. When this communication is effective, shifters using the webDCS to control data taking  
1377 can thus send instructions to the DAQ.

1378

### **1379 A.5.1 V1718 USB Bridge**

1380 In the previous section, the data transfer as been discussed. The importance of the `v1718` object  
1381 (Source Code A.6), used as private member of `DataReader`, was not explicated. VME master  
1382 modules are used for communication purposes as they host the USB port that connects the pow-  
1383 ered crate buffer to the computer were the DAQ is installed. From the source code point of view,  
1384 this object is used to control the communication status, by reading the returned error codes with  
1385 `v1718::CheckStatus()`, or to check for IRQs coming from the TDCs through `v1718::CheckIRQ()`.  
1386 Finally, to ensure that triggers are blocked at the hardware level, a NIM pulse is sent out of one of the  
1387 5 programmable outputs (`v1718::SendBUSY()`) to the VETO of the coïndidence module where the  
1388 trigger signals originate from. As long as this signal is ON, no trigger can reach the TDCs anymore.

```
1389 class v1718{
    private:
        int Handle;
        Data32 Data;           // Data
        CVIRQLevels Level;     // Interrupt level
        CVAddressModifier AM;   // Addressing Mode
        CVDataWidth DataSize;   // Data Format
        Data32 BaseAddress;    // Base Address

    public:
        v1718(IniFile *inifile);
        ~v1718();
        long GetHandle(void) const;
        int SetData(Data16 data);
        int GetData(void);
        int SetLevel(CVIRQLevels level);
        int GetLevel(void);
        int SetAM(CVAddressModifier am);
        CVAddressModifier GetAM(void);
        int SetDatasize(CVDataWidth datasize);
        CVDataWidth GetDataSize(void);
        int SetBaseAddress(Data16 baseaddress);
        Data16 GetBaseAddress(void);
        void CheckStatus(CVErrorCodes status) const;
        bool CheckIRQ();
        void SetPulsers();
        void SendBUSY(BusyLevel level);
};

1390
```

1391 Source Code A.6: Description of C++ object v1718.

## 1392 A.5.2 Configuration file

1393 The DAQ software takes as input a configuration file written using INI standard [41]. This file is  
1394 partly filled with the information provided by the shifters when starting data acquisition using the  
1395 webDCS, as shown by Figure A.6. This information is written in section [**General**] and will later

1396 be stored in the ROOT file that contains the DAQ data as can be seen from Figure A.3. Indeed,  
 1397 another `TTree` called `RunParameters` as well as the 2 histograms `ID`, containing the scan number,  
 1398 start and stop time stamps, and `Triggers`, containing the number of triggers requested by the shifter,  
 1399 are available in the data files. Moreover, `ScanID` and `HV` are then used to construct the file name  
 1400 thanks to the method `DataReader::GetFileName()`.

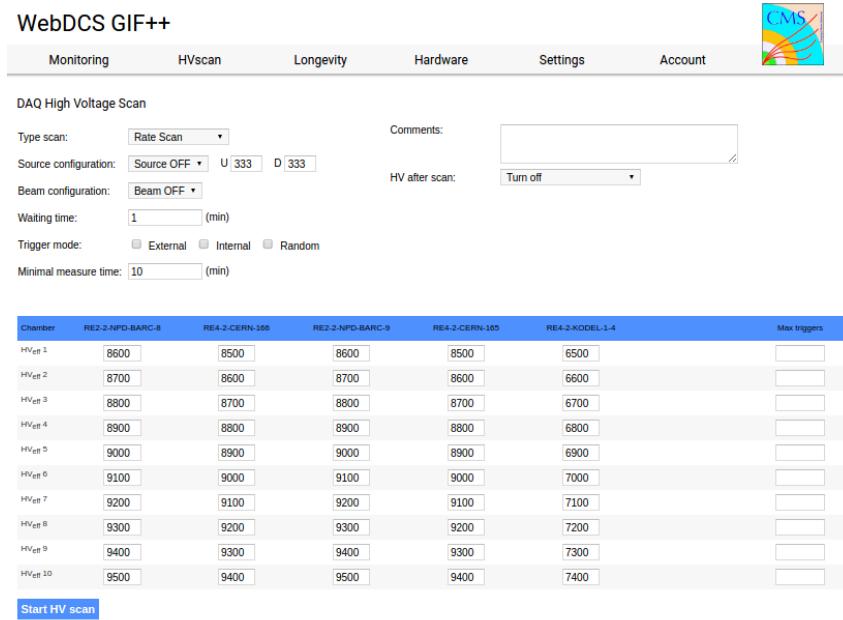


Figure A.6: WebDCS DAQ scan page. On this page, shifters need to choose the type of scan (Rate, Efficiency or Noise Reference scan), the gamma source configuration at the moment of data taking, the beam configuration, and the trigger mode. These information will be stored in the DAQ ROOT output. Are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

1401 The rest of the information is written beforehand in the configuration file template, as explicated  
 1402 in Source Code A.7, and contains the hardware addresses to the different VME modules in the  
 1403 setup as well as settings for the TDCs. As the TDC settings available in the configuration file are not  
 1404 supposed to be modified, an improvement would be to remove them from the configuration file and  
 1405 to hardcode them inside of the DAQ code itself or to place them into a different INI file that would  
 1406 host only the TDC settings to lower the probability for a bad manipulation of the configuration file  
 1407 that can be modified from one of webDCS' menus.

1408

```

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

```

1409

*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.*

1411     In order to retrieve the information of the configuration file, the object `IniFile` has been developed  
 1412     to provide an INI parser, presented in Source Code A.8. It contains private methods returning a  
 1413     boolean to check the type of line written in the file, whether a comment, a group header or a key line  
 1414     (`IniFile::CheckIfComment()`, `IniFile::CheckIfGroup()` and `IniFile::CheckIfToken()`). The  
 1415     key may sometimes be referred to as *token* in the source code. Moreover, the private element  
 1416     `FileData` is a map of `const string` to `string` that allows to store the data contained inside the  
 1417     configuration file via the public method `IniFile::GetFileData()` following the formatting (see  
 1418     method `IniFile::Read()`):

```
1419
 1420     string group, token, value;
 1421     // Get the field values for the 3 strings.
 1422     // Then concatenate group and token together as a single string
 1423     // with a dot separation.
 1424     token = group + "." + token;
 1425     FileData[token] = value;
```

1421     More methods have been written to translate the different keys into the right variable format  
 1422     when used by the DAQ. For example, to get a `float` value out of the configuration file data, knowing  
 1423     the group and the key needed, the method `IniFile::floatType()` can be used. It takes 3 arguments  
 1424     being the group name and key name (both `string`), and a default `float` value used as exception in  
 1425     the case the expected combination of group and key cannot be found in the configuration file. This  
 1426     default value is then used and the DAQ continues on working after sending an alert in the log file for  
 1427     further debugging.

```

1428 typedef map< const string, string > IniFileData;
1429
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();
    };

```

1430       *Source Code A.8: Description of C++ object `IniFile` used as a parser for INI file format.*

### 1431     A.5.3 WebDCS/DAQ intercommunication

1432     When shifters send instructions to the DAQ via the configuration file, it is the webDCS itself that  
1433     gives the start command to the DAQ and then the 2 softwares use inter-process communication  
1434     through file to synchronise themselves. This communication file is represented by the variable **const**  
1435     string \_\_runstatuspath.

1436     On one side, the webDCS sends commands or status that are readout by the DAQ:

- 1437       • INIT, status sent when launching a scan and read via function `CtrlRunStatus(...)`,
- 1438       • START, command to start data taking and read via function `CheckSTART()`,
- 1439       • STOP, command to stop data taking at the end of the scan and read via function `CheckSTOP()`,  
1440        and
- 1441       • KILL, command to kill data taking sent by user and read via function `CheckKILL()`

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

- 1443     ● `DAQ_RDY`, sent with `SendDAQReady()` to signify that the DAQ is ready to receive commands  
1444       from the webDCS,
- 1445     ● `RUNNING`, sent with `SendDAQRunning()` to signify that the DAQ is taking data,
- 1446     ● `DAQ_ERR`, sent with `SendDAQError()` to signify that the DAQ didn't receive the expected com-  
1447       mand from the webDCS or that the launch command didn't have the right number of argu-  
1448       ments,
- 1449     ● `RD_ERR`, sent when the DAQ wasn't able to read the communication file, and
- 1450     ● `WR_ERR`, sent when the DAQ wasn't able to write into the communication file.

#### 1451     **A.5.4 Example of inter-process communication cycle**

1452     Under normal conditions, the webDCS and the DAQ processes exchange commands and status via  
1453       the file hosted at the address `__runstatuspath`, as explained in subsection A.5.3. An example of  
1454       cycle is given in Table A.1. In this example, the steps 3 to 5 are repeated as long as the webDCS tells  
1455       the DAQ to take data. A data taking cycle is the equivalent as what is called a *Scan* in GIFT++ jargon,  
1456       referring to a set a runs with several HV steps. Each repetition of steps 3 to 5 is then equivalent to a  
1457       single *Run*.

