



Universiteit Gent  
Faculteit Wetenschappen  
Vakgroep Fysica en Sterrenkunde

<sup>2</sup> No title yet

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

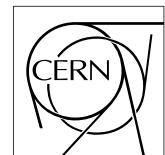
---

<sup>4</sup> Alexis Fagot

5



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





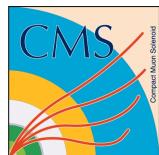


Universiteit Gent  
Faculteit Wetenschappen  
Vakgroep Fysica en Sterrenkunde

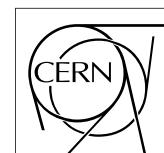
Promotoren: Dr. Michael Tytgat  
Prof. Dr. Dirk Ryckbosch

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

17



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





## Acknowledgements

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

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



# Table of Contents

23

24	<b>Acknowledgements</b>	i
25	<b>Nederlandse samenvatting</b>	vii
26	<b>English summary</b>	ix
27	<b>1 Introduction</b>	1-1
28	1.1 A story of High Energy Physics . . . . .	1-1
29	1.2 Organisation of this study . . . . .	1-1
30	<b>2 Investigating the TeV scale</b>	2-1
31	2.1 The Standard Model of Particle Physics . . . . .	2-2
32	2.1.1 A history of particle physics . . . . .	2-2
33	2.1.2 Construction and test of the model . . . . .	2-11
34	2.1.3 Investigating the TeV scale . . . . .	2-12
35	2.2 The Large Hadron Collider & the Compact Muon Solenoid . . . . .	2-12
36	2.2.1 LHC, the most powerful particle accelerator . . . . .	2-12
37	2.2.1.1 Particle acceleration . . . . .	2-13
38	2.2.1.2 LHC discoveries and LHC physics program . . . . .	2-16
39	2.2.1.3 High Luminosity LHC . . . . .	2-17
40	2.2.2 CMS, a multipurpose experiment . . . . .	2-17
41	2.3 Muon Phase-II Upgrade . . . . .	2-17
42	<b>3 Physics of Resistive plate chambers</b>	3-1
43	3.1 Principle . . . . .	3-1
44	3.1.1 Electron drift velocity . . . . .	3-4
45	3.2 Rate capability and time resolution of Resistive Plate Chambers . . . . .	3-4
46	3.2.1 Operation modes . . . . .	3-4
47	3.2.2 Detector designs and performance . . . . .	3-6
48	3.2.2.1 Double-gap RPC . . . . .	3-7
49	3.2.2.2 Multigap RPC (MRPC) . . . . .	3-8
50	3.2.2.3 Charge distribution and performance limitations . . . . .	3-9
51	3.3 Signal formation . . . . .	3-12
52	3.4 Gas transport parameters . . . . .	3-12
53	<b>4 Longevity studies and Consolidation of the present CMS RPC subsystem</b>	4-1
54	4.1 Resistive Plate Chambers at CMS . . . . .	4-1
55	4.1.1 Overview . . . . .	4-1
56	4.1.2 The present RPC system . . . . .	4-2
57	4.1.3 Pulse processing of CMS RPCs . . . . .	4-3
58	4.2 Testing detectors under extreme conditions . . . . .	4-4

---

59	4.2.1	The Gamma Irradiation Facilities . . . . .	4-7
60	4.2.1.1	GIF . . . . .	4-7
61	4.2.1.2	GIF++ . . . . .	4-9
62	4.3	Preliminary tests at GIF . . . . .	4-11
63	4.3.1	Resistive Plate Chamber test setup . . . . .	4-11
64	4.3.2	Data Acquisition . . . . .	4-13
65	4.3.3	Geometrical acceptance of the setup layout to cosmic muons . . . . .	4-13
66	4.3.3.1	Description of the simulation layout . . . . .	4-14
67	4.3.3.2	Simulation procedure . . . . .	4-16
68	4.3.3.3	Results . . . . .	4-17
69	4.3.4	Photon flux at GIF . . . . .	4-17
70	4.3.4.1	Expectations from simulations . . . . .	4-17
71	4.3.4.2	Dose measurements . . . . .	4-22
72	4.3.5	Results and discussions . . . . .	4-23
73	4.4	Longevity tests at GIF++ . . . . .	4-24
74	4.4.1	Description of the Data Acquisition . . . . .	4-27
75	4.4.2	RPC current, environmental and operation parameter monitoring . . . . .	4-28
76	4.4.3	Measurement procedure . . . . .	4-29
77	4.4.4	Longevity studies results . . . . .	4-29
78	<b>5</b>	<b>Investigation on high rate RPCs</b>	<b>5-1</b>
79	5.1	Rate limitations and ageing of RPCs . . . . .	5-1
80	5.1.1	Low resistivity electrodes . . . . .	5-1
81	5.1.2	Low noise front-end electronics . . . . .	5-1
82	5.2	Construction of prototypes . . . . .	5-1
83	5.3	Results and discussions . . . . .	5-1
84	<b>6</b>	<b>Conclusions and outlooks</b>	<b>6-1</b>
85	6.1	Conclusions . . . . .	6-1
86	6.2	Outlooks . . . . .	6-1
87	<b>A</b>	<b>A data acquisition software for CAEN VME TDCs</b>	<b>A-1</b>
88	A.1	GIF++ DAQ file tree . . . . .	A-1
89	A.2	Usage of the DAQ . . . . .	A-2
90	A.3	Description of the readout setup . . . . .	A-3
91	A.4	Data read-out . . . . .	A-3
92	A.4.1	V1190A TDCs . . . . .	A-4
93	A.4.2	DataReader . . . . .	A-6
94	A.4.3	Data quality flag . . . . .	A-10
95	A.5	Communications . . . . .	A-12
96	A.5.1	V1718 USB Bridge . . . . .	A-13
97	A.5.2	Configuration file . . . . .	A-13
98	A.5.3	WebDCS/DAQ intercommunication . . . . .	A-17
99	A.5.4	Example of inter-process communication cycle . . . . .	A-18
100	A.6	Software export . . . . .	A-18

---

101	<b>B Details on the offline analysis package</b>	<b>B-1</b>
102	B.1 GIF++ Offline Analysis file tree . . . . .	B-1
103	B.2 Usage of the Offline Analysis . . . . .	B-2
104	B.2.1 Output of the offline tool . . . . .	B-3
105	B.2.1.1 ROOT file . . . . .	B-3
106	B.2.1.2 CSV files . . . . .	B-5
107	B.3 Analysis inputs and information handling . . . . .	B-6
108	B.3.1 Dimensions file and IniFile parser . . . . .	B-6
109	B.3.2 TDC to RPC link file and Mapping . . . . .	B-7
110	B.4 Description of GIF++ setup within the Offline Analysis tool . . . . .	B-9
111	B.4.1 RPC objects . . . . .	B-9
112	B.4.2 Trolley objects . . . . .	B-10
113	B.4.3 Infrastructure object . . . . .	B-11
114	B.5 Handeling of data . . . . .	B-12
115	B.5.1 RPC hits . . . . .	B-13
116	B.5.2 Clusters of hits . . . . .	B-14
117	B.6 DAQ data Analysis . . . . .	B-15
118	B.6.1 Determination of the run type . . . . .	B-16
119	B.6.2 Beam time window calculation for efficiency runs . . . . .	B-17
120	B.6.3 Data loop and histogram filling . . . . .	B-18
121	B.6.4 Results calculation . . . . .	B-19
122	B.6.4.1 Rate normalisation . . . . .	B-19
123	B.6.4.2 Rate and activity . . . . .	B-21
124	B.6.4.3 Strip masking tool . . . . .	B-23
125	B.6.4.4 Output CSV files filling . . . . .	B-25
126	B.7 Current data Analysis . . . . .	B-29



<sup>127</sup>

## Nederlandse samenvatting –Summary in Dutch–

<sup>129</sup> Le resume en Neerlandais (j'aurais peut-être pu apprendre la langue juste pour ça...).

<sup>128</sup>



## English summary

<sup>131</sup> Le meme résume mais en Anglais (on commencera par la hein!).



# List of Figures

132

133	2.1	Through the gold foil experiment Rutherford could show that most of the mass of atoms was contained in a positively charged nucleus and could then propose a more accurate atomic model than that of Thomson. . . . .	2-3
134	2.2	Figure 2.2a: Meson octet. Figure 2.2b: Baryon octet. Figure 2.2c: Baryon decuplet. . . . .	2-8
135	2.3	The elementary particles of the Standard Model are showed along with their names and properties. Their interactions with the strong, weak and electromagnetic forces have been explicated as well. The crucial role of the Higgs boson in electroweak symmetry breaking is highlighted, and the properties of the various particles differ in the (high-energy) symmetric phase (top) and the (low-energy) broken-symmetry phase (bottom) are showed. . . . .	2-11
136	2.4	CERN accelerator complex. . . . .	2-13
137	2.5	Pictures of the different accelerators. From top to bottom: first the LINAC 2 and the <i>Pb</i> source of LINAC 3. Then the Booster and the LEIR. Finally, the PS, the SPS and the LHC. . . . .	2-15
138	2.6	Figure 2.6a: schematics of the LHC cryodipoles. 1: Superconducting Coils, 2: Beam pipe, 3: Heat exchanger Pipe, 4: Helium-II Vessel, 5: Superconducting Bus-bar, 6: Iron Yoke, 7: Non-Magnetic Collars, 8: Vacuum Vessel, 9: Radiation Screen, 10: Thermal Shield, 11: Auxiliary Bus-bar Tube, 12: Instrumentation Feed Throughs, 13: Protection Diode, 14: Quadrupole Bus-bars, 15: Spool Piece Bus-bars. Figure 2.6b: magnetic field and resulting motion force applied on the beam particles. . . . .	2-16
139	2.7	Figure 2.7a: picture of the LHC quadrupoles. Figure 2.7b: magnetic fields and resulting focussing force applied on the beam by 2 consecutive quadrupoles. . . . .	2-16
140	2.8	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. . . . .	2-17
141	2.9	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). . . . .	2-18
142	2.10	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. . . . .	2-19
143	3.1	Different phases of the avalanche development in the RPC gas volume subjected to a constant electric field $E_0$ . a) An avalanche is initiated by the primary ionisation caused by the passage of a charged particle through the gas volume. b) Due to its growing size, the avalanche starts to locally influence the electric field. c) The electrons, lighter than the cations reach the anode first. d) The ions reach the cathode. While the charges have not recombined, the electric field in the small region around the avalanche stays affected and locally blind the detector. . . . .	3-2
144			
145			
146			
147			
148			
149			
150			
151			
152			
153			
154			
155			
156			
157			
158			
159			
160			
161			
162			
163			
164			
165			
166			
167			
168			
169			
170			
171			

172	3.2	Effeciency (circles and stars with 30 mV and 100 mV thresholds respectively) and streamer probability (opened circles) as function of the operating voltatge of a 2 mm single gap HPL RPC flushed with a gas mixture containing (a) 5%, (b) 2%, (c) 1% and (d) no $SF_6$ [33]. . . . .	3-3
176	3.3	Movement of the charge carriers in an RPC. Figure 3.3a: Voltage across an RPC whose electrode have a relative permittivity of 5 at the moment the tension s applied. Figure 3.3b: After the charge carriers moved, the electrodes are charged and there is no voltage drop over the electrodes anymore. The full potential is applied on the gas gap only. Figure 3.3c: The streamer discharge initiated by a charged particle transports electrons and cations towards the anode and cathode respectively. . . . .	3-5
182	3.4	Typical oscilloscope pulses in streamer mode (Figure 3.4a) and avalanche mode(Figure 3.4b). In the case of streamer mode, the very small avalanche signal is visible. . . . .	3-5
184	3.5	Rate capability comparison for the streamer and avalanche mode of operation. An order of magnitude in rate capability for a maximal efficiency drop of 10% is gained by using the avalanche mode over the streamer mode. . . . .	3-6
187	3.6	Possible double-gap RPC layouts: a) "standard" 1D double-gap RPC, as used in CMS experiment, where the anodes are facing each other and a 1D read-out plane is sandwiched in between them, b) double read-out double-gap RPC as used in AT-LAS experiment, where the cathodes are facing each other and 2 read-out planes are used on the outer surfaces. This last layout can offer the possibility to use a 2D reconstruction by using orthogonal read-out planes. . . . .	3-7
193	3.7	Comparison of performance of CMS double and single gap RPCs using cosmic muons [44]. Figure 3.7a: Comparison of efficiency sigmoids. Figure 3.7b: Voltage distribution at 95% of maximum efficiency. Figure 3.7c: $\Delta_{10\%}^{90\%}$ distribution. . . . .	3-7
196	3.8	Presentation of ALICE MRPC using 250 $\mu m$ gas gaps, 620 $\mu m$ outer glass electrodes and 550 $\mu m$ inner floating electrodes. More details on the labels are given in [45]. . . . .	3-8
198	3.9	Particle identification applied to electrons in the STAR experiment. The identifica- tion is performed combining ToF and $dE/dx$ measurements [50]. . . . .	3-9
200	3.10	Comparison of the detector performance of ALICE ToF MRPC [51] at fixed applied voltage (in blue) and at fixed effective voltage (in red). The effective voltage is kept fixed by increasing the applied voltage accordingly to the current drawn by the detector.	3-9
203	3.11	Ratio between total induced and drifting charge have been simulated for single gap, double-gap and multigap layouts [52]. The total induced charge for a double-gap RPC is a factor 2 higher than for a multigap. . . . .	3-10
206	3.12	Charge spectra have been simulated for single gap, double-gap and multigap lay- outs [52]. It appears that when single gap shows a decreasing spectrum, double and multigap layouts exhibit a spectrum whose peak is detached from the origin. The detachment gets stronger as the number of gaps increases. . . . .	3-11
210	3.13	The maximal theoretical efficiency is simulated for single gap, double-gap and multi- gap layouts [52] at a constant gap thickness of 2 mm and using an effective Townsend coefficient of $9 \text{ mm}^{-1}$ . . . . .	3-11
213	4.1	Signals from the RPC strips are shaped by the FEE described on Figure 4.1a. Out- put LVDS signals are then read-out by a TDC module connected to a computer or converted into NIM and sent to scalers. Figure 4.1b describes how these converted signals are put in coincidence with the trigger. . . . .	4-3

217	4.2	Description of the principle of a CFD. A comparison of threshold triggering (left) and constant fraction triggering (right) is shown in Figure 4.2a. Constant fraction triggering is obtained thanks to zero-crossing technique as explained in Figure 4.2b. The signal arriving at the input of the CFD is split into three components. A first one is delayed and connected to the inverting input of a first comparator. A second component is connected to the noninverting input of this first comparator. A third component is connected to the noninverting input of another comparator along with a threshold value connected to the inverting input. Finally, the output of both comparators is fed through an AND gate. . . . .	4-4
226	4.3	(4.3a) Extrapolation from 2016 data of single hit rate per unit area in the barrel region. (4.3b) Extrapolation from 2016 data of single hit rate per unit area in the endcap region. . . . .	4-6
229	4.4	Background Fluka simulation compared to 2016 Data at $L = 10^{34} \text{ cm}^{-2} \cdot \text{s}^{-1}$ in the fourth endcap disk region. A mismatch in between simulation and data can be observed. [To be understood.] . . . . .	4-7
232	4.5	Layout of the test beam zone called X5c GIF at CERN. Photons from the radioactive source produce a sustained high rate of random hits over the whole area. The zone is surrounded by 8 m high and 80 cm thick concrete walls. Access is possible through three entry points. Two access doors for personnel and one large gate for material. A crane allows installation of heavy equipment in the area. . . . .	4-8
237	4.6	$^{137}\text{Cs}$ decays by $\beta^-$ emission to the ground state of $^{137}\text{Ba}$ (BR = 5.64%) and via the 662 keV isomeric level of $^{137}\text{Ba}$ (BR = 94.36%) whose half-life is 2.55 min. . . . .	4-9
239	4.7	Floor plan of the GIF++ facility. When the facility downstream of the GIF++ takes electron beam, a beam pipe is installed along the beam line (z-axis). The irradiator can be displaced laterally (its center moves from $x = 0.65 \text{ m}$ to $2.15 \text{ m}$ ), to increase the distance to the beam pipe. . . . .	4-9
243	4.8	Simulated unattenuated current of photons in the xz plane (Figure 4.8a) and yz plane (Figure 4.8b) through the source at $x = 0.65 \text{ m}$ and $y = 0 \text{ m}$ . With angular correction filters, the current of 662 keV photons is made uniform in xy planes. . . . .	4-10
246	4.9	Description of the RPC setup. Dimensions are given in mm. A tent containing RPCs is placed at 1720 mm from the source container. The source is situated in the center of the container. RE-4-2-BARC-161 chamber is 160 mm inside the tent. This way, the distance between the source and the chambers plan is 2060 mm. Figure 4.9a provides a side view of the setup in the xz plane while Figure 4.9b shows a top view in the yz plane. . . . .	4-11
252	4.10	RE-4-2-BARC-161 chamber is inside the tent as described in Figure 4.9. In the top right, the two scintillators used as trigger can be seen. This trigger system has an inclination of $10^\circ$ relative to horizontal and is placed above half-partition B2 of the RPCs. PMT electronics are shielded thanks to lead blocks placed in order to protect them without stopping photons from going through the scintillators and the chamber.	4-12
257	4.11	Hit distributions over all 3 partitions of RE-4-2-BARC-161 chamber is showed on these plots. Top, middle and bottom figures respectively correspond to partitions A, B, and C. These plots show that some events still occur in other half-partitions than B2, which corresponds to strips 49 to 64, in front of which the trigger is placed, contributing to the inefficiency of detection of cosmic muons. In the case of partitions A and C, the very low amount of data can be interpreted as noise. On the other hand, it is clear that a little portion of muons reach the half-partition B1, corresponding to strips 33 to 48. . . . .	4-13

265	4.12 Results are derived from data taken on half-partition B2 only. On the 18 <sup>th</sup> of June 266 data has been taken on chamber RE-2-BARC-161 at building 904 (Prevessin 267 Site) with cosmic muons providing us a reference efficiency plateau of $(97.54 \pm$ 268 $0.15)\%$ represented by a black curve. A similar measurement has been done at GIF 269 on the 21 <sup>st</sup> of July with the same chamber giving a plateau of $(78.52 \pm 0.94)\%$ 270 represented by a red curve. . . . .	4-14
271	4.13 Representation of the layout used for the simulations of the test setup. The RPC is 272 represented as a yellow trapezoid while the two scintillators as blue cuboids looking 273 at the sky. A green plane corresponds to the muon generation plane within the sim- 274 ulation. Figure 4.9a shows a global view of the simulated setup. Figure 4.9b shows 275 a zommed view that allows to see the 2 scintillators as well as the full RPC plane. . . . .	4-15
276	4.14 $\gamma$ flux $F(D)$ is plot using values from table 4.1. As expected, the plot shows similar 277 attenuation behaviours with increasing distance for each absorption factors. . . . .	4-18
278	4.15 Figure 4.15a shows the linear approximation fit done via formulae 4.7 on data from 279 table 4.2. Figure 4.15b shows a comparison of this model with the simulated flux 280 using a and b given in figure 4.15a in formulae 4.4 and the reference value $D_0 =$ 281 $50\text{cm}$ and the associated flux for each absorption factor $F_0^{ABS}$ from table 4.1 . . . . .	4-20
282	4.16 Dose measurements has been done in a plane corresponding to the tents front side. 283 This plan is 1900 mm away from the source. As explained in the first chapter, a 284 lens-shaped lead filter provides a uniform photon flux in the vertical plan orthogonal 285 to the beam direction. If the second line of measured fluxes is not taken into account 286 because of lower values due to experimental equipments in the way between the 287 source and the tent, the uniformity of the flux is well showed by the results. . . . .	4-22
288	4.17 . . . . .	4-23
289	4.18 Evolution of the maximum efficiency for RE2 (4.18a) and RE4 (4.18b) chambers 290 with increasing extrapolated $\gamma$ rate per unit area at working point. Both irradiated 291 (blue) and non irradiated (red) chambers are shown. . . . .	4-25
292	4.19 Evolution of the working point for RE2 (4.19a) and RE4 (4.19b) with increasing 293 extrapolated $\gamma$ rate per unit area at working point. Both irradiated (blue) and non 294 irradiated (red) chambers are shown. . . . .	4-25
295	4.20 Evolution of the maximum efficiency at HL-LHC conditions, i.e. a background hit 296 rate per unit area of $300\text{Hz/cm}^2$ , with increasing integrated charge for RE2 (4.20a) 297 and RE4 (4.20b) detectors. Both irradiated (blue) and non irradiated (red) chambers 298 are shown. The integrated charge for non irradiated detectors is recorded during test 299 beam periods and stays small with respect to the charge accumulated in irradiated 300 chambers. . . . .	4-26
301	4.21 Comparison of the efficiency sigmoid before (triangles) and after (circles) irradiation 302 for RE2 (4.21a) and RE4 (4.21b) detectors. Both irradiated (blue) and non irradiated 303 (red) chambers are shown. . . . .	4-26
304	4.22 Evolution of the Bakelite resistivity for RE2 (4.22a) and RE4 (4.22b) detectors. Both 305 irradiated (blue) and non irradiated (red) chambers are shown. . . . .	4-27
306	4.23 Evolution of the noise rate per unit area for the irradiated chamber RE2-2-BARC-9 307 only. . . . .	4-27
308	A.1 (A.1a) View of the front panel of a V1190A TDC module [58]. (A.1b) View of the 309 front panel of a V1718 Bridge module [59]. (A.1c) View of the front panel of a 6U 310 6021 VME crate [60]. . . . .	A-3
311	A.2 Module V1190A <i>Trigger Matching Mode</i> timing diagram [58]. . . . .	A-4

- 312     A.3 Structure of the ROOT output file generated by the DAQ. The 5 branches (`EventNumber`,  
 313       `number_of_hits`, `Quality_flag`, `TDC_channel` and `TDC_TimeStamp`) are visible on  
 314       the left panel of the ROOT browser. On the right panel is visible the histogram cor-  
 315       responding to the variable `nHits`. In this specific example, there were approximately  
 316       50k events recorded to measure the gamma irradiation rate on the detectors. Each  
 317       event is stored as a single entry in the `TTree`. . . . . A-10
- 318     A.4 The effect of the quality flag is explained by presenting the content of `TBranch`  
 319       `number_of_hits` of a data file without `Quality_flag` in Figure A.4a and the con-  
 320       tent of the same `TBranch` for data corresponding to a `Quality_flag` where all TDCs  
 321       were labelled as `GOOD` in Figure A.4b taken with similar conditions. It can be noted  
 322       that the number of entries in Figure A.4b is slightly lower then in Figure A.4a due  
 323       to the excluded events. . . . . A-12
- 324     A.5 Using the same data as previously showed in Figure A.4, the effect of the quality  
 325       flag is explained by presenting the reconstructed hit multiplicity of a data file with-  
 326       out `Quality_flag` in Figure A.5a and the reconstructed content of the same RPC  
 327       partition for data corresponding to a `Quality_flag` where all TDCs were labelled as  
 328       `GOOD` in Figure A.5b taken with similar conditions. The artificial high content of bin  
 329       0 is completely suppressed. . . . . A-12
- 330     A.6 WebDCS DAQ scan page. On this page, shifters need to choose the type of scan  
 331       (Rate, Efficiency or Noise Reference scan), the gamma source configuration at the  
 332       moment of data taking, the beam configuration, and the trigger mode. These in-  
 333       formation will be stored in the DAQ ROOT output. Are also given the minimal  
 334       measurement time and waiting time after ramping up of the detectors is over before  
 335       starting the data acquisition. Then, the list of HV points to scan and the number of  
 336       triggers for each run of the scan are given in the table underneath. . . . . A-14
- 337     B.1 Example of expected hit time distributions in the cases of efficiency (Figure B.1a)  
 338       and noise/gamma rate per unit area (Figure B.1b) measurements as extracted from  
 339       the raw ROOT files. The unit along the x-axis corresponds to ns. The fact that  
 340       "the" muon peak is not well defined in Figure B.1a is due to the contribution of all  
 341       the RPCs being tested at the same time that don't necessarily have the same signal  
 342       arrival time. Each individual peak can have an offset with the ones of other detectors.  
 343       The inconsistancy in the first 100 ns of both time distributions is an artefact of the  
 344       TDCs and are systematically rejected during the analysis. . . . . B-16
- 345     B.2 The effect of the quality flag is explained by presenting the reconstructed hit multi-  
 346       plicity of a data file without `Quality_flag`. The artificial high content of bin 0 is the  
 347       effect of corrupted data. . . . . B-19
- 348     B.3 Display of the masking tool page on the webDCS. The window on the left allows the  
 349       shifter to edit `ChannelsMapping.csv`. To mask a channel, it only is needed to set the  
 350       3rd field corresponding to the strip to mask to 0. It is not necessary for older mapping  
 351       file formats to add a 1 for each strip that is not masked as the code is versatile and  
 352       the default behaviour is to consider missing mask fields as active strips. The effect  
 353       of the mask is directly visible for noisy channels as the corresponding bin turns red.  
 354       The global effect of masking strips will be an update of the rate value showed on the  
 355       histogram that will take into consideration the rejected channels. . . . . B-24



# List of Tables

356

357	3.1	Properties of the most used electrode materials for RPCs. . . . .	3-4
358	4.1	Total photon flux ( $E\gamma \leq 662$ keV) with statistical error predicted considering a $^{137}\text{Cs}$ activity of 740 GBq at different values of the distance $D$ to the source along the x-axis of irradiation field [55]. . . . .	4-17
359	4.2	Correction factor $c$ is computed thanks to formulae 4.5 taking as reference $D_0 = 50$ cm and the associated flux $F_0^{ABS}$ for each absorption factor available in table 4.1.	4-19
360	4.3	The data at $D_0$ in 1997 is taken from [55]. In a second step, using Equations 4.8 and 4.9, the flux at $D$ can be estimated in 1997. Then, taking into account the attenuation of the source activity, the flux at $D$ can be estimated at the time of the tests in GIF in 2014. Finally, assuming a sensitivity of the RPC to $\gamma$ $s = 2 \cdot 10^{-3}$ , an estimation of the hit rate per unit area is obtained. . . . .	4-21
361	A.1	Inter-process communication cycles in between the webDCS and the DAQ through file string signals. . . . .	A-19
362			
363			
364			
365			
366			
367			
368			
369			



370

## List of Acronyms

371

### List of Acronyms

372

373

#### A

374

375

376

AFL

Almost Full Level

377

378

#### B

379

380

381 BARC

Bhabha Atomic Research Centre

382 BLT

Block Transfer

383 BNL

Brookhaven National Laboratory

384 BR

Branching Ratio

385

386

#### C

387

388

389 CAEN

Costruzioni Apparecchiature Elettroniche Nucleari S.p.A.

390 CERN

European Organization for Nuclear Research

391 CFD

Constant Fraction Discriminator

392 CMS

Compact Muon Solenoid

393 CSC

Cathode Strip Chamber

394

395

#### D

396

397

398 DAQ

Data Acquisition

399 DCS

Detector Control Software

400 DQM

Data Quality Monitoring

401 DT

Drift Tube

402

403

#### F

404

405

406	<b>FEE</b>	Front-End Electronics
407	<b>FEB</b>	Front-End Board
408		
409		
410	<b>G</b>	
411		
412	<b>GE-</b>	Find a good description
413	<b>GE1/1</b>	Find a good description
414	<b>GE2/1</b>	Find a good description
415	<b>GEANT</b>	GEometry ANd Tracking - a series of software toolkit platforms developed by CERN
416		
417	<b>GEM</b>	Gas Electron Multiplier
418	<b>GIF</b>	Gamma Irradiation Facility
419	<b>GIF++</b>	new Gamma Irradiation Facility
420		
421		
422	<b>H</b>	
423		
424	<b>HL-LHC</b>	High Luminosity LHC
425	<b>HPL</b>	High-pressure laminate
426	<b>HV</b>	High Voltage
427		
428		
429	<b>I</b>	
430		
431	<b>iRPC</b>	improved RPC
432	<b>IRQ</b>	Interrupt Request
433	<b>ISR</b>	Intersecting Storage Rings
434		
435		
436	<b>L</b>	
437		
438	<b>LEIR</b>	Low Energy Ion Ring
439	<b>LEP</b>	Large Electron-Positron
440	<b>LHC</b>	Large Hadron Collider
441	<b>LS1</b>	First Long Shutdown
442	<b>LS3</b>	Third Long Shutdown
443	<b>LV</b>	Low Voltage
444	<b>LVDS</b>	Low-Voltage Differential Signaling
445		
446		
447	<b>M</b>	
448		
449	<b>MC</b>	Monte Carlo

450	MCNP	Monte Carlo N-Particle
451	ME-/	Find good description
452	ME0	Find good description
453	MRPC	Multigap RPC
454		
455		
456	<b>N</b>	
457		
458	NIM	Nuclear Instrumentation Module logic signals
459		
460		
461	<b>P</b>	
462		
463	PS	Proton Synchrotron
464	PMT	PhotoMultiplier Tube
465		
466		
467	<b>Q</b>	
468		
469	QCD	Quantum Chromodynamics
470	QED	Quantum Electrodynamics
471		
472		
473	<b>R</b>	
474		
475	RE-/	Find a good description
476	RE2/2	Find a good description
477	RE3/1	Find a good description
478	RE3/2	Find a good description
479	RE4/1	Find a good description
480	RE4/2	Find a good description
481	RE4/3	Find a good description
482	RMS	Root Mean Square
483	ROOT	a framework for data processing born at CERN
484	RPC	Resistive Plate Chamber
485		
486		
487	<b>S</b>	
488		
489	SC	Synchrocyclotron
490	SLAC	Stanford Linear Accelerator Center
491	SM	Standard Model
492	SPS	Super Proton Synchrotron

493

494

495

496

**T**

497 TDC Time-to-Digital Converter  
498 ToF Time-of-flight

499

500

501

502

**W**

503 webDCS Web Detector Control System

# 1

## Introduction

504

505

<sup>506</sup> **1.1 A story of High Energy Physics**

<sup>507</sup> **1.2 Organisation of this study**



# 2

508

509

## Investigating the TeV scale

510 „We may regard the present state of the universe as the effect of the  
511 past and the cause of the future. An intellect which at any given mo-  
512 ment knew all of the forces that animate nature and the mutual posi-  
513 tions of the beings that compose it, if this intellect were vast enough  
514 to submit the data to analysis, could condense into a single formula  
515 the movement of the greatest bodies of the universe and that of the  
516 lightest atom; for such an intellect nothing could be uncertain and  
517 the future just like the past would be present before its eyes.”

518

519 - Pierre Simon de Laplace, *A Philosophical Essay on Probabilities*, 1814

Throughout history, physics experiment became more and more powerful in order to investigate finer details of nature and helped understanding the elementary blocks of matter and the fundamental interactions that bond them in the microscopic world. Nowadays, the Standard Model (SM) of particle physics is the most accurate theory designed to explain the behaviour of particles and was able to make very precise predictions that are constantly verified, although some hints of new physics are visible as bricks are still missing to have a global comprehension of the Universe.

To highlight the limits of the SM and test the different alternative theories, ever more powerful machines are needed. This is in this context that the Large Hadron Collider (LHC) has been thought and built to accelerate and collide particles at energies exceeding anything that had been done before. Higher collision energies and high pile-up imply the use of enormous detectors to measure the properties of the interaction products. The Compact Muon Solenoid (CMS) is a multipurpose experiment that have been designed to study the proton-proton collisions of the LHC and give answers on various high energy physics scenari. Nevertheless, the luminosity delivered by the collider will in the future be increased to levels beyond the original plans to improve its discovery potential giving no choice to experiments such as CMS to upgrade their technologies to cope with the increased radiation levels and detection rates.

## 2.1 The Standard Model of Particle Physics

In this early 21<sup>st</sup> century it is now widely accepted that matter is made of elementary blocks referred to as *elementary particles*. The physics theory that classifies and describes the best the behaviour and interaction of such elementary particles is the so called Standard Model that formalizes 3 of the 4 fundamental interactions (electromagnetic, weak and strong interactions). It's development took place during the 20<sup>th</sup> century thanks to a strong collaboration in between the theoretical and experimental physicists.

### 2.1.1 A history of particle physics

The idea that nature is composed of elementary bricks, called *atomism*, is not contemporary as it was already discussed by Indian or Greek philosophers during antiquity. In Greece, atomism has been rejected by Aristotelianism as the existance of *atoms* would imply the existance of a void that would violate the physical principles of Aristotle philosophy. Aristotelianism has been considered as a reference in the european area until the 15<sup>th</sup> century and the italian *Rinascimento* where antic text and history started to be more deeply studied. The re-discovery of Platon's philosophy would allow to open the door to alternative theories and give a new approach to natural sciences where experimentation would become central. A new era of knowledge was starting. By the begining of the 17<sup>th</sup> century, atomism was re-discovered by philosophers and the very first attempt to estimate an *atom* size would be provided by Magnetus in 1646. Although his *atoms* correspond to what would nowadays be called *molecules*, Magnetus achieved feats by calculating that the number of molecules in a grain of incense would be of the order of  $10^{18}$  simply by considering the time necessary to smell it everywhere in a large church after the stick was lit on. It is now known that this number only falls short by 1 order of magnitude.

An alternative philosophy to atomism popularized by Descartes was corpuscularianism. Built on ever divisible corpuscles, contrary to atoms, it's principles would be mainly used by alchemists like Newton who would later develop a corpuscular theory of light. Boyle would combine together ideas of both atomism or corpuscularianism leading to mechanical philosophy. The 18<sup>th</sup> century have

seen the development of engineering providing philosophical thought experiments with repeatable demonstration and a new point of view to explain the composition of matter and Lavoisier would greatly contribute to chemistry and atomism by publishing in 1789 a list of 33 chemical elements corresponding to what is now called *atoms*. In the early 19<sup>th</sup> century Dalton would summarize the knowledge on composition of matter and Fraunhofer would invent the spectrometer and discover the spectral lines. The rise of atomic physics, chemistry and mathematical formalism would unravel the different atomic elements and ultimately, the 20<sup>th</sup> century would see the very first sub-atomic particles.

**Discovery of the inner structure of the atom**

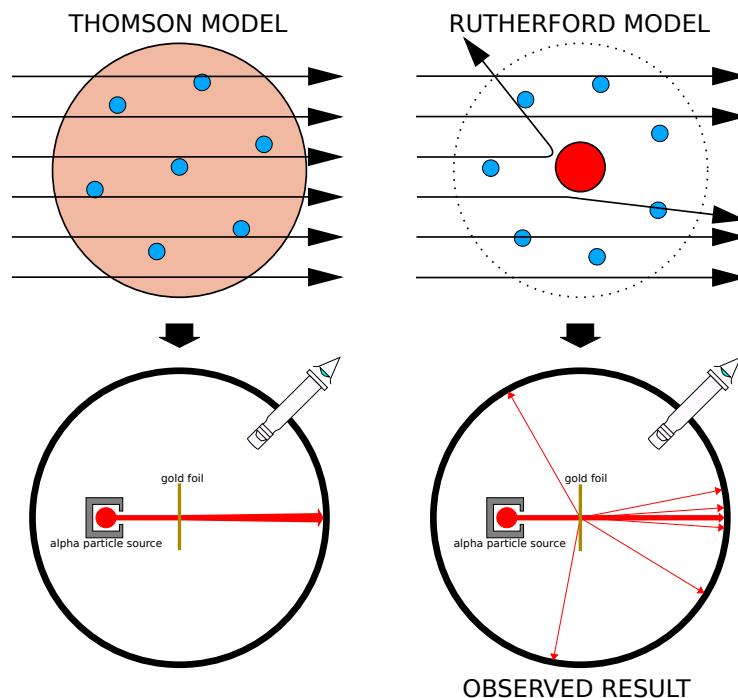


Figure 2.1: Through the gold foil experiment Rutherford could show that most of the mass of atoms was contained in a positively charged nucleus and could then propose a more accurate atomic model than that of Thomson.

The negative *electron* would be the first to be discovered in 1897 by Thomson after 3 decades of research on cathode rays by proving that the electrification observed in an electroscope, as reported by Perrin, was due to the rays themselves and that they had to be composed of electrically charged particles. In 1900, Becquerel would show the *beta rays* emitted by radium had the same charge over mass ratio than what measured by Thomson for cathode rays, pointing to electrons as a constituent of atoms. In 1907, Rutherford and Royds showed that *alpha* particles, once captured in a tube and subjected to an electric spark causing an electron avalanche, where helium ions as they could combine with 2 electrons to form a  ${}^4\text{He}$ . This discovery was directly followed by the constraint of the atom structure in 1909 through the gold foil experiment in which the deflection angle of alpha particles fired at a very thin gold foil was measured and highlighted atoms where mainly empty with

581 nearly all its mass contained into a tiny positively charged *nucleus*. With these two observations,  
 582 he could formulate the Rutherford model of the atom in 1911, shown together with the Thomson  
 583 plum pudding model in Figure 2.1. The link in between atomic number and number of positive and  
 584 negative charges contained into the atoms would fast be understood and the different kind of element  
 585 transmutation appeared to be purely nuclear processes making clear that the electromagnetic nature  
 586 of chemical transformation could not possibly change nuclei. Thus a new branch in physics appeared  
 587 to study nuclei exclusively: the nuclear physics.

588 Moreover, in 1913 quantum physics would be introduced into the atomic model by Bohr based  
 589 on the assumptions of Plank to explain spectral lines, and other observed quantum effects. The same  
 590 year, Moseley would confirm Bohr's model and Debye would extend it by introducing elliptical  
 591 orbits.

592 By studying alpha emission and the product of their interaction with nitrogen gas, Rutherford  
 593 reported in 1919 the very first nuclear reaction leading to the discovery that the hydrogen nucleus was  
 594 composed of a single positively charged particle that was later baptised *proton*. This idea came from  
 595 1815 Prout's hypothesis proposing that all atoms are composed of "*protyles*" (i.e. hydrogen atoms).  
 596 By using scintillation detectors, Rutherford could highlight typical hydrogen nuclei signature and  
 597 understand that the impact of alpha particles with nitrogen would knock out an hydrogen nucleus  
 598 and produce an oxygen 17, as explicated in Formula 2.1 and would then postulate that protons are  
 599 building bricks of all elements.