1458     At any moment during the data taking, for any reason, the shifter can decide that the data taking  
1459       needs to be stopped before it reached the end of the scheduled cycle. Thus at any moment on the  
1460       cycle, the content of the inter-process communication file will be changed to `KILL` and the DAQ will  
1461       shut down right away. The DAQ checks for `KILL` signals every 5s after the TDCs configuration is  
1462       over. So far, the function `CheckKILL()` has been used only inside of the data taking loop of method  
1463       `DataReader::Run()` and thus, if the shifter decides to KILL the data taking during the TDC con-  
1464       figuration phase or the HV ramping in between 2 HV steps, the DAQ will not be stopped smoothly  
1465       and a *force kill* command will be sent to stop the DAQ process that is still awake on the computer.  
1466       Improvements can be brought on this part of the software to make sure that the DAQ can safely  
1467       shutdown at any moment.

## 1470     **A.6 Software export**

1471     In section A.2 was discussed the fact that the DAQ as written in its last version is not a standalone  
1472       software. It is possible to make it a standalone program that could be adapted to any VME setup  
1473       using V1190A and V1718 modules by creating a GUI for the software or by printing the log mes-  
1474       sages that are normally printed in the webDCS through the log file, directly into the terminal. This  
1475       method was used by the DAQ up to version 3.0 moment where the webDCS was completed. Also, it  
1476       is possible to check branches of DAQ v2.X to have example of communication through a terminal.

1477     DAQ v2.X is nonetheless limited in it's possibilities and requires a lot of offline manual interven-  
1478       tions from the users. Indeed, there is no communication of the software with the detectors' power  
1479       supply system that would allow for a user a predefine a list of voltages to operate the detectors at

step	actions of webDCS	status of DAQ	__runstatuspath
1	launch DAQ 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.

1481 and loop over to take data without any further manual intervention. In v2.X, the data is taken for a  
1482 single detector setting and at the end of each run, the softwares asks the user if he intends on taking  
1483 more runs. If so, the software invites the user to set the operating voltages accordingly to what is  
1484 necessary and to manual update the configuration file in consequence. This working mode can be a  
1485 very first approach before an evolution and has been successfully used by colleagues from different  
1486 collaborations.

1487  
1488 For a more robust operation, it is recommended to develop a GUI or a web application to inter-  
1489 face the DAQ. Moreover, to limit the amount of manual interventions, and thus the probability to  
1490 make mistakes, it is also recommended to add an extra feature into the DAQ by installing the HV  
1491 Wrapper library provided by CAEN of which an example of use in a similar DAQ software devel-  
1492 opped by a master student of UGent, and called TinyDAQ, is provided on UGent's github. Then, this  
1493 HV Wrapper will help you communicating with and give instructions to a CAEN HV powered crate  
1494 and can be added into the DAQ at the same level where the communication with the user was made  
1495 in DAQ v2.X. In case you are using another kind of power system for your detectors, it is stringly  
1496 adviced to use HV modules or crates that can be remotely controled via a using C++ libraries.

1497

# B

1498

1499

## Details on the offline analysis package

1500 The data collected in GIF++ thanks to the DAQ described in Appendix A is difficult to interpret by  
1501 a human user that doesn't have a clear idea of the raw data architecture of the ROOT data files. In  
1502 order to render the data human readable, a C++ offline analysis tool was designed to provide users  
1503 with detector by detector histograms that give a clear overview of the parameters monitored during  
1504 the data acquisition [42]. In this appendix, details about this software in the context of GIF++, as of  
1505 how the software was written and how it functions will be given.

### 1506 B.1 GIF++ Offline Analysis file tree

1507 GIF++ Offline Analysis source code is fully available on github at [https://github.com/aafagot/GIF\\_OfflineAnalysis](https://github.com/aafagot/GIF_OfflineAnalysis). The software requires ROOT as non-optionnal dependency  
1508 as it takes ROOT files in input and write an output ROOT file containing histograms. To compile the  
1509 GIF++ Offline Analysis project is compiled with cmake. To compile, first a build/ directory must  
1510 be created to compile from there:

```
1512 mkdir build  
1513 cd build  
1514 cmake ..  
1515 make  
1516 make install
```

1514 To clean the directory and create a new build directory, the bash script cleandir.sh can be used:

```
1515  
1516 ./cleandir.sh
```

1517 The source code tree is provided below along with comments to give an overview of the files' con-  
1518 tent. The different objects created for this project (`Infrastructure`, `Trolley`, `RPC`, `Mapping`, `RPCHit`,  
1519 `RPCCluster` and `Inifile`) will be described in details in the following sections.

1520

```

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..... DEFINITION OF OFFLINE LOG MESSAGES
        │   ├── OfflineAnalysis.cc..... DEFINITION OF DATA ANALYSIS MACRO
        │   ├── RPCTracker.cc..... DEFINITION OF OBJECT RPC
        │   ├── RPCHit.cc..... DEFINITION 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

```

1521

## B.2 Usage of the Offline Analysis

1522

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:

1524

1525

```
Scan00XXXX_HVY
```

1526

1527

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

1528 of the GIF++ webDCS. Usually, the offline analysis tool is automatically called by the webDCS at  
 1529 the end of data taking or by a user from the webDCS panel if an update of the tool was brought.  
 1530 Nonetheless, an expert can locally launch the analysis for tests on the GIF++ computer, or a user can  
 1531 get the code on its local machine from github and download data from the webDCS for its own anal-  
 1532 ysis. To launch the code, the following command can be used from the `GIF_OfflineAnalysis` folder:

1533  
 1534     `bin/offlineanalysis /path/to/Scan00XXXX_HVY`

1535 where, `/path/to/Scan00XXXX_HVY` refers to the local data files. Then, the offline tool will by itself  
 1536 take care of finding all available ROOT data files present in the folder, as listed below:

- 1537
  - 1538         ● `Scan00XXXX_HVY_DAQ.root` containing the TDC data as described in Appendix ?? (events, hit  
     and timestamp lists), and
  - 1539         ● `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.

## 1542     **B.2.1 Output of the offline tool**

### 1543     **B.2.1.1 ROOT file**

1544 The analysis gives in output ROOT datafiles that are saved into the data folder and called using the  
 1545 naming convention `Scan00XXXX_HVY_Offline.root`. Inside those, a list of `TH1` histograms can be  
 1546 found. Its size will vary as a function of the number of detectors in the setup as each set of histograms  
 1547 is produced detector by detector. For each partition of each chamber, can be found:

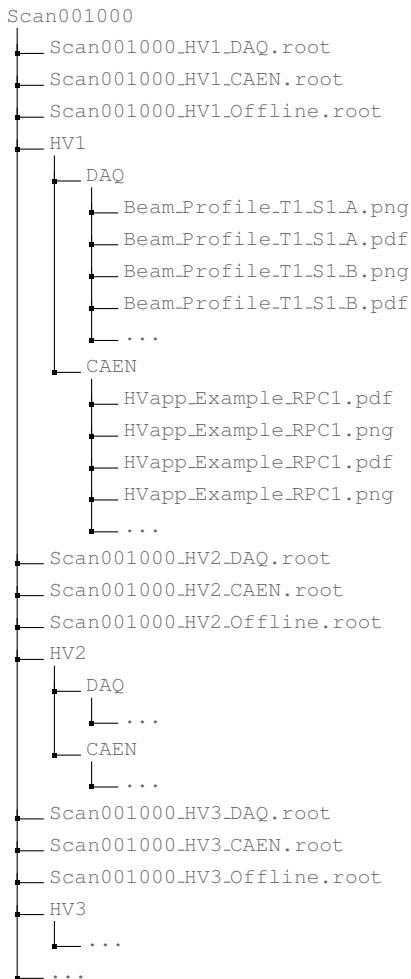
- 1548
  - 1549         ● `Time_Profile_Tt_Sc_p` shows the time profile of all recorded events (number of events per  
     time bin),
  - 1550         ● `Hit_Profile_Tt_Sc_p` shows the hit profile of all recorded events (number of events per chan-  
     nel),
  - 1552         ● `Hit_Multiplicity_Tt_Sc_p` shows the hit multiplicity (number of hits per event) of all recorded  
     events (number of occurrences per multiplicity bin),
  - 1554         ● `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,
  - 1558         ● `Strip_Activity_Tt_Sc_p` shows noise/gamma activity for each strip (normalised version of  
     previous histogram - strip activity = strip rate / average partition rate),
  - 1560         ● `Strip_Homogeneity_Tt_Sc_p` shows the *homogeneity* of a given partition ( $\text{homogeneity} = \exp(-\text{strip rates standard deviation}(\text{strip rates in partition}/\text{average partition rate}))$ ),
  - 1562         ● `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,

- 1566     ● `mask_Strip_Activity_Tt_Sc_p` shows noise/gamma activity per unit area for each masked  
1567       strip with respect to the average rate of active strips,
- 1568     ● `NoiseCSize_H_Tt_Sc_p` shows noise/gamma cluster size, a cluster being constructed out of  
1569       adjacent strips giving a signal at the *same time* (hits within a time window of 25 ns),
- 1570     ● `NoiseCMult_H_Tt_Sc_p` shows noise/gamma cluster multiplicity (number of reconstructed  
1571       clusters per event),
- 1572     ● `Chip_Mean_Noise_Tt_Sc_p` shows the same information than `Strip_Mean_Noise_Tt_Scp` us-  
1573       ing a different binning (1 chip corresponds to 8 strips),
- 1574     ● `Chip_Activity_Tt_Sc_p` shows the same information than `Strip_Activity_Tt_Scp` using  
1575       chip binning,
- 1576     ● `Chip_Homogeneity_Tt_Sc_p` shows the homogeneity of a given partition using chip binning,
- 1577     ● `Beam_Profile_Tt_Sc_p` shows the estimated beam profile when taking efficiency scan. This  
1578       is obtained by filtering `Time_Profile_Tt_Sc_p` to only consider the muon peak where the  
1579       noise/gamma background has been subtracted. The resulting hit profile corresponds to the  
1580       beam profile on the detector channels,
- 1581     ● `L0_Efficiency_Tt_Sc_p` shows the level 0 efficiency that was estimated **without** muon track-  
1582       ing,
- 1583     ● `MuonCSize_H_Tt_Sc_p` shows the level 0 muon cluster size that was estimated **without** muon  
1584       tracking, and
- 1585     ● `MuonCMult_H_Tt_Sc_p` shows the level 0 muon cluster multiplicity that was estimated **without**  
1586       muon tracking.

1587     In the histogram labels,  $t$  stands for the trolley number (1 or 3),  $c$  for the chamber slot label in  
1588       trolley  $t$  and  $p$  for the partition label (A, B, C or D depending on the chamber layout) as explained  
1589       in Chapter 5.3.

1590     In the context of GIF++, an extra script called by the webDCS is called to extract the histograms  
1591       from the ROOT files. The histograms are then stored in PNG and PDF formats into the correspond-  
1592       ing folder (a single folder per HV step, so per ROOT file). the goal is to then display the histograms  
1593       on the Data Quality Monitoring (DQM) page of the webDCS in order for the users to control the  
1594       quality of the data taking at the end of data taking. An example of histogram organisation is given  
1595       below:

1596



**1598      Here can put some screens from the webDCS to show the DQM and the plots available to users.**

**1599**

### **1600      B.2.1.2 CSV files**

**1601      Moreover, up to 3 CSV files can be created depending on which ones of the 3 input files were in the**

**1602      data folder:**

**1603      • Offline-Corrupted.csv , is used to keep track of the amount of data that was corrupted and**

**1604      removed from old data format files that don't contain any data quality flag.**

**1605      • Offline-Current.csv , contains the summary of the currents and voltages applied on each**

**1606      RPC HV channel.**

**1607      • Offline-L0-EffC1.csv , is used to write the efficiencies, cluster size and cluster multiplicity**

**1608      of efficiency runs. Note that L0 refers here to *Level 0* and means that the results of efficiency and**

**1609      clusterization are a first approximation calculated without performing any muon tracking in**

1610 between the different detectors. This offline tool provides the user with a preliminar calculation  
 1611 of the efficiency and of the muon event parameters. Another analysis software especially  
 1612 dedicated to muon tracking is called on selected data to retrieve the results of efficiency and  
 1613 muon clusterization using a tracking algorithm to discriminate noise or gamma from muons  
 1614 as muons are the only particles that pass through the full setup, leaving hits than can be used  
 1615 to reconstruct their tracks.

- 1616 • `Offline-Rate.csv`, is used to write the noise or gamma rates measured in the detector readout  
 1617 partitions.

1618 Note that these 4 CSV files are created along with their *headers* (`Offline-[...]-Header.csv`  
 1619 containing the names of each data columns) and are automatically merged together when the offline  
 1620 analysis tool is called from the webDCS, contrary to the case where the tool is runned locally from  
 1621 the terminal as the merging bash script is then not called. Thus, the resulting files, used to make  
 1622 official plots, are:

- 1623 • `Corrupted.csv`,  
 1624 • `Current.csv`,  
 1625 • `L0-EffCl.csv`.  
 1626 • `Rate.csv`.

## 1627 B.3 Analysis inputs and information handling

1628 The usage of the Offline Analysis tool as well as its output have been presented in the previous section.  
 1629 It is now important to dig further and start looking at the source code and the inputs necessary  
 1630 for the tool to work. Indeed, other than the raw ROOT data files that are analysed, more information  
 1631 needs to be imported inside of the program to perform the analysis such as the description of the  
 1632 setup inside of GIFT++ at the time of data taking (number of trolleys, of RPCs, dimensions of the  
 1633 detectors, etc...) or the mapping that links the TDC channels to the coresponding RPC channels in  
 1634 order to translate the TDC information into human readable data. 2 files are used to transmit all this  
 1635 information:  
 1636

- 1637 • `Dimensions.ini`, that provides the necessary setup and RPC information, and  
 1638 • `ChannelsMapping.csv`, that gives the link between the TDC and RPC channels as well as the  
 1639 *mask* for each channel (masked or not?).

### 1640 B.3.1 Dimensions file and InFile parser

1641 This input file, present in every data folder, allows the analysis tool to know of the number of active  
 1642 trolleys, the number of active RPCs in those trolleys, and the details about each RPCs such as  
 1643 the number of RPC gaps, the number of pseudo-rapidity partitions (for CMS-like prototypes), the  
 1644 number of strips per partion or the dimensions. To do so, there are 3 types of groups in the INI file  
 1645 architecture. A first general group, appearing only once at the head of the document, gives information  
 1646 about the number of active trolleys as well as their IDs, as presented in Source Code B.1. For

1647 each active trolley, a group similar to Source Code B.2 can be found containing information about  
 1648 the number of active detectors in the trolley and their IDs. Each trolley group as a `Tt` name format,  
 1649 where `t` is the trolley ID. Finally, for each detector stored in slots of an active trolley, there is a group  
 1650 providing information about their names and dimensions, as shown in Source Code B.3. Each slot  
 1651 group as a `TtSs` name format, where `s` is the slot ID of trolley `t` where the active RPC is hosted.

```
1652 [General]
1653 nTrolleys=2
  TrolleysID=13
```

1654 *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.*

```
1655 [T1]
  nSlots=4
  SlotsID=1234
```

1656 *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.*

```
1657 [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
```

1658 *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.*

1659 This information is readout and stored in a C++ object called `IniFile`, that parses the information  
 1660 in the INI input file and stores it into a local buffer for later use. This INI parser is the exact same  
 1661 one that was previously developed for the Gif++ DAQ and described in Appendix A.5.2.

### 1662 **B.3.2 TDC to RPC link file and Mapping**

1663 The same way the INI dimension file information is stored using `map`, the channel mapping and mask  
 1664 information is stored and accessed through `map`. First of all, the mapping CSV file is organised into  
 1665 3 columns separated by tabulations (and not by commas, as expected for CSV files as it is easier using  
 1666 streams to read tab or space separated data using C++):

1667

1668      RPC\_channel            TDC\_channel            mask

1669      using as formatting for each field:

1670      TSCCC                  TCCC                  M

1672      TSCCC is a 5-digit integer where  $T$  is the trolley ID,  $s$  the slot ID in which the RPC is held insite  
 1673      the trolley  $T$  and CCC is the RPC channel number, or *strip* number, that can take values up to  
 1674      3-digits depending on the detector,

1675      TCCC is a 4 digit integer where  $T$  is the TDC ID, CCC is the TDC channel number that can take values  
 1676      in between 0 and 127, and

1677      M is a 1-digit integer indicating if the channel should be considered ( $M = 1$ ) or discarded ( $M = 0$ )  
 1678      during analysis.

1679      This mapping and masking information is readout and stored thanks to the object `Mapping`, pre-  
 1680      sented in Source Code B.4. Similarly to `IniFile` objects, this class has private methods. The first  
 1681      one, `Mapping::CheckIfNewLine()` is used to find the newline character '`\n`' or return character  
 1682      '`\r`' (depending on which kind of operating system interacted with the file). This is used for the  
 1683      simple reason that the masking information has been introduced only during the year 2017 but the  
 1684      channel mapping files exist since 2015 and the very beginning of data taking at GIF++. This means  
 1685      that in the older data folders, before the upgrade, the channel mapping file only had 2 columns, the  
 1686      RPC channel and the TDC channel. For compatibility reasons, this method helps controling the  
 1687      character following the readout of the 2 first fields of a line. In case any end of line character is  
 1688      found, no mask information is present in the file and the default  $M = 1$  is used. On the contrary, if  
 1689      the next character was a tabulation or a space, the mask information is present.