600 With this assumption and the discovery of isotopes together with Aston, elements with identical  
 601 atomic number but different masses, Rutherford would propose that all elements' nuclei but hydrogen's are composed of both charged particles, protons, and of chargeless particles, which he called  
 602 *neutrons*, and that these neutral particles would help maintaining nuclei as one, as charged protons  
 603 were likely to electrostatically repulse each other, and introduced the idea of a new force, a *nuclear*  
 604 force. Though the first idea concerning neutrons was a bond state of protons and electrons as it was  
 605 known that the beta decay, emitting electrons, was taking place in the nucleus, it was then showed  
 606 that such a model would hardly be possible due to Heisenberg's uncertainty principle and by the  
 607 recently measured *spin* of both protons and electrons. The spin, discovered through the study of  
 608 the emission spectrum of alkali metals, would be understood as a "two-valued quantum degree of  
 609 freedom" and formalized by Pauli and extended by Dirac, to take the relativist case into account.  
 610 Measured to be  $\frac{1}{2}\hbar$  for both, it was impossible to arrange an odd number of half integer spins and  
 611 obtain a global nucleus spin that would be integer. Finally, in 1932, following the discovery of a new  
 612 neutral radiation, Chadwick could discover the neutron as an uncharged particle with a mass similar  
 613 to that of the proton whose half integer spin would reveal to be the solution to explain the nuclear  
 614 spin.  
 615

### 616 Development of the Quantum Electrodynamics

617 Historically, the development of the quantum theory revolved around the question of emission and  
 618 absorption of discrete amount of energy through light. Einstein used the initial intuition of Plank  
 619 about the black-body radiation to develop in 1905 a model to explain the photoelectric effect in  
 620 which light was described by discrete quanta now called *photons*. For this model, Einstein introduced  
 621 the concept of wave-particle duality as classical theory was not able to describe the phenomenon.  
 622 With the new understanding of atoms and of their structure, classical theories also proved unable

623 to explain atoms stability. Indeed, using classical mechanics, electrons orbiting around a nucleus  
 624 should radiate an energy proportional to their angular momentum and thus lose energy through  
 625 time and the spectrum of energy emission should then be continuous, but it was known since the  
 626 19<sup>th</sup> century and the discovery of spectral lines that the emission spectrum of material was discrete.

627 This was Bohr who first suggested that a quantum description of the atom was necessary in 1913.  
 628 Using the correspondence principle stating that at large enough numbers the quantum calculations  
 629 should give the same results than the classical theory, he proposed the very first quantum model  
 630 of the hydrogen atom explaining the line spectrum by introducing the principal quantum number  
 631  $n$  describing the electron shell. This model would then be improved by Sommerfeld that would  
 632 quantize the z-component of the angular momentum, leading to the second and third quantum  
 633 numbers, or azimuthal and magnetic quantum number,  $l$  and  $m$  defining for the second the orbital  
 634 angular momentum of the electrons on their shells and thus, the shape of the orbital, and for the third  
 635 the available orbital on the subshell for each electron. Nevertheless, although the model was not only  
 636 limited to spherical orbitals anymore, making the atom more realistic, the Zeeman effect couldn't be  
 637 completely explained by just using  $n$ ,  $l$  and  $m$ . A solution would be brought after the discovery of  
 638 Pauli in 1924, as Uhlenbeck, Goudsmit, and Kramers proposed in 1925 the idea of intrinsic rotation  
 639 of the electron, introducing a new angular momentum vector associated to the particle itself, and  
 640 not to the orbital, and associated to a new quantic number  $s$ , the *spin* projection quantum number  
 641 explaining the lift of degeneracy to an even number of energy levels.

642 The introduction of the *spin* happened 1 year after another attempt of improvement of the theory  
 643 was made by De Broglie in his PhD thesis. The original formulation of the quantum theory only  
 644 considered photons as energy quanta behaving as both waves and particles. De Broglie proposed  
 645 that all matter are described by waves and that their momentum is proportional to the oscillation of  
 646 quantized electromagnetic field oscillators. This interpretation was able to reproduce the previous  
 647 version of the quantum energy levels by showing that the quantum condition involves an integer  
 648 multiple of  $2\pi$ , as shown by Formula 2.2.

$$p = \hbar k \Leftrightarrow \int pdx = \hbar \int kdx = 2\pi\hbar n \quad (2.2)$$

649 Although the intuition of De Broglie about the wave-particle duality of all matter, his interpreta-  
 650 tion was semiclassical and it's in 1926 that the first fully quantum mechanical wave-equation would  
 651 be introduced by Schrödinger to describe electron-like particles, reproducing the previous semiclas-  
 652 sical formulation without inconsistencies. This complexe equation describes the evolution of the  
 653 wave function  $\Psi$  of the quantum system, defined by its position vector  $\mathbf{r}$  and time  $t$  as an energy  
 654 conservation law, in which the hamiltonian of the system  $\hat{H}$  is explicit, by solving the Equation 2.3.

$$i\hbar \frac{\partial}{\partial t} |\Psi(\mathbf{r}, t)\rangle = \hat{H} |\Psi(\mathbf{r}, t)\rangle \quad (2.3)$$

655 In 1927, Dirac would go further in his paper about emission and absorption of radiation by  
 656 proposing a second quantization not only of the physical process at play but also of the electromag-  
 657 netic field, providing the ingredients to the first formulation of *Quantum Electrodynamics (QED)*  
 658 and the description of photon emission by electrons dropping into a lower energy state in which the  
 659 final number of particles is different than the initial one. To complete this model to the many-body  
 660 wave functions of identical particles, Jordan included creation and annihilation operators for fields  
 661 obeying Fermi-Dirac statistics leading to a model describing particles that would be referred to as  
 662 *fermions*. Nevertheless, in order to properly treat electromagnetism, the incorporation of the relativ-

ity theory developed by Einstein. Including gravity into quantum physics still is a challenge nowadays, but in 1928 Pauli and Jordan would show that special relativity's coordinate transformations could be applied to quantum fields as the field commutators were Lorentz invariant. Finally derived the same year, the Dirac equation, shown as Equation 2.4, similarly to Schrödinger's equation, is a single-particle equation but it incorporates special relativity in addition to quantum mechanics rules. It features the  $4 \times 4$  gamma matrices  $\gamma^\mu$  built using  $2 \times 2$  Pauli matrices and unitary matrix, the 4-gradient  $\partial_\mu$ , the rest mass  $m$  of any half integer spin massive particle described by the wave function  $\psi(x, t)$ , also called a Dirac spinor, and the speed of light  $c$ . In addition to perfectly reproduce the results obtained with quantum mechanics so far, it also provided with *negative-energy solutions* that would later be interpreted as a new form of matter, *antimatter* and give a theoretical justification to the Pauli equation that was phenomenologically constructed to account for the spin as in the non-relativistic limit, the Dirac equation is similar.

$$i\hbar\gamma^\mu\partial_\mu\psi - mc\psi = 0 \quad (2.4)$$

The successes of the QED was soon followed with theoretical problems as computations of any physical process involving photons and charged particles were showed to be only reliable at the first order of perturbation theory. At higher order of the theory, divergent contributions were appearing giving nonsensical results. Only two effects were contributing to these infinities.

- The self-energy of the electron (or positron), the energy that the particle has due its own interaction with its environment.
- The vacuum polarization, virtual electron–positron pairs produced by a background electromagnetic field in the vacuum which is not an "empty" space. These virtual pairs affect the charge and current distributions generated by the original electromagnetic field.

Solving this apparent problem was done by carefully defining the concepts of each observables, for example mass or charge, as these quantities are understood within the context of a non-interacting field equation, and that from the experiment point of view, they are abstractions as what is measured are "renormalized observables" shifted from there "bare" value by the interaction taking place in the measuring process. The infinities needed to be connected to corrections of mass and charge as those are fixed to finite values by experiment. This was the intuition of Bethe in 1947 who successfully computed the effect of such *renormalization* in the non-relativist case. Fully covariant formulations of QED including renormalization was achieved by 1949 by Tomonaga, Schwinger, Feynman and Dyson and Feynman is now famous for his association of diagrams to the term of the scattering matrix, greatly simplifying the representation and computation of interactions as the diagrams directly corresponded the measurable physical processes and would then be used in every quantum field theories. With the resolution of infinities, QED had mostly reached its final form, being still today the most accurate physical theory and would serve as a model to build all other quantum field theories.

### Development of the quark model and Quantum Chromodynamics

To explain the nuclear force that holds *nucleons* (protons and neutrons) together, Yukawa theoretically proposed in 1934 the existence of a force carrier called *meson* due to it's predicted mass in the range in between the electron and nucleon masses. Discovered in 1936 by Anderson and Neddermeyer, and confirmed using bubble chambers in 1937 by Street and Stevenson, a first meson

703 candidate was observed in the decay products of cosmic rays. Assuming it had the same electric  
 704 charge than electrons and protons, this particle was observed to have a curvature due to magnetic  
 705 field that was sharper than protons but smoother than electrons resulting in a mass in between that  
 706 of electrons and protons. But its properties were not compatible with Yukawa's theory, which was  
 707 emphasized by the discovery of a new candidate in 1947, again in cosmic ray products using photo-  
 708 graphic emulsions.

709 This new candidate, although it had a similar mass than the already believed *meson*, would rather  
 710 decay into it. For distinction, the first candidate would then be renamed "*mu meson*" when the second  
 711 would be the "*pi meson*". The *mu meson* was behaving like a heavy electron and didn't participate  
 712 in the strong interaction whereas the pion was believed to be the carrier of the nuclear interaction.  
 713 This lead to classify the *mu* in a new category of particles called *leptons* together with the electron  
 714 that shared similar properties and *the neutrino*, and be renamed *muon*. The *pi meson* was finally  
 715 found to be a triplet of particles: a positively charged, a negatively charged, and a neutral particle.  
 716 The neutral *pi meson* has been more difficult to identify as it wouldn't leave tracks on emulsions nor  
 717 on bubble chambers and needed to be studied via it's decay products. It was ultimately identified in  
 718 University of California's cyclotron in 1950 through the observation of its decay into 2 photons.

719 Also discovered in 1947 but in cloud chamber photographs, the *K meson* as also been an im-  
 720 portant step towards the establishment of the Standard Model. A triplet of particle, 2 charged and a  
 721 neutral, with a mass roughly half that of a proton, were reported. These particles were baptised *K*  
 722 *meson* in contrast to the "light" *pi* and *mu* "L-mesons". The particularity of the *K* were there very  
 723 slow decays with a typical lifetime of the order of  $10^{-10}$ s much greater than the  $10^{-23}$ s of *pi*-proton  
 724 reactions. The concept of *strangeness*, a new quantum number was then introduced by Pais as an  
 725 attempt to explain this phenomenon as *strange* particles appeared as a pair production of a strange  
 726 and anti-strange particle.

727 With the development of synchrotrons, the particle *zoo* would grow to several dozens during the  
 728 1950s as higher energies were reachable through acceleration. In 1961, a first classification system,  
 729 called Eightfold Way, was proposed by Gell-Mann and finding its roots in the Gell-Mann–Nishijima  
 730 formula, which relates the electric charge  $Q$ , the third component of the isospin  $I_3$ , the *baryon*  
 731 number  $B$  and the strangeness  $S$ , as explicitated in Formula 2.5. The isospin was a quantum number  
 732 introduced in 1932 to explain symmetries of the newly discovered neutron using representation  
 733 theory of SU(2). The baryon number, was introduced by Nishijima as a quantum number for baryons,  
 734 i.e. particles of the same family as nucleons. The mesons were classified in an octet and baryons of  
 735 spin  $\pm \frac{1}{2}$  and  $\pm \frac{3}{2}$  were respectively classified into an octet and a decuplet, as shown in Figure 2.2. To  
 736 complete the baryon decuplet, Gell-Mann predicted the existance of baryon  $\Omega^-$  which would later  
 737 be discovered in 1964.

$$Q = I_3 + \frac{1}{2}(B + S) \quad (2.5)$$

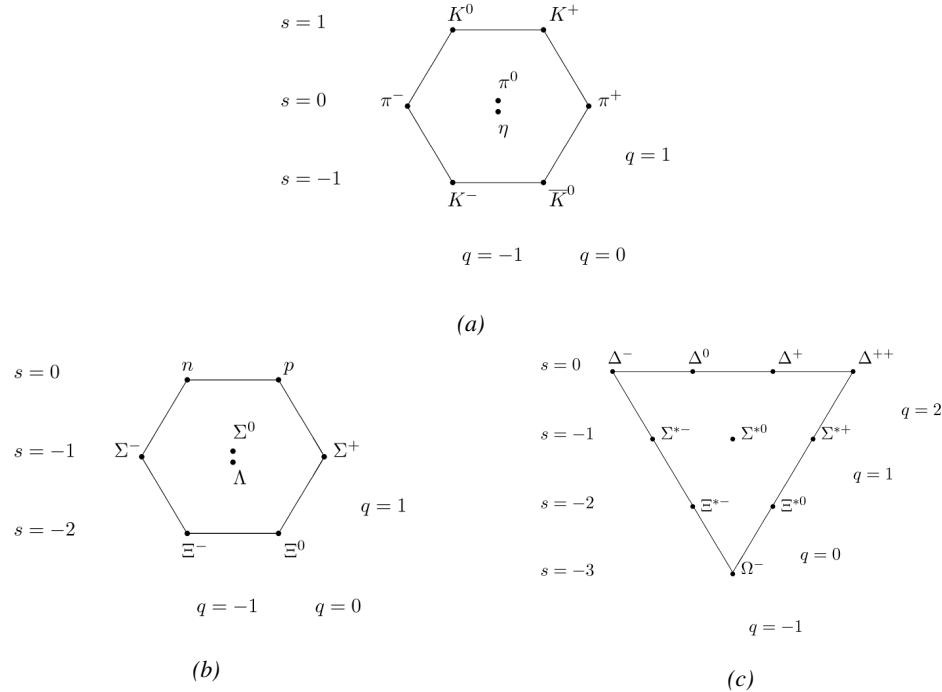


Figure 2.2: Figure 2.2a: Meson octet. Figure 2.2b: Baryon octet. Figure 2.2c: Baryon decuplet.

Strong of this classification using an SU(3) flavor symmetry, Gell-Mann, and independently Zweig, would propose a full theoretical model in which *hadrons* (strongly interacting particles, i.e. mesons and baryons) were not elementary particles anymore. They would rather be composed with 3 flavors of particles called *quarks* and there anti-particles. The 3 flavors were called *up*, *down* and *strange*. *Up* and *down* would be used to explain the nucleons and non-strange mesons, while *strange* would come into the composition of hadrons showing strangeness. *Up* and *down* flavors would be discovered in 1968 thanks to the deep inelastic scattering experiments conducted at the Stanford Linear Accelerator Center (SLAC), and *strange* could only be indirectly validated even though it provided a robust explanation to *kaon* (K) and *pion* ( $\pi$ ). However, in the decade following the Gell-Mann-Zweig quark model proposition, several improvement to the model were brought, first by Glashow and Bjorken the same year that predicted the existence of a fourth quark flavor, the *charm*, that would equalize the then known number of quarks and leptons and finally in 1973 by Kobayashi and Maskawa that would increase the number of quarks to 6 to explain the experimental observation of CP violation. These two quarks would be referred to as *top* and *bottom* for the first time in 1975. It's only after these additions to the quark model that finally the *charm* would be discovered in 1974 at both SLAC and Brookhaven National Laboratory (BNL). A meson where the *charm* was bound with an *anti-charm*, called  $J/\psi$ , would help convince the physics community of the validity of the model. The *top* would be discovered soon after in 1977 in Fermilab and indicate the existence of the *bottom* that would resist to discovery until Fermilab's experiments CDF and D $\emptyset$  in 1995 due to its very large mass and the energy needed to produce it.

As remarked by Struminsky, the original quark model proposal composed of 3 quarks should possess an additional quantum number due to mesons such as  $\Omega^-$  or  $\Delta^{++}$ . Indeed, these mesons are composed of 3 identical quarks, respectively 3 *strange* and *up* quark, with parallel spins, which

should be forbidden by the exclusion principle. Independentle, Greenberg and Han-Nambu proposed an additional SU(3) degree of freedom possessed by the quarks, that would later be refered to as *color charge gauge*, that could interact through *gluons*, the gauge boson octet corresponding to this degree of freedom. Nevertheless, as observing free quarks proved to be impossible, two visions of the quarks were argued mainly due to the failures to observe these particles free to prove their existence. On one side, Gell-Mann proposed to see quarks as mathematical construct instead of real particles, as they are always confined, implying that quantum field theory would not describe entirely the strong interaction. Opposed to this vision, Feynman on the contrary argued that quarks were real particles, that he would call *partons*, that should be described as all other particles by a distribution of position and momentum. The implications of quarks as point-like particles would be verified at SLAC and the concept of *color* would be added to the quark model in 1973 by Fritzsch and Leutwyler together with Gell-Mann to propose a description of the strong interaction through the theory of Quantum Chromodynamics (QCD). The discovery the same year of asymptotic freedom within the QCD by Groos, Politzer and Wilczek, allowed for very precise predictions thanks to the perturbation theory. Nowadays, the confinement of quarks is studied in experiments such as ALICE, through exploration of the quark-gluon plasma.

### 777      **The Weak interaction, spontaneous symmetry breaking, the Higgs mechanism and the Elec- 778      troweak unification**

779      The weak interaction is the process that causes radioactive decays. Thanks to the neutron discovery,  
 780      Fermi could explain in 1934 the beta radiations through the beta decay process in which the neutron  
 781      decays into a proton by emitting an electron. Though the missing energy observed during this  
 782      process triggered a huge debate about the apparent non conservation of energy, momentum and spin  
 783      of the process, Fermi, as Pauli before him, proposed that the missing energy was due to a neutral  
 784      not yet discovered particle that would then be baptised *neutrino*. The impossibility to detect such  
 785      a particle would leave some members of the scientific community sceptical, but hints of energy  
 786      conservation and of the existence of the neutrino were provided by measuring the energy spectrum  
 787      of electrons emitted through beta decay, as there was a strict limit on their energy. It's only 30 years  
 788      later in 1953 that it would be discovered by the team of Cowan and Reines using the principle of  
 789      inverse beta decay described through Formula 2.6. The experiment consisted in placing water tanks  
 790      sandwiched in between liquid scintillators near a nuclear reactor with an estimated neutrino flux of  
 791       $5 \times 10^{13} \text{ s}^{-1} \text{ cm}^{-2}$ . However, in order to explain the absence of some reactions in the experiment  
 792      of Cowan and Reines, and constraint the beta decay theory of Fermi and extend it to the case of  
 793      the muon, Konopinski and Mahmoud proposed in 1953 that the muon decay would eject a particle  
 794      similar to the neutrino and thus predicted the existence of a muon neutrino that would be different  
 795      than the one involved in the beta decay, related to the electron. With this, the idea of lepton number  
 796      would arise. The muon neutrino would successfully be detected in 1962 by Lederman, Schwartz and  
 797      Steinberger.

$$\bar{\nu} + p \rightarrow n + e^+ \quad (2.6)$$

798      The theory could not be valid though as the probability of interaction, called cross-section, would  
 799      have been increasing without bond with the square of the energy. Fermi assumed in a two vector  
 800      current coupling but Lee and Yang noted that an axial current could appear and would violate parity.  
 801      The experiment of Wu in 1956 would confirm the parity violation and Gamow and Teller would try to  
 802      account for it by describing Fermi's interaction through allowed (parity-violating) and superallowed

803 (parity-conserving) decays. But the success of QED as a quantum field theory would spark the  
804 development of such a theory to describe the weak interaction.

805 As previously discussed, the great success of QED was built on an underlying symmetry, interpreted  
806 as a gauge invariance so that the effect of the force is the same in all space-time coordinates,  
807 and of the possibility to renormalize it in order to absorb the infinities. Independently in 1958,  
808 Glashow, and Salam and Ward used 1957 Schwinger ideas about vector intermediary for the decay  
809 processes, could find a way to unite both the electromagnetic and weak interaction into a gauge  
810 theory involving 4 gauge bosons, 3 of which were massive and carried out the weak interaction and  
811 a massless boson carrying the electromagnetic interaction. Among the 3 massive bosons, 2 were  
812 charged and 1 was neutral, similarly to the previously theorized *pi meson* vector of the Yukawa  
813 model and all have a mass much greater than nucleons and thus a very short life time implying a  
814 finite very short range contrary to the contact interaction originally proposed by Fermi.

815 Breakthrough in other fields of physics contributed in giving theoretical support and interpretation  
816 to the unified electroweak theory. The stepping stone would be the use of spontaneous symmetry  
817 breaking that was inspired to Nambu at the end of the 1950s following the development of the BCS  
818 superconductivity mechanism in 1957. Cooper had shown that BCS pairs, pairs of electrons bound  
819 together at low temperature, could have lower energy than the Fermi energy and where responsible  
820 for superconductivity. This lead to the discovery of Goldstone-Nambu bosons as a result of the  
821 spontaneous breaking of the chiral symmetry in a theory describing nucleons and mesons developed  
822 by Nambu and Jona-Lasinio in 1961, and now understood as a low-energy approximation of  
823 QCD. Similarly to mechanism of energy gap appearance in superconductivity, the nucleon mass  
824 is suggested to the result of a self-energy of a fermion field and is studied through a four-fermion  
825 interaction in which, as a consequence of the symmetry, bound states of nucleon-antinucleon pairs  
826 appear and can be regarded as virtual pions. Though the symmetry is maintained in the equations,  
827 the ground state is not preserved. Goldstone would later the same year show that the bound states  
828 corresponds to spinless bosons with zero mass.

829 Although the model in itself didn't revolutionize particle physics, spontaneous symmetry breaking  
830 would be generalized to quantum field theories. As all fundamental interactions are described  
831 using gauge theories based on underlying symmetries, processes such as the chiral symmetry breaking  
832 would be introduced soon after the publication of Nambu and Jona-Lasinio. In 1962, Anderson,  
833 following an idea of Schwinger who suggested that zero-mass vector bosons were not necessarily  
834 required to describe the conservation of baryons contrary to the bosons emerging from chiral symmetry  
835 breaking, discussed the implications of spontaneous symmetry breaking in particles physics.  
836 A model was finally independently built in 1964 by Brout and Englert, Higgs, and Guralnik, Hagen,  
837 and Kibble, who discovered that combining an additional field into a gauge theory in order to break  
838 the symmetry, the resulting gauge bosons acquire a nonzero mass. Moreover, Higgs stated that this  
839 implied the existence of at least one new massive, i.e. self-interacting, scalar boson, that are now  
840 known as *Higgs bosons* corresponding to this additional field. The Higgs mechanism today specifically  
841 refers to the process through which the gauge bosons of the weak interaction acquire mass. In  
842 1968, Weinberg could point to the Higgs mechanism to integrate a Higgs field into a new version  
843 of the electroweak theory in which the spontaneous symmetry breaking mechanism of the Higgs  
844 field would explicitly explain the masses of the weak interaction gauge bosons and the zero-mass  
845 of photons.

846 **2.1.2 Construction and test of the model**

847 The Standard Model of particle physics was built in the middle of the 1970s after the experimental  
 848 confirmation of the existence of quarks. It is a model based on the assembly of the models previously  
 849 introduced and describing the fundamental interactions and their gauge bosons as well as the way  
 850 elementary particles interact with the fields associated with these force carriers. In this sense, the  
 851 development of QED and the unification of the electroweak interaction, of the Yukawa interaction  
 852 and of QCD, and of the Higgs mechanism made it possible to explain most of contemporary physics.  
 853 This picture of the SM is summarized through Figure 2.3.

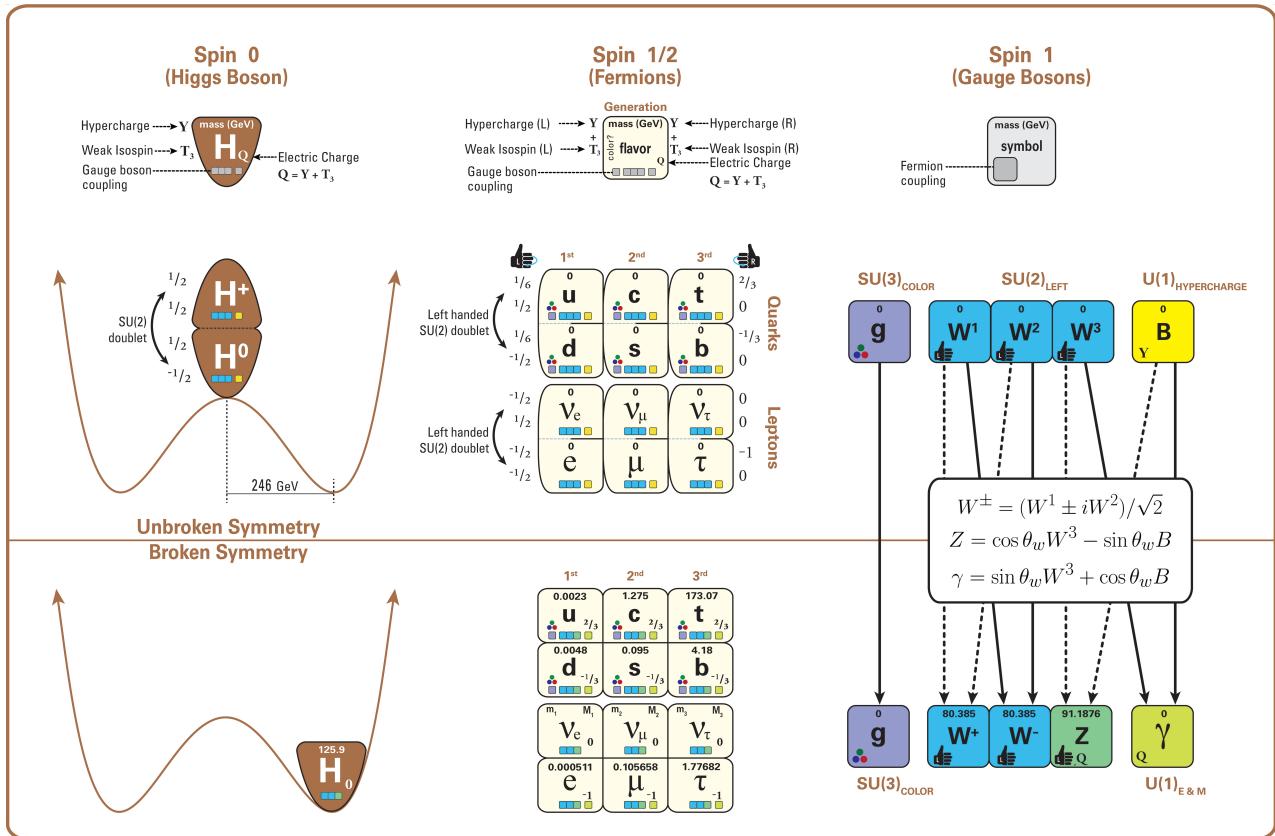


Figure 2.3: The elementary particles of the Standard Model are showed along with their names and properties. Their interactions with the strong, weak and electromagnetic forces have been explicated as well. in the left column, the scalar higgs boson is depicted, while the central is focused on the matter particles, the fermions, and the right on the force carriers, the gauge bosons. The role of the Higgs boson in electroweak symmetry breaking is highlighted, and the corresponding way properties of the various particles differ in the (high-energy) symmetric phase (top) and the (low-energy) broken-symmetry phase (bottom) are showed.

854 Nevertheless, with the state of experimental physics nowadays, hints of physics beyond the SM  
 855 have rised.

856    **2.1.3 Investigating the TeV scale**

857    **2.2 The Large Hadron Collider & the Compact Muon Solenoid**

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

872    **2.2.1 LHC, the most powerful particle accelerator**

873    The LHC has always been considered as an option to the future of CERN. At the moment of the  
874    construction of the LEP beneath the border between France and Switzerland, the tunnel was built in  
875    order to accomodate what would be a Large Hadron Collider with a dipole field of 10 T and a beam  
876    energy in between 8 and 9 TeV [7] directly followed in 1985 with the creation of a 'Working Group  
877    on the Scientific and Technological Future of CERN' to investigate such a collider [8]. The decision  
878    was finally taken almost 10 years later, in 1994, to construct the LHC in the LEP tunnel [9] and the  
879    approval of the 4 main experiments that would take place at the 4 interaction points would come in  
880    1997 [10] and 1998 [11]:

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

887    These large scale experiments, as well as the full CERN accelerator complex, are displayed on  
888    Figure 2.4.

## CERN's Accelerator Complex

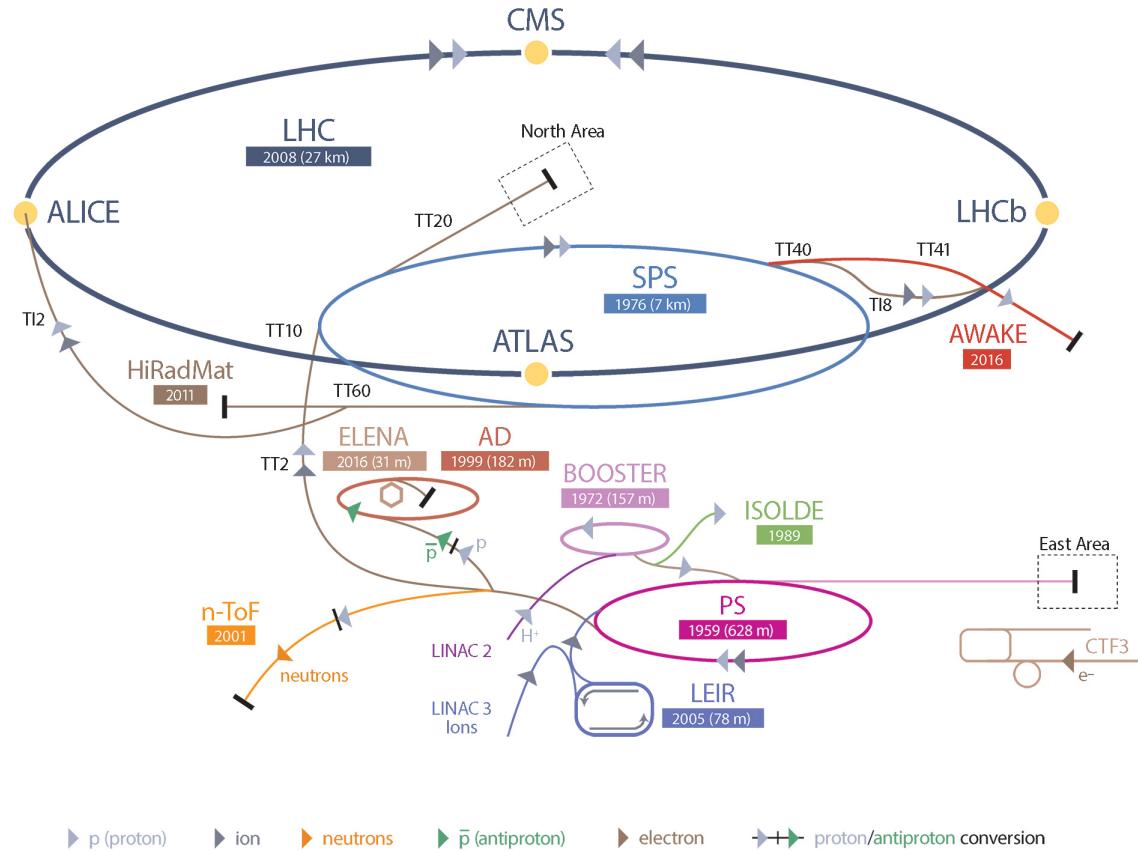


Figure 2.4: CERN accelerator complex.

The LHC is a 27 km long hadron collider and the most powerful accelerator used for particle physics since 2008 [16]. The LHC was originally designed to collide protons at a center-of-mass energy of 14 TeV and luminosity of  $10^{34} \text{ cm}^{-2}\text{s}^{-1}$ , as well as  $Pb$  ions at a center-of-mass energy of 2.8 TeV/A with a peak luminosity of  $10^{27} \text{ cm}^{-2}\text{s}^{-1}$ . Run 1 of LHC, when the center-of-mass energy only was half of the nominal LHC energy, was enough for both CMS and ATLAS to discover the Higgs boson [17] and for LHCb to discover pentaquarks [18] and confirm the existence of tetraquarks [19]. Nevertheless, after the Third Long Shutdown (LS3) (2024-2026), the accelerator will be in the so called High Luminosity LHC (HL-LHC) configuration [20], increasing its instantaneous luminosity to  $10^{35} \text{ cm}^{-2}\text{s}^{-1}$  for  $pp$  collisions and to  $4.5 \times 10^{27} \text{ cm}^{-2}\text{s}^{-1}$ , boosting the discovery potential of the LHC.

### 2.2.1.1 Particle acceleration

The LHC is the last of a long series of accelerating devices. Before being accelerated by the LHC, the particles need to pass through different acceleration stages. All these acceleration stages are

902 visible on Figure 2.4 and pictures of the accelerators are showed in Figure 2.5.

903

904       The story of accelerated protons at CERN starts with a bottle of hydrogen gas injected into the  
905 source chamber of the linear particle accelerator *LINAC 2* in which a strong electric field strips the  
906 electron off the hygroden molecules only to keep their nuclei, the protons. The cylindrical conductors,  
907 alternatively positively or negatively charged by radiofrequency cavities, accelerate protons by  
908 pushing them from behing and pulling them from the front and ultimately give them an energy of  
909 50 MeV, increasing their mass by 5% in the process.

910

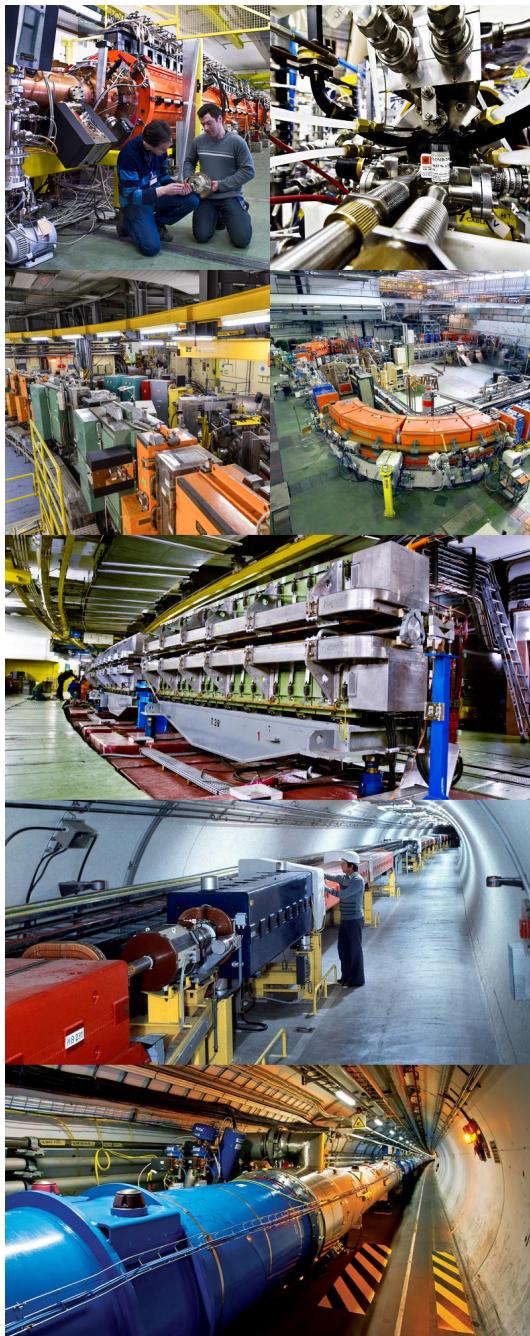
911       When exiting the LINAC 2, the protons are divided into 4 bunches and injected into the 4 su-  
912 perimposed synchrotron rings of the *Booster* where they are then accelerated to reach an energy of  
913 1.4 GeV before being injected into the *PS*. Before the Booster was operational in 1972, the protons  
914 were directly injected into the PS from the LINAC 2 but the low injection energy limited the amount  
915 of protons that could be accelerated at once by the PS. With the Booster, the PS accepts approxi-  
916 mately 100 times more particles.

917

918       The 4 proton bunches are thus sent as one to the PS where their energy eventually reaches  
919 26 GeV. Since the 70s, the main goal of this 628 m circumference synchrotron has been to sup-  
920 ply other machines with accelerated particles. Nowadays, not only the PS accelerates protons, it also  
921 accelerates heavy ions from the *Low Energy Ion Ring (LEIR)*. Indeed, the LHC experiments are not  
922 only designed to study *pp*-collisions but also *Pb*-collisions. Lead is first injected into the dedicated  
923 linear collider *LINAC 3*, that accelerate the ions using the same principle than LINAC 2. Electrons  
924 are striped off the lead ions all along the acceleration process and eventually, only bare nuclei are  
925 injected in the LEIR whose goal is to transform the long ion pulses received into short dense bunches  
926 for LHC. Ions injected and stored in the PS were aceelerated by the LEIR from 4.2 MeV to 72 MeV.

927

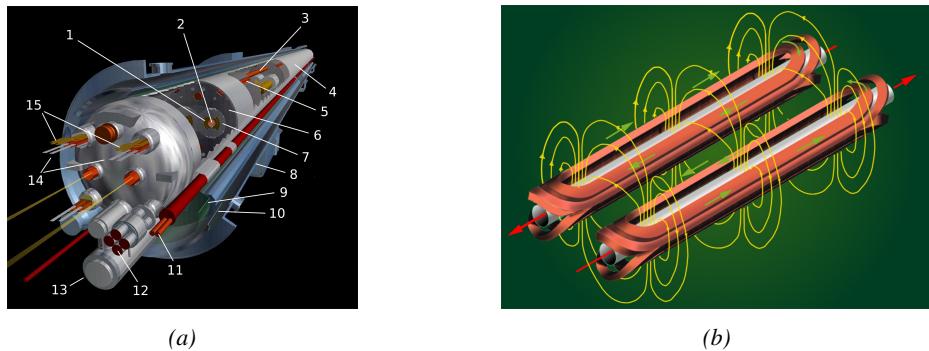
928       Directly following the PS, is finally the last acceleration stage before the LHC, the 7 km long  
929 *SPS*. The SPS accelerates the protons to 450 GeV and inject proton in both LHC accelerator rings  
930 that will increase their energy up to 7 TeV. When the LHC runs with heavy lead ions for ALICE  
931 and LHCb, ions are injected and accelerated to reach the energy of 2.8 TeV/A.



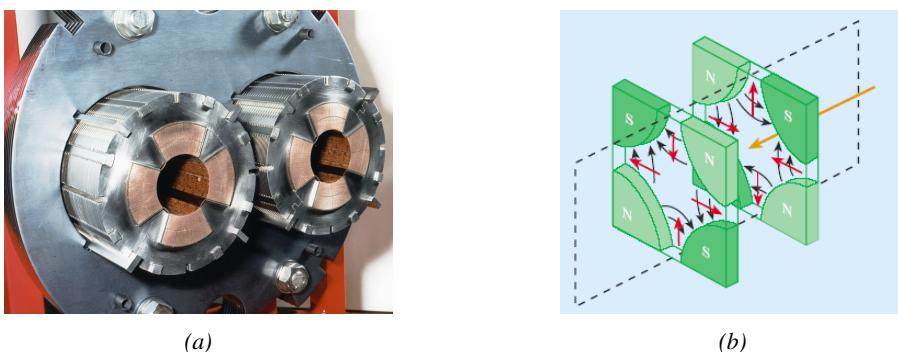
*Figure 2.5: Pictures of the different accelerators. From top to bottom: first the LINAC 2 and the Pb source of LINAC 3. Then the Booster and the LEIR. Finally, the PS, the SPS and the LHC.*

932      The LHC beams are not continuous and are rather organised in bunches of particles. When in  $pp$ -  
 933      collision mode, the beams are composed of 2808 bunches of  $1.15 \times 10^{11}$  protons separated by 25 ns.  
 934      When in  $Pb$  collision mode, the 592  $Pb$  bunches are on the contrary composed of  $2.2 \times 10^8$  ions  
 935      separated by 100 ns. The two parallel proton beams of the LHC are contained in a single twin-

bore magnet due to the space restriction in the LEP tunnel. Indeed, building 2 completely separate accelerator rings next to each other was impossible. The dipoles of the 1232 twin-bore magnets are showed in Figure 2.6 alongside the magnetic field generated along the dipole section to accelerate the particles. The dipoles generate a nominal field of 8.33 T, needed to give protons and lead nucleons their nominal energy. Some 392 quadrupoles, presented in Figure 2.7, are also used to focus the beams, as well as other multipoles to correct smaller imperfections.



*Figure 2.6: Figure 2.6a: schematics of the LHC cryodipoles. 1: Superconducting Coils, 2: Beam pipe, 3: Heat exchanger Pipe, 4: Helium-II Vessel, 5: Superconducting Bus-bar, 6: Iron Yoke, 7: Non-Magnetic Collars, 8: Vacuum Vessel, 9: Radiation Screen, 10: Thermal Shield, 11: Auxiliary Bus-bar Tube, 12: Instrumentation Feed Throughs, 13: Protection Diode, 14: Quadrupole Bus-bars, 15: Spool Piece Bus-bars. Figure 2.6b: magnetic field and resulting motion force applied on the beam particles.*



*Figure 2.7: Figure 2.7a: picture of the LHC quadrupoles. Figure 2.7b: magnetic fields and resulting focussing force applied on the beam by 2 consecutive quadrupoles.*

### 942 2.2.1.2 LHC discoveries and LHC physics program

943 The very first proton beam successfully circulated in the LHC in September 2008 directly followed  
 944 by an incident leading to mechanical damage that would delay the LHC program for a year until  
 945 November 2009.

946 **2.2.1.3 High Luminosity LHC**

947 **2.2.2 CMS, a multipurpose experiment**

948 **2.3 Muon Phase-II Upgrade**

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

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

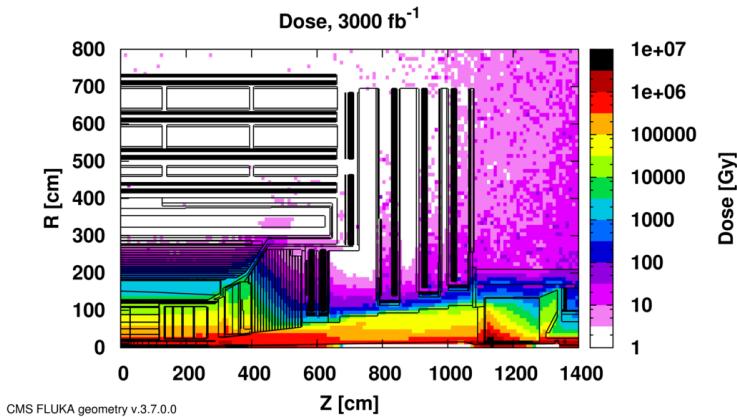


Figure 2.8: Absorbed dose in the CMS cavern after an integrated luminosity of  $3000 \text{ fb}^{-1}$ .  $R$  is the transverse distance from the beamline and  $Z$  is the distance along the beamline from the Interaction Point at  $Z=0$ .

963 The measurement of small production cross-section and/or decay branching ratio processes, such  
 964 as the Higgs boson coupling to charge leptons or the  $B_s \rightarrow \mu^+ \mu^-$  decay, is of major interest and  
 965 specific upgrades in the forward regions of the detector will be required to maximize the physics  
 966 acceptance on the largest possible solid angle. To ensure proper trigger performance within the  
 967 present coverage, the muon system will be completed with new chambers. In figure 2.9 one can  
 968 see that the existing Cathode Strip Chambers (CSCs) will be completed by Gas Electron Multipliers  
 969 (GEMs) and Resistive Plate Chambers (RPCs) in the pseudo-rapidity region  $1.6 < |\eta| < 2.4$  to  
 970 complete its redundancy as originally scheduled in the CMS Technical Proposal [23].

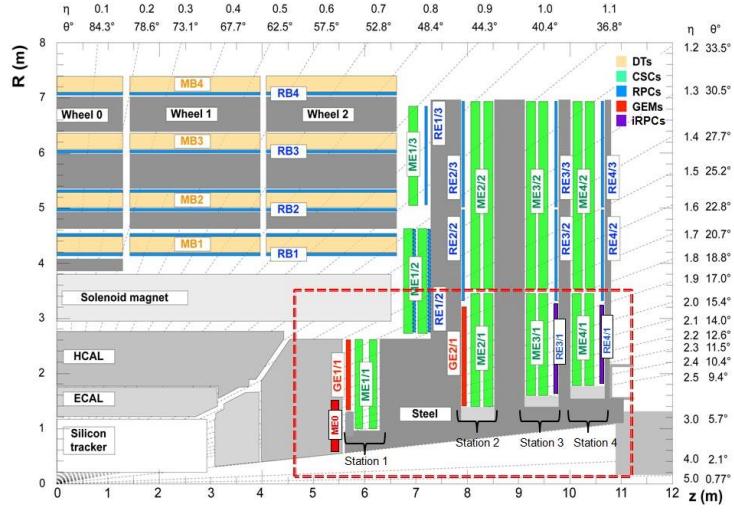
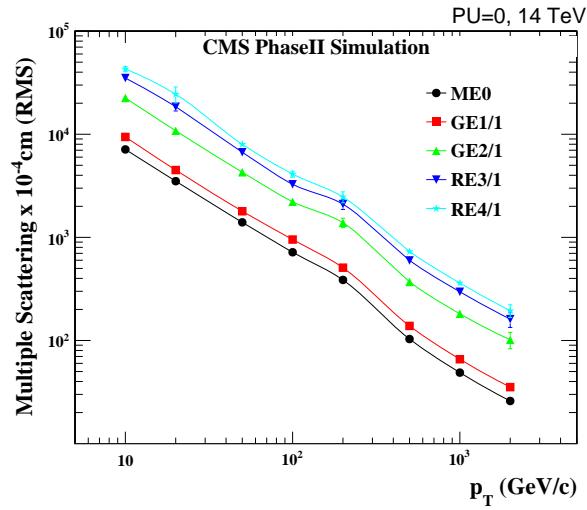


Figure 2.9: A quadrant of the muon system, showing DTs (yellow), RPCs (light blue), and CSCs (green). The locations of new forward muon detectors for Phase-II are contained within the dashed box and indicated in red for GEM stations (ME0, GE1/1, and GE2/1) and dark blue for improved RPC (iRPC) stations (RE3/1 and RE4/1).

971        RPCs are used by the CMS first level trigger for their good timing performances. Indeed, a very  
 972        good bunch crossing identification can be obtained with the present CMS RPC system, given their  
 973        fast response of the order of 1 ns. In order to contribute to the precision of muon momentum mea-  
 974        surements, muon chambers should have a spatial resolution less or comparable to the contribution  
 975        of multiple scattering [21]. Most of the plausible physics is covered only considering muons with  
 976         $p_T < 100$  GeV thus, in order to match CMS requirements, a spatial resolution of  $\mathcal{O}(\text{few mm})$  the  
 977        proposed new RPC stations, as shown by the simulation in figure 2.10. According to preliminary  
 978        designs, RE3/1 and RE4/1 readout pitch will be comprised between 3 and 6 mm and 5  $\eta$ -partitions  
 979        could be considered.



*Figure 2.10: RMS of the multiple scattering displacement as a function of muon p<sub>T</sub> for the proposed forward muon stations. All of the electromagnetic processes such as bremsstrahlung and magnetic field effect are included in the simulation.*



# 3

980

981

## Physics of Resistive plate chambers

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

993 Through its development history, different operating modes [27–29] and new detector designs [30–  
994 32] have been discovered, leading to further improvement of the rate capability of such a detector.  
995 Moreover, the addition of  $SF_6$  into the gas mix improved the stability of operation of the RPC [33,  
996 34].

997 The low developing costs and easily achievable large detection areas offered by RPCs, as well as  
998 the wide range of possible designs, made them a natural choice to as muon chambers and/or trigger  
999 detectors in multipurpose experiments such as CMS [21] or ATLAS [35], time-of-flight detectors in  
1000 ALICE [36], calorimeter with CALICE [37] or even detectors for volcanic muography with ToMu-  
1001 Vol [38].

### 1002 3.1 Principle

1003 RPCs are ionisation detectors composed of two parallel resistive plate electrodes in between which  
1004 a constant electric field is set. The space in between the electrodes, referred as *gap*, is filled with a  
1005 dense gas that is used to generate primary ionization into the gas volume. The free charge carriers  
1006 (electrons and cations) created by the ionization of the gas molecules are then accelerated towards

1007 the electrodes by the electric field, as shown in Figure 3.1 [39].

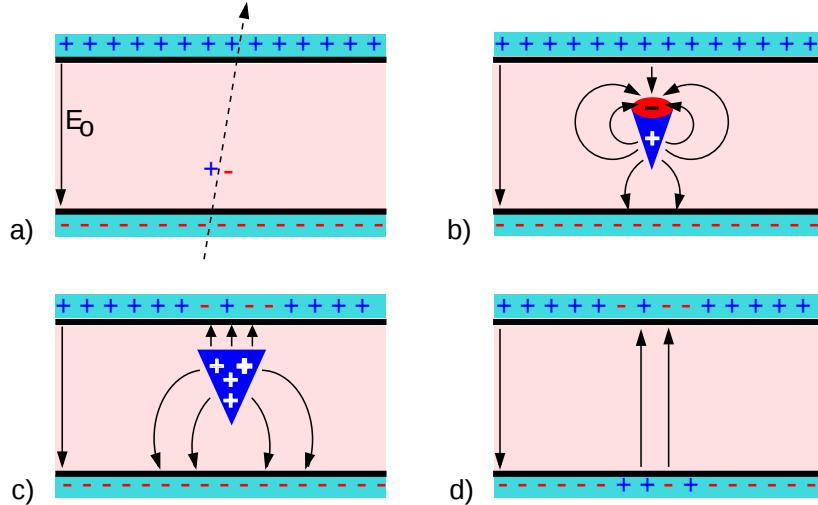


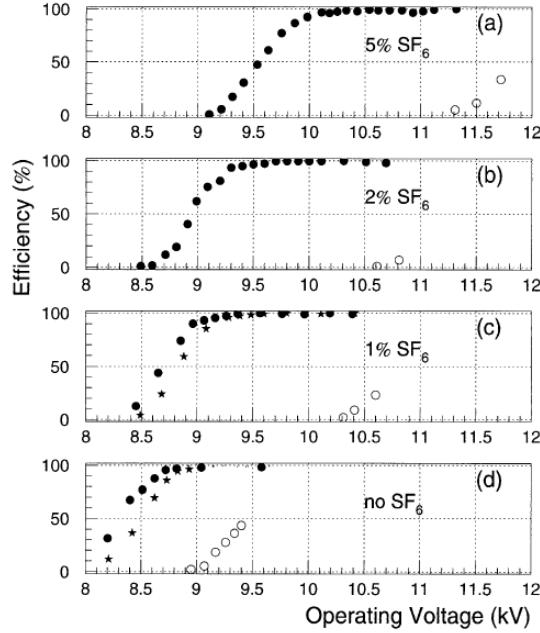
Figure 3.1: Different phases of the avalanche development in the RPC gas volume subjected to a constant electric field  $E_0$ . a) An avalanche is initiated by the primary ionisation caused by the passage of a charged particle through the gas volume. b) Due to its growing size, the avalanche starts to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

1008     RPCs being passive detectors, a current on pick-up copper read-out placed outside of the gas  
 1009 volume is induced by the charge accumulation during the growth of the avalanche. As a result,  
 1010 the time resolution of the detector is substantially increased as the output signal is generated while  
 1011 the electrons are still in movement. The advantage of a constant electric field, over multi-wire  
 1012 proportional chambers, is that the electrons are being fully accelerated from the moment charge  
 1013 carriers are freed and feel the full strength of the electric field that doesn't depend on the distance to  
 1014 the readout and that the output signal doesn't need for the electrons to be physically collected.

1015     The typical gas mixture RPCs are operated with is generally composed of 3 gas compounds.

- 1016     • Tetrafluoroethane ( $C_2F_4H_2$ ), also referred to as *Freon*, is the principal compound of the RPC  
 1017       gas mixtures, with a typical fraction above 90%. It is used for it's high effective Townsend  
 1018       coefficient and the great average fast charge that allows to operate the detector with a high  
 1019       threshold with respect to argon, for example, that has similar effective Townsend coefficient  
 1020       but suffers from a lower fast charge. To operate with similar conditions, argon would require a  
 1021       higher electric field leading to a higher fraction of streamers, thus limiting the rate capability  
 1022       of the detector [40].
- 1023     • Isobutane (i- $C_4H_{10}$ ), only present in a few percent in the gas mixtures, is used for its UV  
 1024       quenching properties [41] helping to prevent streamers due to UV photon emission during the  
 1025       avalanche growth.
- 1026     • Sulfur hexafluoride, ( $SF_6$ ), referred to simply as *SF6*, is used in very little quantities for its  
 1027       high electronegativity. Excess of electrons are being absorbed by the compound and streamers

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



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

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

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

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

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

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

1044 the developing cost of such materials is quite high due to the very low demand. Thus, High-pressure  
 1045 laminate (HPL) is often the choice for high rate experiments using very large RPC detection areas.  
 1046 Other experiments working at cosmic muon fluxes can safely operate with ordinary float glass.

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

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

### 1047 3.1.1 Electron drift velocity

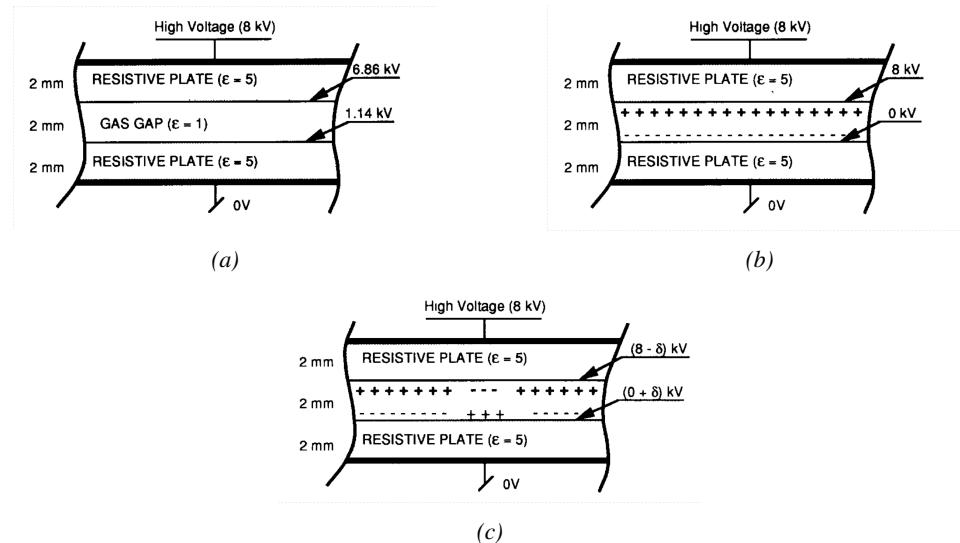
1048 Talk about the electron drift velocity and mention the time resolution of RPCs.

## 1049 3.2 Rate capability and time resolution of Resistive Plate Cham- 1050 bers

1051 As already previously discussed, the electrode material plays a key role in the max intrinsic rate  
 1052 capability of RPCs. R&D is being done to develop at always cheaper costs material with lower  
 1053 resistivity. Nevertheless, the amount of charge released, i.e. the size of the discharge, if reduced  
 1054 leads to a smaller blind area in the detector, increasing the rate capability of the detector.

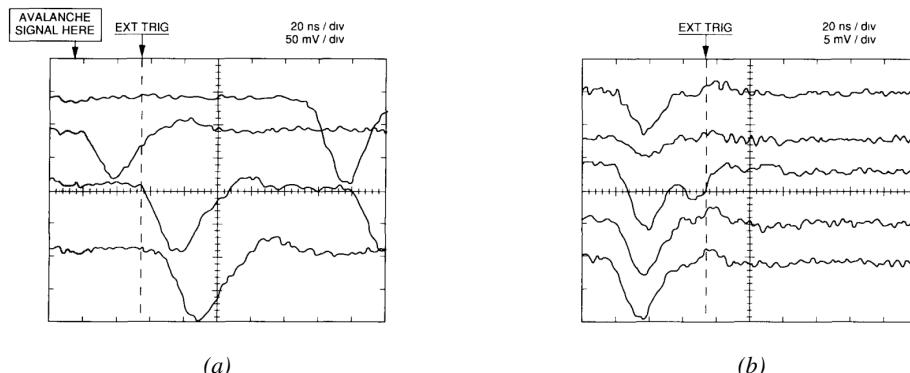
### 1055 3.2.1 Operation modes

1056 RPCs where developed early 1980s. At that time it was using an operating mode now referred to  
 1057 as *streamer mode*. Streamers are large discharges that develop in between the 2 electrodes enough  
 1058 to locally discharge the electrodes. If the electric field inside of the gas volume is strong enough,  
 1059 with electrons being fast compared to ions, a large and dense cloud of positive ions will develop  
 1060 nearby the anode and extend toward the cathode while the electrons are being collected, eventually  
 1061 leading to a streamer discharge due to the increase of field seen at the cathode. the field is then strong  
 1062 enough so that electrons are pulled out of the cathode. Electrodes, though they are a unique volume  
 1063 of resistive material, can be assimilated to capacitors. At the moment an electric field is applied in  
 1064 between their outer surfaces, the charge carriers inside of the volume will start moving leading to  
 1065 a situation where there is no voltage across the electrodes and a higher density of negative charges,  
 1066 i.e. electrons, on the inner surface of the cathode. Finally, when a streamer discharge occurs, these  
 1067 electrons are partially released in the gas volume contributing to increase the discharge strength until  
 1068 the formation of a conductive plasma, the streamer. This can be understood through Figure 3.3 [27].  
 1069 Streamer signals are very convenient in terms of read-out as no amplification is required with output  
 1070 pulses amplitudes of the order of a few tens to few hundreds of mV as can be seen on Figure 3.4.



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

1071 When the electric field is reduced though, the electronic gain is small until the electrons get close  
 1072 enough to the anode and the positive ion cloud is much smaller. The electric field cannot rise to the  
 1073 point a field emission of electrons on the cathode is possible. The resulting signal is weak, of the  
 1074 order of a few mv as shown on Figure 3.4, and requires amplification. This is the *avalanche mode*  
 1075 of RPC operation.



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

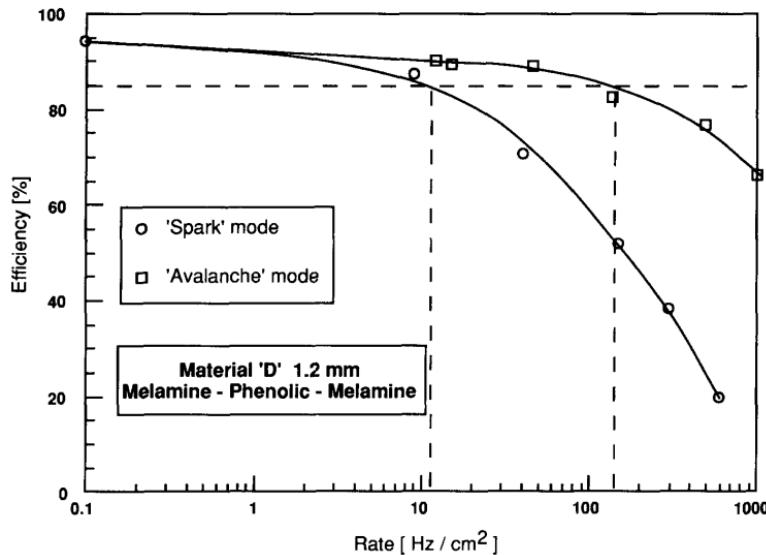


Figure 3.5: Rate capability comparison for the streamer and avalanche mode of operation. An order of magnitude in rate capability for a maximal efficiency drop of 10% is gained by using the avalanche mode over the streamer mode.

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

### 3.2.2 Detector designs and performance

Different RPC design have been used and each of them present its own advantages. Historically, the first type of RPC to have been developed is what is referred now to *narrow gap* RPC [24, 43]. After the avalanche mode has been discovered [27], it has been proven that increasing the width of the gas gap lead to higher rate capability, due to lower charge deposition per avalanche, and lower power dissipation [43]. Nevertheless, by increasing the gas gap width, the time resolution of the detector decreases. This is a natural result if the increase of active gas volume in the detector is taken into account. Indeed, for a given threshold, only the small fraction of gas closest to the cathode will provide enough gain to have a detectable signal. In the case of a wider gas volume, the active region is then larger and a larger time jitter is introduced with the variation of starting position of the avalanche, as discussed in [30]. To solve improve both the time resolution and the rate capability, different methods were used trying to get advantages of both narrow and wide gap RPCs.

1095 **3.2.2.1 Double-gap RPC**

1096 Double-gap RPCs are made out of 2 narrow RPC detectors. The 2 RPC gaps are stacked on top of  
 1097 each other as shown in Figure 3.6. This detector layout, popularized by the two multipurpose experiments  
 1098 CMS [21] and ATLAS [35] at LHC, can be used as an OR system in which each individual  
 1099 chamber participates in the output signal. The gain of such a detector is greatly reduced with respect  
 1100 to single-gap RPCs with an efficiency plateau reached at lower voltage, as visible on Figure 3.7.

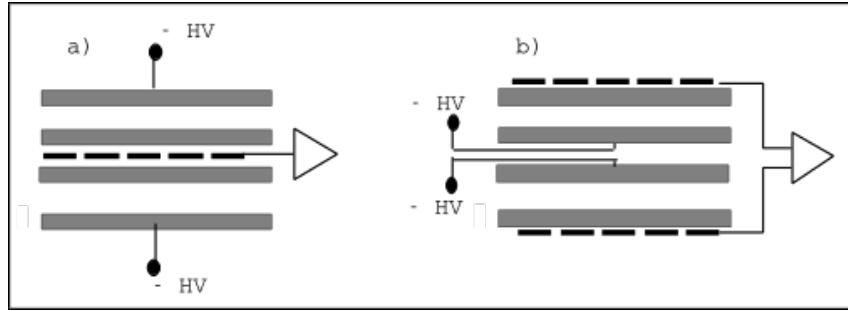


Figure 3.6: Possible double-gap RPC layouts: a) "standard" 1D double-gap RPC, as used in CMS experiment, where the anodes are facing each other and a 1D read-out plane is sandwiched in between them, b) double read-out double-gap RPC as used in ATLAS experiment, where the cathodes are facing each other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

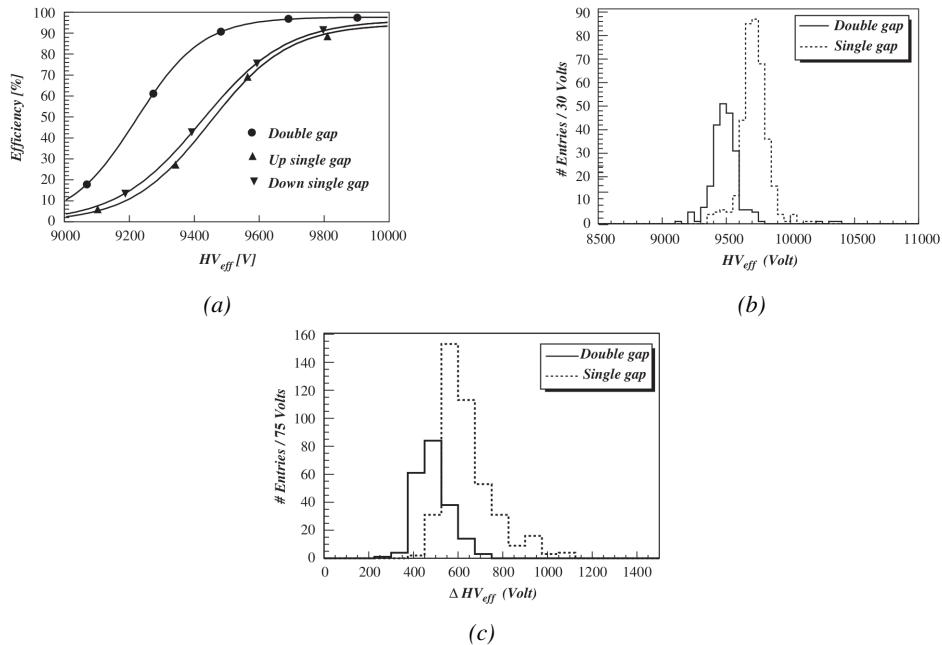


Figure 3.7: Comparison of performance of CMS double and single gap RPCs using cosmic muons [44]. Figure 3.7a: Comparison of efficiency sigmoids. Figure 3.7b: Voltage distribution at 95% of maximum efficiency. Figure 3.7c:  $\Delta_{10\%}^{90\%}$  distribution.

### 3.2.2.2 Multigap RPC (MRPC)

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

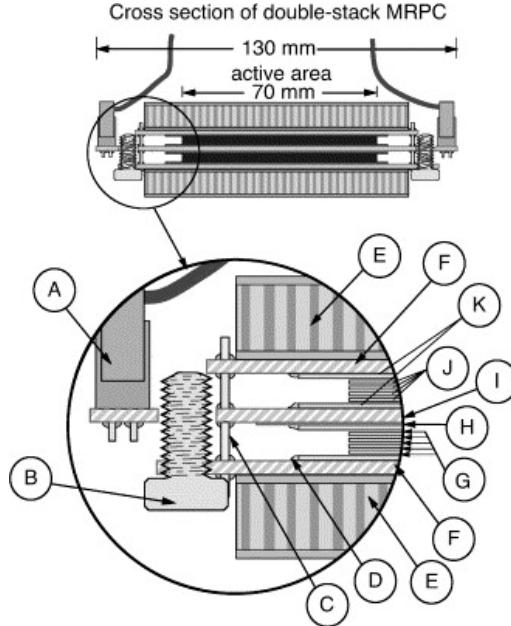


Figure 3.8: Presentation of ALICE MRPC using  $250 \mu\text{m}$  gas gaps,  $620 \mu\text{m}$  outer glass electrodes and  $550 \mu\text{m}$  inner floating electrodes. More details on the labels are given in [45].

Sometimes used as a double multigap RPC, taking advantage of the OR of double gap RPCs, the MRPC is mainly used as ToF detector [45–49] due to its excellent timing properties that allow to perform particle identification as explained by Williams in [50]. The principle of particle identification using ToF is simply the measurement of the velocity of a particle. Indeed, particles are defined by their mass (for the parameter of interest here, their electric charge being measured using the bending angle of the particles traveling through a magnetic field) and this mass can be calculated by measuring the velocity  $\beta$  and momentum of the particle:

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

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

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

1117 An example of particle identification is given for the case of STAR experiment in Figure 3.9.

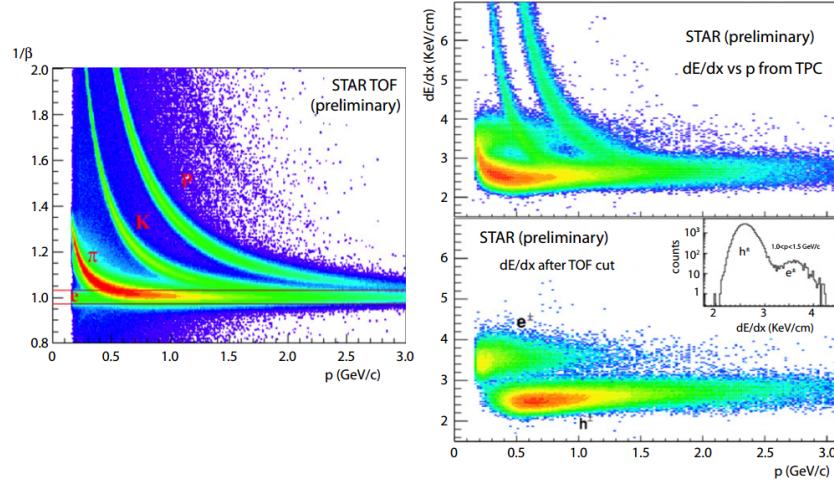


Figure 3.9: Particle identification applied to electrons in the STAR experiment. The identification is performed combining ToF and  $dE/dx$  measurements [50].

1118 Another benefice of using such small gas gaps is the strong reduction of the average avalanche  
1119 volume and thus of the blind spot on MRPCs leading to an improved rate capability. Multigaps can  
1120 sustain backgrounds of several kHz/cm<sup>2</sup> as demonstrated in Figure 3.10.

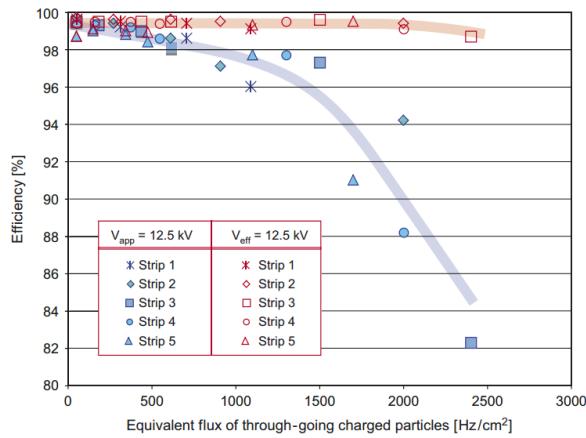


Figure 3.10: Comparison of the detector performance of ALICE ToF MRPC [51] at fixed applied voltage (in blue) and at fixed effective voltage (in red). The effective voltage is kept fixed by increasing the applied voltage accordingly to the current drawn by the detector.

### 1121 3.2.2.3 Charge distribution and performance limitations

1122 The direct consequence of the different RPC layouts is a variation of intrinsic time resolution of the  
1123 RPC as the gap size decreases. An advantage is given to multigaps whose design use sub-millimeter  
1124 gas volumes providing very consistent signals.

1125 On the charge spectrum point of view, each layout has its own advantages. When the double-gap  
 1126 has the highest induced over drifting charge ratio, as seen in Figure 3.11, the multigap has a charge  
 1127 spectrum strongly detached from the origin, as visible in Figure 3.12. A high induced over drifting  
 1128 charge ratio means that the double gap can be safely operated at high threshold or that at similar  
 1129 threshold it can be operated with a twice smaller drifting charge, meaning a higher rate capability.  
 1130 On the other hand, the strong detachment of the charge spectrum from the origin in the MRPC case  
 1131 allows to reach a higher efficiency with increasing threshold as most of the induced charge is not low  
 1132 due to the convolution of several single gap spectra. The range of stable efficiency increases with  
 1133 the number of gap, as presented in Figure 3.13.

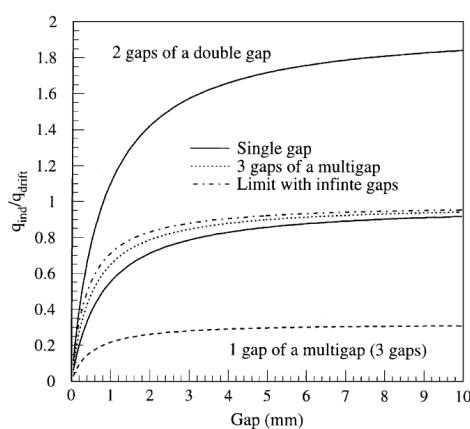


Figure 3.11: Ratio between total induced and drifting charge have been simulated for single gap, double-gap and multigap layouts [52]. The total induced charge for a double-gap RPC is a factor 2 higher than for a multigap.

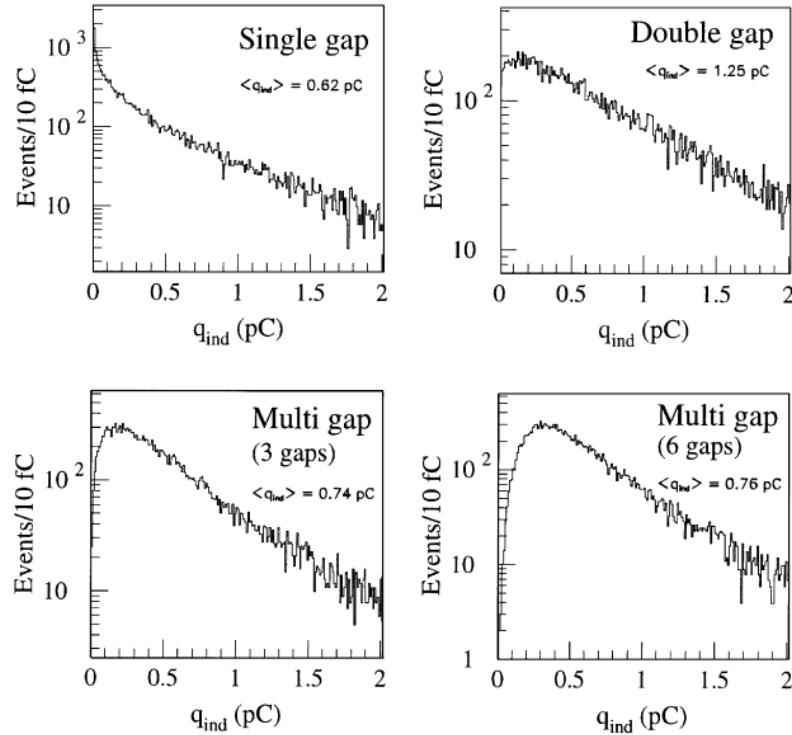


Figure 3.12: Charge spectra have been simulated for single gap, double-gap and multigap layouts [52]. It appears that when single gap shows a decreasing spectrum, double and multigap layouts exhibit a spectrum whose peak is detached from the origin. The detachment gets stronger as the number of gaps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

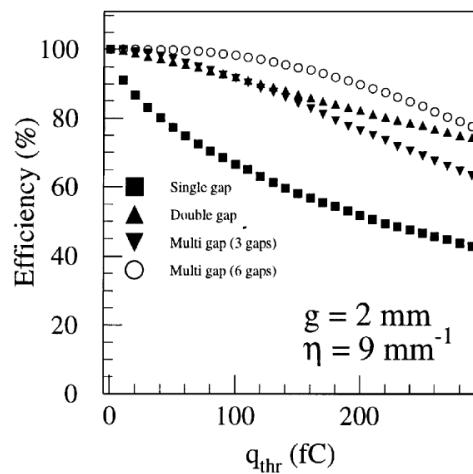


Figure 3.13: The maximal theoretical efficiency is simulated for single gap, double-gap and multigap layouts [52] at a constant gap thickness of 2 mm and using an effective Townsend coefficient of  $9 \text{ mm}^{-1}$ .

<sup>1134</sup> **3.3 Signal formation**

<sup>1135</sup> **3.4 Gas transport parameters**

# 4

1136

1137

1138

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

### 4.1 Resistive Plate Chambers at CMS

#### 4.1.1 Overview

1141 The Resistive Plate Chambers (RPC) system, located in both barrel and endcap regions, provides a  
1142 fast, independent muon trigger with a looser  $p_T$  threshold over a large portion of the pseudorapidity  
1143 range ( $|\eta| < 1.6$ ) [add reconstruction].

1144

1145 During High-Luminosity LHC (HL-LHC) operations the expected conditions in terms of back-  
1146 ground and pile-up will make the identification and correct  $P_T$  assignment a challenge for the Muon  
1147 system. The goal of RPC upgrade is to provide additional hits to the Muon system with precise tim-  
1148 ing. All these informations will be elaborated by the trigger system in a global way enhancing the  
1149 performance of the trigger in terms of efficiency and rate control. The RPC Upgrade is based on two  
1150 projects: an improved Link Board System and the extension of the RPC coverage up to  $|\eta| = 2.4$ .  
1151 [FIXME 2.4 or 2.5?]