1690      Once the 3 fields have been readout, the second private method `Mapping::CheckIfTDCCh()` is  
 1691      used to control that the TDC channel is an existing TDC channel. Finally, the information is stored  
 1692      into 3 different maps (`Link`, `ReverseLink` and `Mask`) thanks to the public method `Mapping::Read()`.  
 1693      `Link` allows to get the RPC channel by knowing the TDC channel while `ReverseLink` does the op-  
 1694      posite by returning the TDC channel by knowing the RPC channel. Finally, `Mask` returns the mask  
 1695      associated to a given RPC channel.

```

1696 typedef map<Uint,Uint> MappingData;

1697 class Mapping {
1698     private:
1699         bool          CheckIfNewLine(char next);
1700         bool          CheckIfTDCCh(Uint channel);
1701         string        FileName;
1702         MappingData  Link;
1703         MappingData  ReverseLink;
1704         MappingData  Mask;
1705         int           Error;
1706
1707     public:
1708         Mapping();
1709         Mapping(string baseName);
1710         ~Mapping();
1711
1712         void SetFileName(const string filename);
1713         int  Read();
1714         Uint GetLink(Uint tdcchannel);
1715         Uint GetReverse(Uint rpcchannel);
1716         Uint GetMask(Uint rpcchannel);
1717     };

```

1698 *Source Code B.4: Description of C++ object Mapping used as a parser for the channel mapping and mask file.*

## 1699 B.4 Description of GIF++ setup within the Offline Analysis tool

1700 In the previous section, the tool input files have been discussed. The dimension file information is  
 1701 stored in a map hosted by the `IniFile` object. But this information is then used to create a series of  
 1702 new objects that helps defining the GIF++ infrastructure directly into the Offline Analysis. Indeed,  
 1703 from the `RPC`, to the more general `Infrastructure`, every element of the GIF++ infrastrucutre is  
 1704 recreated for each data analysis based on the information provided in input. All this information  
 1705 about the infrastructure will be used to assign each hit signal to a specific strip channel of a specific  
 1706 detector, and having a specific active area. This way, rate per unit area calculation is possible.  
 1707

### 1708 B.4.1 RPC objects

1709 `RPC` objects have been developped to represent physical active detectors in GIF++ at the moment  
 1710 of data taking. Thus, there are as many `RPC` objects created during the analysis than there were  
 1711 active `RPCs` tested during a run. Each `RPC` hosts the information present in the corresponding INI  
 1712 slot group, as shown in B.3, and organises it using a similar architecture. This can be seen from  
 1713 Source B.5.

1714 To make the object more compact, the lists of gap labels, of gap active areas and strip active  
 1715 areas are stored into `vector` dynamical containers. `RPC` objects are always contructed thanks to the  
 1716 dimension file information stored into the `IniFILE` and their ID, using the format `TtSs`. Using the  
 1717 `RPC` ID, the constructor calls the methods of `IniFILE` to initialise the `RPC`. The other constructors  
 1718 are not used but exist in case of need. Finally, some getters have been written to access the different  
 1719 private parameters storing the detector information.

```

1720
class RPC{
    private:
        string      name;           //RPC name as in webDCS database
        Uint        nGaps;          //Number of gaps in the RPC
        Uint        nPartitions;    //Number of partitions in the RPC
        Uint        nStrips;         //Number of strips per partition
        vector<string> gaps;       //List of gap labels (BOT, TOP, etc...)
        vector<float>  gapGeo;        //List of gap active areas
        vector<float>  stripGeo;      //List of strip active areas

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

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

1721

```

1722 *Source Code B.5: Description of C++ objects RPC that describe each active detectors used during data taking.*

## 1723 B.4.2 Trolley objects

1724 Trolley objects have been developped to represent physical active trolleys in GIFT++ at the moment  
 1725 of data taking. Thus, there are as many trolley objects created during the analysis than there were  
 1726 active trolleys hosting tested RPCs during a run. Each Trolley hosts the information present in the  
 1727 corresponding INI trolley group, as shown in B.2, and organises it using a similar architecture. In  
 1728 addition to the information hosted in the INI file, these object have a dynamical container of RPC  
 1729 objects, representing the active detectors the active trolley was hosting at the time of data taking.  
 1730 This can been seen from Source Code B.6.

1731 Trolley objects are always contructed thanks to the dimension file information stored into the  
 1732 IniFILE and their ID, using the format Tt. Using the Trolley ID, the constructor calls the methods  
 1733 of IniFILE to initialise the Trolley. Retrieving the information of the RPC IDs via SlotsID, a new  
 1734 RPC is constructed and added to the container RPCs for each character in the ID string. The other  
 1735 constructors are not used but exist in case of need. Finally, some getters have been written to access  
 1736 the different private parameters storing the trolley and detectors information.

```

1737
class Trolley{
    private:
        Uint          nSlots; //Number of active RPCs in the considered trolley
        string        SlotsID; //Active RPC IDs written into a string
        vector<RPC*> RPCs;   //List of active RPCs

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

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

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

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

```

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

### 1740 B.4.3 Infrastructure object

1741 The `Infrastructure` object has been developped to represent the GIFT++ bunker area dedicated to  
 1742 CMS RPC experiments. With this very specific object, all the information about the CMS RPC  
 1743 setup within GIFT++ at the moment of data taking is stored. It hosts the information present in the  
 1744 corresponding INI general group, as shown in B.1, and organises it using a similar architecture. In  
 1745 addition to the information hosted in the INI file, this object have a dynamical container of `Trolley`  
 1746 objects, representing the active tolleys in GIFT++ area. This can be seen from Source Code B.7.

1747 The `Infrastructure` object is always contructed thanks to the dimension file information stored  
 1748 into the `IniFILE`. Retrieving the information of the trolley IDs via `TrolleysID`, a new `Trolley` is  
 1749 constructed and added to the container `Trolleys` for each character in the ID `string`. By extension,  
 1750 it is easy to understand that the process described in Section B.4.2 for the construction of RPCs  
 1751 takes place when a trolley is constructed. The other constructors are not used but exist in case of  
 1752 need. Finally, some getters have been written to access the different private parameters storing the  
 1753 infrastructure, tolleys and detectors information.

```

1754
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);

1755
        //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++.

## 1757 B.5 Handeling of data

1758 As discussed in Appendix A.4.2, the raw data as a `TTree` architecture where every entry is related to  
 1759 a trigger signal provided by a muon or a random pulse, whether the goal of the data taking was to  
 1760 measure the performance of the detector or the noise/gamma background respectively. Each of these  
 1761 entries, referred also as events, contain a more or less full list of hits in the TDC channels to which  
 1762 the detectors are connected. To this list of hits corresponds a list of time stamps, marking the arrival  
 1763 of the hits within the TDC channel.

1764 The infrastructure of the CMS RPC experiment within GIF++ being defined, combining the  
 1765 information about the raw data with the information provided by both the mapping/mask file and the  
 1766 dimension file allows to build new physical objects that will help in computing efficiency or rates.

### 1767 B.5.1 RPC hits

1768 The raw data stored in the ROOT file as output of the GIFT++ DAQ, is readout by the analysis tool  
 1769 using the structure `RAWData` presented in Source Code B.9 that differs from the structure presented  
 1770 in Appendix A.4.2 as it is not meant to hold all of the data contained in the ROOT file. In this sense,  
 1771 this structure is in the case of the offline analysis tool not a dynamical object and will only be storing  
 1772 a single event contained in a single entry of the `TTree`.