1152 The Link Board system, that will be described in section xxx, is responsible to process, syn-  
1153 chronize and zero-suppress the signals coming from the RPC front end boards. The Link Board  
1154 components have been produced between 2006 and 2007 and will be subjected to aging and failure  
1155 in the long term. The upgraded Link Board system will overcome the aging problems described in  
1156 section xxx and will allow for a more precise timing information to the RPC hits from 25 to 1 ns [ref  
1157 section xxx].

1158 The extension of the RPC system up to  $|\eta| = 2.1$  was already planned in the CMS TDR [ref  
1159 cmstdr] and staged because of budget limitations and expected background rates higher than the rate  
1160 capability of the present CMS RPCs in that region. An extensive R&D program has been done in  
1161 order to develop an improved RPC that fulfills the CMS requirements. Two new RPC layers in the  
1162 innermost ring of stations 3 and 4 will be added with benefits to the neutron-induced background

1163 reduction and efficiency improvement for both trigger and offline reconstruction.

### 1164 4.1.2 The present RPC system

1165 The RPC system is organized in 4 stations called RB1 to RB4 in the barrel region, and RE1 to RE4  
 1166 in the endcap region. The innermost barrel stations, RB1 and RB2, are instrumented with 2 layers  
 1167 of RPCs facing the innermost (RB1in and RB2in) and outermost (RB1out and RB2out) sides of the  
 1168 DT chambers. Every chamber is then divided from the read-out point of view into 2 or 3  $\eta$  partitions  
 1169 called “rolls”. The RPC system consist of 480 barrel chambers and 576 endcap chambers. Details  
 1170 on the geometry are discussed in the paper [ref to geo paper].

1171 The CMS RPC chamber is a double-gap, operated in avalanche mode to ensure reliable operation  
 1172 at high rates. Each RPC gap consists of two 2-mm-thick resistive High-Pressure Laminate (HPL)  
 1173 plates separated by a 2-mm-thick gas gap. The outer surface of the HPL plates is coated with a thin  
 1174 conductive graphite layer, and a voltage is applied. The RPCs are operated with a 3-component,  
 1175 non-flammable gas mixture consisting of 95.2% freon ( $C_2H_2F_4$ , known as R134a), 4.5% isobutane  
 1176 ( $i-C_4H_{10}$ ), and 0.3% sulphur hexafluoride ( $SF_6$ ) with a relative humidity of 40% - 50%. Readout  
 1177 strips are aligned in  $\eta$  between the 2 gas gaps. [\[Add a sentence on FEBs.\]](#)

1178 The discriminated signals coming from the Front End boards feed via twisted cables (10 to 20 m  
 1179 long) the Link Board System located in UXC on the balconies around the detector. The Link System  
 1180 consist of the 1376 Link Boards (LBs) and the 216 Control Boards (CBs), placed in 108 Link Boxes.  
 1181 The Link Box is a custom crate (6U high) with 20 slots (for two CBs and eighteen LBs). The Link  
 1182 Box contains custom backplane to which the cables from the chambers are connected, as well as the  
 1183 cables providing the LBs and CBs power supply and the cables for the RPC FEBs control with use  
 1184 of the I2C protocol (trough the CB). The backplane itself contains only connectors (and no any other  
 1185 electronic devices).

1186 The Link Board has 96 input channels (one channel corresponds to one RPC strip). The input  
 1187 signals are the  $\sim 100$  ns binary pulses which are synchronous to the RPC hits, but not to the LHC  
 1188 clock (which drives the entire CMS electronics). Thus the first step of the FEB signals processing  
 1189 is synchronization, i.e. assignment of the signals to the BXes (25 ns periods). Then the data are  
 1190 compressed with a simple zero-suppressing algorithm (the input channels are grouped into 8 bit  
 1191 partitions, only the partitions with at least one nonzero bit are selected for each BX). Next, the non-  
 1192 empty partitions are time-multiplexed i.e. if there are more than one such partition in a given BX,  
 1193 they are sent one-by-one in consecutive BXes. The data from 3 neighbouring LBs are concentrated  
 1194 by the middle LB which contains the optical transmitter for sending them to the USC over a fiber at  
 1195 1.6 Gbps.

1196 The Control Boards provide the communication of the control software with the LBs via the  
 1197 FEC/CCU system. The CBs are connected into token rings, each ring consists of 12 CBs of one  
 1198 detector tower and a FEC mezzanine board placed on the CCS board located in the VME crate in  
 1199 the USC. In total, there are 18 rings in the entire Link System. The CBs also perform automatic  
 1200 reloading of the LB's firmware which is needed in order to avoid accumulation of the radiation  
 1201 induced SEUs in the LBs firmware.

1202 Both LBs and CB are based on the Xilinx Spartan III FPGAs, the CB additionally contains  
 1203 radiation-tolerant (FLASH based) FPGA Actel ProAsicPlus.

1204 The High Voltage power system is located in USC, not exposed to radiation and easily accessible  
 1205 for any reparation. A single HV channel powers 2 RPC chambers both in the barrel and endcap  
 1206 regions. The Low Voltage boards are located in UXC on the balconies and provide the voltage to the

<sup>1207</sup> front end electronics.

### <sup>1208</sup> 4.1.3 Pulse processing of CMS RPCs

<sup>1209</sup> Signals induced by cosmic particle in the RPC strips are shaped by standard CMS RPC Front-End  
<sup>1210</sup> Electronics (FEE) following the scheme of Figure 4.1. On a first stage, analogic signals are amplified  
<sup>1211</sup> and then sent to the Constant Fraction Discriminator (CFD) described in Figure 4.2. At the end of  
<sup>1212</sup> the chain, 100 ns long pulses are sent in the LVDS output. These output signal are sent on one side to  
<sup>1213</sup> a V1190A Time-to-Digital Converter (TDC) module from CAEN and on the other to an OR module  
<sup>1214</sup> to count the number of detected signals. Trigger and hit coïncidences are monitored using scalers.  
<sup>1215</sup> The TDC is used to store the data into ROOT files. These files are thus analysed to understand the  
<sup>1216</sup> detectors performance.

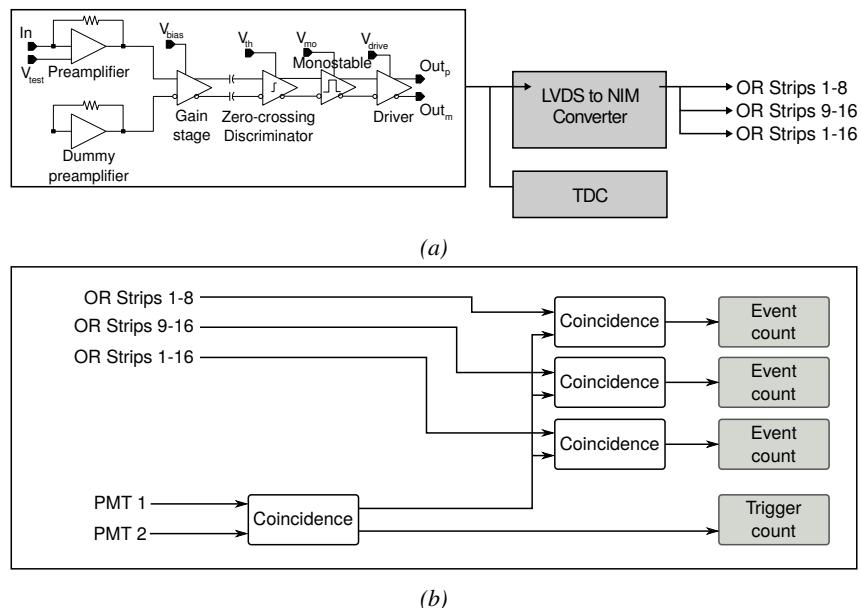
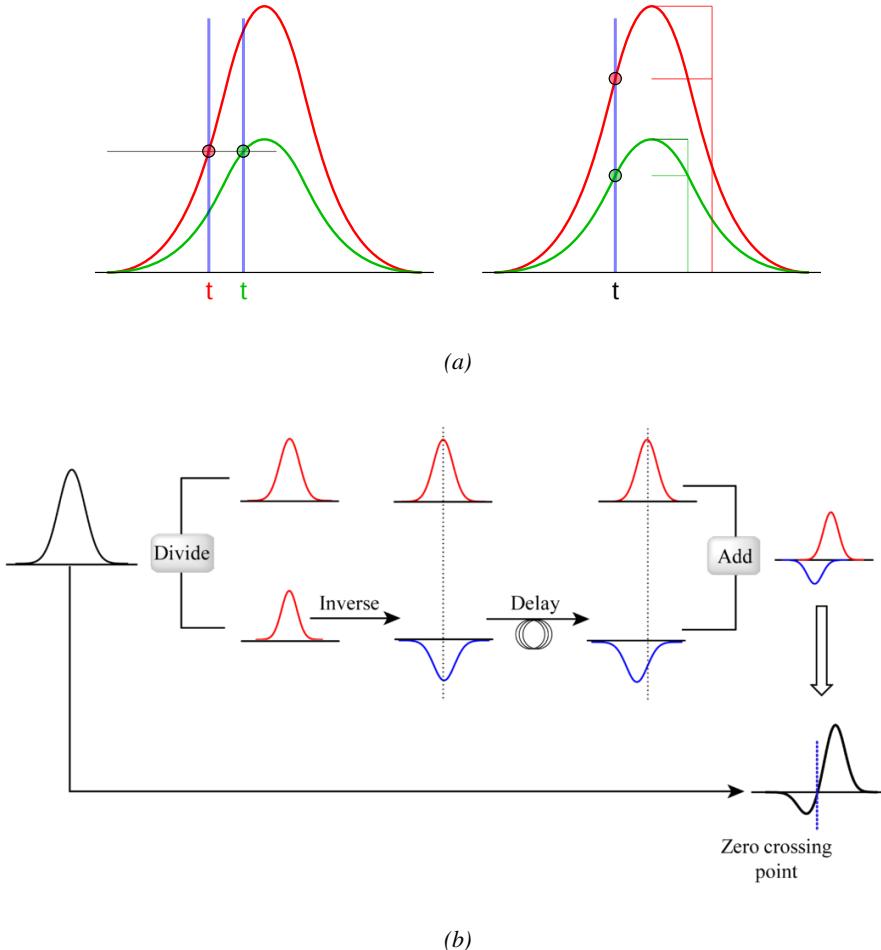


Figure 4.1: Signals from the RPC strips are shaped by the FEE described on Figure 4.1a. Output LVDS signals are then read-out by a TDC module connected to a computer or converted into NIM and sent to scalers. Figure 4.1b describes how these converted signals are put in coincidence with the trigger.



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

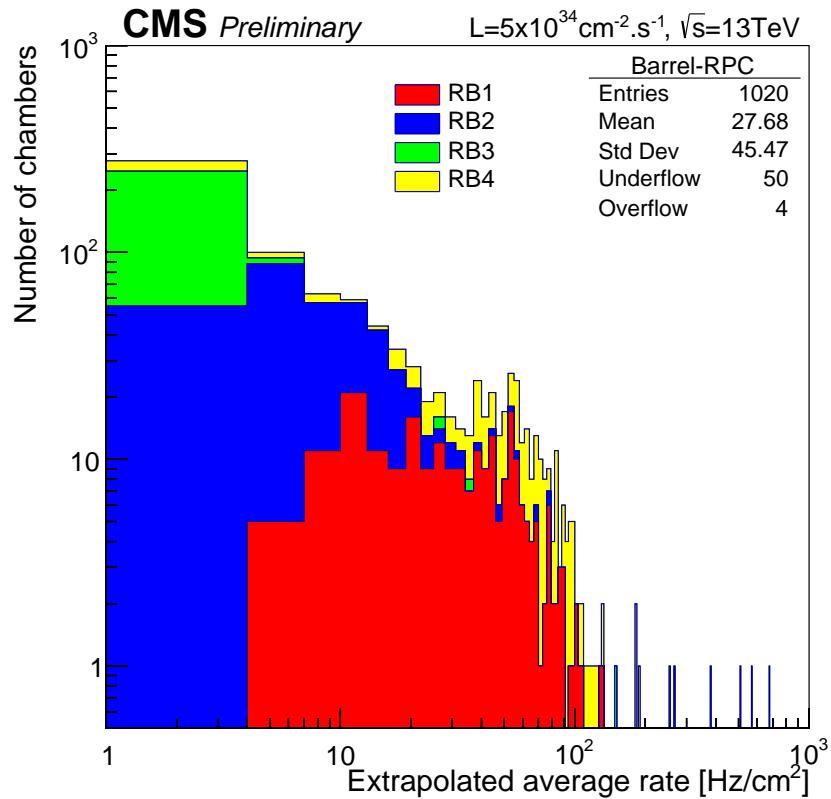
## 1217 4.2 Testing detectors under extreme conditions

1218 The upgrade from LHC to HL-LHC will increase the peak luminosity from  $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$  to reach  
 1219  $5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ , increasing in the same way the total expected background to which the RPC  
 1220 system will be subjected to. Composed of low energy gammas and neutrons from  $p\text{-}p$  collisions, low  
 1221 momentum primary and secondary muons, puch-through hadrons from calorimeters, and particles  
 1222 produced in the interaction of the beams with collimators, the background will mostly affect the  
 1223 regions of CMS that are the closest to the beam line, i.e. the RPC detectors located in the endcaps.  
 1224 [To update.]

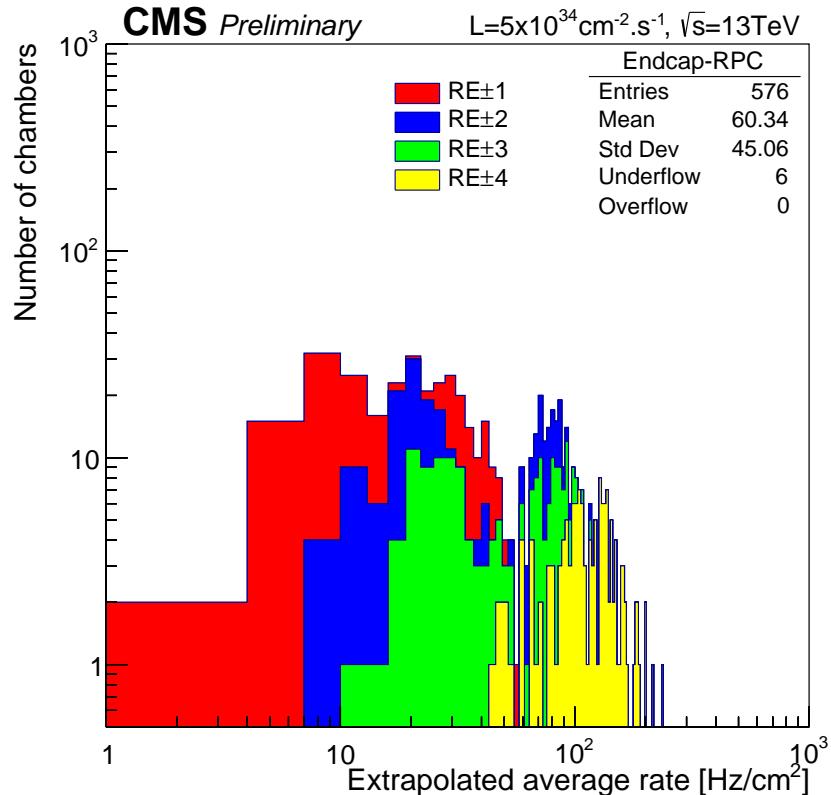
1225

1226        The 2016 data allowed to study the values of the background rate in all RPC system. In Fig-  
1227        ure 4.3, the distribution of the chamber background hit rate per unit area is shown at a luminosity  
1228        of  $5 \times 10^{34} cm^{-2}.s^{-1}$  linearly extrapolating from data collected in 2016 [ref mentioning the linear  
1229        dependency of rate vs lumi]. The maximum rate per unit area at HL-LHC conditions is expected to  
1230        be of the order of  $600 Hz/cm^2$  (including a safety factor 3). Nevertheless, Fluka simulations have  
1231        conducted in order to understand the background at HL-LHC conditions. The comparison to the  
1232        data has shown, in Figure 4.4, a discrepancy of a factor 2 even though the order of magnitude is  
1233        consistent. [Understand mismatch.]

1234



(a)



(b)

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

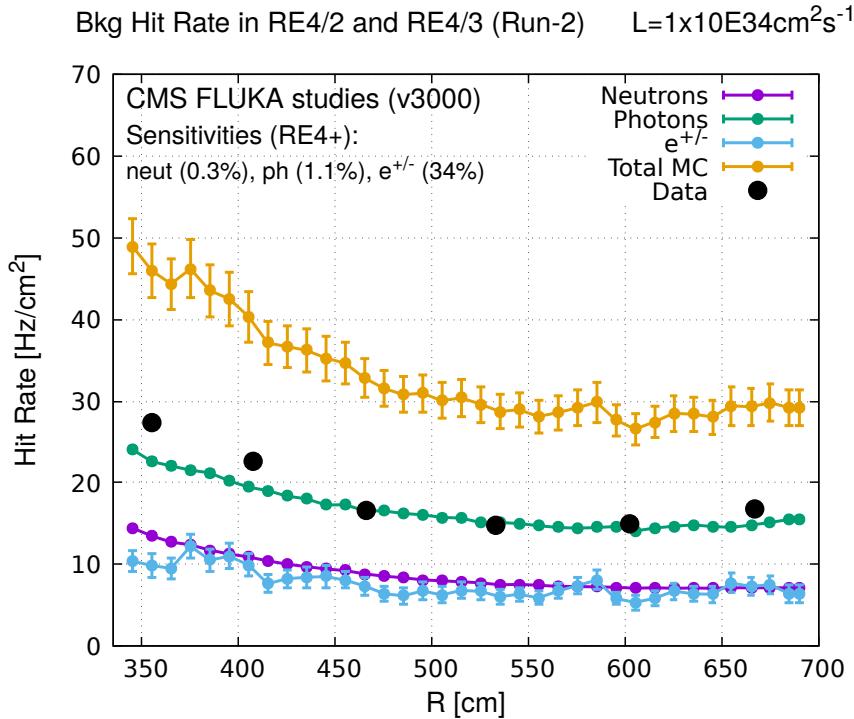


Figure 4.4: Background Fluka simulation compared to 2016 Data at  $L = 10^{34} \text{ cm}^{-2} \cdot \text{s}^{-1}$  in the fourth endcap disk region. A mismatch in between simulation and data can be observed. [\[To be understood.\]](#)

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

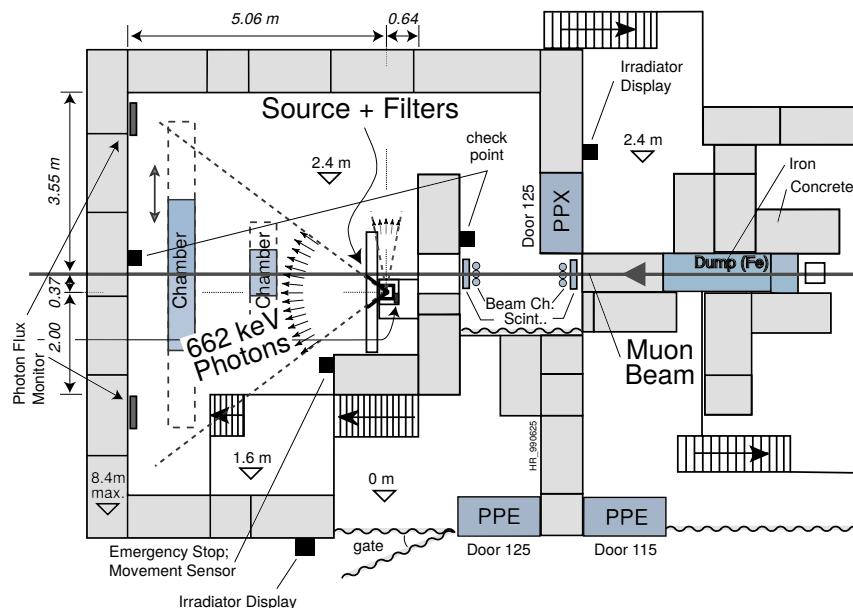
1242

## 1243 4.2.1 The Gamma Irradiation Facilities

### 1244 4.2.1.1 GIF

1245 Located in the SPS West Area at the downstream end of the X5 test beam, the Gamma Irradiation  
 1246 Facility (GIF) was a test area in which particle detectors were exposed to a particle beam in presence  
 1247 of an adjustable gamma background [55]. Its goal was to reproduce background conditions these  
 1248 detectors would suffer in their operating environment at LHC. GIF layout is shown in Figure 4.5.  
 1249 Gamma photons are produced by a strong  $^{137}\text{Cs}$  source installed in the upstream part of the zone  
 1250 inside a lead container. The source container includes a collimator, designed to irradiate a  $6 \times 6 \text{ m}^2$   
 1251 area at 5 m maximum to the source. A thin lens-shaped lead filter helps providing with a uniform  
 1252 outgoing flux in a vertical plane, orthogonal to the beam direction. The principal collimator hole  
 1253 provides a pyramidal aperture of  $74^\circ \times 74^\circ$  solid angle and provides a photon flux in a pyramidal vol-

1254      ume along the beam axis. The photon rate is controled by further lead filters allowing the maximum  
 1255      rate to be limited and to vary within a range of four orders of magnitude. Particle detectors under test  
 1256      are then placed within the pyramidal volume in front of the source, perpendicularly to the beam line  
 1257      in order to profit from the homogeneous photon flux. Adjusting the background flux of photons can  
 1258      then be done by using the filters and choosing the position of the detectors with respect to the source.  
 1259



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

1260      As described on Figure 4.6, the  $^{137}\text{Cs}$  source emits a 662 keV photon in 85% of the decays. An  
 1261      activity of 740 GBq was measured on the 5<sup>th</sup> March 1997. To estimate the strength of the flux in  
 1262      2014, it is necessary to consider the nuclear decay through time assiciated to the Cesium source  
 1263      whose half-life is well known ( $t_{1/2} = (30.05 \pm 0.08)$  y). The GIF tests where done in between the  
 1264      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  
 1265      the activity from 740 GBq in 1997 to 494 GBq in 2014.  
 1266

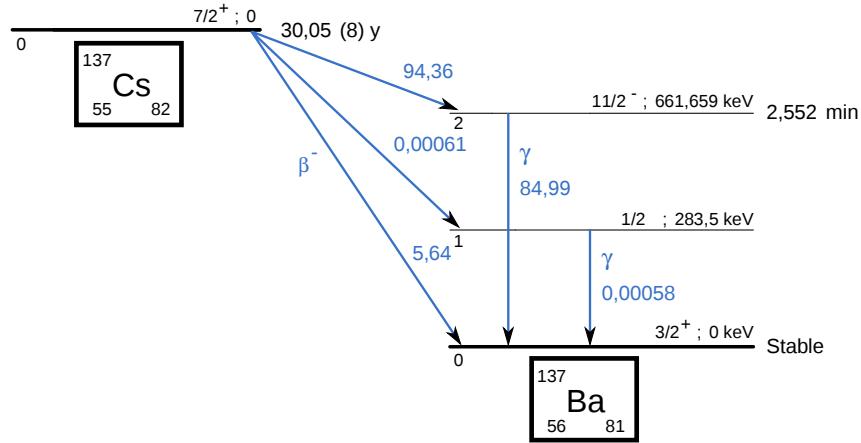


Figure 4.6:  $^{137}\text{Cs}$  decays by  $\beta^-$  emission to the ground state of  $^{137}\text{Ba}$  ( $\text{BR} = 5.64\%$ ) and via the 662 keV isomeric level of  $^{137}\text{Ba}$  ( $\text{BR} = 94.36\%$ ) whose half-life is 2.55 min.

#### 4.2.1.2 GIF++

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

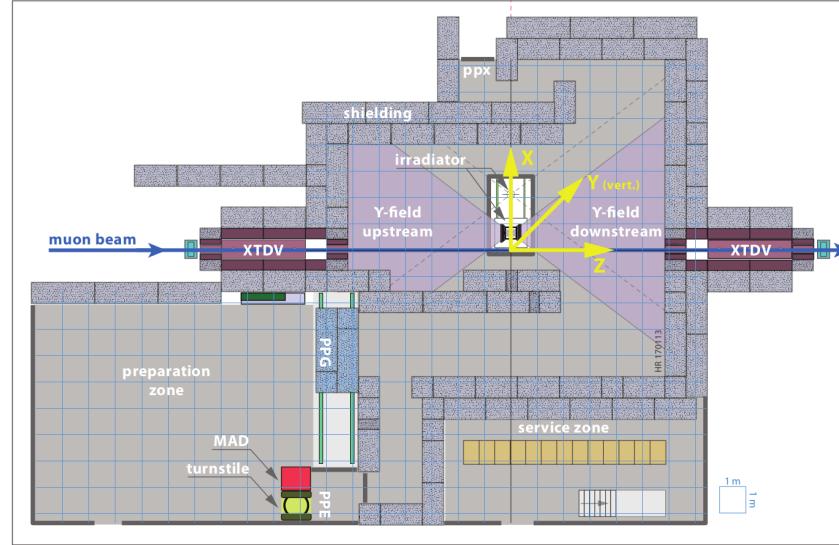


Figure 4.7: Floor plan of the GIF++ facility. When the facility downstream of the GIF++ takes electron beam, a beam pipe is installed along the beam line ( $z$ -axis). The irradiator can be displaced laterally (its center moves from  $x = 0.65 \text{ m}$  to  $2.15 \text{ m}$ ), to increase the distance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

1275 The source activity was measured to be about 13.5 TBq in March 2016. The photon flux being  
 1276 far greater than HL-LHC expectations, GIF++ provides an excellent facility for accelerated aging  
 1277 tests of muon detectors.

1278

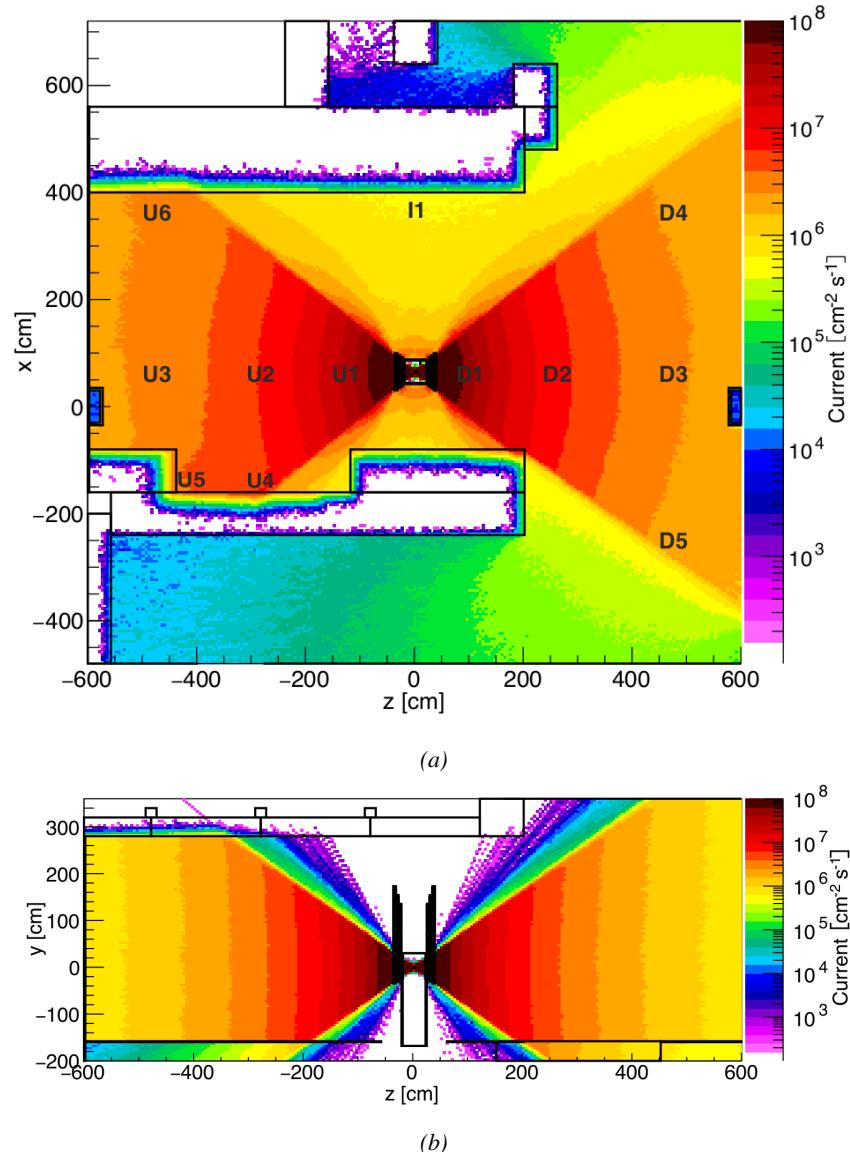


Figure 4.8: Simulated unattenuated current of photons in the xz plane (Figure 4.8a) and yz plane (Figure 4.8b) through the source at  $x = 0.65 \text{ m}$  and  $y = 0 \text{ m}$ . With angular correction filters, the current of 662 keV photons is made uniform in xy planes.

1279 The source is situated in the muon beam line with the muon beam being available a few times a  
 1280 year. The H4 beam, composed of muons with a momentum of about 150 GeV/c, passes through the  
 1281 GIF++ zone and is used to study the performance of the detectors. Its flux is of 104 particles/ $\text{s cm}^2$

1282 focused in an area similar to  $10 \times 10 \text{ cm}^2$ . Therefore, with properly adjusted filters, one can imitate  
 1283 the HL-LHC background and study the performance of muon detectors with their trigger/readout  
 1284 electronics in HL-LHC environment.

1285

## 1286 4.3 Preliminary tests at GIF

### 1287 4.3.1 Resistive Plate Chamber test setup

1288 During summer 2014, preliminary tests have been conducted in the GIF area on a newly produced  
 1289 RE4/2 chamber labelled RE-4-2-BARC-161. This chamber has been placed into a trolley covered  
 1290 with a tent. The position of the RPC inside the tent and of the tent related to the source is described  
 1291 in Figure 4.9. To test this CMS RPC, three different absorber settings were used. First of all,  
 1292 measurements were done with fully opened source. Then, to complete this preliminary study, the  
 1293 gamma flux has been attenuated from a factor 2 and a factor 5. The expected gamma flux at the level  
 1294 of our detector will be discussed in subsection ??.

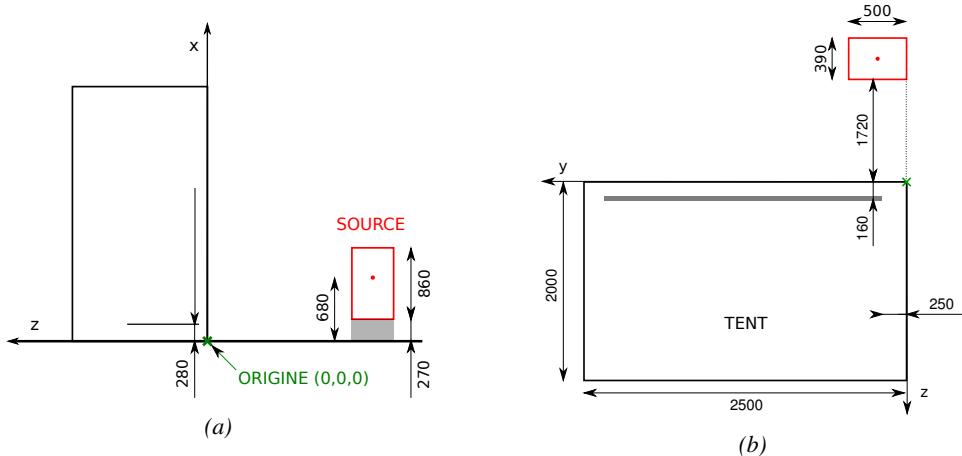
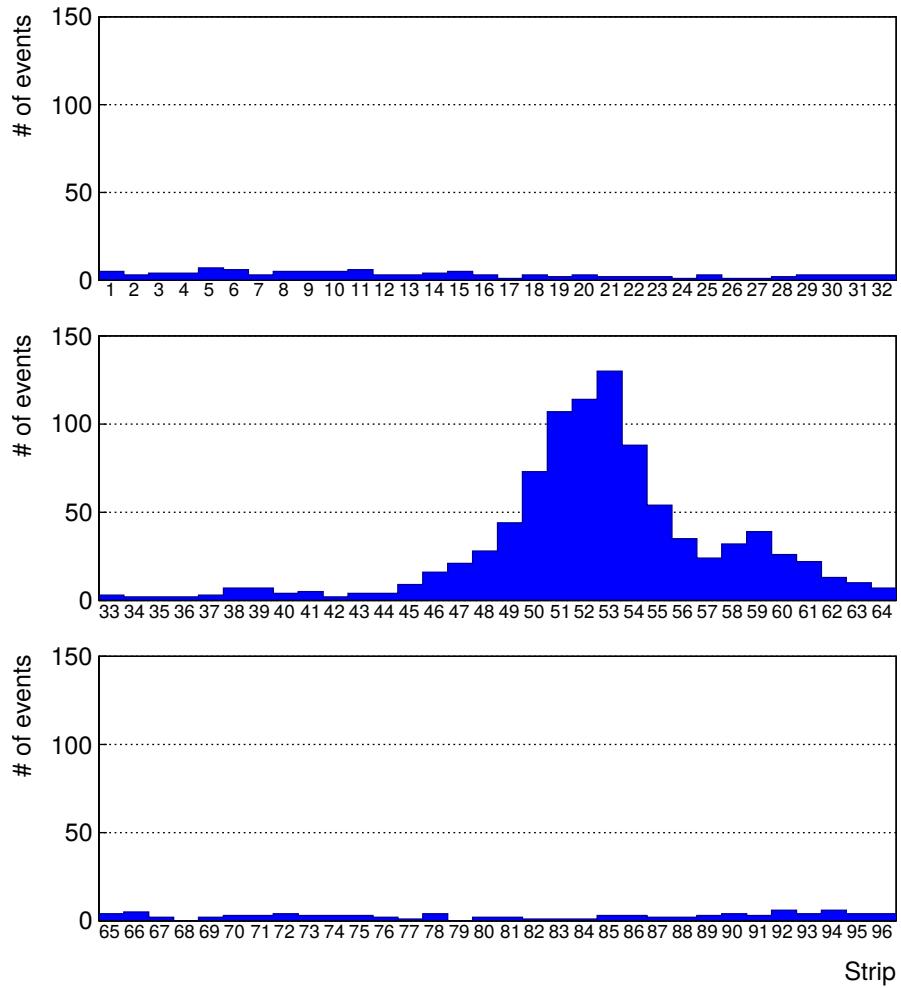


Figure 4.9: Description of the RPC setup. Dimensions are given in mm. A tent containing RPCs is placed at 1720 mm from the source container. The source is situated in the center of the container. RE-4-2-BARC-161 chamber is 160 mm inside the tent. This way, the distance between the source and the chambers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



*Figure 4.10: RE-4-2-BARC-I61 chamber is inside the tent as described in Figure 4.9. In the top right, the two scintillators used as trigger can be seen. This trigger system has an inclination of 10° relative to horizontal and is placed above half-partition B2 of the RPCs. PMT electronics are shielded thanks to lead blocks placed in order to protect them without stopping photons from going through the scintillators and the chamber.*

1295 At the time of the tests, the beam not being operationnal anymore, a trigger composed of 2 plastic  
 1296 scintillators has been placed in front of the setup with an inclination of 10 deg with respect to the  
 1297 detector plane in order to look at cosmic muons. Using this particular trigger layout, shown on Fig-  
 1298 ure 4.10, leads to a cosmic muon hit distribution into the chamber similar to the one in Figure 4.11.  
 1299 Measured without gamma irradiation, two peaks can be seen on the profil of partition B, centered  
 1300 on strips 52 and 59. Section ?? will help us understand that these two peaks are due respectively to  
 1301 forward and backward coming cosmic particles where forward coming particles are first detected by  
 1302 the scintillators and then the RPC while the backward coming muons are first detected in the RPC.



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

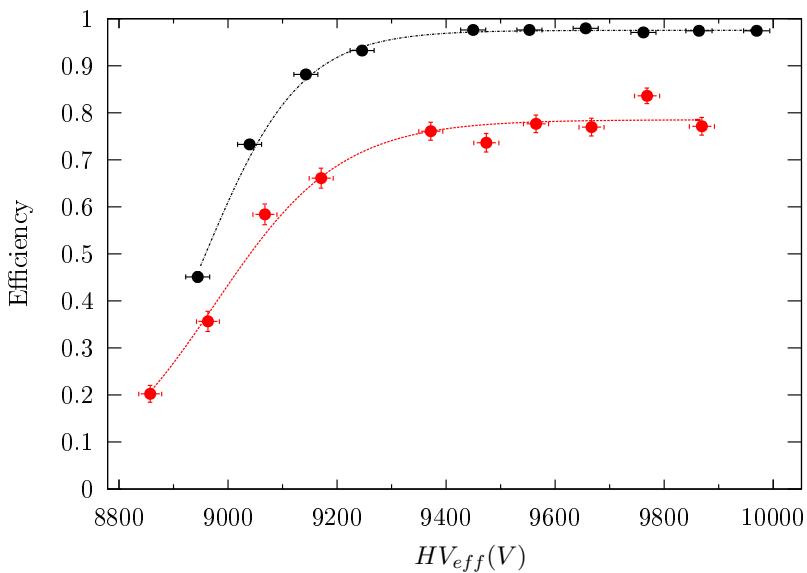
### 4.3.2 Data Acquisition

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

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

1310 protect the photomultipliers from gammas as can be seen from Figure 4.10.

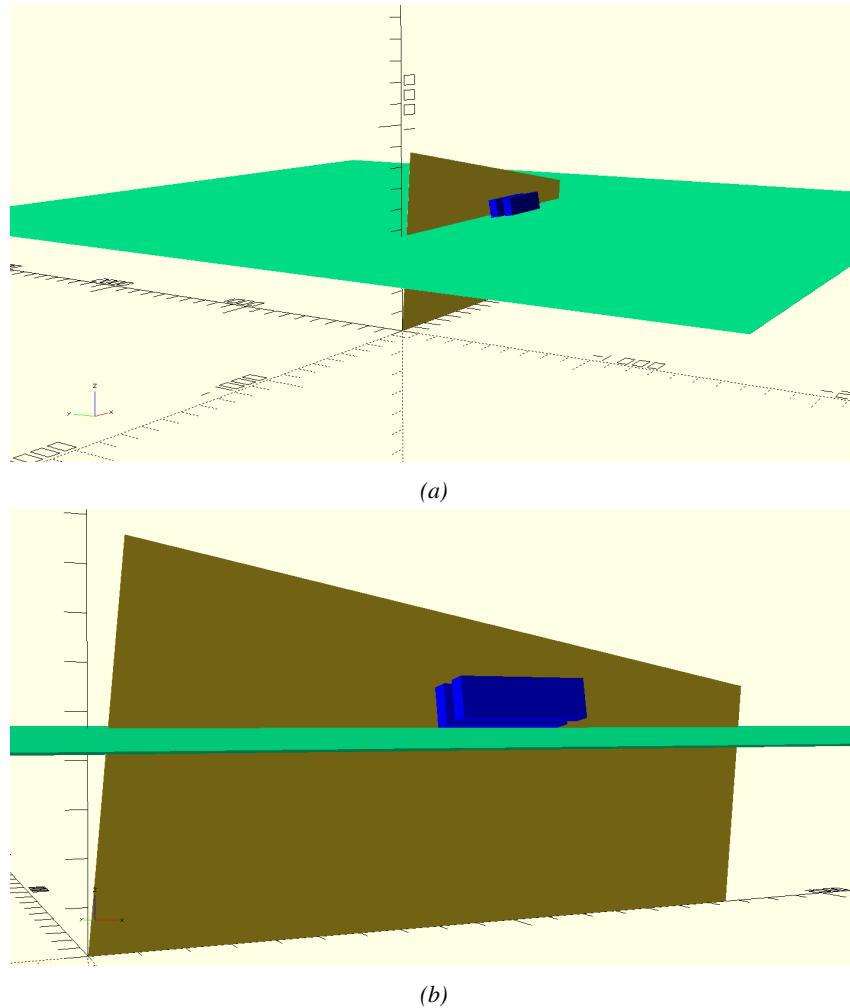
1311 An inclination has been given to the cosmic telescope to maximize the muon flux. A good com-  
 1312 promise had to be found between good enough muon flux and narrow enough hit distribution to  
 1313 be sure to contain all the events into only one half partitions as required from the limited available  
 1314 readout hardware. Nevertheless, a consequence of the misplaced trigger, that can be seen as a loss  
 1315 of events in half-partition B1 in Figure 4.11, is an inefficiency. Nevertheless, the inefficiency of ap-  
 1316 proximately 20 % highlighted in Figure 4.12 by comparing the performance of chamber BARC-161  
 1317 in 904 and at GIF without irradiation seems too important to be explained only by the geometri-  
 1318 cal acceptance of the setup itself. Simulations have been conducted to show how the setup brings  
 1319 inefficiency.



1320 *Figure 4.12: Results are derived from data taken on half-partition B2 only. On the 18<sup>th</sup> of June 2014, data  
 1321 has been taken on chamber RE-2-BARC-161 at building 904 (Prevessin Site) with cosmic muons providing us a  
 1322 reference efficiency plateau of  $(97.54 \pm 0.15)\%$  represented by a black curve. A similar measurement has been  
 1323 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  
 1324 red curve.*

### 1320 4.3.3.1 Description of the simulation layout

1321 The layout of GIF setup has been reproduced and incorporated into a Monte Carlo (MC) simulation  
 1322 to study the influence of the disposition of the telescope on the final distribution measured by the  
 1323 RPC. A 3D view of the simulated layout is given into Figure 4.13. Muons are generated randomly  
 1324 in a horizontal plane located at a height corresponding to the lowest point of the PMTs. This way,  
 1325 the needed size of the plane in order to simulate events happening at very big azimuthal angles (i.e.  
 1326  $\theta \approx \pi$ ) can be kept relatively small. The muon flux is designed to follow the usual  $\cos^2\theta$  distribution  
 1327 for cosmic particle. The goal of the simulation is to look at muons that pass through the muon  
 1328 telescope composed of the two scintillators and define their distribution onto the RPC plane. During  
 1329 the reconstruction, the RPC plane is then divided into its strips and each muon track is assigned to a  
 1330 strip.



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

1331 In order to further refine the quality of the simulation and understand deeper the results the  
 1332 dependance of the distribution has been studied for a range of telescope inclinations. Moreover,  
 1333 the threshold applied on the PMT signals has been included into the simulation in the form of a  
 1334 cut. In the approximation of uniform scintillators, it has been considered that the threshold can be  
 1335 understood as the minimum distance particles need to travel through the scintillating material to give  
 1336 a strong enough signal. Particles that travel a distance smaller than the set "threshold" are thus not  
 1337 detected by the telescope and cannot trigger the data taking. Finally, the FEE threshold also has  
 1338 been considered in a similar way. The mean momentum of horizontal cosmic rays is higher than  
 1339 those of vertical ones but the stopping power of matter for momenta ranging from 1 GeV to 1 TeV  
 1340 stays comparable. It is then possible to assume that the mean number of primary  $e^-$ /ion pairs per  
 1341 unit length will stay similar and thus, depending on the applied discriminator threshold, muons with

1342 the shortest path through the gas volume will deposit less charge and induce a smaller signal on the  
 1343 pick-up strips that could eventually not be detected. These two thresholds also restrain the overall  
 1344 geometrical acceptance of the system.

1345 **4.3.3.2 Simulation procedure**

1346 The simulation software has been designed using C++ and the output data is saved into ROOT  
 1347 histograms. Simulations start for a threshold  $T_{scint}$  varying in a range from 0 to 45 mm in steps  
 1348 of 5 mm, where  $T_{scint} = 0$  mm corresponds to the case where there isn't any threshold apply on  
 1349 the input signal while  $T_{scint} = 45$  mm, which is the scintillator thickness, is the case where muons  
 1350 cannot arrive orthogonally onto the scintillator surface. For a given  $T_{scint}$ , a set of RPC thresholds  
 1351 are considered. The RPC threshold,  $T_{RPC}$  varies from 2 mm, the thickness of the gas volume, to  
 1352 3 mm in steps of 0.25 mm. For each  $(T_{scint}; T_{RPC})$  pair,  $N_\mu = 10^8$  muons are randomly generated  
 1353 inside the muon plane described in the previous paragraph with an azimuthal angle  $\theta$  chosen to follow  
 1354 a  $\cos^2\theta$  distribution.

1355 Planes are associated to each surface of the scintillators. Knowing muon position into the muon  
 1356 plane and its direction allows us, by assuming that muons travel in a straight line, to compute the  
 1357 intersection of the muon track with these planes. Applying conditions to the limits of the surfaces  
 1358 of the scintillator faces then gives us an answer to whether or not the muon passed through the  
 1359 scintillators. In the case the muon has indeed passed through the telescope, the path through each  
 1360 scintillator is computed and muons whose path was shorter than  $T_{scint}$  are rejected and are thus  
 1361 considered as having not interacted with the setup.

1362 On the contrary, if the muon is labeled as good, its position within the RPC plane is computed  
 1363 and the corresponding strip, determined by geometrical tests in the case the distance through the  
 1364 gas volume was enough not to be rejected because of  $T_{RPC}$ , gets a hit and several histograms  
 1365 are filled in order to keep track of the generation point on the muon plane, the intersection points  
 1366 of the reconstructed muons within the telescope, or on the RPC plane, the path traveled through  
 1367 each individual scintillator or the gas volume, as well as other histograms. Moreover, muons fill  
 1368 different histograms whether they are forward or backward coming muons. They are discriminated  
 1369 according to their direction components. When a muon is generated, an  $(x, y, z)$  position is assigned  
 1370 into the muon plane as well as a  $(\theta; \phi)$  pair that gives us the direction it's coming from. This way,  
 1371 muons satisfying the condition  $0 \leq \phi < \pi$  are designated as backward coming muons while muons  
 1372 satisfying  $\pi \leq \phi < 2\pi$  as forward coming muons.

1373 This simulation is then repeated for different telescope inclinations ranging in between 4 and 20°  
 1374 and varying in steps of 2°. Due to this inclination and to the vertical position of the detector under  
 1375 test, the muon distribution reconstructed in the detector plane is asymmetrical. The choice as been  
 1376 made to chose a skew distribution formula to fit the data built as the multiplication of gaussian and  
 1377 sigmoidal curves together. A typical gaussian formula is given as 4.1 and has three free parameters  
 1378 as  $A_g$ , its amplitude,  $\bar{x}$ , its mean value and  $\sigma$ , its root mean square. Sigmoidal curves as given by  
 1379 formula 4.2 are functions converging to 0 and  $A_s$  as  $x$  diverges. The inflection point is given as  $x_i$   
 1380 and  $\lambda$  is proportional to the slope at  $x = x_i$ . In the limit where  $\lambda \rightarrow \infty$ , the sigmoid becomes a  
 1381 step function.

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

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

1382 Finally, a possible representation of a skew distribution is given by formula 4.3 and is the product  
 1383 of 4.1 and 4.2. Naturally, here  $A_{sk} = A_g \times A_s$  and represents the theoretical maximum in the limit  
 1384 where the skew tends to a gaussian function.

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

### 1385 4.3.3.3 Results

#### 1386 Influence of $T_{scint}$ on the muon distribution

#### 1387 Influence of $T_{RPC}$ on the muon distribution

#### 1388 Influence of the telescope inclination on the muon distribution

#### 1389 Comparison to data taken at GIF without irradiation

### 1390 4.3.4 Photon flux at GIF

#### 1391 4.3.4.1 Expectations from simulations

1392 In order to understand and evaluate the  $\gamma$  flux in the GIF area, simulations had been conducted in  
 1393 1999 and published by S. Agosteo et al [55]. Table 4.1 presented in this article gives us the  $\gamma$  flux  
 1394 for different distances  $D$  to the source. This simulation was done using GEANT and a Monte Carlo  
 1395 N-Particle (MCNP) transport code, and the flux  $F$  is given in number of  $\gamma$  per unit area and unit time  
 1396 along with the estimated error from these packages expressed in %.

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

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

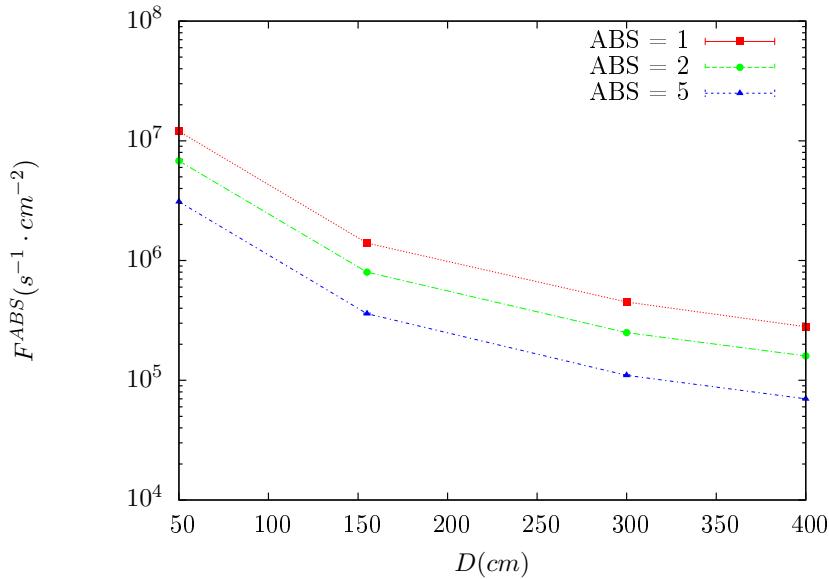


Figure 4.14:  $\gamma$  flux  $F(D)$  is plot using values from table 4.1. As expected, the plot shows similar attenuation behaviours with increasing distance for each absorption factors.

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

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

By rewriting Equation 4.4, it comes that :

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

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

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

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

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

<sup>1408</sup> For the range of  $D/D_0$  values available, it is possible to use a simple linear fit to get the evolution  
<sup>1409</sup> of  $c$ . The linear fit will then use only 2 free parameters,  $a$  and  $b$ , as written in Equation 4.7. This gives  
<sup>1410</sup> us the results showed in Figure 4.15. Figure 4.15b confirms that using only a linear fit to extract  $c$  is  
<sup>1411</sup> enough as the evolution of the rate that can be obtained superimposes well on the simulation points.

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

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

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

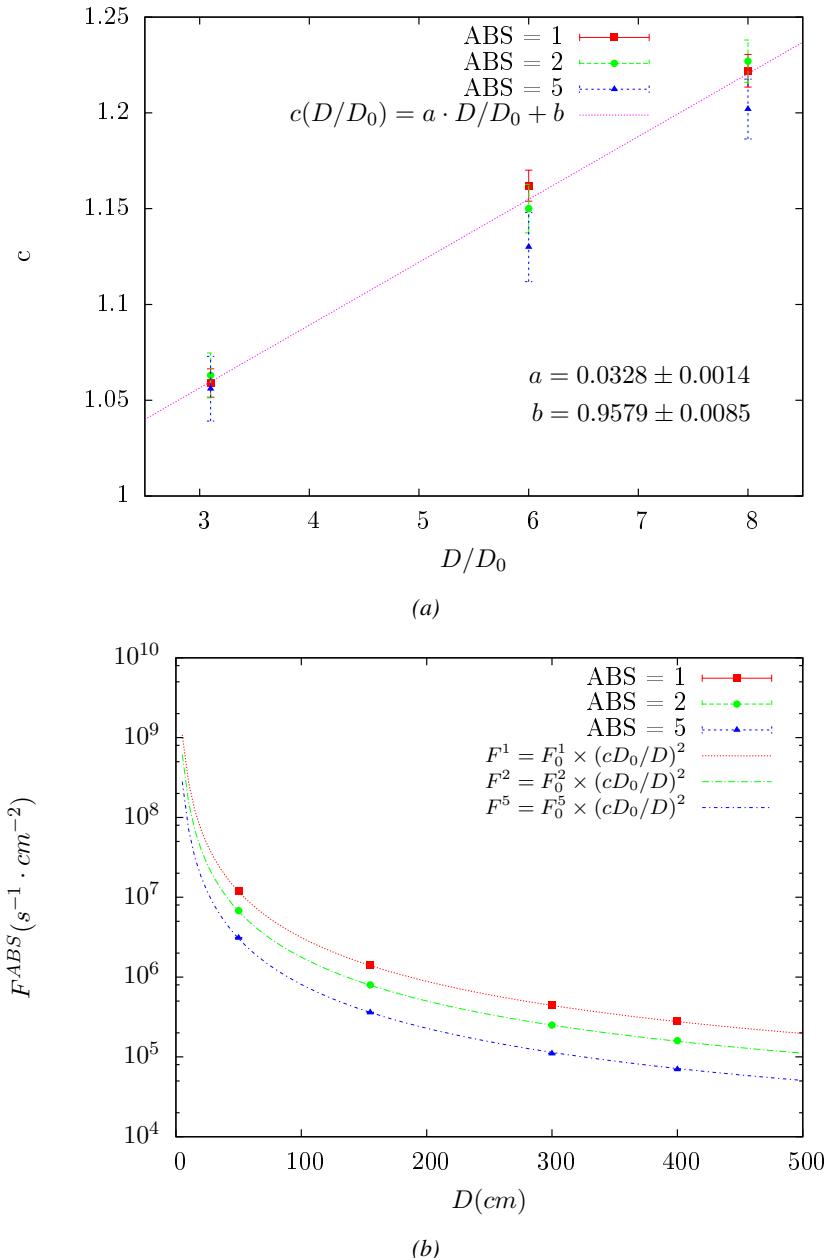


Figure 4.15: Figure 4.15a shows the linear approximation fit done via formulae 4.7 on data from table 4.2. Figure 4.15b shows a comparison of this model with the simulated flux using  $a$  and  $b$  given in figure 4.15a in formulae 4.4 and the reference value  $D_0 = 50\text{cm}$  and the associated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

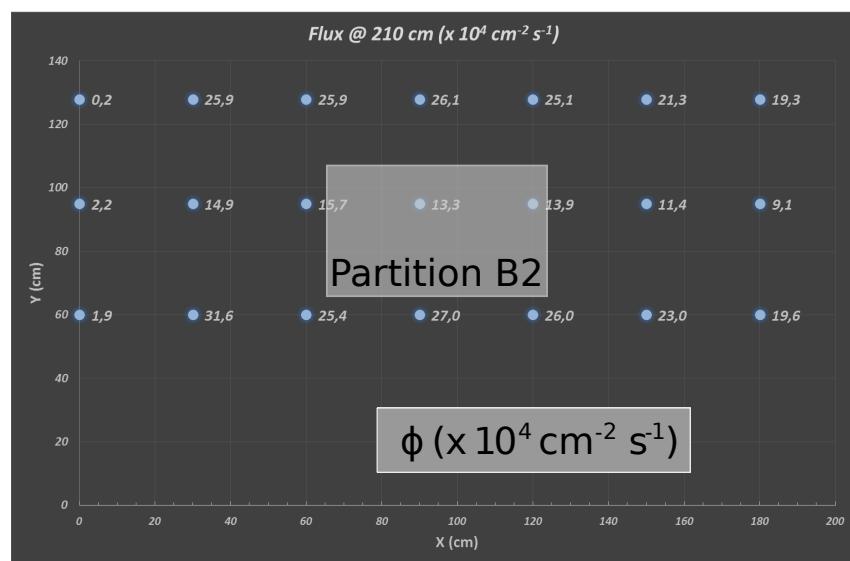
In the case of the 2014 GIF tests, the RPC plane is located at a distance  $D = 206\text{ cm}$  to the source.  
 Moreover, to estimate the strength of the flux in 2014, it is necessary to consider the nuclear decay through time associated to the Cesium source whose half-life is well known ( $t_{1/2}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

1416 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  
 1417 resulting in an attenuation of the activity from 740 GBq in 1997 to 494 GBq in 2014. All the needed  
 1418 information to extrapolate the flux through our detector in 2014 has now been assembled, leading  
 1419 to the Table 4.3. It is interesting to note that for a common RPC sensitivity to  $\gamma$  of  $2 \cdot 10^{-3}$ , the  
 1420 order of magnitude of the estimated hit rate per unit area is of the order of the kHz for the fully  
 1421 opened source. Moreover, taking profit of the two working absorbers, it will be possible to scan  
 1422 background rates at 0 Hz,  $\sim 300$  Hz as well as  $\sim 600$  Hz. Without source, a good estimate of the  
 1423 intrinsic performance will be available. Then at 300 Hz, the goal will be to show that the detectors  
 1424 fulfill the performance certification of CMS RPCs. Then a first idea of the performance of the  
 1425 detectors at higher background will be provided with absorption factors 2 ( $\sim 600$  Hz) and 1 (no  
 1426 absorption). *[Here I will also put a reference to the plot showing the estimated background rate at  
 1427 the level of RE3/1 in the case of HL-LHC but this one being in another chapter, I will do it later.]*

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

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

1428 **4.3.4.2 Dose measurements**



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

<sup>1429</sup> **4.3.5 Results and discussions**

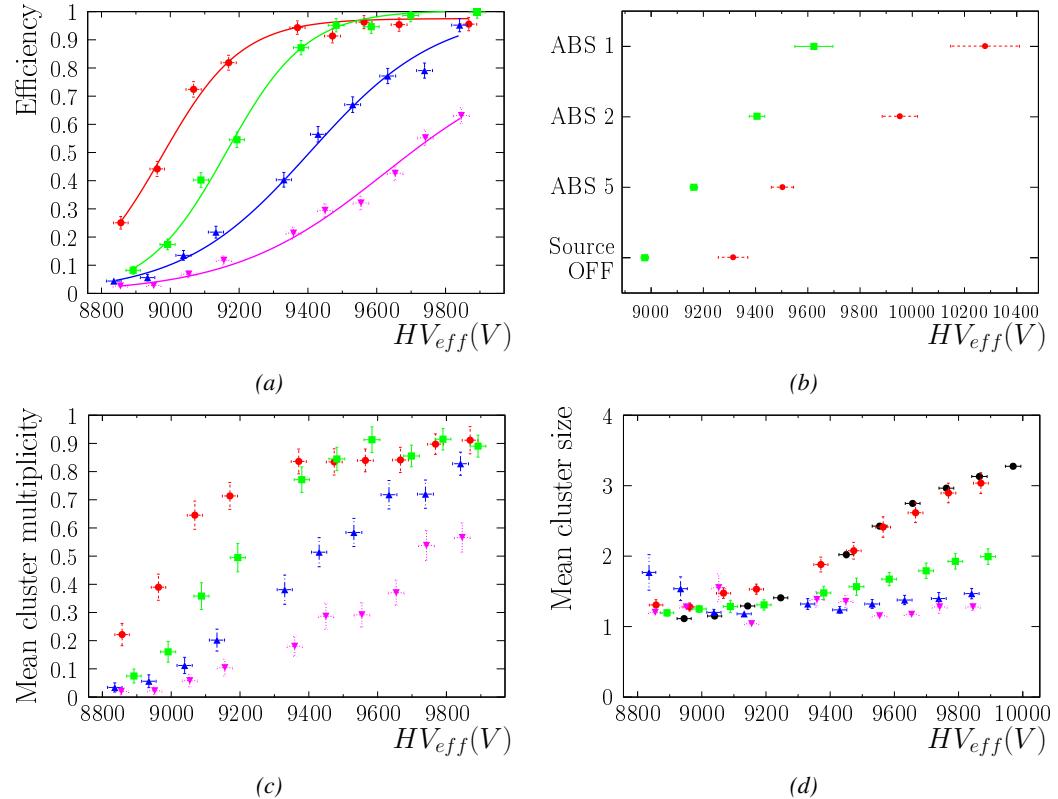


Figure 4.17

<sup>1430</sup> **4.4 Longevity tests at GIF++**

<sup>1431</sup> Longevity studies imply a monitoring of the performance of the detectors probed using a high inten-  
<sup>1432</sup> sity muon beam in a irradiated environment by periodically measuring their rate capability, the dark  
<sup>1433</sup> current running through them and the bulk resistivity of the Bakelite composing their electrodes.  
<sup>1434</sup> GIF++, with its very intense  $^{137}\text{Cs}$  source, provides the perfect environment to perform such kind  
<sup>1435</sup> of tests. Assuming a maximum acceleration factor of 3, it is expected to accumulate the equivalent  
<sup>1436</sup> charge in 1.7 years.

<sup>1437</sup> As the maximum background is found in the endcap, the choice naturally was made to focus the  
<sup>1438</sup> GIF++ longevity studies on endcap chambers. Most of the RPC system was installed in 2007. Nev-  
<sup>1439</sup> ertheless, the large chambers in the fourth endcap (RE4/2 and RE4/3) have been installed during  
<sup>1440</sup> LS1 in 2014. The Bakelite of these two different productions having different properties, four spare  
<sup>1441</sup> chambers of the present system were selected, two RE2,3/2 spares and two RE4/2 spares. Having  
<sup>1442</sup> two chambers of each type allows to always keep one of them non irradiated as reference, the per-  
<sup>1443</sup> formance evolution of the irradiated chamber being then compared through time to the performance  
<sup>1444</sup> of the non irradiated one.

<sup>1445</sup> The performance of the detectors under different level of irradiation is measured periodically dur-  
<sup>1446</sup> ing dedicated test beam periods using the H4 muon beam. In between these test beam periods, the  
<sup>1447</sup> two RE2,3/2 and RE4/2 chambers selected for this study are irradiated by the  $^{137}\text{Cs}$  source in order  
<sup>1448</sup> to accumulate charge and the gamma background is monitored, as well as the currents. The two  
<sup>1449</sup> remaining chambers are kept non-irradiated as reference detectors. Due to the limited gas flow in  
<sup>1450</sup> GIF++, the RE4 chamber remained non-irradiated until end of November 2016 where a new mass  
<sup>1451</sup> flow controller has been installed allowing for bigger volumes of gas to flow in the system.

<sup>1452</sup> Figures 4.18 and 4.19 give us for different test beam periods, and thus for increasing integrated  
<sup>1453</sup> charge through time, a comparison of the maximum efficiency, obtained using a sigmoid-like func-  
<sup>1454</sup> tion, and of the working point of both irradiated and non irradiated chambers [**SIGMOID2005**]. No  
<sup>1455</sup> aging is yet to see from this data, the shifts in  $\gamma$  rate per unit area in between irradiated and non  
<sup>1456</sup> irradiated detectors and RE2 and RE4 types being easily explained by a difference of sensitivity due  
<sup>1457</sup> to the various Bakelite resistivities of the HPL electrodes used for the electrode production.

<sup>1458</sup> Collecting performance data at each test beam period allows us to extrapolate the maximum effi-  
<sup>1459</sup> ciency for a background hit rate of  $300 \text{ Hz/cm}^2$  corresponding to the expected HL-LHC conditions.  
<sup>1460</sup> Aging effects could emerge from a loss of efficiency with increasing integrated charge over time,  
<sup>1461</sup> thus Figure 4.20 helps us understand such degradation of the performance of irradiated detectors in  
<sup>1462</sup> comparison with non irradiated ones. The final answer for an eventual loss of efficiency is given in  
<sup>1463</sup> Figure 4.21 by comparing for both irradiated and non irradiated detectors the efficiency sigmoids  
<sup>1464</sup> before and after the longevity study. Moreover, to complete the performance information, the Bake-  
<sup>1465</sup> lite resistivity is regularly measured thanks to  $Ag$  scans (Figure 4.22) and the noise rate is monitored  
<sup>1466</sup> weekly during irradiation periods (Figure 4.23). At the end of 2016, no signs of aging were observed  
<sup>1467</sup> and further investigation is needed to get closer to the final integrated charge requirements proposed  
<sup>1468</sup> for the longevity study of the present CMS RPC sub-system.

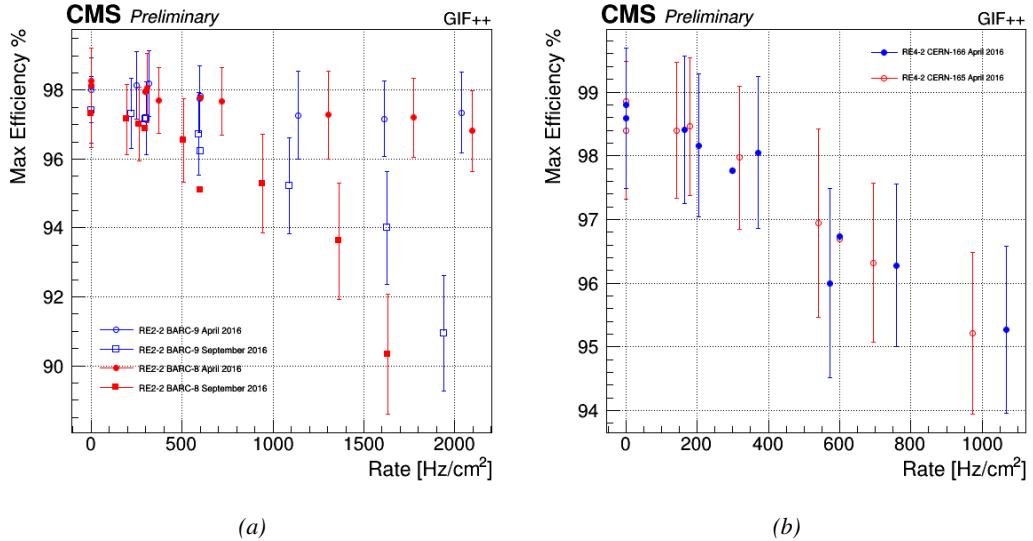


Figure 4.18: Evolution of the maximum efficiency for RE2 (4.18a) and RE4 (4.18b) chambers with increasing extrapolated  $\gamma$  rate per unit area at working point. Both irradiated (blue) and non irradiated (red) chambers are shown.

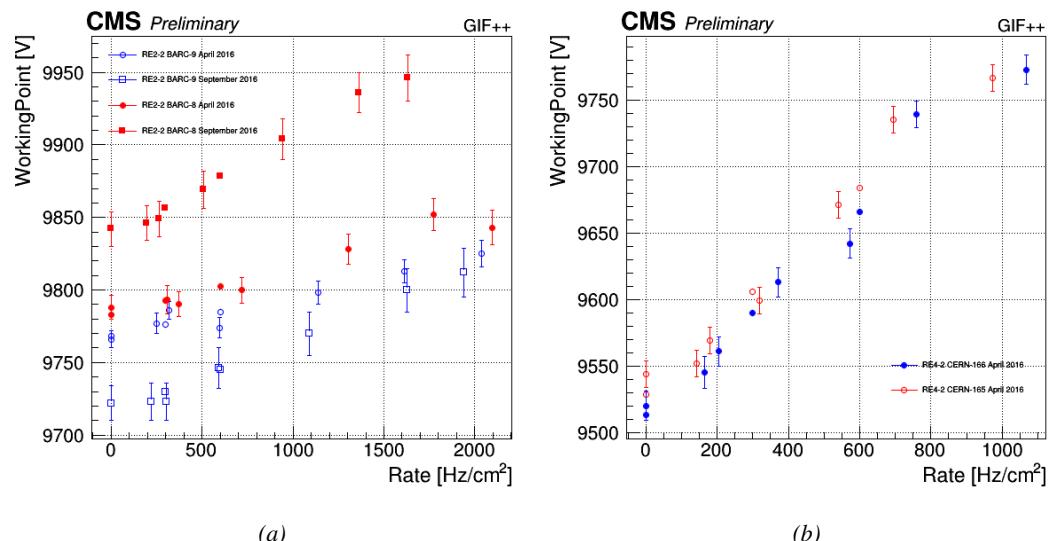
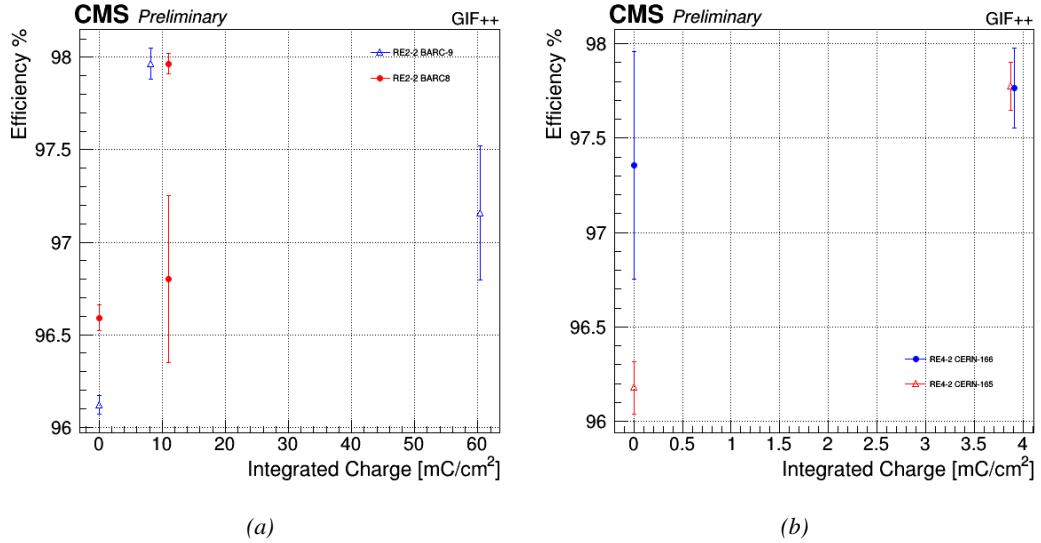
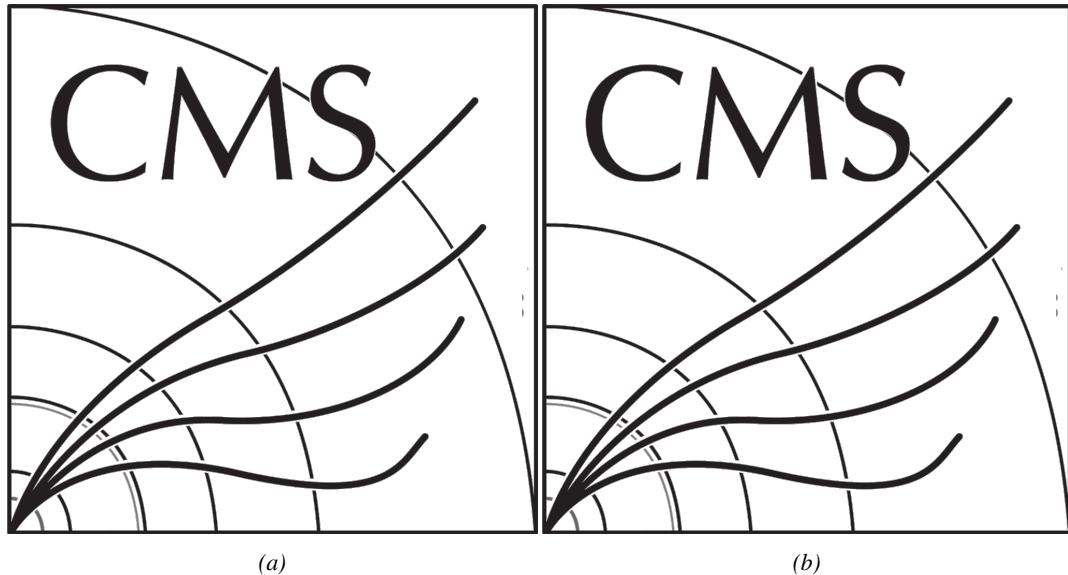


Figure 4.19: Evolution of the working point for RE2 (4.19a) and RE4 (4.19b) with increasing extrapolated  $\gamma$  rate per unit area at working point. Both irradiated (blue) and non irradiated (red) chambers are shown.



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



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

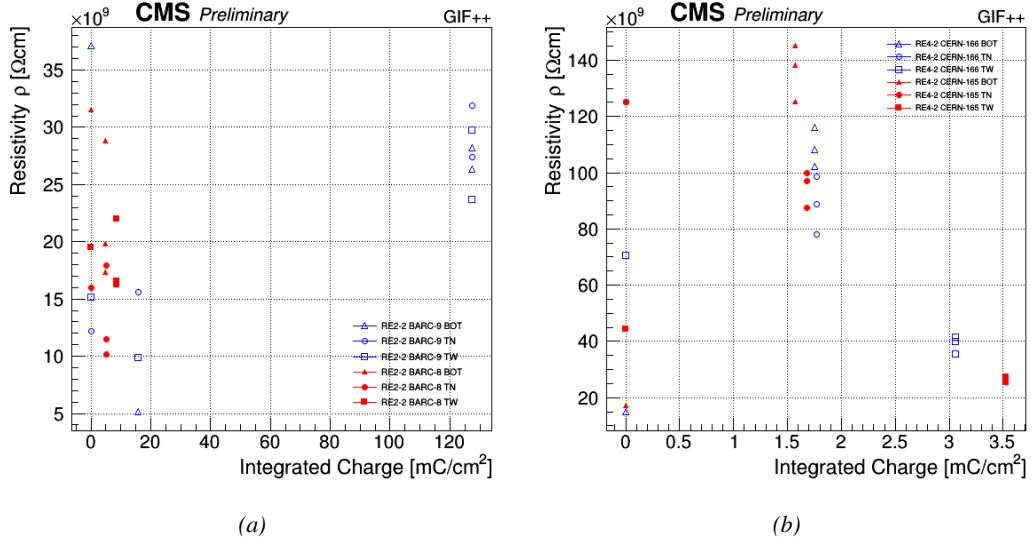


Figure 4.22: Evolution of the Bakelite resistivity for RE2 (4.22a) and RE4 (4.22b) detectors. Both irradiated (blue) and non-irradiated (red) chambers are shown.

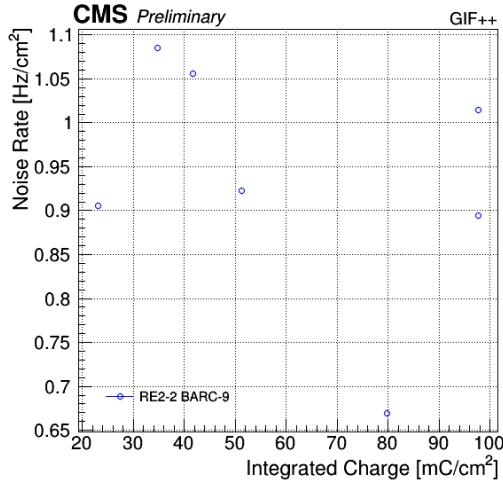


Figure 4.23: Evolution of the noise rate per unit area for the irradiated chamber RE2-2-BARC-9 only.

#### 4.4.1 Description of the Data Acquisition

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

1477 moved outside the beam line and is placed in a distance of 5.2 m to the source, which irradiates the  
 1478 bunker using an attenuation filter of 2.2 which corresponds to a fluence of  $10^7 \text{ gamma/cm}^2$ .

1479 During GIF++ operation, the data collected can be divided into different categories as several  
 1480 parameters are monitored in addition to the usual RPC performance data. On one hand, to know  
 1481 the performance of a chamber, it is need to measure its efficiency and to know the background  
 1482 conditions in which it is operated. To do this, the hit signals from the chamber are recorded and  
 1483 stored in a ROOT file via a Data Acquisition (DAQ) software. On the other hand, it is also very  
 1484 important to monitor parameters such as environmental pressure and temperature, gas temperature  
 1485 and humidity, RPC HV, LV, and currents, or even source and beam status. This is done through the  
 1486 GIF++ web Detector Control Software (DCS) that stores this information in a database.