```
1773
1774 class RPCHit {
1775     private:
1776         Uint Channel;      //RPC channel according to mapping (5 digits)
1777         Uint Trolley;      //0, 1 or 3 (1st digit of the RPC channel)
1778         Uint Station;      //Slot where is held the RPC in Trolley (2nd digit)
1779         Uint Strip;        //Physical RPC strip where the hit occurred (last 3
1780             digits)
1781         Uint Partition;    //Readout partition along eta segmentation
1782         float TimeStamp;   //Time stamp of the arrival in TDC
1783
1784     public:
1785         //Constructors, destructor & operator =
1786         RPCHit();
1787         RPCHit(Uint channel, float time, Infrastructure* Infra);
1788         RPCHit(const RPCHit& other);
1789         ~RPCHit();
1790         RPCHit& operator=(const RPCHit& other);
1791
1792         //Get RPCHit members
1793         Uint GetChannel();
1794         Uint GetTrolley();
1795         Uint GetStation();
1796         Uint GetStrip();
1797         Uint GetPartition();
1798         float GetTime();
1799     };
2000
2001     typedef vector<RPCHit> HitList;
2002     typedef struct GIFHitList { HitList rpc[NTROLLEYS][NSLOTS][NPARTITIONS]; } →
2003         GIFHitList;
2004
2005     bool SortHitbyStrip(RPCHit h1, RPCHit h2);
2006     bool SortHitbyTime(RPCHit h1, RPCHit h2);
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
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
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
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
3139
3139
3140
3141
3142
3143
3144
3145
3146
3147
3147
3148
3149
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3159
3160
3161
3162
3163
3164
3165
3166
3167
3167
3168
3169
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3179
3180
3181
3182
3183
3184
3185
3186
3187
3187
3188
3189
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
3217
3218
3219
3219
3220
3221
3222
3223
3224
3225
3226
3227
3227
3228
3229
3229
3230
3231
3232
3233
3234
3235
3236
3237
3237
3238
3239
3239
3240
3241
3242
3243
3244
3245
3246
3247
3247
3248
3249
3249
3250
3251
3252
3253
3254
3255
3256
3257
3257
3258
3259
3259
3260
3261
3262
3263
3264
3265
3266
3267
3267
3268
3269
3269
3270
3271
3272
3273
3274
3275
3276
3276
3277
3278
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3287
3288
3289
3289
3290
3291
3292
3293
3294
3295
3296
3297
3297
3298
3299
3299
3300
3301
3302
3303
3304
3305
3306
3307
3307
3308
3309
3309
3310
3311
3312
3313
3314
3315
3316
3317
3317
3318
3319
3319
3320
3321
3322
3323
3324
3325
3326
3327
3327
3328
3329
3329
3330
3331
3332
3333
3334
3335
3336
3337
3337
3338
3339
3339
3340
3341
3342
3343
3344
3345
3346
3347
3347
3348
3349
3349
3350
3351
3352
3353
3354
3355
3356
3357
3357
3358
3359
3359
3360
3361
3362
3363
3364
3365
3366
3367
3367
3368
3369
3369
3370
3371
3372
3373
3374
3375
3376
3376
3377
3378
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3387
3388
3389
3389
3390
3391
3392
3393
3394
3395
3396
3397
3397
3398
3399
3399
3400
3401
3402
3403
3404
3405
3406
3407
3407
3408
3409
3409
3410
3411
3412
3413
3414
3415
3416
3417
3417
3418
3419
3419
3420
3421
3422
3423
3424
3425
3426
3427
3427
3428
3429
3429
3430
3431
3432
3433
3434
3435
3436
3437
3437
3438
3439
3439
3440
3441
3442
3443
3444
3445
3446
3447
3447
3448
3449
3449
3450
3451
3452
3453
3454
3455
3456
3457
3457
3458
3459
3459
3460
3461
3462
3463
3464
3465
3466
3467
3467
3468
3469
3469
3470
3471
3472
3473
3474
3475
3476
3477
3477
3478
3479
3479
3480
3481
3482
3483
3484
3485
3486
3487
3487
3488
3489
3489
3490
3491
3492
3493
3494
3495
3496
3497
3497
3498
3499
3499
3500
3501
3502
3503
3504
3505
3506
3507
3507
3508
3509
3509
3510
3511
3512
3513
3514
3515
3516
3517
3517
3518
3519
3519
3520
3521
3522
3523
3524
3525
3526
3527
3527
3528
3529
3529
3530
3531
3532
3533
3534
3535
3536
3537
3537
3538
3539
3539
3540
3541
3542
3543
3544
3545
3546
3547
3547
3548
3549
3549
3550
3551
3552
3553
3554
3555
3556
3557
3557
3558
3559
3559
3560
3561
3562
3563
3564
3565
3566
3567
3567
3568
3569
3569
3570
3571
3572
3573
3574
3575
3576
3577
3577
3578
3579
3579
3580
3581
3582
3583
3584
3585
3586
3587
3587
3588
3589
3589
3590
3591
3592
3593
3594
3595
3596
3597
3597
3598
3599
3599
3600
3601
3602
3603
3604
3605
3606
3607
3607
3608
3609
3609
3610
3611
3612
3613
3614
3615
3616
3617
3617
3618
3619
3619
3620
3621
3622
3623
3624
3625
3626
3627
3627
3628
3629
3629
3630
3631
3632
3633
3634
3635
3636
3637
3637
3638
3639
3639
3640
3641
3642
3643
3644
3645
3646
3647
3647
3648
3649
3649
3650
3651
3652
3653
3654
3655
3656
3657
3657
3658
3659
3659
3660
3661
3662
3663
3664
3665
3666
3667
3667
3668
3669
3669
3670
3671
3672
3673
3674
3675
3676
3677
3677
3678
3679
3679
3680
3681
3682
3683
3684
3685
3686
3687
3687
3688
3689
3689
3690
3691
3692
3693
3694
3695
3696
3697
3697
3698
3699
3699
3700
3701
3702
3703
3704
3705
3706
3707
3707
3708
3709
3709
3710
3711
3712
3713
3714
3715
3716
3717
3717
3718
3719
3719
3720
3721
3722
3723
3724
3725
3726
3727
3727
3728
3729
3729
3730
3731
3732
3733
3734
3735
3736
3737
3737
3738
3739
3739
3740
3741
3742
3743
3744
3745
3746
3747
3747
3748
3749
3749
3750
3751
3752
3753
3754
3755
3756
3757
3757
3758
3759
3759
3760
3761
3762
3763
3764
3765
3766
3767
3767
3768
3769
3769
3770
3771
3772
3773
3774
3775
3776
3777
3777
3778
3779
3779
3780
3781
3782
3783
3784
3785
3786
3787
3787
3788
3789
3789
3790
3791
3792
3793
3794
3795
3796
3797
3797
3798
3799
3799
3800
3801
3802
3803
3804
3805
3806
3807
3807
3808
3809
3809
3810
3811
3812
3813
3814
3815
3816
3817
3817
3818
3819
3819
3820
3821
3822
3823
3824
3825
3826
3827
3827
3828
3829
3829
3830
3831
3832
3833
3834
3835
3836
3837
3837
3838
3839
3839
3840
3841
3842
3843
3844
3845
3846
3847
3847
3848
3849
3849
3850
3851
3852
3853
3854
3855
3856
3857
3857
3858
3859
3859
3860
3861
3862
3863
3864
3865
3866
3867
3867
3868
3869
3869
3870
3871
3872
3873
3874
3875
3876
3877
3877
3878
3879
3879
3880
3881
3882
3883
3884
3885
3886
3887
3887
3888
3889
3889
3890
3891
3892
3893
3894
3895
3896
3897
3897
3898
3899
3899
3900
3901
3902
3903
3904
3905
3906
3907
3907
3908
3909
3909
3910
3911
3912
3913
3914
3915
3916
3917
3917
3918
3919
3919
3920
3921
3922
3923
3924
3925
3926
3927
392
```

```

1781 TTree* dataTree = (TTree*)dataFile.Get("RAWData");
1782 RAWData data;
1783
1784 dataTree->SetBranchAddress("EventNumber", &data.iEvent);
1785 dataTree->SetBranchAddress("number_of_hits", &data.TDCNHits);
1786 dataTree->SetBranchAddress("Quality_flag", &data.QFlag);
1787 dataTree->SetBranchAddress("TDC_channel", &data.TDCCh);
1788 dataTree->SetBranchAddress("TDC_TimeStamp", &data.TDCTS);

```

1783       *Source Code B.10: Example of link in between RAWData and TTree.*

1784       The data is then analysed entry by entry and to each element of the TDC channel list, a `RPCHit` is  
1785       constructed by linking each TDC channel to the corresponding RPC channel thanks to the `Mapping`  
1786       object. The information carried by the RPC channel format allows to easily retrieve the trolley and  
1787       slot from which the hit was recorded (see section B.3.2). Using these 2 values, the readout partition  
1788       can be found by knowing the strip channel and comparing it with the number of partitions and strips  
1789       per partition stored into the `Infrastructure` object.

1790       Thus `RPCHit` objects are then stored into 3D dynamical list called `GIFHitList` (Source Code B.9)  
1791       where the 3 dimensions refer to the 3 layers of the readout in `GIF++` : in the bunker there are *trolleys*  
1792       ( $\tau$ ) holding detectors in *slots* ( $s$ ) and each detector readout is divided into 1 or more pseudo-rapidity  
1793       *partitions* ( $p$ ). Using these 3 information allows to assign an address to each readout partition and  
1794       this address will point to a specific hit list.

1795

## 1796     B.5.2 Clusters of hits

1797       All the hits contained in the ROOT file have been sorted into the different hit lists through the  
1798       `GIFHitList`. At this point, it is possible to start looking for clusters. A cluster is a group of adjacent  
1799       strips getting hits within a time window of 25 ns. These strips are then assumed to be part of the same  
1800       physical avalanche signal generated by a muon passing through the chamber or by the interaction of  
1801       a gamma stopping into the electrodes of the RPCs.

1802       To keep the cluster information, `RPCCluster` objects have been defined as shown in Source  
1803       Code B.11. Using the information of each individual `RPCHit` taken out of the hit list, it stores  
1804       the cluster size (number of adjacent strips composing the cluster), the first and last hit, the center for  
1805       spatial reconstruction and finally the start and stop time stamps as well as the time spread in between  
1806       the first and last hit.