1487 Two different types of tests are conducted on RPCs via the DAQ. Indeed, the performance of the  
 1488 detectors is measured periodically during dedicated test beam periods using the H4 muon beam. In  
 1489 between these test beam periods, when the beam is not available, the chambers are irradiated by the  
 1490  $^{137}\text{Cs}$  in order to accumulate deposited charge and the gamma background is measured.

1491 RPCs under test are connected through LVDS cables to V1190A Time-to-Digital Converter  
 1492 (TDC) modules manufactured by CAEN. These modules, located in the rack area outside of the  
 1493 bunker, get the logic signals sent by the chambers and save them into their buffers. Due to the  
 1494 limited size of the buffers, the collected data is regularly erased and replaced. A trigger signal is  
 1495 needed for the TDC modules to send the useful data to the DAQ computer via a V1718 CAEN USB  
 1496 communication module.

1497 In the case of performance test, the trigger signal used for data acquisition is generated by the  
 1498 coincidence of three scintillators. A first one is placed upstream outside of the bunker, a second one  
 1499 is placed downstream outside of the bunker, while a third one is placed in front of the trolley, close by  
 1500 the chambers. Every time a trigger is sent to the TDCs, i.e. every time a muon is detected, knowing  
 1501 the time delay in between the trigger and the RPC signals, signals located in the right time window  
 1502 are extracted from the buffers and saved for later analysis. Signals are taken in a time window of  
 1503 400 ns centered on the muon peak (here we could show a time spectrum). On the other hand, in the  
 1504 case of background rate measurement, the trigger signal needs to be "random" not to measure muons  
 1505 but to look at gamma background. A trigger pulse is continuously generated at a rate of 300 Hz using  
 1506 a dual timer. To integrate an as great as possible time, all signals contained within a time window of  
 1507 10us prior to the random trigger signal are extracted form the buffers and saved for further analysis  
 1508 (here another time spectrum to illustrate could be useful, maybe even place both spectrum together  
 1509 as a single Figure).

1510 The signals sent to the TDCs correspond to hit collections in the RPCs. When a particle hits  
 1511 a RPC, it induce a signal in the pickup strips of the RPC readout. If this signal is higher than the  
 1512 detection threshold, a LVDS signal is sent to the TDCs. The data is then organised into 4 branches  
 1513 keeping track of the event number, the hit multiplicity for the whole setup, and the time and channel  
 1514 profile of the hits in the TDCs.

#### 1515 4.4.2 RPC current, environmental and operation parameter monitoring

1516 In order to take into account the variation of pressure and temperature between different data taking  
 1517 periods the applied voltage is corrected following the relationship :

$$1518 \quad HV_{eff} = HV_{app} \times \left( 0.2 + 0.8 \cdot \frac{P_0}{P} \times \frac{T}{T_0} \right) \quad (4.10)$$

1518 where  $T_0$  (=293 K) and  $P_0$  (=990 mbar) are the reference values.

1519 **4.4.3 Measurement procedure**

1520 Insert a short description of the online tools (DAQ, DCS, DQM).

1521 Insert a short description of the offline tools : tracking and efficiency algorithm.

1522 Identify long term aging effects we are monitoring the rates per strip.

1523 **4.4.4 Longevity studies results**



# 5

1524

1525

## Investigation on high rate RPCs

1526 **5.1 Rate limitations and ageing of RPCs**

1527 **5.1.1 Low resistivity electrodes**

1528 **5.1.2 Low noise front-end electronics**

1529 **5.2 Construction of prototypes**

1530 **5.3 Results and discussions**



# 6

1531

1532

## Conclusions and outlooks

1533 **6.1 Conclusions**

1534 **6.2 Outlooks**



# A

1535

1536

1537

## A data acquisition software for CAEN VME TDCs

1538 Certifying detectors in the perspective of HL-LHC required to develop tools for the GIF++ experi-  
1539 ment. Among them was the C++ Data Acquisition (DAQ) software that allows to make the commu-  
1540 nications in between a computer and TDC modules in order to retrieve the RPC data [57]. In this  
1541 appendix, details about this software, as of how the software was written, how it functions and how  
1542 it can be exported to another similar setup, will be given.

### 1543 A.1 GIF++ DAQ file tree

1544 GIF++ DAQ source code is fully available on github at [https://github.com/afagot/GIF\\_](https://github.com/afagot/GIF_DAQ)  
1545 DAQ. The software requires 3 non-optional dependencies:

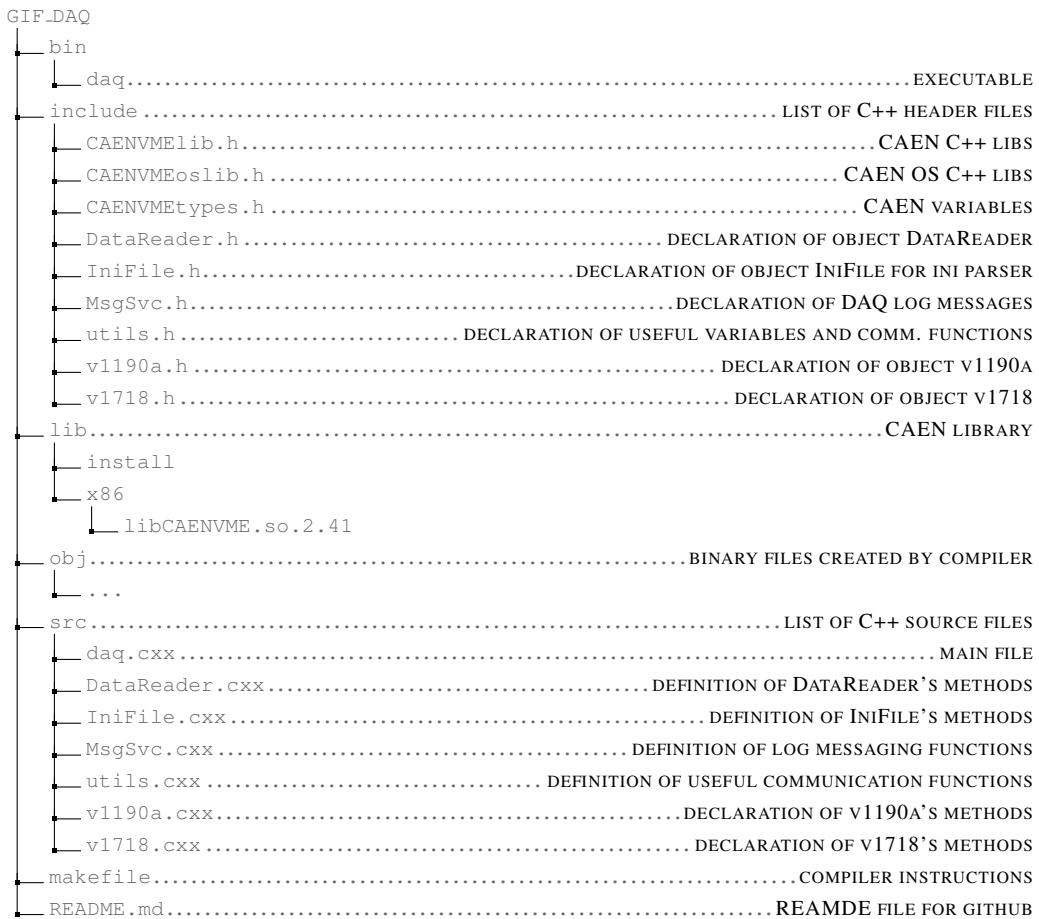
- 1546 • CAEN USB Driver, to mount the VME hardware,  
1547 • CAEN VME Library, to communicate with the VME hardware, and  
1548 • ROOT, to organize the collected data into a TTree.

1549 The CAEN VME library will not be packaged by distributions and will need to be installed man-  
1550 ually. To compile the GIF++ DAQ project via a terminal, from the DAQ folder use the command:

1551      make

1553 The source code tree is provided below along with comments to give an overview of the files' con-  
1554 tent. The different objects created for this project (`v1718`, `v1190a`, `IniFile` & `DataReader`) will be  
1555 described in details in the following sections.

1556



## 1557 A.2 Usage of the DAQ

1558 GIF++ DAQ, as used in GIF++, is not a standalone software. Indeed, the system being more complex,  
 1559 the DAQ only is a sub-layer of the software architecture developed to control and monitor  
 1560 the RPCs that are placed into the bunker for performance study in an irradiated environment. The top  
 1561 layer of GIF++ is a Web Detector Control System (webDCS) application. The DAQ is only called  
 1562 by the webDCS when data needs to be acquired. The webDCS operates the DAQ through command  
 1563 line. To start the DAQ, the webDCS calls:

1564  
 1565   bin/daq /path/to/the/log/file/in/the/output/data/folder

1566 where /path/to/the/log/file/in/the/output/data/folder is the only argument required. This  
 1567 log file is important for the webDCS as this file contains all the content of the communication of the  
 1568 webDCS and the different systems monitored by the webDCS. Its content is constantly displayed  
 1569 during data taking for the users to be able to follow the operations. The communication messages  
 1570 are normally sent to the webDCS log file via the functions declared in file MsgSvc.h, typically  
 1571 MSG\_INFO(string message).

1572

### 1573 A.3 Description of the readout setup

1574 The CMS RPC setup at GIF++ counts 5 V1190A Time-to-Digital Converter (TDC) manufactured  
 1575 by CAEN [58]. V1190A are VME units accepting 128 independent Multi-Hit/Multi-Event TDC  
 1576 channels whose signals are treated by 4 100 ps high performance TDC chips developed by CERN  
 1577 / ECP-MIC Division. The communication between the computer and the TDCs to transfer data is  
 1578 done via a V1718 VME master module also manufactured by CAEN and operated from a USB  
 1579 port [59]. These VME modules are all hosted into a 6U VME 6021 powered crate manufactured by  
 1580 W-Ie-Ne-R than can accommodate up to 21 VME bus cards [60]. These 3 components of the DAQ  
 1581 setup are shown in Figure A.1.

1582

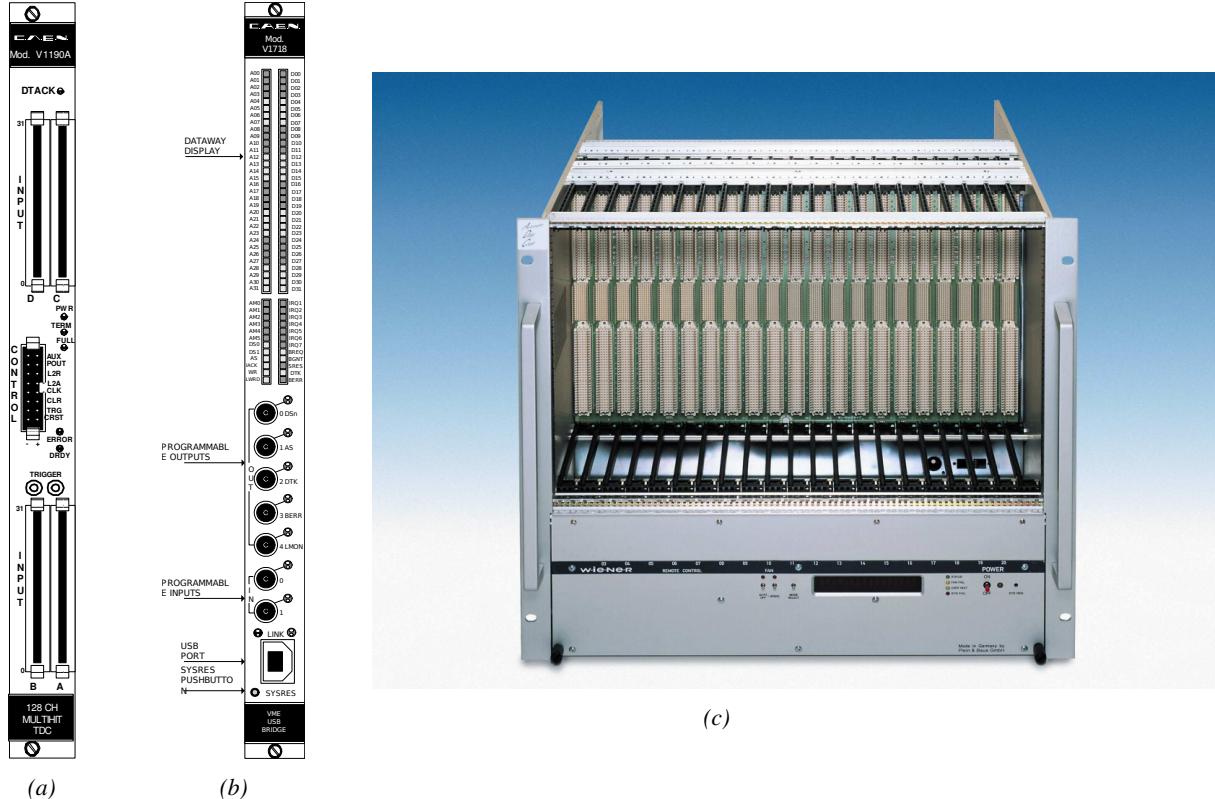


Figure A.1: (A.1a) View of the front panel of a V1190A TDC module [58]. (A.1b) View of the front panel of a V1718 Bridge module [59]. (A.1c) View of the front panel of a 6U 6021 VME crate [60].

1583

### A.4 Data read-out

1584 To efficiently perform a data readout algorithm, C++ objects to handle the VME modules (TDCs  
 1585 and VME bridge) have been created along with objects to store data and read the configuration file

1586 that comes as an input of the DAQ software.

1587

#### 1588 A.4.1 V1190A TDCs

1589 The DAQ used at GIF takes profit of the *Trigger Matching Mode* offered by V1190A modules.  
 1590 This setting is enabled through the method `v1190a::SetTrigMatching (int ntdcs)` where `ntdcs`  
 1591 is the total number of TDCs in the setup this setting needs to be enabled for (Source Code A.1). A  
 1592 trigger matching is performed in between a trigger time tag, a trigger signal sent into the TRIGGER  
 1593 input of the TDC visible on Figure A.1a, and the channel time measurements, signals recorded from  
 1594 the detectors under test in our case. Control over this data acquisition mode, explained through  
 1595 Figure A.2, is offered via 4 programmable parameters:

- 1596 • **match window:** the matching between a trigger and a hit is done within a programmable time  
 1597 window. This is set via the method

1598   `void v1190a::SetTrigWindowWidth(Uint windowHeight, int ntdcs)`

- 1599 • **window offset:** temporal distance between the trigger tag and the start of the trigger matching  
 1600 window. This is set via the method

1601   `void v1190a::SetTrigWindowWidth(Uint windowHeight, int ntdcs)`

- 1602 • **extra search margin:** an extended time window is used to ensure that all matching hits are  
 1603 found. This is set via the method

1604   `void v1190a::SetTrigSearchMargin(Uint searchMargin, int ntdcs)`

- 1605 • **reject margin:** older hits are automatically rejected to prevent buffer overflows and to speed  
 1606 up the search time. This is set via the method

1607   `void v1190a::SetTrigRejectionMargin(Uint rejectMargin, int ntdcs)`

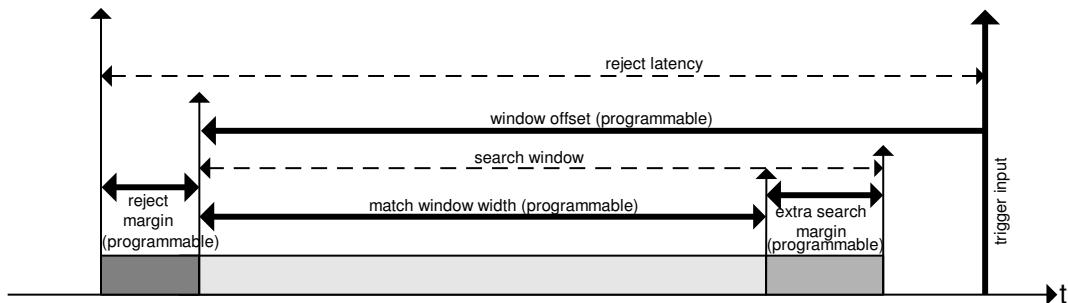


Figure A.2: Module V1190A Trigger Matching Mode timing diagram [58].

1608 Each of these 4 parameters are given in number of clocks, 1 clock being 25 ns long. It is easy to  
 1609 understand at this level that there are 3 possible functioning settings:

- 1610 • **1:** the match window is entirely contained after the trigger signal,

- 1611 • **2:** the match window overlaps the trigger signal, or

- 1612 • **3:** the match window is entirely contained before the trigger signal as displayed on Figure A.2.

1613 In both the first and second cases, the sum of the window width and of the offset can be set to  
1614 a maximum of 40 clocks, which corresponds to 1  $\mu$ s. Evidently, the offset can be negative, allowing  
1615 for a longer match window, with the constraint of having the window ending at most 1  $\mu$ s after the  
1616 trigger signal. In the third case, the maximum negative offset allowed is of 2048 clocks (12 bit) cor-  
1617 responding to 51.2  $\mu$ s, the match window being strictly smaller than the offset. In the case of GIF++,  
1618 the choice has been made to use this last setting by delaying the trigger signal. During the studies  
1619 performed in GIF++, both the efficiency of the RPCs, probed using a muon beam, and the noise or  
1620 gamma background rate are monitored. The extra search and reject margins are left unused.  
1621 To probe the efficiency of RPC detectors, the trigger time tag is provided by the coïncidence of  
1622 scintillators when a bunch of muons passes through GIF++ area is used to trigger the data acquisi-  
1623 tion. For this measurement, it is useful to reduce the match window width only to contain the muon  
1624 information. Indeed, the delay in between a trigger signal and the detection of the corresponding  
1625 muon in the RPC being very contant (typically a few tens of ns due to jitter and cable length), the  
1626 muon signals are very localised in time. Thus, due to a delay of approximalety 325 ns in between  
1627 the muons and the trigger, the settings where chosen to have a window width of 24 clocks (600 ns)  
1628 centered on the muon peak thanks to a negative offset of 29 clocks (725 ns).  
1629 On the otherhand, monitoring the rates don't require for the DAQ to look at a specific time window.  
1630 It is important to integrate enough time to have a robust measurement of the rate as the number of  
1631 hits per time unit. The triggerring signal is provided by a pulse generator at a frequency of 300 Hz  
1632 to ensure that the data taking occurs in a random way, with respect to beam physics, to probe only  
1633 the irradiation spectrum on the detectors. The match window is set to 400 clocks (10  $\mu$ s) and the  
1634 negative offset to 401 clocks as it needs to exceed the value of the match window.

```

1635
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);
};

1636

```

1637       *Source Code A.1: Description of C++ object v1190a.*

1638       The v1190a object, defined in the DAQ software as in Source Code A.1, offers the possibility to  
 1639       concatenate all TDCs in the readout setup into a single object containing a list of hardware addresses  
 1640       (addresses to access the TDCs' buffer through the VME crate) and each constructor and method acts  
 1641       on the list of TDCs.

1642

#### 1643     A.4.2 DataReader

1644       Enabled thanks to v1190a::SetBlockTransferMode(Data16 mode, int ntdcs), the data transfer  
 1645       is done via Block Transfer (BLT). Using BLT allows to tranfer a fixed number of events called a  
 1646       *block*. This is used together with an Almost Full Level (AFL) of the TDCs' output buffers, defined

1647 through `v1190a::SetIRQ(Data32 level, Data32 count, int ntdcs)`. This AFL gives the maxi-  
 1648 mum amount of 32735 words (16 bits, corresponding to the depth of a TDC output buffer) that can  
 1649 writen in a buffer before an Interrupt Request (IRQ) is generated and seen by the VME Bridge,  
 1650 stopping the data acquisition to transfer the content of each TDC buffers before resuming. For each  
 1651 trigger, 6 words or more are written into the TDC buffer:

- 1652     • a **global header** providing information of the event number since the beginning of the data  
       acquisition,
- 1654     • a **TDC header**,
- 1655     • the **TDC data** (*if any*), 1 for each hit recorded during the event, providing the channel and the  
       time stamp associated to the hit,
- 1657     • a **TDC error** providing error flags,
- 1658     • a **TDC trailer**,
- 1659     • a **global trigger time tag** that provides the absolute trigger time relatively to the last reset,  
       and
- 1661     • a **global trailer** providing the total word count in the event.

1662     As previously described in Section ??, CMS RPC FEEs provide us with 100 ns long LVDS out-  
 1663 put signals that are injected into the TDCs' input. Any avalanche signal that gives a signal above the  
 1664 FEEs threshold is thus recorded by the TDCs as a hit within the match window. Each hit is assigned  
 1665 to a specific TDC channel with a time stamp, with a precision of 100 ps. The reference time,  $t_0 = 0$ ,  
 1666 is provided by the beginning of the match window. Thus for each trigger, coming from a scintillator  
 1667 coïncidence or the pulse generator, a list of hits is stored into the TDCs' buffers and will then be  
 1668 transferred into a ROOT Tree.

1669     When the BLT is used, it is easy to understand that the maximum number of words that have  
 1670 been set as ALF will not be a finite number of events or, at least, the number of events that would  
 1671 be recorded into the TDC buffers will not be a multiple of the block size. In the last BLT cycle to  
 1672 tranfer data, the number of events to transfer will most propably be lower than the block size. In that  
 1673 case, the TDC can add fillers at the end of the block but this option requires to send more data to the  
 1674 computer and is thus a little slower. Another solution is to finish the transfer after the last event by  
 1675 sending a bus error that states that the BLT reached the last event in the pile. This method has been  
 1677 chosen in GIF++.

1678     Due to irradiation, an event in GIF++ can count up to 300 words per TDC. A limit of 4096 words  
 1679 (12 bits) has been set to generate IRQ which represent from 14 to almost 700 events depending on  
 1680 the average of hits collected per event. Then the block size has been set to 100 events with enabled  
 1682 bus errors. When an AFL is reached for one of the TDCs, the VME bridge stops the acquisition by  
 1683 sending a BUSY signal.

1685     The data is then transferred one TDC at a time into a structure called `RAWData` (Source Code A.2).

```
1686
1687     struct RAWData{
1688         vector<int>           *EventList;
1689         vector<int>           *NHitsList;
1690         vector<int>           *QFlagList;
1691         vector<vector<int>>   *Channellist;
1692         vector<vector<float>>  *TimeStampList;
1693     };
```

1688                 *Source Code A.2: Description of data holding C++ structure `RAWData`.*

1689     In order to organize the data transfer and the data storage, an object called `DataReader` was  
1690     created (Source Code A.3). On one hand, it has `v1718` and `v1190a` objects as private members for  
1691     communication purposes, such as VME modules settings via the configuration file `*iniFile` or data  
1692     read-out through `v1190a::Read()` and on the other hand, it contains the structure `RAWData` that allows  
1693     to organise the data in vectors reproducing the tree structure of a ROOT file.

```
1694
1695     class DataReader
1696     {
1697         private:
1698             bool      StopFlag;
1699             IniFile *iniFile;
1700             Data32  MaxTriggers;
1701             v1718   *VME;
1702             int      nTDCs;
1703             v1190a  *TDCs;
1704             RAWData TDCData;
1705
1706         public:
1707             DataReader();
1708             virtual ~DataReader();
1709             void      SetIniFile(string inifilename);
1710             void      SetMaxTriggers();
1711             Data32  GetMaxTriggers();
1712             void      SetVME();
1713             void      SetTDC();
1714             int      GetQFlag(Uint it);
1715             void      Init(string inifilename);
1716             void      FlushBuffer();
1717             void      Update();
1718             string  GetFileName();
1719             void      WriteRunRegistry(string filename);
1720             void      Run();
1721     };
```

1696                 *Source Code A.3: Description of C++ object `DataReader`.*

1697     Each event is transferred from `TDCData` and saved into branches of a ROOT `TTree` as 3 integers  
1698     that represent the event ID (`EventCount`), the number of hits read from the TDCs (`nHits`), and the  
1699     quality flag that provides information for any problem in the data transfer (`qflag`), and 2 lists of  
1700     `nHits` elements containing the fired TDC channels (`TDCCh`) and their respective time stamps (`TDCTS`),  
1701     as presented in Source Code A.4. The ROOT file file is named using information contained into  
1702     the configuration file, presented in section A.5.2. The needed information is extracted using method  
1703     `DataReader::GetFileName()` and allow to build the output filename format `ScanXXXXXX_HVX_DAQ.root`

1704 where ScanXXXXXX is a 6 digit number representing the scan number into GIFT++ database and HVX  
 1705 the HV step within the scan that can be more than a single digit. An example of ROOT data file is  
 1706 provided with Figure A.3.

```
1707
RAWData TDCData;
TFile *outputFile = new TFile(outputFileName.c_str(),"recreate");
TTree *RAWDataTree = new TTree("RAWData","RAWData");

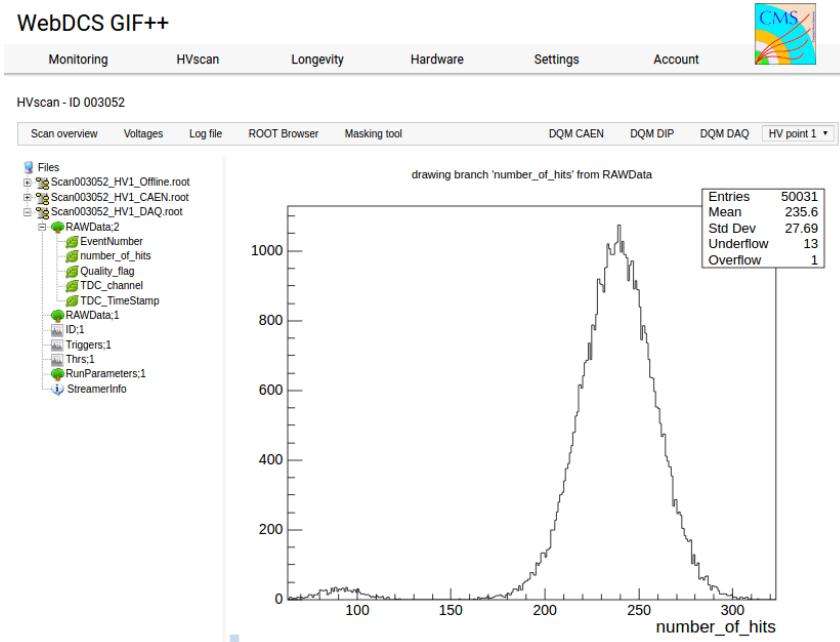
int EventCount = -9;
int nHits = -8;
int qflag = -7;
vector<int> TDCCh;
vector<float> TDCTS;

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);

1708
//...
//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();
}
```

1709 *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.*

#### 1710 A.4.3 Data quality flag

1711 Among the parameters that are recorded for each event, the quality flag, defined in Source Code A.5,  
 1712 is determined on the fly by checking the data recorded by every single TDC. From method `v1190a::Read()`,  
 1713 it can be understood that the content of each TDC buffer is readout one TDC at a time. Entries are  
 1714 created in the data list for the first TDC and then, when the second buffer is readout, events corre-  
 1715 sponding to entries that have already been created to store data for the previous TDC are added to  
 1716 the existing list element. On the contrary, when an event entry has not been yet created in the data  
 1717 list, a new entry is created.

```
1718 typedef enum _QualityFlag {
 1719     GOOD      = 1,
     CORRUPTED = 0
 } QualityFlag;
```

1720 *Source Code A.5: Definition of the quality flag `enum`.*

1721 It is possible that each TDC buffer contains a different number of events. In cases where the first  
 1722 element in the buffer list is an event for corresponds to a new entry, the difference in between the  
 1723 entry from the buffer and the last entry in the data list is recorded and checked. If it is greater than 1,  
 1724 what should never be the case, the quality flag is set to CORRUPTED for this TDC and an empty entry  
 1725 is created in the place of the missing ones. Missing entries are believe to be the result of a bad hold

1726 on the TDC buffers at the moment of the readout. Indeed, the software hold is effective only on 1  
 1727 TDC at a time and no solution as been found yet to completely block the writting in the buffers when  
 1728 an IRQ is received.

1729 At the end of each BLT cycle, the ID of the last entry stored for each TDC buffer is not recorded.  
 1730 When starting the next cycle, if the first entry in the pile corresponds to an event already existing  
 1731 in the list, the readout will start from this list element and will not be able to check the difference  
 1732 in between this entry's ID and the one of the last entry that was recorded for this TDC buffer in  
 1733 the previous cycle. In the case events were missing, the flag stays at its initial value of 0, which is  
 1734 similar to CORRUPTED and it is assumed that then this TDC will not contribute to `number_of_hits`,  
 1735 `TDC_channel` or `TDC_TimeStamp`.

1736 Finally, since there will be 1 `RAWData` entry per TDC for each event (meaning `nTDCs` entries,  
 1737 referring to `DataReader` private attribute), the individual flags of each TDC will be added together.  
 1738 The final format is an integer composed `nTDCs` digits where each digit is the flag of a specific TDC.  
 1739 This is constructed using powers of 10 like follows:

```
1740     TDC 0: QFlag = 100 × _QualityFlag
1741     TDC 1: QFlag = 101 × _QualityFlag
1742     ...
1743     TDC N: QFlag = 10N × _QualityFlag
```

1744 and the final flag to be with N digits:

```
1745     QFlag = n....3210
```

1746 each digit being 1 or 0. Below is given an example with a 4 TDCs setup.

```
1747     If all TDCs were good : QFlag = 1111,
1748     but if TDC 2 was corrupted : QFlag = 1011.
```

1749 When data taking is over and the data contained in the dynamical `RAWData` structure is transferred  
 1750 to the ROOT file, all the 0s are changed into 2s by calling the method `DataReader::GetQFlag()`.  
 1751 This will help translating the flag without knowing the number of TDCs beforehand. Indeed, a flag  
 1752 111 could be due to a 3 TDC setup with 3 good individual TDC flags or to a more than 3 TDC setup  
 1753 with TDCs those ID is greater than 2 being CORRUPTED, thus giving a 0.

1754 The quality flag has been introduced quite late, in October 2017 only, to the list of GIFT++ DAQ  
 1755 parameters to be recorded into the output ROOT file. Before this addition, the missing data, corrupting  
 1756 the quality for the offline analysis, was contributing to artificially fill data with lower multiplicity.  
 1757 Looking at `TBranch number_of_hits` provides an information about the data of the full GIFT++  
 1758 setup. When a TDC is not able to transfer data for a specific event, the effect is a reduction of the  
 1759 total number of hits recorded in the full setup, this is what can be seen from Figure A.4. After offline  
 1760 reconstruction detector by detector, the effect of missing events can be seen in the artificially filled  
 1761 bin at multiplicity 0 shown in Figure A.5. Nonetheless, for data with high irradiation levels, as it is  
 1762 the case for Figure A.5a, discarding the fake multiplicity 0 data can be done easily during the offline  
 1763 analysis. At lower radiation, the missing events contribution becomes more problematic as the multi-  
 1764 tiplicity distribution overlaps the multiplicity 0 and that in the same time the proportion of missing

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

1768

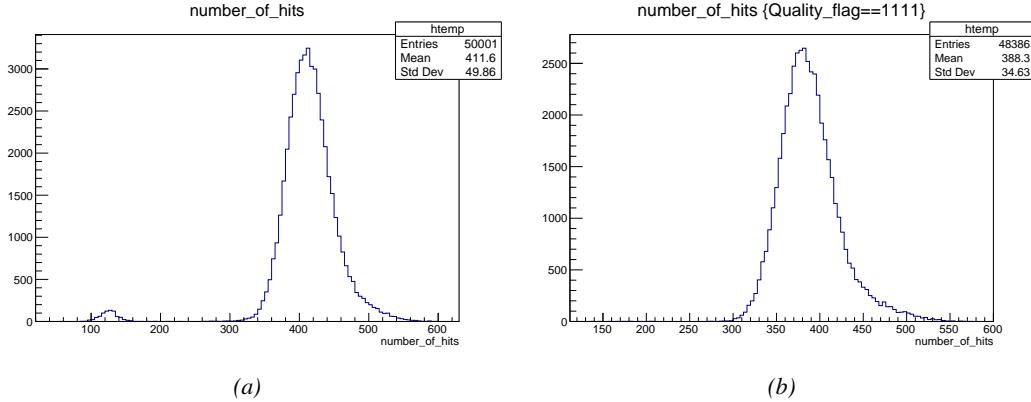


Figure A.4: The effect of the quality flag is explained by presenting the content of TBranch `number_of_hits` of a data file without `Quality_flag` in Figure A.4a and the content of the same TBranch for data corresponding to a `Quality_flag` where all TDCs were labelled as `GOOD` in Figure A.4b taken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

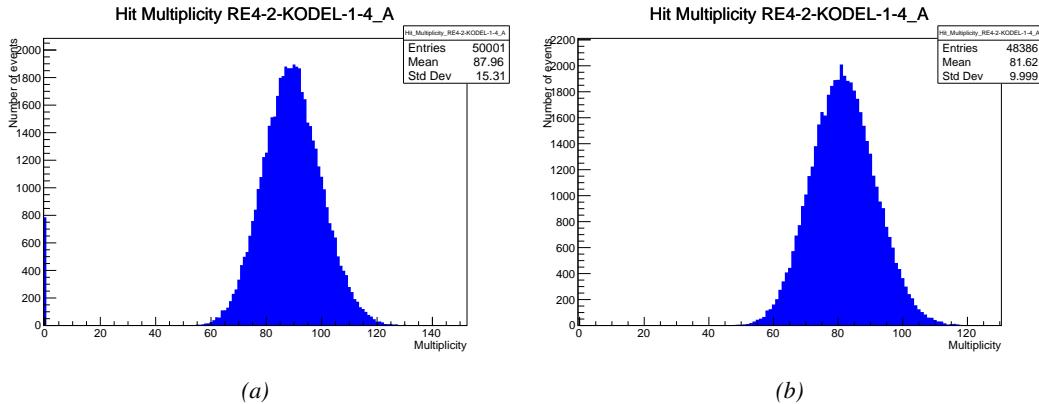


Figure A.5: Using the same data as previously showed in Figure A.4, the effect of the quality flag is explained by presenting the reconstructed hit multiplicity of a data file without `Quality_flag` in Figure A.5a and the reconstructed content of the same RPC partition for data corresponding to a `Quality_flag`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

## A.5 Communications

1769 To ensure data readout and dialog in between the machine and the TDCs or in between the webDCS  
1770 and the DAQ, different communication solutions were used. First of all, it is important to have a

1772 module to allow the communication in between the TDCs and the computer from which the DAQ  
 1773 operates. When this communication is effective, shifters using the webDCS to control data taking  
 1774 can thus send instructions to the DAQ.

1775

### 1776 A.5.1 V1718 USB Bridge

1777 In the previous section, the data transfer has been discussed. The importance of the `v1718` object  
 1778 (Source Code A.6), used as private member of `DataReader`, was not explicated. VME master  
 1779 modules are used for communication purposes as they host the USB port that connects the pow-  
 1780 ered crate buffer to the computer where the DAQ is installed. From the source code point of view,  
 1781 this object is used to control the communication status, by reading the returned error codes with  
 1782 `v1718::CheckStatus()`, or to check for IRQs coming from the TDCs through `v1718::CheckIRQ()`.  
 1783 Finally, to ensure that triggers are blocked at the hardware level, a NIM pulse is sent out of one of the  
 1784 5 programmable outputs (`v1718::SendBUSY()`) to the VETO of the coincidence module where the  
 1785 trigger signals originate from. As long as this signal is ON, no trigger can reach the TDCs anymore.  
 1786

```
1787 class v1718{
1788     private:
1789         int             Handle;
1790         Data32          Data;           // Data
1791         CVIRQLevels    Level;         // Interrupt level
1792         CVAddressModifier AM;          // Addressing Mode
1793         CVDataWidth     DataSize;       // Data Format
1794         Data32          BaseAddress;   // Base Address
1795
1796     public:
1797         v1718(IniFile *inifile);
1798         ~v1718();
1799         long            GetHandle(void) const;
1800         int             SetData(Data16 data);
1801         Data16          GetData(void);
1802         int             SetLevel(CVIRQLevels level);
1803         CVIRQLevels    GetLevel(void);
1804         int             SetAM(CVAddressModifier am);
1805         CVAddressModifier GetAM(void);
1806         int             SetDatasize(CVDataWidth datasize);
1807         CVDataWidth     GetDataSize(void);
1808         int             SetBaseAddress(Data16 baseaddress);
1809         Data16          GetBaseAddress(void);
1810         void            CheckStatus(CVErrorCodes status) const;
1811         void            CheckIRQ();
1812         void            SetPulsers();
1813         void            SendBUSY(BusyLevel level);
1814     };
1815 }
```

1788 *Source Code A.6: Description of C++ object v1718.*

### 1789 A.5.2 Configuration file

1790 The DAQ software takes as input a configuration file written using INI standard [61]. This file is  
 1791 partly filled with the information provided by the shifters when starting data acquisition using the  
 1792 webDCS, as shown by Figure A.6. This information is written in section [`General`] and will later

1793 be stored in the ROOT file that contains the DAQ data as can be seen from Figure A.3. Indeed,  
 1794 another `TTree` called `RunParameters` as well as the 2 histograms `ID`, containing the scan number,  
 1795 start and stop time stamps, and `Triggers`, containing the number of triggers requested by the shifter,  
 1796 are available in the data files. Moreover, `ScanID` and `HV` are then used to construct the file name  
 1797 thanks to the method `DataReader::GetFileName()`.

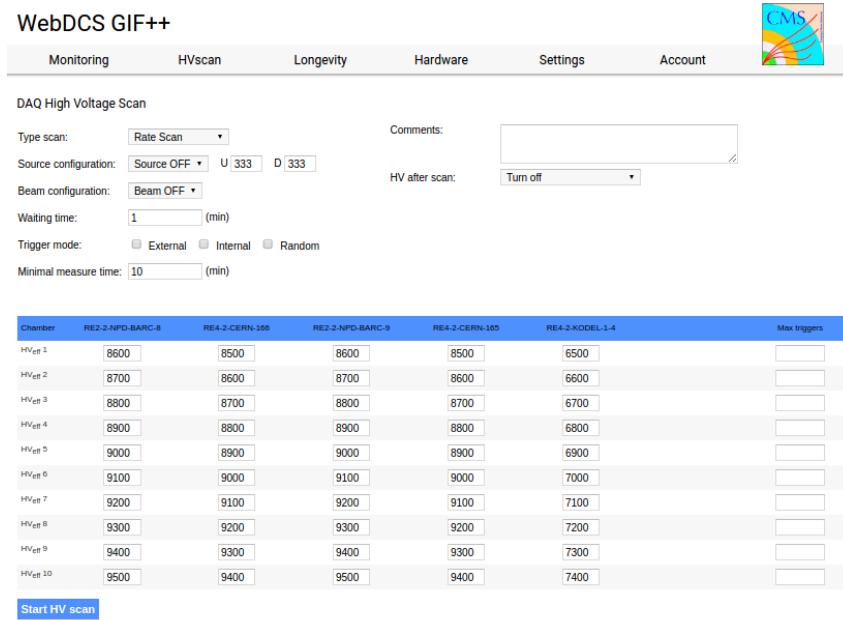


Figure A.6: WebDCS DAQ scan page. On this page, shifters need to choose the type of scan (Rate, Efficiency or Noise Reference scan), the gamma source configuration at the moment of data taking, the beam configuration, and the trigger mode. These information will be stored in the DAQ ROOT output. Are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

1798 The rest of the information is written beforehand in the configuration file template, as explicated  
 1799 in Source Code A.7, and contains the hardware addresses to the different VME modules in the  
 1800 setup as well as settings for the TDCs. As the TDC settings available in the configuration file are not  
 1801 supposed to be modified, an improvement would be to remove them from the configuration file and  
 1802 to hardcode them inside of the DAQ code itself or to place them into a different INI file that would  
 1803 host only the TDC settings to lower the probability for a bad manipulation of the configuration file  
 1804 that can be modified from one of webDCS' menus.

1805

```

[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
TriggerTimeSubstraction=0b1
TdcDetectionMode=0b01
TdcResolution=0b10
TdcDeadTime=0b00
TdcHeadTrailer=0b1
TdcEventSize=0b1001
TdcTestMode=0b0
BLTMode=1

```

1806

*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.*

1808     In order to retrieve the information of the configuration file, the object `IniFile` has been developed  
 1809     to provide an INI parser, presented in Source Code A.8. It contains private methods returning a  
 1810     boolean to check the type of line written in the file, whether a comment, a group header or a key line  
 1811     (`IniFile::CheckIfComment()`, `IniFile::CheckIfGroup()` and `IniFile::CheckIfToken()`). The  
 1812     key may sometimes be referred to as *token* in the source code. Moreover, the private element  
 1813     `FileData` is a map of `const string` to `string` that allows to store the data contained inside the  
 1814     configuration file via the public method `IniFile::GetFileData()` following the formatting (see  
 1815     method `IniFile::Read()`):

```
1816
  string group, token, value;
  // Get the field values for the 3 strings.
  // Then concatenate group and token together as a single string
  // with a dot separation.
  token = group + "." + token;
  FileData[token] = value;
```

1818     More methods have been written to translate the different keys into the right variable format  
 1819     when used by the DAQ. For example, to get a `float` value out of the configuration file data, knowing  
 1820     the group and the key needed, the method `IniFile::floatType()` can be used. It takes 3 arguments  
 1821     being the group name and key name (both `string`), and a default `float` value used as exception in  
 1822     the case the expected combination of group and key cannot be found in the configuration file. This  
 1823     default value is then used and the DAQ continues on working after sending an alert in the log file for  
 1824     further debugging.

```

1825 typedef map< const string, string > IniFileData;
1826
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();
    };

```

1827       *Source Code A.8: Description of C++ object `IniFile` used as a parser for INI file format.*

### 1828     A.5.3 WebDCS/DAQ intercommunication

1829     When shifters send instructions to the DAQ via the configuration file, it is the webDCS itself that  
 1830     gives the start command to the DAQ and then the 2 softwares use inter-process communication  
 1831     through file to synchronise themselves. This communication file is represented by the variable **const**  
 1832     string \_\_runstatuspath.

1833     On one side, the webDCS sends commands or status that are readout by the DAQ:

- 1834       • INIT, status sent when launching a scan and read via function `CtrlRunStatus(...)`,
- 1835       • START, command to start data taking and read via function `CheckSTART()`,
- 1836       • STOP, command to stop data taking at the end of the scan and read via function `CheckSTOP()`,  
 1837        and
- 1838       • KILL, command to kill data taking sent by user and read via function `CheckKILL()`

1839 and on the other, the DAQ sends status that are controled by the webDCS:

- 1840     ● `DAQ_RDY`, sent with `SendDAQReady()` to signify that the DAQ is ready to receive commands  
1841       from the webDCS,
- 1842     ● `RUNNING`, sent with `SendDAQRunning()` to signify that the DAQ is taking data,
- 1843     ● `DAQ_ERR`, sent with `SendDAQError()` to signify that the DAQ didn't receive the expected com-  
1844       mand from the webDCS or that the launch command didn't have the right number of argu-  
1845       ments,
- 1846     ● `RD_ERR`, sent when the DAQ wasn't able to read the communication file, and
- 1847     ● `WR_ERR`, sent when the DAQ wasn't able to write into the communication file.

#### 1848     **A.5.4 Example of inter-process communication cycle**

1849 Under normal conditions, the webDCS and the DAQ processes exchange commands and status via  
1850 the file hosted at the address `__runstatuspath`, as explained in subsection A.5.3. An example of  
1851 cycle is given in Table A.1. In this example, the steps 3 to 5 are repeated as long as the webDCS tells  
1852 the DAQ to take data. A data taking cycle is the equivalent as what is called a *Scan* in GIFT++ jargon,  
1853 referring to a set a runs with several HV steps. Each repetition of steps 3 to 5 is then equivalent to a  
1854 single *Run*.

1855

1856     At any moment during the data taking, for any reason, the shifter can decide that the data taking  
1857 needs to be stopped before it reached the end of the scheduled cycle. Thus at any moment on the  
1858 cycle, the content of the inter-process communication file will be changed to `KILL` and the DAQ will  
1859 shut down right away. The DAQ checks for `KILL` signals every 5s after the TDCs configuration is  
1860 over. So far, the function `CheckKILL()` has been used only inside of the data taking loop of method  
1861 `DataReader::Run()` and thus, if the shifter decides to KILL the data taking during the TDC con-  
1862 figuration phase or the HV ramping in between 2 HV steps, the DAQ will not be stopped smoothly  
1863 and a *force kill* command will be sent to stop the DAQ process that is still awake on the computer.  
1864 Improvements can be brought on this part of the software to make sure that the DAQ can safely  
1865 shutdown at any moment.

1866

## 1867     **A.6 Software export**

1868 In section A.2 was discussed the fact that the DAQ as written in its last version is not a standalone  
1869 software. It is possible to make it a standalone program that could be adapted to any VME setup  
1870 using V1190A and V1718 modules by creating a GUI for the software or by printing the log mes-  
1871 sages that are normally printed in the webDCS through the log file, directly into the terminal. This  
1872 method was used by the DAQ up to version 3.0 moment where the webDCS was completed. Also, it  
1873 is possible to check branches of DAQ v2.X to have example of communication through a terminal.

1874

1875     DAQ v2.X is nonetheless limited in it's possibilities and requires a lot of offline manual interven-  
1876 tions from the users. Indeed, there is no communication of the software with the detectors' power  
1877 supply system that would allow for a user a predefine a list of voltages to operate the detectors at

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

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

1878 and loop over to take data without any further manual intervention. In v2.X, the data is taken for a  
1879 single detector setting and at the end of each run, the softwares asks the user if he intends on taking  
1880 more runs. If so, the software invites the user to set the operating voltages accordingly to what is  
1881 necessary and to manual update the configuration file in consequence. This working mode can be a  
1882 very first approach before an evolution and has been successfully used by colleagues from different  
1883 collaborations.

1884  
1885 For a more robust operation, it is recommended to develop a GUI or a web application to inter-  
1886 face the DAQ. Moreover, to limit the amount of manual interventions, and thus the probability to  
1887 make mistakes, it is also recommended to add an extra feature into the DAQ by installing the HV  
1888 Wrapper library provided by CAEN of which an example of use in a similar DAQ software devel-  
1889 opped by a master student of UGent, and called TinyDAQ, is provided on UGent's github. Then, this  
1890 HV Wrapper will help you communicating with and give instructions to a CAEN HV powered crate  
1891 and can be added into the DAQ at the same level where the communication with the user was made  
1892 in DAQ v2.X. In case you are using another kind of power system for your detectors, it is stringly  
1893 adviced to use HV modules or crates that can be remotely controled via a using C++ libraries.

1894

# B

1895

1896

## Details on the offline analysis package

1897 The data collected in GIF++ thanks to the DAQ described in Appendix A is difficult to interpret by  
1898 a human user that doesn't have a clear idea of the raw data architecture of the ROOT data files. In  
1899 order to render the data human readable, a C++ offline analysis tool was designed to provide users  
1900 with detector by detector histograms that give a clear overview of the parameters monitored during  
1901 the data acquisition [62]. In this appendix, details about this software in the context of GIF++, as of  
1902 how the software was written and how it functions will be given.

### 1903 B.1 GIF++ Offline Analysis file tree

1904 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  
1905 as it takes ROOT files in input and write an output ROOT file containing histograms. To compile the  
1906 GIF++ Offline Analysis project is compiled with cmake. To compile, first a build/ directory must  
1907 be created to compile from there:

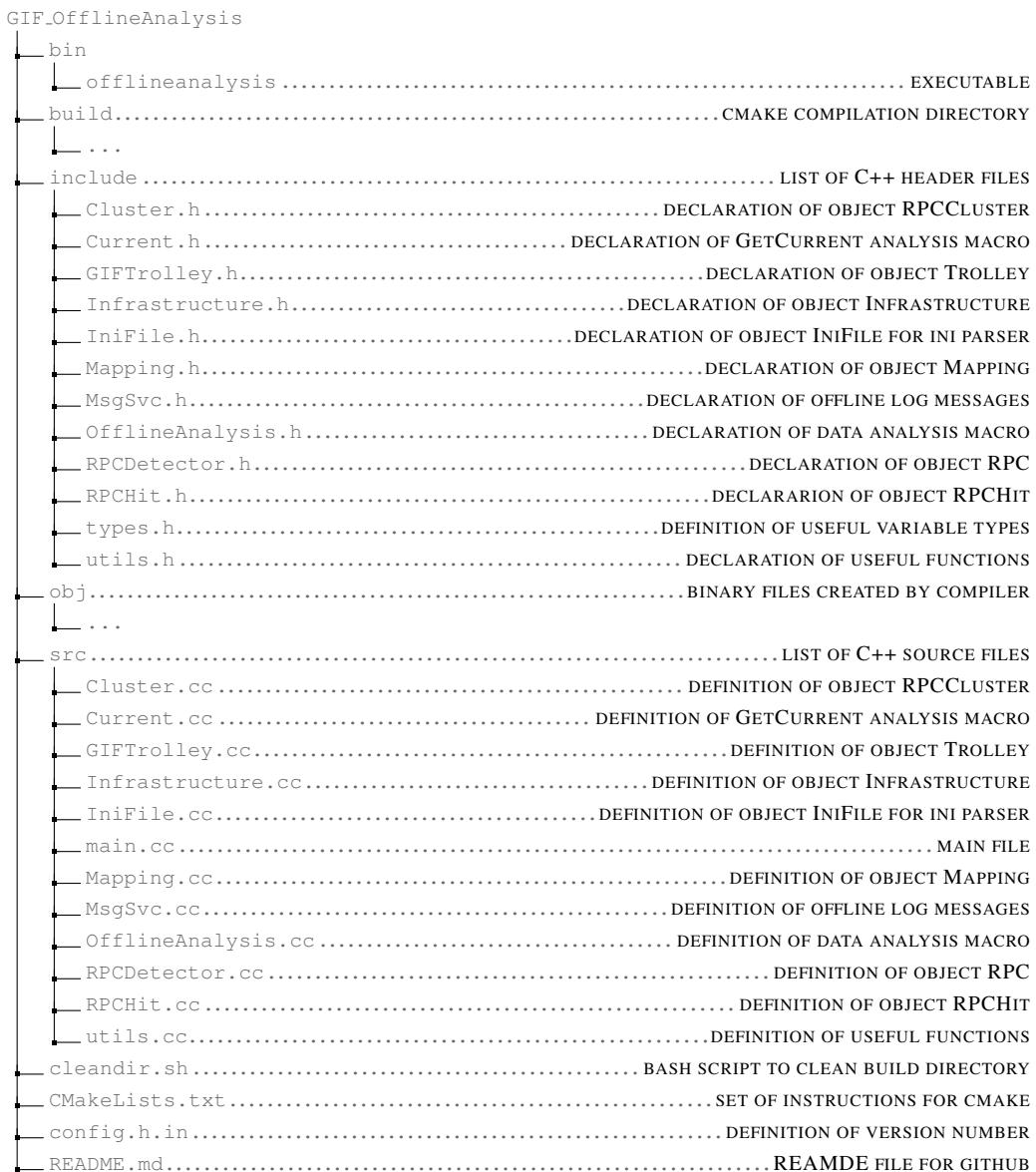
```
1909 mkdir build  
1910 cd build  
1911 cmake ..  
1912 make  
1913 make install
```

1911 To clean the directory and create a new build directory, the bash script cleandir.sh can be used:

```
1912  
1913 ./cleandir.sh
```

1914 The source code tree is provided below along with comments to give an overview of the files' con-  
1915 tent. The different objects created for this project (`Infrastructure`, `Trolley`, `RPC`, `Mapping`, `RPCHit`,  
1916 `RPCCluster` and `Inifile`) will be described in details in the following sections.

1917



1918

## B.2 Usage of the Offline Analysis

1919

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:

1920

1921

1922

```
Scan00XXXX_HVY
```

1923

1924

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

1925 of the GIF++ webDCS. Usually, the offline analysis tool is automatically called by the webDCS at  
 1926 the end of data taking or by a user from the webDCS panel if an update of the tool was brought.  
 1927 Nonetheless, an expert can locally launch the analysis for tests on the GIF++ computer, or a user can  
 1928 get the code on its local machine from github and download data from the webDCS for its own anal-  
 1929 ysis. To launch the code, the following command can be used from the `GIF_OfflineAnalysis` folder:

```
1930
1931 bin/offlineanalysis /path/to/Scan00XXXX_HVY
```

1932 where, `/path/to/Scan00XXXX_HVY` refers to the local data files. Then, the offline tool will by itself  
 1933 take care of finding all available ROOT data files present in the folder, as listed below:

- 1934     ● `Scan00XXXX_HVY_DAQ.root` containing the TDC data as described in Appendix ?? (events, hit  
 1935         and timestamp lists), and
- 1936     ● `Scan00XXXX_HVY_CAEN.root` containing the CAEN mainframe data recorded by the monitor-  
 1937         ing tool webDCS during data taking (HVs and currents of every HV channels). This file is  
 1938         created independently of the DAQ.

## 1939     **B.2.1 Output of the offline tool**

### 1940     **B.2.1.1 ROOT file**

1941 The analysis gives in output ROOT datafiles that are saved into the data folder and called using the  
 1942 naming convention `Scan00XXXX_HVY_Offline.root`. Inside those, a list of `TH1` histograms can be  
 1943 found. Its size will vary as a function of the number of detectors in the setup as each set of histograms  
 1944 is produced detector by detector. For each partition of each chamber, can be found:

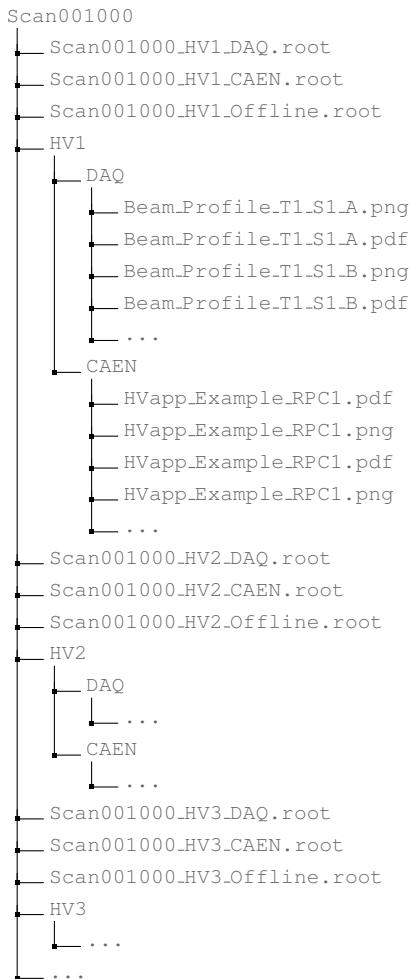
- 1945     ● `Time_Profile_Tt_Sc_p` shows the time profile of all recorded events (number of events per  
 1946         time bin),
- 1947     ● `Hit_Profile_Tt_Sc_p` shows the hit profile of all recorded events (number of events per chan-  
 1948         nel),
- 1949     ● `Hit_Multiplicity_Tt_Sc_p` shows the hit multiplicity (number of hits per event) of all recorded  
 1950         events (number of occurrences per multiplicity bin),
- 1951     ● `Strip_Mean_Noise_Tt_Sc_p` shows noise/gamma rate per unit area for each strip in a se-  
 1952         lected time range. After filters are applied on `Time_Profile_Tt_Sc_p`, the filtered version  
 1953         of `Hit_Profile_Tt_Sc_p` is normalised to the total integrated time and active detection area  
 1954         of a single channel,
- 1955     ● `Strip_Activity_Tt_Sc_p` shows noise/gamma activity for each strip (normalised version of  
 1956         previous histogram - strip activity = strip rate / average partition rate),
- 1957     ● `Strip_Homogeneity_Tt_Sc_p` shows the *homogeneity* of a given partition ( $\text{homogeneity} = \exp(-\text{strip rates standard deviation(strip rates in partition/average partition rate)})$ ),
- 1959     ● `mask_Strip_Mean_Noise_Tt_Sc_p` shows noise/gamma rate per unit area for each masked  
 1960         strip in a selected time range. Offline, the user can control the noise/gamma rate and decide to  
 1961         mask the strips that are judged to be noisy or dead. This is done via the *Masking Tool* provided  
 1962         by the webDCS,

- 1963     ● `mask_Strip_Activity_Tt_Sc_p` shows noise/gamma activity per unit area for each masked  
1964     strip with respect to the average rate of active strips,
- 1965     ● `NoiseCSize_H_Tt_Sc_p` shows noise/gamma cluster size, a cluster being constructed out of  
1966     adjacent strips giving a signal at the *same time* (hits within a time window of 25 ns),
- 1967     ● `NoiseCMult_H_Tt_Sc_p` shows noise/gamma cluster multiplicity (number of reconstructed  
1968     clusters per event),
- 1969     ● `Chip_Mean_Noise_Tt_Sc_p` shows the same information than `Strip_Mean_Noise_Tt_Scp` us-  
1970     ing a different binning (1 chip corresponds to 8 strips),
- 1971     ● `Chip_Activity_Tt_Sc_p` shows the same information than `Strip_Activity_Tt_Scp` using  
1972     chip binning,
- 1973     ● `Chip_Homogeneity_Tt_Sc_p` shows the homogeneity of a given partition using chip binning,
- 1974     ● `Beam_Profile_Tt_Sc_p` shows the estimated beam profile when taking efficiency scan. This  
1975     is obtained by filtering `Time_Profile_Tt_Sc_p` to only consider the muon peak where the  
1976     noise/gamma background has been subtracted. The resulting hit profile corresponds to the  
1977     beam profile on the detector channels,
- 1978     ● `L0_Efficiency_Tt_Sc_p` shows the level 0 efficiency that was estimated **without** muon track-  
1979     ing,
- 1980     ● `MuonCSize_H_Tt_Sc_p` shows the level 0 muon cluster size that was estimated **without** muon  
1981     tracking, and
- 1982     ● `MuonCMult_H_Tt_Sc_p` shows the level 0 muon cluster multiplicity that was estimated **without**  
1983     muon tracking.

1984     In the histogram labels, *t* stands for the trolley number (1 or 3), *c* for the chamber slot label in  
1985     trolley *t* and *p* for the partition label (A, B, C or D depending on the chamber layout) as explained  
1986     in Chapter ??.

1987     In the context of GIF++, an extra script called by the webDCS is called to extract the histograms  
1988     from the ROOT files. The histograms are then stored in PNG and PDF formats into the correspond-  
1989     ing folder (a single folder per HV step, so per ROOT file). the goal is to then display the histograms  
1990     on the Data Quality Monitoring (DQM) page of the webDCS in order for the users to control the  
1991     quality of the data taking at the end of data taking. An example of histogram organisation is given  
1992     below:

1994



**1995      Here can put some screens from the webDCS to show the DQM and the plots available to users.**

**1996**

### **1997      B.2.1.2 CSV files**

**1998      Moreover, up to 3 CSV files can be created depending on which ones of the 3 input files were in the  
1999      data folder:**

- 2000      • Offline-Corrupted.csv , is used to keep track of the amount of data that was corrupted and  
2001      removed from old data format files that don't contain any data quality flag.**
- 2002      • Offline-Current.csv , contains the summary of the currents and voltages applied on each  
2003      RPC HV channel.**
- 2004      • Offline-L0-EffC1.csv , is used to write the efficiencies, cluster size and cluster multiplicity  
2005      of efficiency runs. Note that L0 refers here to *Level 0* and means that the results of efficiency and  
2006      clusterization are a first approximation calculated without performing any muon tracking in**

2007 between the different detectors. This offline tool provides the user with a preliminar calculation  
 2008 of the efficiency and of the muon event parameters. Another analysis software especially  
 2009 dedicated to muon tracking is called on selected data to retrieve the results of efficiency and  
 2010 muon clusterization using a tracking algorithm to discriminate noise or gamma from muons  
 2011 as muons are the only particles that pass through the full setup, leaving hits than can be used  
 2012 to reconstruct their tracks.

- 2013 • `Offline-Rate.csv`, is used to write the noise or gamma rates measured in the detector readout  
 2014 partitions.

2015 Note that these 4 CSV files are created along with their *headers* (`Offline-[...]-Header.csv`  
 2016 containing the names of each data columns) and are automatically merged together when the offline  
 2017 analysis tool is called from the webDCS, contrary to the case where the tool is runned locally from  
 2018 the terminal as the merging bash script is then not called. Thus, the resulting files, used to make  
 2019 official plots, are:

- 2020 • `Corrupted.csv`,  
 2021 • `Current.csv`,  
 2022 • `L0-EffCl.csv`.  
 2023 • `Rate.csv`.

## 2024 B.3 Analysis inputs and information handling

2025 The usage of the Offline Analysis tool as well as its output have been presented in the previous section.  
 2026 It is now important to dig further and start looking at the source code and the inputs necessary  
 2027 for the tool to work. Indeed, other than the raw ROOT data files that are analysed, more information  
 2028 needs to be imported inside of the program to perform the analysis such as the description of the  
 2029 setup inside of GIFT++ at the time of data taking (number of trolleys, of RPCs, dimensions of the  
 2030 detectors, etc...) or the mapping that links the TDC channels to the coresponding RPC channels in  
 2031 order to translate the TDC information into human readable data. 2 files are used to transmit all this  
 2032 information:  
 2033

- 2034 • `Dimensions.ini`, that provides the necessary setup and RPC information, and  
 2035 • `ChannelsMapping.csv`, that gives the link between the TDC and RPC channels as well as the  
 2036 *mask* for each channel (masked or not?).

### 2037 B.3.1 Dimensions file and InFile parser

2038 This input file, present in every data folder, allows the analysis tool to know of the number of active  
 2039 trolleys, the number of active RPCs in those trolleys, and the details about each RPCs such as  
 2040 the number of RPC gaps, the number of pseudo-rapidity partitions (for CMS-like prototypes), the  
 2041 number of strips per partion or the dimensions. To do so, there are 3 types of groups in the INI file  
 2042 architecture. A first general group, appearing only once at the head of the document, gives information  
 2043 about the number of active trolleys as well as their IDs, as presented in Source Code B.1. For

2044 each active trolley, a group similar to Source Code B.2 can be found containing information about  
 2045 the number of active detectors in the trolley and their IDs. Each trolley group as a `Tt` name format,  
 2046 where `t` is the trolley ID. Finally, for each detector stored in slots of an active trolley, there is a group  
 2047 providing information about their names and dimensions, as shown in Source Code B.3. Each slot  
 2048 group as a `TtSs` name format, where `s` is the slot ID of trolley `t` where the active RPC is hosted.

```
2049 [General]
2050 nTrolleys=2
  TrolleysID=13
```

2051 *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.*

```
2052 [T1]
  nSlots=4
  SlotsID=1234
```

2053 *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.*

```
2054 [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
```

2055 *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.*

2056 This information is readout and stored in a C++ object called `IniFile`, that parses the information  
 2057 in the INI input file and stores it into a local buffer for later use. This INI parser is the exact same  
 2058 one that was previously developed for the Gif++ DAQ and described in Appendix A.5.2.

### 2059 B.3.2 TDC to RPC link file and Mapping

2060 The same way the INI dimension file information is stored using `map`, the channel mapping and mask  
 2061 information is stored and accessed through `map`. First of all, the mapping CSV file is organised into  
 2062 3 columns separated by tabulations (and not by commas, as expected for CSV files as it is easier using  
 2063 streams to read tab or space separated data using C++):

2064

2065        RPC\_channel              TDC\_channel              mask

2066        using as formatting for each field:

2067  
2068        TSCCC              TCCC              M

2069        TSCCC is a 5-digit integer where  $T$  is the trolley ID,  $s$  the slot ID in which the RPC is held insite  
2070        the trolley  $T$  and CCC is the RPC channel number, or *strip* number, that can take values up to  
2071        3-digits depending on the detector,

2072        TCCC is a 4 digit integer where  $T$  is the TDC ID, CCC is the TDC channel number that can take values  
2073        in between 0 and 127, and

2074        M is a 1-digit integer indicating if the channel should be considered ( $M = 1$ ) or discarded ( $M = 0$ )  
2075        during analysis.

2076        This mapping and masking information is readout and stored thanks to the object `Mapping`, pre-  
2077        sented in Source Code B.4. Similarly to `IniFile` objects, this class has private methods. The first  
2078        one, `Mapping::CheckIfNewLine()` is used to find the newline character '`\n`' or return character  
2079        '`\r`' (depending on which kind of operating system interacted with the file). This is used for the  
2080        simple reason that the masking information has been introduced only during the year 2017 but the  
2081        channel mapping files exist since 2015 and the very beginning of data taking at GIF++. This means  
2082        that in the older data folders, before the upgrade, the channel mapping file only had 2 columns, the  
2083        RPC channel and the TDC channel. For compatibility reasons, this method helps controling the  
2084        character following the readout of the 2 first fields of a line. In case any end of line character is  
2085        found, no mask information is present in the file and the default  $M = 1$  is used. On the contrary, if  
2086        the next character was a tabulation or a space, the mask information is present.

2087        Once the 3 fields have been readout, the second private method `Mapping::CheckIfTDCCh()` is  
2088        used to control that the TDC channel is an existing TDC channel. Finally, the information is stored  
2089        into 3 different maps (`Link`, `ReverseLink` and `Mask`) thanks to the public method `Mapping::Read()`.  
2090        `Link` allows to get the RPC channel by knowing the TDC channel while `ReverseLink` does the op-  
2091        posite by returning the TDC channel by knowing the RPC channel. Finally, `Mask` returns the mask  
2092        associated to a given RPC channel.

```

2093 typedef map<Uint,Uint> MappingData;

2094 class Mapping {
2095     private:
2096         bool          CheckIfNewLine(char next);
2097         bool          CheckIfTDCCh(Uint channel);
2098         string        FileName;
2099         MappingData  Link;
2100         MappingData  ReverseLink;
2101         MappingData  Mask;
2102         int           Error;
2103
2104     public:
2105         Mapping();
2106         Mapping(string baseName);
2107         ~Mapping();

2108         void SetFileName(const string filename);
2109         int  Read();
2110         Uint GetLink(Uint tdcchannel);
2111         Uint GetReverse(Uint rpcchannel);
2112         Uint GetMask(Uint rpcchannel);
2113     };

```

2095 *Source Code B.4: Description of C++ object Mapping used as a parser for the channel mapping and mask file.*

## 2096 B.4 Description of GIF++ setup within the Offline Analysis tool

2097 In the previous section, the tool input files have been discussed. The dimension file information is  
 2098 stored in a map hosted by the `IniFile` object. But this information is then used to create a series of  
 2099 new objects that helps defining the GIF++ infrastructure directly into the Offline Analysis. Indeed,  
 2100 from the `RPC`, to the more general `Infrastructure`, every element of the GIF++ infrastrucutre is  
 2101 recreated for each data analysis based on the information provided in input. All this information  
 2102 about the infrastructure will be used to assign each hit signal to a specific strip channel of a specific  
 2103 detector, and having a specific active area. This way, rate per unit area calculation is possible.  
 2104

### 2105 B.4.1 RPC objects

2106 `RPC` objects have been developped to represent physical active detectors in GIF++ at the moment  
 2107 of data taking. Thus, there are as many `RPC` objects created during the analysis than there were  
 2108 active `RPCs` tested during a run. Each `RPC` hosts the information present in the corresponding INI  
 2109 slot group, as shown in B.3, and organises it using a similar architecture. This can be seen from  
 2110 `Source Code B.5`.

2111 To make the object more compact, the lists of gap labels, of gap active areas and strip active  
 2112 areas are stored into `vector` dynamical containers. `RPC` objects are always contructed thanks to the  
 2113 dimension file information stored into the `IniFILE` and their ID, using the format `TtSs`. Using the  
 2114 `RPC` ID, the constructor calls the methods of `IniFILE` to initialise the `RPC`. The other constructors  
 2115 are not used but exist in case of need. Finally, some getters have been written to access the different  
 2116 private parameters storing the detector information.

```

2117
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);
};

2118

```

2119 *Source Code B.5: Description of C++ objects RPC that describe each active detectors used during data taking.*

## 2120 B.4.2 Trolley objects

2121 Trolley objects have been developped to represent physical active trolleys in GIFT++ at the moment  
 2122 of data taking. Thus, there are as many trolley objects created during the analysis than there were  
 2123 active trolleys hosting tested RPCs during a run. Each Trolley hosts the information present in the  
 2124 corresponding INI trolley group, as shown in B.2, and organises it using a similar architecture. In  
 2125 addition to the information hosted in the INI file, these object have a dynamical container of RPC  
 2126 objects, representing the active detectors the active trolley was hosting at the time of data taking.  
 2127 This can been seen from Source Code B.6.

2128 Trolley objects are always contructed thanks to the dimension file information stored into the  
 2129 IniFILE and their ID, using the format Tt. Using the Trolley ID, the constructor calls the methods  
 2130 of IniFILE to initialise the Trolley. Retrieving the information of the RPC IDs via SlotsID, a new  
 2131 RPC is constructed and added to the container RPCs for each character in the ID string. The other  
 2132 constructors are not used but exist in case of need. Finally, some getters have been written to access  
 2133 the different private parameters storing the trolley and detectors information.

```

2134 class Trolley{
2135     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
2136
    public:
        //Constructors, destructor and operator =
        Trolley();
        Trolley(string ID, IniFile* geofile);
        Trolley(const Trolley& other);
        ~Trolley();
        Trolley& operator=(const Trolley& other);
2137
        //Get GIFTrolley members
        Uint  GetNSlots();
        string GetSlotsID();
        Uint   GetSlotID(Uint s);
2138
        //Manage RPC list
        RPC*  GetRPC(Uint r);
        void  DeleteRPC(Uint r);
2139
        //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.*

### 2137 B.4.3 Infrastructure object

2138 The Infrastructure object has been developped to represent the GIFT++ bunker area dedicated to  
 2139 CMS RPC experiments. With this very specific object, all the information about the CMS RPC  
 2140 setup within GIFT++ at the moment of data taking is stored. It hosts the information present in the  
 2141 corresponding INI general group, as shown in B.1, and organises it using a similar architecture. In  
 2142 addition to the information hosted in the INI file, this object have a dynamical container of Trolley  
 2143 objects, representing the active tolleys in GIFT++ area. This can be seen from Source Code B.7.

2144 The Infrastructure object is always contructed thanks to the dimension file information stored  
 2145 into the IniFILE. Retrieving the information of the trolley IDs via TrolleysID, a new Trolley is  
 2146 constructed and added to the container Trolleys for each character in the ID string. By extension,  
 2147 it is easy to understand that the process described in Section B.4.2 for the construction of RPCs  
 2148 takes place when a trolley is constructed. The other constructors are not used but exist in case of  
 2149 need. Finally, some getters have been written to access the different private parameters storing the  
 2150 infrastructure, tolleys and detectors information.

```

2151
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);

2152
        //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++.

## 2154 B.5 Handeling of data

2155 As discussed in Appendix A.4.2, the raw data as a `TTree` architecture where every entry is related to  
 2156 a trigger signal provided by a muon or a random pulse, whether the goal of the data taking was to  
 2157 measure the performance of the detector or the noise/gamma background respectively. Each of these  
 2158 entries, referred also as events, contain a more or less full list of hits in the TDC channels to which  
 2159 the detectors are connected. To this list of hits corresponds a list of time stamps, marking the arrival  
 2160 of the hits within the TDC channel.

2161 The infrastructure of the CMS RPC experiment within GIF++ being defined, combining the  
 2162 information about the raw data with the information provided by both the mapping/mask file and the  
 2163 dimension file allows to build new physical objects that will help in computing efficiency or rates.

### 2164 B.5.1 RPC hits

2165 The raw data stored in the ROOT file as output of the GIFT++ DAQ, is readout by the analysis tool  
 2166 using the structure `RAWData` presented in Source Code B.9 that differs from the structure presented  
 2167 in Appendix A.4.2 as it is not meant to hold all of the data contained in the ROOT file. In this sense,  
 2168 this structure is in the case of the offline analysis tool not a dynamical object and will only be storing  
 2169 a single event contained in a single entry of the `TTree`.

```
2170
2171 class RPCHit {
2172     private:
2173         Uint Channel;      //RPC channel according to mapping (5 digits)
2174         Uint Trolley;      //0, 1 or 3 (1st digit of the RPC channel)
2175         Uint Station;      //Slot where is held the RPC in Trolley (2nd digit)
2176         Uint Strip;        //Physical RPC strip where the hit occurred (last 3
2177             digits)
2178         Uint Partition;    //Readout partition along eta segmentation
2179         float TimeStamp;   //Time stamp of the arrival in TDC
2180
2181     public:
2182         //Constructors, destructor & operator =
2183         RPCHit();
2184         RPCHit(Uint channel, float time, Infrastructure* Infra);
2185         RPCHit(const RPCHit& other);
2186         ~RPCHit();
2187         RPCHit& operator=(const RPCHit& other);
2188
2189         //Get RPCHit members
2190         Uint GetChannel();
2191         Uint GetTrolley();
2192         Uint GetStation();
2193         Uint GetStrip();
2194         Uint GetPartition();
2195         float GetTime();
2196     };
2197
2198     typedef vector<RPCHit> HitList;
2199     typedef struct GIFHitList { HitList rpc[NTROLLEYS][NSLOTS][NPARTITIONS]; } →
2200         GIFHitList;
2201
2202     bool SortHitbyStrip(RPCHit h1, RPCHit h2);
2203     bool SortHitbyTime(RPCHit h1, RPCHit h2);
2204 }
```

2172 *Source Code B.8: Description of C++ object RPCHit.*

```
2173
2174     struct RAWData{
2175         int iEvent;          //Event i
2176         int TDCNHits;       //Number of hits in event i
2177         int QFlag;           //Quality flag list (1 flag digit per TDC)
2178         vector<Uint> *TDCCh; //List of channels giving hits per event
2179         vector<float> *TDCTS; //List of the corresponding time stamps
2180     };
2181 }
```

2174 *Source Code B.9: Description of C++ structure RAWData.*

2175 Each member of the structure is then linked to the corresponding branch of the ROOT data tree,  
 2176 as shown in the example of Source Code B.10, and using the method `GetEntry(int i)` of the ROOT  
 2177 class `TTree` will update the state of the members of `RAWData`.

```

2178 TTree* dataTree = (TTree*)dataFile.Get("RAWData");
2179 RAWData data;
2180
2181 dataTree->SetBranchAddress("EventNumber", &data.iEvent);
2182 dataTree->SetBranchAddress("number_of_hits", &data.TDCNHits);
2183 dataTree->SetBranchAddress("Quality_flag", &data.QFlag);
2184 dataTree->SetBranchAddress("TDC_channel", &data.TDCCh);
2185 dataTree->SetBranchAddress("TDC_TimeStamp", &data.TDCTS);

```

2180       *Source Code B.10: Example of link in between RAWData and TTree.*

2181       The data is then analysed entry by entry and to each element of the TDC channel list, a `RPCHit` is  
2182       constructed by linking each TDC channel to the corresponding RPC channel thanks to the `Mapping`  
2183       object. The information carried by the RPC channel format allows to easily retrieve the trolley and  
2184       slot from which the hit was recorded (see section B.3.2). Using these 2 values, the readout partition  
2185       can be found by knowing the strip channel and comparing it with the number of partitions and strips  
2186       per partition stored into the `Infrastructure` object.

2187       Thus `RPCHit` objects are then stored into 3D dynamical list called `GIFHitList` (Source Code B.9)  
2188       where the 3 dimensions refer to the 3 layers of the readout in `GIF++` : in the bunker there are *trolleys*  
2189       ( $\tau$ ) holding detectors in *slots* ( $s$ ) and each detector readout is divided into 1 or more pseudo-rapidity  
2190       *partitions* ( $p$ ). Using these 3 information allows to assign an address to each readout partition and  
2191       this address will point to a specific hit list.