```
1807 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.

## B.6 DAQ data Analysis

1823 All the ingredients to analyse GIF++ data have been defined. This section will focus on the different  
1824 part of the analysis performed on the data, from determining the type of data the tool is dealing with

1825 to calculating the rate in each detector or reconstructing muon or gamma clusters.

### 1826 B.6.1 Determination of the run type

1827 In GIF++, both the performance of the detectors in detecting muons in an irradiated environment and  
 1828 the gamma background can be independantly measured. These corresponds to different run types  
 1829 and thus, to different TDC settings giving different data to look at.

1830

1831 In the case of performance measurements, the trigger for data taking is provided by the coïncidence  
 1832 of several scintillators when muons from the beam passing through the area are detected. Data  
 1833 is collected in a 600 ns wide window around the arrival of muons in the RPCs. The expected time  
 1834 distribution of hits is shown in Figure B.1a. The muon peak is clearly visible in the center of the  
 1835 distribution and is to be extracted from the gamma background that composes the flat part of the  
 1836 distribution.

1837 On the other hand, gamma background or noise measurements are focussed on the non muon  
 1838 related physics and the trigger needs to be independant from the muons to give a good measurement  
 1839 of the gamma/noise distribution as seen by the detectors. The trigger is then provided by a pulse  
 1840 generator at a frequency of 300 Hz whose pulse is not likely to be on time with a muon. In order  
 1841 to increase the integrated time without increasing the acquisition time too much, the width of the  
 1842 acquisition windows are increased to 10  $\mu$ s. The time distribution of the hits is expected to be flat, as  
 1843 shown by Figure B.1b.

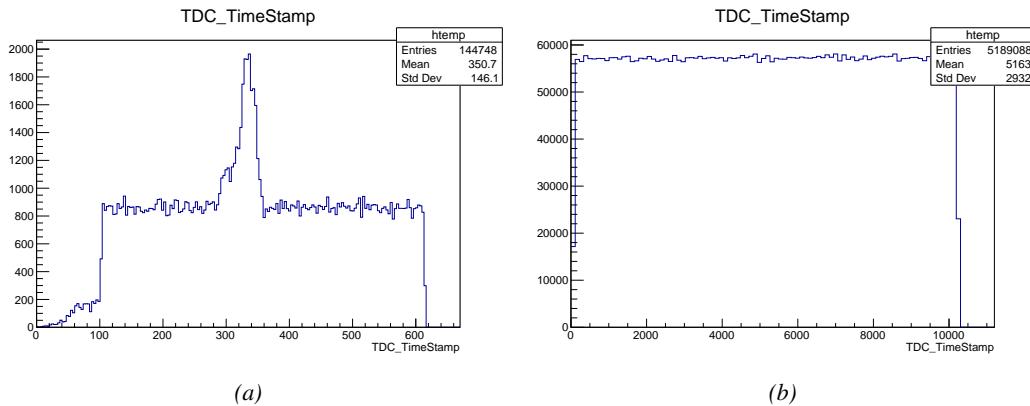


Figure B.1: Example of expected hit time distributions in the cases of efficiency (Figure B.1a) and noise/gamma rate per unit area (Figure B.1b) measurements as extracted from the raw ROOT files. The unit along the x-axis corresponds to ns. The fact that "the" muon peak is not well defined in Figure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

1844 The ROOT files include a TTree called RunParameters containing, among other things, the in-  
 1845 formation related to the type of run. The run type can then be accessed as described by Source  
 1846 Code B.12 and the function IsEfficiencyRun() is then used to determine if the run file is an effi-  
 1847 ciency run or, on the contrary, another type of run (noise or gamma measurement).

```

1848     TTree* RunParameters = (TTree*)dataFile.Get("RunParameters");
1849     TString* RunType = new TString();
1850     RunParameters->SetBranchAddress("RunType", &RunType);
1851     RunParameters->GetEntry(0);

```

1850       *Source Code B.12: Access to the run type contained in TTree\* RunParameters.*

1851       Finally, the data files will have a slightly different content whether it was collected before or after  
 1852       October 2017 and the upgrade of the DAQ software that brought a new information into the ROOT  
 1853       output. This is discussed in Appendix A.4.3 and implies that the analysis will differ a little depending  
 1854       on the data format. Indeed, as no information on the data quality is stored, in older data files, the cor-  
 1855       rections for missing events has to be done at the end of the analysis. The information about the type  
 1856       of data format is stored in the variable **bool** `isNewFormat` by checking the list of branches contained  
 1857       in the data tree via the methods `TTree::GetListOfBranches()` and `TCollection::Contains()`.

## 1858     **B.6.2 Beam time window calculation for efficiency runs**

1859       Knowing the run type is important first of all to know the width of the acquisition window to be used  
 1860       for the rate calculation and finally to be able to seek for muons. Indeed, the peak that appears in the  
 1861       time distribution for each detectors is then fitted to extract the most probable time window in which  
 1862       the tool should look for muon hits. The data outside of this time window is then used to evaluate the  
 1863       noise or gamma background the detector was subjected to during the data taking. Computing the  
 1864       position of the peak is done calling the function `SetBeamWindow()` defined in file `src/RPCHit.cc` that  
 1865       loops a first time on the data. The data is first sorted in a 3D array of 1D histograms (`GIFH1Array`, see  
 1866       include/types.h). Then the location of the highest bin is determined using `TH1::GetMaximumBin()`  
 1867       and is used to define a window in which a gaussian fit will be applied to compute the peak width.  
 1868       This window is a 80 ns defined by Formula B.1 around the central bin.

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

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

1869       Before the fit is performed, the average number of noise/gamma hits per bin is evaluated using  
 1870       the data outside of the fit window. Excluding the first 100 ns, the average number of hits per bin  
 1871       due to the noise or gamma is defined by Formula B.2 after extracting the amount of hits in the time  
 1872       windows  $[100; t_{low}]$  and  $[t_{high}; 600]$  thanks to the method `TH1::Integral()`. This average number  
 1873       of hits is then subtracted to every bin of the 1D histogram, in order to *clean* it from the noise or  
 1874       gamma contribution as much as possible to improve the fit quality. Bins where  $\langle n_{hits} \rangle$  is greater  
 1875       than the actual bin content are set to 0.

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

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

1876       Finally, the fit parameters are extracted and saved for each detector in 3D arrays of **float**  
 1877       (`muonPeak`, see include/types.h), a first one for the mean arrival time of the muons, `PeakTime`,

1878 and a second one for the width of the peak, `PeakWidth`. The width is defined as  $6\sigma$  of the gaussian  
 1879 fit. The same settings are applied to every partitions of the same detector. To determine which one  
 1880 of the detector's partitions is directly illuminated by the beam, the peak height of each partition is  
 1881 compared and the highest one is then used to define the peak settings.

### 1882 **B.6.3 Data loop and histogram filling**

1883 3D arrays of histogram are created to store the data and display it on the DQM of G4F++ webDCS  
 1884 for the use of shifters. These histograms, presented in section B.2.1.1, are filled while looping on  
 1885 the data. Before starting the analysis loop, it is necessary to control the entry quality for the new  
 1886 file formats featuring `QFlag`. If the `QFlag` value for this entry shows that 1 TDC or more have a  
 1887 `CORRUPTED` flag, then this event is discarded. The loss of statistics is low enough to be neglected.  
 1888 `QFlag` is controlled using the function `IsCorruptedEvent()` defined in `src/utils.cc`. As explained  
 1889 in Appendix A.4.3, each digit of this integer represents a TDC flag that can be 1 or 2. Each 2 is  
 1890 the sign of a `CORRUPTED` state. Then, the data is accessed entry by entry in the ROOT `TTree` using  
 1891 `RAWData` and each hit in the hit list is assigned to a detector channel and saved in the corresponding  
 1892 histograms. In the first part of the analysis, in which the loop over the ROOT file's content is  
 1893 performed, the different steps are:

1894 **1- RPC channel assignment and control:** a check is done on the RPC channel extracted thanks  
 1895 to the mapping via the method `Mapping::GetLink()`. If the channel is not initialised and is 0, or if  
 1896 the TDC channel was greater than 5127, the hit is discarded. This means there was a problem in the  
 1897 mapping. Often a mapping problem leads to the crash of the offline tool.

1898 **2- Creation of a `RPCHit` object:** to easily get the trolley, slot and partition in which the hit has  
 1899 been assigned, this object is particularly helpful.

1900 **3- General histograms are filled:** the hit is filled into the time distribution and the general hit  
 1901 distribution histograms, and if the arrival time is within the first 100 ns, it is discarded and nothing  
 1902 else happens and the loop proceeds with the next hit in the list.

1903 **4- Multiplicity counter:** the hit multiplicity counter of the corresponding detectors incremented.

1904 **5-a- Efficiency runs - Is the hit within the peak window? :** if the peak is contained in the peak  
 1905 window previously defined in section B.6.2, the hit is filled into the beam hit profile histogram of  
 1906 the corresponding chamber, added into the list of muon hits and increments the counter of *in time*  
 1907 hits. The term *in time* here refers to the hits that are likely to be muons by arriving in the expected  
 1908 time window. If the hit is outside of the peak window, it is filled into the noise profile histogram  
 1909 of the corresponding detector, added into the list of noise/gamma hits and increments the counter of  
 1910 noise/gamma hits.

1911 **5-b- Noise/gamma rate runs - Noise histograms are filled:** the hit is filled into the noise profile  
 1912 histogram of the corresponding detector, added into the list of noise/gamma hits and increments the  
 1913 counter of noise/gamma hits.

1915 After the loop on the hit list of the entry is over, the next step is to clusterize the 3D lists filled  
 1916 in the previous steps. A 3D loop is then started over the active trolley, slot and RPC partitions to  
 1917 access these objects. Each `NoiseHitList` and `MuonHitList`, in case of efficiency run, are clusterized  
 1918 as described in section B.5.2. There corresponding cluster size and multiplicity histograms are filled  
 1919 at the end of the clustering process. Then, the efficiency histogram is filled in case of efficiency run.  
 1920 The selection is simply made by checking whether the RPC detected signals in the peak window  
 1921 during this event. Nevertheless, it is useful to highlight that at this level, it is not possible yet to  
 1922 discriminate in between a muon hit and noise or gamma hit. Thus, `MuonCSize_H`, `MuonCMult_H`  
 1923 and `Efficiency0_H` are subjected to noise and gamma contamination. This contamination will be  
 1924 estimated and corrected at the moment the results will be written into output CSV files. Finally, the  
 1925 loop ends on the filling of the general hit multiplicity histogram.

## 1926 **B.6.4 Results calculation**

1927 As mentioned in section B.2.1, the analysis of DAQ data provides the user with 3 CSV files and  
 1928 a ROOT file associated to each and every ROOT data file. The fourth CSV file is provided by the  
 1929 extraction of the CEAN main frame data monitored during data taking and will be discussed later.  
 1930 After looping on the data in the previous part of the analysis macro, the output files are created and a  
 1931 3D loop on each RPC readout partitions is started to extract the histograms parameters and compute  
 1932 the final results.

1933

### 1934 **B.6.4.1 Rate normalisation**

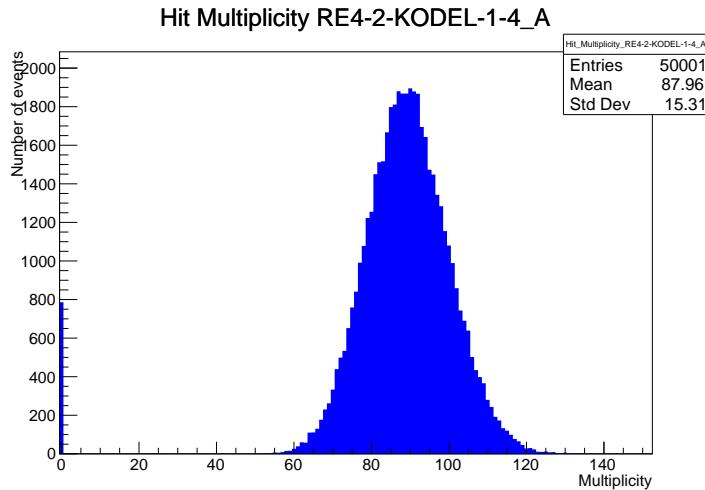


Figure B.2: The effect of the quality flag is explained by presenting the reconstructed hit multiplicity of a data file without `Quality_flag`. The artificial high content of bin 0 is the effect of corrupted data.

1935 To analyse old data format files, not containing any quality flag, it is needed to estimate the amount  
 1936 of corrupted data via a fit as the corrupted data will always fill events with a fake "0 multiplicity".  
 1937 Indeed, as no hits were stored in the DAQ ROOT files, these events artificially contribute to fill  
 1938 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 5.3, 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.

```

1961
if(!isNewFormat){
    TF1* GaussFit = new TF1("gaussfit","[0]*exp(-0.5*((x-[1])/[2])**2)",0,Xmax);
    GaussFit->SetParameter(0,100);
    GaussFit->SetParameter(1,10);
    GaussFit->SetParameter(2,1);
    HitMultiplicity_H.rpc[T][S][p]->Fit(GaussFit,"LIQR","",0.5,Xmax);

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

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

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

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

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

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

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

```

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

### **1964 B.6.4.2 Rate and activity**

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

- 1970     ● the strip rate, defined as the number of hits recorded in the bin normalised like described in  
 1971       the previous section, using the variable `rate_norm`, and

- 1972     ● the strip activity, defined as the number of hits recorded in the bin normalised to the average  
 1973       number of hits per bin contained in the partition histogram, using the variable `averageNhit`.  
 1974       This value provides an information on the homogeneity of the detector response to the gamma  
 1975       background or of the detector noise. An activity of 1 corresponds to an average response.  
 1976       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);
}

```

1978 *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.*

1979     On each detector partitions, which are readout by a single FEE, all the channels are not processed  
 1980       by the same chip. Each chip can give a different noise response and thus, histograms using a chip  
 1981       binning are used to investigate chip related noise behaviours. The average values of the strip rate  
 1982       or activity grouped into a given chip are extracted using the using the function `GetChipBin()` and  
 1983       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;
}

```

1986 *Source Code B.15: Function used to compute the content of a bin for an histogram using chip binning.*

```

1987   for(UInt ch = 0; ch < (nStripsPart/NSTRIPSCHIP); ch++) {
1988     ChipMeanNoiseProf_H.rpc[T][S][p]->
1989       SetBinContent(ch+1,GetChipBin(StripNoiseProfile_H.rpc[T][S][p],ch));
1990     ChipActivity_H.rpc[T][S][p]->
1991       SetBinContent(ch+1,GetChipBin(StripActivity_H.rpc[T][S][p],ch));
1992   }

```

*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.

```

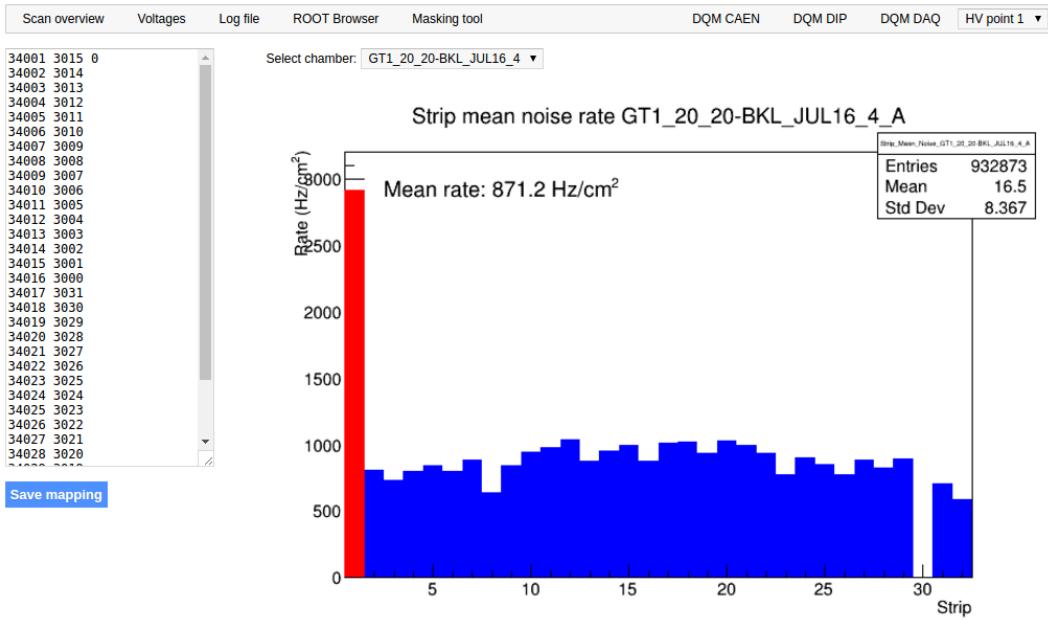
2001   float MeanPartSDev = GetTH1StdDev(StripNoiseProfile_H.rpc[T][S][p]);
2002   float strip_homog = (MeanPartRate==0)
2003     ? 0.
2004     : exp(-MeanPartSDev/MeanPartRate);
2005   StripHomogeneity_H.rpc[T][S][p]->Fill("exp -#left(#frac{\#sigma_{Strip}
2006     \rightarrow Rate}{\#mu_{Strip Rate}}\#right)",strip_homog);
2007   StripHomogeneity_H.rpc[T][S][p]->GetYaxis()->SetRangeUser(0.,1.);
2008
2009   float ChipStDevMean = GetTH1StdDev(ChipMeanNoiseProf_H.rpc[T][S][p]);
2010   float chip_homog = (MeanPartRate==0)
2011     ? 0.
2012     : exp(-ChipStDevMean/MeanPartRate);
2013   ChipHomogeneity_H.rpc[T][S][p]->Fill("exp -#left(#frac{\#sigma_{Chip}
2014     \rightarrow Rate}{\#mu_{Chip Rate}}\#right)",chip_homog);
2015   ChipHomogeneity_H.rpc[T][S][p]->GetYaxis()->SetRangeUser(0.,1.);

```

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

#### 2004    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.*

2010      From the code point of view, the function `GetTH1Mean()` is used to retrieve the mean rate par-  
 2011      tition by partition after the rates have been calculated strip by strip and filled into the histograms  
 2012      `StripNoiseProfile_H.rpc[T][S][p]`, as described through Source Code B.18.

2013      Once the mask for each rejected channel has been updated, the shifter can manually run the of-  
 2014      fline tool again to update the DQM plots, now including the masked strips, as well the rate results  
 2015      written in the output CSV file `Offline-Rate.csv`. If not done during the shifts, the strip masking  
 2016      procedure needs to be carefully done by the person in charge of data analysis on the scans that were  
 2017      selected to produce the final results.

```

2018
float GetTH1Mean(TH1* H) {
    int nBins = H->GetNbinsX();
    int nActive = nBins;
    float mean = 0.;

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

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

    return mean;
}

```

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.

#### 2021 B.6.4.4 Output CSV files filling

2022 All the histograms have been filled. Parameters will then be extracted from them to compute the  
 2023 final results that will later be used to produce plots. Once the results have been computed, the very  
 2024 last step of the offline macro is to write these values into the corresponding CSV outputs. Aside of  
 2025 the file `Offline-Corrupted.csv`, 2 CSV files are being written by the macro `offlineAnalysis()`,  
 2026 `Offline-Rates.csv` and `Offline-L0-EffCl.csv` that respectively contain information about noise  
 2027 or gamma rates, cluster size and multiplicity, and about level 0 reconstruction of the detector effi-  
 2028 ciency, muon cluster size and multiplicity. Details on the computation and file writing are respec-  
 2029 tively given in Sources Codes B.19 and B.20.

2030 **Noise/gamma background variables** are computed and written in the output file for each detector  
 2031 partitions. A detector average of the hit and cluster rate is also provided, as shown through Sources  
 2032 Code B.19. The variables that are written for each partition are:

- 2033 • The mean partition hit rate per unit area, `MeanPartRate`, that is extracted from the histogram  
   2034   `StripNoiseProfile_H` as the mean value along the y-axis, as described in section B.6.4.3. No  
   2035   error is recorded for the hit rate as this is considered a single measurement. No statistical error  
   2036   can be associated to it and the systematics are unknown.
- 2037 • The mean cluster size, `cSizePart`, is extracted from the histogram `NoiseCSize_H` and it's  
   2038   statistical error, `cSizePartErr`, is taken to be  $2\sigma$  of the total distribution.
- 2039 • The mean cluster multiplicity per trigger, `cMultPart`, is extracted from the histogram `NoiseCMult_H`  
   2040   and it's statistical error, `cMultPartErr`, is taken to be  $2\sigma$  of the total distribution. It is impor-  
   2041   tant to point to the fact that this variable gives an information that is dependent on the buffer  
   2042   window width used for each trigger for the calculation.
- 2043 • The mean cluster rate per unit area, `ClustPartRate`, is defined as the mean hit rate normalised

2044 to the mean cluster size and it's statistical error, `ClustPartRateErr`, is then obtained using the  
 2045 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';
  }
}

```

2047 *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.*

2048 **Muon performance variables** are computed and written in the output file for each detector partitions as shown through Sources Code B.20. The variables that are written for each partition are:

- 2050  
2051     • The muon efficiency, `eff`, extracted from the histogram `Efficiency0_H`. It is reminded that  
2052       this offline tool doesn't include any tracking algorithm to identify muons from the beam and  
2053       only relies on the hits arriving in the time window corresponding to the beam time. The con-  
2054       tent of the efficiency histogram is thus biased by the noise/gamma background contribution  
2055       into this window and is thus corrected by estimating the muon data content in the peak re-  
2056       gion knowing the noise/gamma content in the rate calculation region. Both time windows  
2057       being different, the choice was made to normalise the noise/gamma background calculation  
2058       window to it's equivalent beam window in order to have comparable values using the variable  
2059       `windowRatio`. Finally, to estimate the data ratio in the peak region, the variable `DataRatio`  
2060       is defined as the ratio in between the estimated mean cluster multiplicity of the muons in the  
2061       peak region, `MuonCM`, and of the total mean cluster multiplicity in the peak region, `PeakCM`.  
2062       `MuonCM` is itself defined as the difference in between the total mean cluster multiplicity in the  
2063       peak region and the normalised mean noise/gamma cluster multiplicity calculated outside of  
2064       the peak region. The statistical error related to the efficiency, `eff_err`, is computed using a  
2065       binomial distribution, as the efficiency measure the probability of "success" and "failure" to  
2066       detect muons.
- 2066  
2067     • The mean muon cluster size, `MuonCS`, is calculated using the total mean cluster size and multi-  
2068       plicity in the peak region, respectively extracted from histograms `MuonCSize_H` and `MuonCMult_H`,  
2069       the noise/gamma background mean cluster size and normalised multiplicity, extracted from  
2070       `NoiseCSize_H` and `NoiseCMult_H`, and of the estimated muon cluster multiplicity `MuonCM` pre-  
2071       viously explicated. The associated statistical error, `MuonCM_err`, is calculated using the propa-  
2072       gation of errors of the mentioned variables.
- 2073  
2074     • The mean muon cluster multiplicity in the peak region, `MuonCM`, explicated above whose sta-  
2075       tistical error, `MuonCM_err`, is the sum of statistical error associated to the total mean clus-  
2076       ter multiplicity in the peak reagion, `PeakCM_err`, and of the mean noise/gamma cluster size,  
2077       `NoiseCM_err`.

2076       In addition to these 2 CSV files, the histograms are saved in ROOT file `Scan00XXXX_HVY_Offline.root`  
2077       as explained in section B.2.1.1.

2078

```

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';
        }
    }
}

```

2079

*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.*

2080

## 2081 B.7 Current data Analysis

2082 Detectors under test at GIF++ are connected both to a CAEN HV power supply and to a CAEN  
2083 ADC that reads the currents inside of the RPC gaps bypassing the supply cable. During data tak-  
2084 ing, the webDCS records into a ROOT file called `Scan00XXXX_HVY_CAEN.root` histograms with the  
2085 monitored parameters of both CAEN devices. Are recorded for each RPC channels (in most cases,  
2086 a channel corresponds to an RPC gap):

- 2087 • the effective voltage,  $HV_{eff}$ , set by the webDCS using the PT correction on the CAEN power  
2088 supply,
- 2089 • the applied voltage,  $HV_{app}$ , monitored by the CAEN power supply, and the statistical error  
2090 related to the variations of this value through time to follow the variation of the environmental  
2091 parameters defined as the RMS of the histogram divided by the square root of the number of  
2092 recorded points,
- 2093 • the monitored current,  $I_{mon}$ , monitored by the CAEN power supply, and the statistical error  
2094 related to the variations of this value through time to follow the variation of the environmental  
2095 parameters defined as the RMS of the histogram divided by the square root of the number of  
2096 recorded points,
- 2097 • the corresponding current density,  $J_{mon}$ , defined as the monitored current per unit area,  
2098  $J_{mon} = I_{mon}/A$ , where  $A$  is the active area of the corresponding gap,
- 2099 • the ADC current,  $I_{ADC}$ , recorded through the CAEN ADC module that monitors the dark  
2100 current in the gap itself. First of all, the resolution of such a module is better than that of  
2101 CAEN power supplies and moreover, the current is not read-out through the HV supply line  
2102 but directly at the chamber level giving the real current inside of the detector. The statistical  
2103 error is defined as the RMS of the histogram distribution divided by the square root of the  
2104 number of recorded points.

2105 Once extracted through a loop over the element of GIF++ infrastructure via the C++ macro  
2106 `GetCurrent()`, these parameters, organised in 9 columns per detector HV supply line, are written in  
2107 the output CSV file `Offline-Current.csv`. The macro can be found in the file `Current.cc`.