2192

## 2193     B.5.2 Clusters of hits

2194       All the hits contained in the ROOT file have been sorted into the different hit lists through the  
2195       `GIFHitList`. At this point, it is possible to start looking for clusters. A cluster is a group of adjacent  
2196       strips getting hits within a time window of 25 ns. These strips are then assumed to be part of the same  
2197       physical avalanche signal generated by a muon passing through the chamber or by the interaction of  
2198       a gamma stopping into the electrodes of the RPCs.

2199       To keep the cluster information, `RPCCluster` objects have been defined as shown in Source  
2200       Code B.11. Using the information of each individual `RPCHit` taken out of the hit list, it stores  
2201       the cluster size (number of adjacent strips composing the cluster), the first and last hit, the center for  
2202       spatial reconstruction and finally the start and stop time stamps as well as te time spread in between  
2203       the first and last hit.

```

2204
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);

```

2206           *Source Code B.11: Description of C++ object Cluster.*

2207       To investigate the hit list of a given detector partition, the function `Clusterization()` defined  
 2208       in `include/Cluster.h` needs the hits in the list to be time sorted. This is achieved by calling func-  
 2209       tion `sort()` of library `<algorithm>` using the comparator `SortHitbyTime(RPCHit h1, RPCHit h2)`  
 2210       defined in `include/RPCHit.h` that returns `true` if the time stamp of hit `h1` is lower than that of `h2`.  
 2211       A first isolation of strips is made only based on time information. All the hits within the 25 ns win-  
 2212       dow are taken separately from the rest. Then, this sub-list of hits is sorted this time by ascending  
 2213       strip number, using this time the comparator `SortHitbyStrip(RPCHit h1, RPCHit h2)`. Finally, the  
 2214       groups of adjacent strips are used to construct `RPCCluster` objects that are then stored in a temporary  
 2215       list of clusters that is at the end of the process used to know how many clusters were reconstructed  
 2216       and to fill their sizes into an histogram that will allows to know the mean size of muon or gamma  
 2217       clusters.  
 2218

## 2219     B.6 DAQ data Analysis

2220     All the ingredients to analyse GIF++ data have been defined. This section will focus on the different  
 2221     part of the analysis performed on the data, from determining the type of data the tool is dealing with

2222 to calculating the rate in each detector or reconstructing muon or gamma clusters.

### 2223 B.6.1 Determination of the run type

2224 In GIF++, both the performance of the detectors in detecting muons in an irradiated environment and  
 2225 the gamma background can be independantly measured. These corresponds to different run types  
 2226 and thus, to different TDC settings giving different data to look at.

2227 In the case of performance measurements, the trigger for data taking is provided by the coïncidence  
 2228 of several scintillators when muons from the beam passing through the area are detected. Data  
 2229 is collected in a 600 ns wide window around the arrival of muons in the RPCs. The expected time  
 2230 distribution of hits is shown in Figure B.1a. The muon peak is clearly visible in the center of the  
 2231 distribution and is to be extracted from the gamma background that composes the flat part of the  
 2232 distribution.  
 2233

2234 On the other hand, gamma background or noise measurements are focussed on the non muon  
 2235 related physics and the trigger needs to be independant from the muons to give a good measurement  
 2236 of the gamma/noise distribution as seen by the detectors. The trigger is then provided by a pulse  
 2237 generator at a frequency of 300 Hz whose pulse is not likely to be on time with a muon. In order  
 2238 to increase the integrated time without increasing the acquisition time too much, the width of the  
 2239 acquisition windows are increased to 10  $\mu$ s. The time distribution of the hits is expected to be flat, as  
 2240 shown by Figure B.1b.

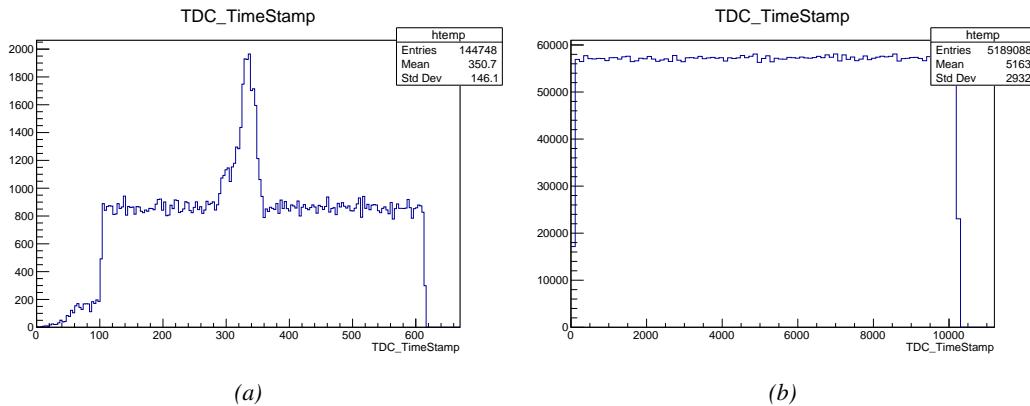


Figure B.1: Example of expected hit time distributions in the cases of efficiency (Figure B.1a) and noise/gamma rate per unit area (Figure B.1b) measurements as extracted from the raw ROOT files. The unit along the x-axis corresponds to ns. The fact that "the" muon peak is not well defined in Figure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

2241 The ROOT files include a TTree called RunParameters containing, among other things, the in-  
 2242 formation related to the type of run. The run type can then be accessed as described by Source  
 2243 Code B.12 and the function IsEfficiencyRun() is then used to determine if the run file is an effi-  
 2244 ciency run or, on the contrary, another type of run (noise or gamma measurement).

```

2245     TTree* RunParameters = (TTree*)dataFile.Get("RunParameters");
2246     TString* RunType = new TString();
2247     RunParameters->SetBranchAddress("RunType", &RunType);
2248     RunParameters->GetEntry(0);

```

2247       *Source Code B.12: Access to the run type contained in TTree\* RunParameters.*

2248       Finally, the data files will have a slightly different content whether it was collected before or after  
 2249       October 2017 and the upgrade of the DAQ software that brought a new information into the ROOT  
 2250       output. This is discussed in Appendix A.4.3 and implies that the analysis will differ a little depending  
 2251       on the data format. Indeed, as no information on the data quality is stored, in older data files, the cor-  
 2252       rections for missing events has to be done at the end of the analysis. The information about the type  
 2253       of data format is stored in the variable **bool** `isNewFormat` by checking the list of branches contained  
 2254       in the data tree via the methods `TTree::GetListOfBranches()` and `TCollection::Contains()`.

## 2255     **B.6.2 Beam time window calculation for efficiency runs**

2256       Knowing the run type is important first of all to know the width of the acquisition window to be used  
 2257       for the rate calculation and finally to be able to seek for muons. Indeed, the peak that appears in the  
 2258       time distribution for each detectors is then fitted to extract the most probable time window in which  
 2259       the tool should look for muon hits. The data outside of this time window is then used to evaluate the  
 2260       noise or gamma background the detector was subjected to during the data taking. Computing the  
 2261       position of the peak is done calling the function `SetBeamWindow()` defined in file `src/RPCHit.cc` that  
 2262       loops a first time on the data. The data is first sorted in a 3D array of 1D histograms (`GIFH1Array`, see  
 2263       `include/types.h`). Then the location of the highest bin is determined using `TH1::GetMaximumBin()`  
 2264       and is used to define a window in which a gaussian fit will be applied to compute the peak width.  
 2265       This window is a 80 ns defined by Formula B.1 around the central bin.

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

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

2266       Before the fit is performed, the average number of noise/gamma hits per bin is evaluated using  
 2267       the data outside of the fit window. Excluding the first 100 ns, the average number of hits per bin  
 2268       due to the noise or gamma is defined by Formula B.2 after extracting the amount of hits in the time  
 2269       windows  $[100; t_{low}]$  and  $[t_{high}; 600]$  thanks to the method `TH1::Integral()`. This average number  
 2270       of hits is then subtracted to every bin of the 1D histogram, in order to *clean* it from the noise or  
 2271       gamma contribution as much as possible to improve the fit quality. Bins where  $\langle n_{\text{hits}} \rangle$  is greater  
 2272       than the actual bin content are set to 0.

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

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

2273       Finally, the fit parameters are extracted and saved for each detector in 3D arrays of **float**  
 2274       (`muonPeak`, see `include/types.h`), a first one for the mean arrival time of the muons, `PeakTime`,

2275 and a second one for the width of the peak, `PeakWidth`. The width is defined as  $6\sigma$  of the gaussian  
 2276 fit. The same settings are applied to every partitions of the same detector. To determine which one  
 2277 of the detector's partitions is directly illuminated by the beam, the peak height of each partition is  
 2278 compared and the highest one is then used to define the peak settings.

### 2279 **B.6.3 Data loop and histogram filling**

2280 3D arrays of histogram are created to store the data and display it on the DQM of G4F++ webDCS  
 2281 for the use of shifters. These histograms, presented in section B.2.1.1, are filled while looping on  
 2282 the data. Before starting the analysis loop, it is necessary to control the entry quality for the new  
 2283 file formats featuring `QFlag`. If the `QFlag` value for this entry shows that 1 TDC or more have a  
 2284 `CORRUPTED` flag, then this event is discarded. The loss of statistics is low enough to be neglected.  
 2285 `QFlag` is controlled using the function `IsCorruptedEvent()` defined in `src/utils.cc`. As explained  
 2286 in Appendix A.4.3, each digit of this integer represents a TDC flag that can be 1 or 2. Each 2 is  
 2287 the sign of a `CORRUPTED` state. Then, the data is accessed entry by entry in the ROOT `TTree` using  
 2288 `RAWData` and each hit in the hit list is assigned to a detector channel and saved in the corresponding  
 2289 histograms. In the first part of the analysis, in which the loop over the ROOT file's content is  
 2290 performed, the different steps are:

2291 **1- RPC channel assignment and control:** a check is done on the RPC channel extracted thanks  
 2292 to the mapping via the method `Mapping::GetLink()`. If the channel is not initialised and is 0, or if  
 2293 the TDC channel was greater than 5127, the hit is discarded. This means there was a problem in the  
 2294 mapping. Often a mapping problem leads to the crash of the offline tool.

2295 **2- Creation of a `RPCHit` object:** to easily get the trolley, slot and partition in which the hit has  
 2296 been assigned, this object is particularly helpful.

2297 **3- General histograms are filled:** the hit is filled into the time distribution and the general hit  
 2298 distribution histograms, and if the arrival time is within the first 100 ns, it is discarded and nothing  
 2299 else happens and the loop proceeds with the next hit in the list.

2300 **4- Multiplicity counter:** the hit multiplicity counter of the corresponding detectors incremented.

2301 **5-a- Efficiency runs - Is the hit within the peak window? :** if the peak is contained in the peak  
 2302 window previously defined in section B.6.2, the hit is filled into the beam hit profile histogram of  
 2303 the corresponding chamber, added into the list of muon hits and increments the counter of *in time*  
 2304 hits. The term *in time* here refers to the hits that are likely to be muons by arriving in the expected  
 2305 time window. If the hit is outside of the peak window, it is filled into the noise profile histogram  
 2306 of the corresponding detector, added into the list of noise/gamma hits and increments the counter of  
 2307 noise/gamma hits.

2308 **5-b- Noise/gamma rate runs - Noise histograms are filled:** the hit is filled into the noise profile  
 2309 histogram of the corresponding detector, added into the list of noise/gamma hits and increments the  
 2310 counter of noise/gamma hits.

2311

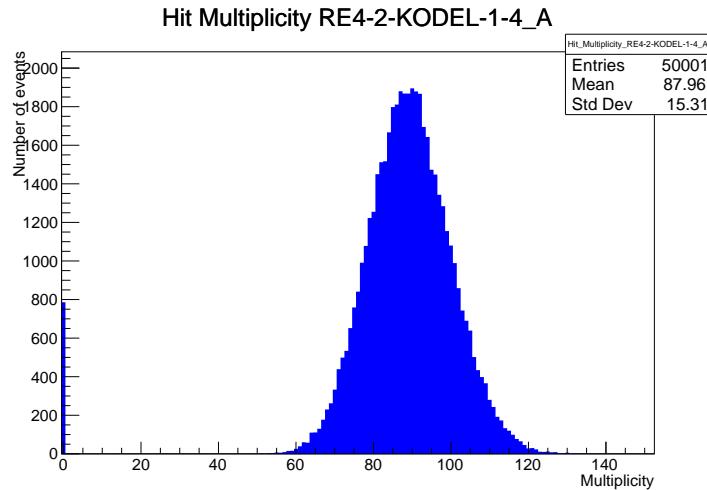
2312 After the loop on the hit list of the entry is over, the next step is to clusterize the 3D lists filled  
 2313 in the previous steps. A 3D loop is then started over the active trolley, slot and RPC partitions to  
 2314 access these objects. Each `NoiseHitList` and `MuonHitList`, in case of efficiency run, are clusterized  
 2315 as described in section B.5.2. There corresponding cluster size and multiplicity histograms are filled  
 2316 at the end of the clustering process. Then, the efficiency histogram is filled in case of efficiency run.  
 2317 The selection is simply made by checking whether the RPC detected signals in the peak window  
 2318 during this event. Nevertheless, it is useful to highlight that at this level, it is not possible yet to  
 2319 discriminate in between a muon hit and noise or gamma hit. Thus, `MuonCSize_H`, `MuonCMult_H`  
 2320 and `Efficiency0_H` are subjected to noise and gamma contamination. This contamination will be  
 2321 estimated and corrected at the moment the results will be written into output CSV files. Finally, the  
 2322 loop ends on the filling of the general hit multiplicity histogram.

#### 2323 **B.6.4 Results calculation**

2324 As mentioned in section B.2.1, the analysis of DAQ data provides the user with 3 CSV files and  
 2325 a ROOT file associated to each and every ROOT data file. The fourth CSV file is provided by the  
 2326 extraction of the CEAN main frame data monitored during data taking and will be discussed later.  
 2327 After looping on the data in the previous part of the analysis macro, the output files are created and a  
 2328 3D loop on each RPC readout partitions is started to extract the histograms parameters and compute  
 2329 the final results.

2330

##### 2331 **B.6.4.1 Rate normalisation**



2332 *Figure B.2: The effect of the quality flag is explained by presenting the reconstructed hit multiplicity of a data  
 2333 file without `Quality_flag`. The artificial high content of bin 0 is the effect of corrupted data.*

2334 To analyse old data format files, not containing any quality flag, it is needed to estimate the amount  
 2335 of corrupted data via a fit as the corrupted data will always fill events with a fake "0 multiplicity".  
 Indeed, as no hits were stored in the DAQ ROOT files, these events artificially contribute to fill  
 the bin corresponding to a null multiplicity, as shown in Figure B.2. In the case the mean of the

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

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

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

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

```

2358
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;

2359
    double nSinglehit = (double)HitMultiplicity_H.rpc[T][S][p]->GetBinContent(1);
    double lowMultRatio = nSinglehit / (double)nEntries;
    bool isMultLOW = lowMultRatio > 0.4;

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

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

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

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

```

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

#### 2361 B.6.4.2 Rate and activity

2362 At this point, the strip rate histograms, StripNoiseProfile\_H.rpc[T][S][p], only contain an in-
2363 formation about the total number of noise or rate hits each channel received during the data taking.
2364 As described in Source Code B.14, a loop on the strip channels will be used to normalise the content
2365 of the rate distribution histogram for each detector partitions. The initial number of hits recorded for
2366 a given bin will be extracted and 2 values will be computed:

- 2367     ● the strip rate, defined as the number of hits recorded in the bin normalised like described in  
 2368       the previous section, using the variable `rate_norm`, and

- 2369     ● the strip activity, defined as the number of hits recorded in the bin normalised to the average  
 2370       number of hits per bin contained in the partition histogram, using the variable `averageNhit`.  
 2371       This value provides an information on the homogeneity of the detector response to the gamma  
 2372       background or of the detector noise. An activity of 1 corresponds to an average response.  
 2373       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);
}

```

2375     *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.*

2376     On each detector partitions, which are readout by a single FEE, all the channels are not processed  
 2377     by the same chip. Each chip can give a different noise response and thus, histograms using a chip  
 2378     binning are used to investigate chip related noise behaviours. The average values of the strip rate  
 2379     or activity grouped into a given chip are extracted using the using the function `GetChipBin()` and  
 2380     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;
}

```

2383     *Source Code B.15: Function used to compute the content of a bin for an histogram using chip binning.*

```

2384     for(UInt ch = 0; ch < (nStripsPart/NSTRIPSCHIP); ch++) {
2385         ChipMeanNoiseProf_H.rpc[T][S][p]->
2386             SetBinContent(ch+1,GetChipBin(StripNoiseProfile_H.rpc[T][S][p],ch));
2387         ChipActivity_H.rpc[T][S][p]->
2388             SetBinContent(ch+1,GetChipBin(StripActivity_H.rpc[T][S][p],ch));
2389     }

```

*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.

```

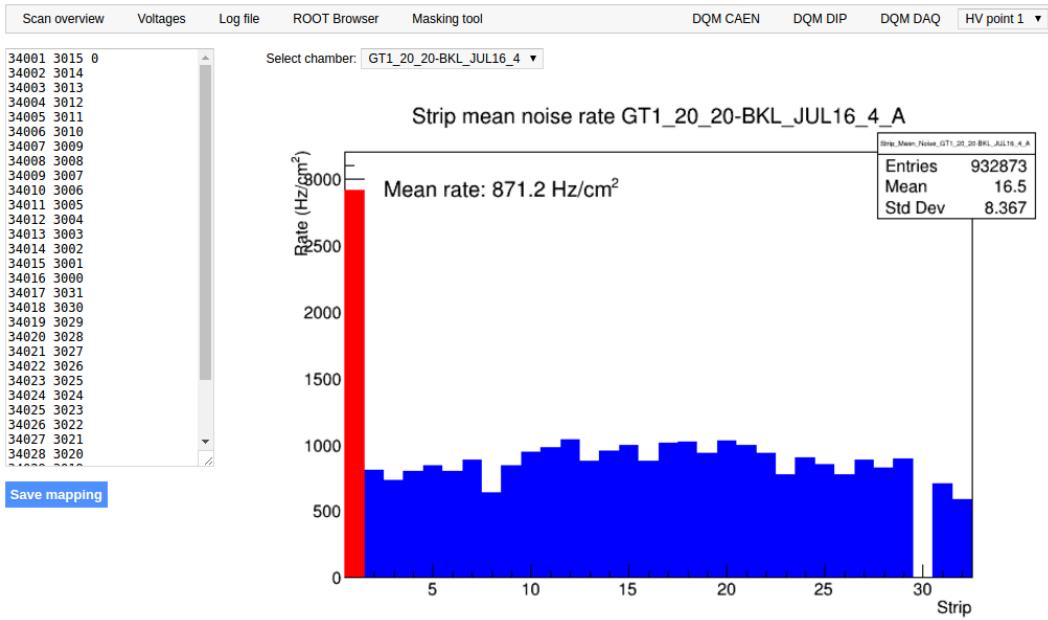
2398     float MeanPartSDev = GetTH1StdDev(StripNoiseProfile_H.rpc[T][S][p]);
2399     float strip_homog = (MeanPartRate==0)
2400         ? 0.
2401         : exp(-MeanPartSDev/MeanPartRate);
2402     StripHomogeneity_H.rpc[T][S][p]->Fill("exp -#left(#frac{\#sigma_{Strip}
2403         \rightarrow Rate}{\#mu_{Strip Rate}}\#right)",strip_homog);
2404     StripHomogeneity_H.rpc[T][S][p]->GetYaxis()->SetRangeUser(0.,1.);
2405
2406     float ChipStDevMean = GetTH1StdDev(ChipMeanNoiseProf_H.rpc[T][S][p]);
2407     float chip_homog = (MeanPartRate==0)
2408         ? 0.
2409         : exp(-ChipStDevMean/MeanPartRate);
2410     ChipHomogeneity_H.rpc[T][S][p]->Fill("exp -#left(#frac{\#sigma_{Chip}
2411         \rightarrow Rate}{\#mu_{Chip Rate}}\#right)",chip_homog);
2412     ChipHomogeneity_H.rpc[T][S][p]->GetYaxis()->SetRangeUser(0.,1.);

```

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

#### 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.*

2407 From the code point of view, the function `GetTH1Mean()` is used to retrieve the mean rate par-  
 2408 tition by partition after the rates have been calculated strip by strip and filled into the histograms  
 2409 `StripNoiseProfile_H.rpc[T][S][p]`, as described through Source Code B.18.

2410 Once the mask for each rejected channel has been updated, the shifter can manually run the of-  
 2411 fline tool again to update the DQM plots, now including the masked strips, as well the rate results  
 2412 written in the output CSV file `Offline-Rate.csv`. If not done during the shifts, the strip masking  
 2413 procedure needs to be carefully done by the person in charge of data analysis on the scans that were  
 2414 selected to produce the final results.

```

2415
2416     float GetTH1Mean(TH1* H) {
2417         int nBins = H->GetNbinsX();
2418         int nActive = nBins;
2419         float mean = 0.;

2420         for(int b = 1; b <= nBins; b++) {
2421             float value = H->GetBinContent(b);
2422             mean += value;
2423             if(value == 0.) nActive--;
2424         }

2425         if(nActive != 0) mean /= (float)nActive;
2426         else mean = 0.;

2427         return mean;
2428     }

```

*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.

#### 2418 B.6.4.4 Output CSV files filling

2419 All the histograms have been filled. Parameters will then be extracted from them to compute the  
 2420 final results that will later be used to produce plots. Once the results have been computed, the very  
 2421 last step of the offline macro is to write these values into the corresponding CSV outputs. Aside of  
 2422 the file `Offline-Corrupted.csv`, 2 CSV files are being written by the macro `offlineAnalysis()`,  
 2423 `Offline-Rates.csv` and `Offline-L0-EffCl.csv` that respectively contain information about noise  
 2424 or gamma rates, cluster size and multiplicity, and about level 0 reconstruction of the detector effi-  
 2425 ciency, muon cluster size and multiplicity. Details on the computation and file writing are respec-  
 2426 tively given in Sources Codes B.19 and B.20.

2427 **Noise/gamma background variables** are computed and written in the output file for each detector  
 2428 partitions. A detector average of the hit and cluster rate is also provided, as shown through Sources  
 2429 Code B.19. The variables that are written for each partition are:

- 2430 • The mean partition hit rate per unit area, `MeanPartRate`, that is extracted from the histogram  
 2431 `StripNoiseProfile_H` as the mean value along the y-axis, as described in section B.6.4.3. No  
 2432 error is recorded for the hit rate as this is considered a single measurement. No statistical error  
 2433 can be associated to it and the systematics are unknown.
- 2434 • The mean cluster size, `cSizePart`, is extracted from the histogram `NoiseCSize_H` and it's  
 2435 statistical error, `cSizePartErr`, is taken to be  $2\sigma$  of the total distribution.
- 2436 • The mean cluster multiplicity per trigger, `cMultPart`, is extracted from the histogram `NoiseCMult_H`  
 2437 and it's statistical error, `cMultPartErr`, is taken to be  $2\sigma$  of the total distribution. It is impor-  
 2438 tant to point to the fact that this variable gives an information that is dependent on the buffer  
 2439 window width used for each trigger for the calculation.
- 2440 • The mean cluster rate per unit area, `ClustPartRate`, is defined as the mean hit rate normalised

2441 to the mean cluster size and it's statistical error, `ClustPartRateErr`, is then obtained using the  
 2442 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';
  }
}

```

2444 *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.*

2445 **Muon performance variables** are computed and written in the output file for each detector parti-  
 2446 tions as shown through Sources Code B.20. The variables that are written for each partition are:

- 2447     ● The muon efficiency, `eff`, extracted from the histogram `Efficiency0_H`. It is reminded that  
2448       this offline tool doesn't include any tracking algorithm to identify muons from the beam and  
2449       only relies on the hits arriving in the time window corresponding to the beam time. The con-  
2450       tent of the efficiency histogram is thus biased by the noise/gamma background contribution  
2451       into this window and is thus corrected by estimating the muon data content in the peak re-  
2452       gion knowing the noise/gamma content in the rate calculation region. Both time windows  
2453       being different, the choice was made to normalise the noise/gamma background calculation  
2454       window to it's equivalent beam window in order to have comparable values using the variable  
2455       `windowRatio`. Finally, to estimate the data ratio in the peak region, the variable `DataRatio`  
2456       is defined as the ratio in between the estimated mean cluster multiplicity of the muons in the  
2457       peak region, `MuonCM`, and of the total mean cluster multiplicity in the peak region, `PeakCM`.  
2458       `MuonCM` is itself defined as the difference in between the total mean cluster multiplicity in the  
2459       peak region and the normalised mean noise/gamma cluster multiplicity calculated outside of  
2460       the peak region. The statistical error related to the efficiency, `eff_err`, is computed using a  
2461       binomial distribution, as the efficiency measure the probability of "success" and "failure" to  
2462       detect muons.
- 2463     ● The mean muon cluster size, `MuonCS`, is calculated using the total mean cluster size and multi-  
2464       plicity in the peak region, respectively extracted from histograms `MuonCSize_H` and `MuonCMult_H`,  
2465       the noise/gamma background mean cluster size and normalised multiplicity, extracted from  
2466       `NoiseCSize_H` and `NoiseCMult_H`, and of the estimated muon cluster multiplicity `MuonCM` pre-  
2467       viously explicated. The associated statistical error, `MuonCM_err`, is calculated using the propa-  
2468       gation of errors of the mentioned variables.
- 2469     ● The mean muon cluster multiplicity in the peak region, `MuonCM`, explicated above whose sta-  
2470       tistical error, `MuonCM_err`, is the sum of statistical error associated to the total mean clus-  
2471       ter multiplicity in the peak reagion, `PeakCM_err`, and of the mean noise/gamma cluster size,  
2472       `NoiseCM_err`.

2473       In addition to these 2 CSV files, the histograms are saved in ROOT file `Scan00XXXX_HVY_Offline.root`  
2474       as explained in section B.2.1.1.

2475

```

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';
        }
    }
}

```

2476

2477

*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.*

## 2478 B.7 Current data Analysis

2479 Detectors under test at GIF++ are connected both to a CAEN HV power supply and to a CAEN  
2480 ADC that reads the currents inside of the RPC gaps bypassing the supply cable. During data tak-  
2481 ing, the webDCS records into a ROOT file called `Scan00XXXX_HVY_CAEN.root` histograms with the  
2482 monitored parameters of both CAEN devices. Are recorded for each RPC channels (in most cases,  
2483 a channel corresponds to an RPC gap):

- 2484 • the effective voltage,  $HV_{eff}$ , set by the webDCS using the PT correction on the CAEN power  
2485 supply,
- 2486 • the applied voltage,  $HV_{app}$ , monitored by the CAEN power supply, and the statistical error  
2487 related to the variations of this value through time to follow the variation of the environmental  
2488 parameters defined as the RMS of the histogram divided by the square root of the number of  
2489 recorded points,
- 2490 • the monitored current,  $I_{mon}$ , monitored by the CAEN power supply, and the statistical error  
2491 related to the variations of this value through time to follow the variation of the environmental  
2492 parameters defined as the RMS of the histogram divided by the square root of the number of  
2493 recorded points,
- 2494 • the corresponding current density,  $J_{mon}$ , defined as the monitored current per unit area,  
2495  $J_{mon} = I_{mon}/A$ , where  $A$  is the active area of the corresponding gap,
- 2496 • the ADC current,  $I_{ADC}$ , recorded through the CAEN ADC module that monitors the dark  
2497 current in the gap itself. First of all, the resolution of such a module is better than that of  
2498 CAEN power supplies and moreover, the current is not read-out through the HV supply line  
2499 but directly at the chamber level giving the real current inside of the detector. The statistical  
2500 error is defined as the RMS of the histogram distribution divided by the square root of the  
2501 number of recorded points.

2502 Once extracted through a loop over the element of GIF++ infrastructure via the C++ macro  
2503 `GetCurrent()`, these parameters, organised in 9 columns per detector HV supply line, are written in  
2504 the output CSV file `Offline-Current.csv`. The macro can be found in the file `Current.cc`.

## References

- 2506 [1] T. Massam et al. “Experimental observation of antideuteron production”. In: *Il Nuovo Cimento A* 63 (1965), pp. 10–14.
- 2507
- 2508 [2] UA1 Collaboration. “Experimental observation of isolated large transverse energy electrons with associated missing energy at  $s = 540 \text{ GeV}$ ”. In: *Physics Letters B* 122 (1983), pp. 103–116.
- 2509
- 2510
- 2511 [3] UA2 Collaboration. “Observation of single isolated electrons of high transverse momentum in events with missing transverse energy at the CERN pp collider”. In: *Physics Letters B* 122 (1983), pp. 476–485.
- 2512
- 2513
- 2514 [4] UA1 Collaboration. “Experimental observation of lepton pairs of invariant mass around  $95 \text{ GeV}/c^2$  at the CERN SPS collider”. In: *Physics Letters B* 126 (1983), pp. 398–410.
- 2515
- 2516 [5] UA2 Collaboration. “Evidence for  $Z_0 \rightarrow e^+e^-$  at the CERN pp collider”. In: *Physics Letters B* 129 (1983), pp. 130–140.
- 2517
- 2518 [6] ALEPH Collaboration. “Determination of the number of light neutrino species”. In: *Physics Letters B* 231 (1989), pp. 519–529.
- 2519
- 2520 [7] CERN, ed. (1985).
- 2521 [8] CERN, ed. (1986).
- 2522 [9] CERN, ed. (1994).
- 2523 [10] CERN, ed. (1998).
- 2524 [11] CERN, ed. (1999).
- 2525 [12] CERN. Geneva. LHC Experiments Committee. *Letter of Intent for A Large Ion Collider Experiment [ALICE]*, note = CERN-LHCC-93-016. Tech. rep. ALICE Collaboration, 1993.
- 2526
- 2527 [13] CERN. Geneva. LHC Experiments Committee. *ATLAS : technical proposal for a general-purpose pp experiment at the Large Hadron Collider at CERN*, note = CERN-LHCC-94-43. Tech. rep. ATLAS Collaboration, 1994.
- 2528
- 2529
- 2530 [14] CERN. Geneva. LHC Experiments Committee. *CMS : letter of intent by the CMS Collaboration for a general purpose detector at LHC*, note = CERN-LHCC-92-003. Tech. rep. CMS Collaboration, 1992.
- 2531
- 2532
- 2533 [15] CERN. Geneva. LHC Experiments Committee. *LHCb : letter of intent*. Tech. rep. CERN-LHCC-95-5. LHCb Collaboration, 1995.
- 2534
- 2535 [16] L. Evans and P. Bryant. “LHC Machine”. In: *JINST* 3 (2008). S08001.
- 2536 [17] CMS Collaboration ATLAS Collaboration. “Combined Measurement of the Higgs Boson Mass in  $pp$  Collisions at  $\sqrt{s} = 7$  and  $8 \text{ TeV}$  with the ATLAS and CMS Experiments”. In: *Physical Review Letters* 114 (2015). 191803.
- 2537
- 2538
- 2539 [18] LHCb Collaboration. “Observation of  $J/\psi p$  Resonances Consistent with Pentaquark States in  $\Lambda_b^0 \rightarrow J/\psi K^- p$  Decays”. In: *Physical Review Letters* 115 (2015). 072001.
- 2540

- 2541 [19] LHCb Collaboration. “Observation of  $J/\psi\phi$  Structures Consistent with Exotic States from  
2542 Amplitude Analysis of  $B^+ \rightarrow J/\psi\phi K^+$  Decays”. In: *Physical Review Letters* 118 (2017).  
2543 022003.
- 2544 [20] CERN. Geneva. *High-Luminosity Large Hadron Collider (HL-LHC) Technical Design Re-*  
2545 *port V. 0.1*. Tech. rep. CERN-2017-007-M. 2017.
- 2546 [21] CERN. Geneva. LHC Experiments Committee. *The CMS muon project : Technical Design*  
2547 *Report*. Tech. rep. CERN-LHCC-97-032. CMS Collaboration, 1997.
- 2548 [22] CERN. Geneva. LHC Experiments Committee. *Technical Proposal for the Phase-II Upgrade*  
2549 *of the CMS Detector*. Tech. rep. CERN-LHCC-2015-010. CMS Collaboration, 2015.
- 2550 [23] CERN. Geneva. LHC Experiments Committee. *CMS, the Compact Muon Solenoid : technical*  
2551 *proposal*. Tech. rep. CERN-LHCC-94-38. CMS Collaboration, 1994.
- 2552 [24] R. Santonico and R. Cardarelli. “Development of resistive plate counters”. In: *Nucl. Instr.*  
2553 *Meth. Phys. Res.* 187 (1981), pp. 377–380.
- 2554 [25] Yu.N. Pestov and G.V. Fedotovich. *A picosecond time-of-flight spectrometer for the VEPP-2M*  
2555 *based on local-discharge spark counter*. Tech. rep. SLAC-TRANS-0184. SLAC, 1978.
- 2556 [26] W.W. Ash, ed. *Spark Counter With A Localized Discharge*. Vol. SLAC-R-250. 1982, pp. 127–  
2557 131.
- 2558 [27] I. Crotty et al. “The non-spark mode and high rate operation of resistive parallel plate cham-  
2559 bers”. In: *NIMA* 337 (1993), pp. 370–381.
- 2560 [28] I. Crotty et al. “Further studies of avalanche mode operation of resistive parallel plate cham-  
2561 bers”. In: *NIMA* 346 (1994), pp. 107–113.
- 2562 [29] R. Cardarelli et al. “Avalanche and streamer mode operation of resistive plate chambers”. In:  
2563 *NIMA* 382 (1996), pp. 470–474.
- 2564 [30] E. Cerron Zeballos et al. “A new type of resistive plate chamber: The multigap RPC”. In:  
2565 *NIMA* 374 (1996), pp. 132–135.
- 2566 [31] M.C.S. Williams. “The development of the multigap resistive plate chamber”. In: *Nucl. Phys.*  
2567 *B* 61 (1998), pp. 250–257.
- 2568 [32] H. Czyrkowski et al. “New developments on resistive plate chambers for high rate operation”.  
2569 In: *NIMA* 419 (1998), pp. 490–496.
- 2570 [33] P. Camarri et al. “Streamer suppression with SF6 in RPCs operated in avalanche mode”. In:  
2571 *NIMA* 414 (1998), pp. 317–324.
- 2572 [34] E. Cerron Zeballos et al. “Effect of adding SF6 to the gas mixture in a multigap resistive plate  
2573 chamber”. In: *NIMA* 419 (1998), pp. 475–478.
- 2574 [35] CERN. Geneva. LHC Experiments Committee. *ATLAS muon spectrometer: Technical design*  
2575 *report*. Tech. rep. CERN-LHCC-97-22. ATLAS Collaboration, 1997.
- 2576 [36] CERN. Geneva. LHC Experiments Committee. *ALICE Time-Of-Flight system (TOF) : Tech-*  
2577 *nical Design Report*. Tech. rep. CERN-LHCC-2000-012. ALICE Collaboration, 2000.
- 2578 [37] The CALICE collaboration. “First results of the CALICE SDHCAL technological proto-  
2579 type”. In: *JINST* 11 (2016).
- 2580 [38] PoS, ed. *Density Imaging of Volcanoes with Atmospheric Muons using GRPCs*. International  
2581 Europhysics Conference on High Energy Physics - HEP 2011. 2011.
- 2582 [39] C. Lippmann. “Detector Physics of Resistive Plate Chambers”. PhD thesis. Johann Wolfgang  
2583 Goethe-Universität, 2003.

## BIBLIOGRAPHY

---

- 2584 [40] M. Abbrescia et al. “Properties of C<sub>2</sub>H<sub>2</sub>F<sub>4</sub>-based gas mixture for avalanche mode operation  
2585 of resistive plate chambers”. In: *NIMA* 398 (1997), pp. 173–179.
- 2586 [41] G.Battistoni et al. “Sensitivity of streamer mode to single ionization electrons”. In: *NIMA*  
2587 235 (1985), pp. 91–97.
- 2588 [42] W. Riegler. “Induced signals in resistive plate chambers”. In: *NIMA* 491 (2002), pp. 258–271.
- 2589 [43] E. Cerron Zeballos et al. “A comparison of the wide gap and narrow gap resistive plate  
2590 chamber”. In: *NIMA* 373 (1996), pp. 35–42.
- 2591 [44] M. Abbrescia et al. “Cosmic ray tests of double-gap resistive plate chambers for the CMS  
2592 experiment”. In: *NIMA* 550 (2005), pp. 116–126.
- 2593 [45] ALICE Collaboration. “A study of the multigap RPC at the gamma irradiation facility at  
2594 CERN”. In: *NIMA* 490 (2002), pp. 58–70.
- 2595 [46] B. Bonner et al. “A multigap resistive plate chamber prototype for time-of-flight for the STAR  
2596 experiment at RHIC”. In: *NIMA* 478 (2002), pp. 176–179.
- 2597 [47] S. Yang et al. “Test of high time resolution MRPC with different readout modes for the  
2598 BESIII upgrade”. In: *NIMA* 763 (2014), pp. 190–196.
- 2599 [48] A. Akindinov et al. “RPC with low-resistive phosphate glass electrodes as a candidate for  
2600 the CBM TOF”. In: *NIMA* 572 (2007), pp. 676–681.
- 2601 [49] JINST, ed. *Development of the MRPC for the TOF system of the MultiPurpose Detector.*  
2602 RPC2016: XII Workshop on Resistive Plate Chambers and Related Detectors. 2016.
- 2603 [50] M.C.S. Williams. “Particle identification using time of flight”. In: *Journal of Physics G* 39  
2604 (2012).
- 2605 [51] A. Alici et al. “Aging and rate effects of the Multigap RPC studied at the Gamma Irradiation  
2606 Facility at CERN”. In: *NIMA* 579 (2007), pp. 979–988.
- 2607 [52] M. Abbrescia et al. “The simulation of resistive plate chambers in avalanche mode: charge  
2608 spectra and efficiency”. In: *NIMA* 431 (1999), pp. 413–427.
- 2609 [53] M. Abbrescia et al. “Study of long-term performance of CMS RPC under irradiation at the  
2610 CERN GIF”. In: *NIMA* 533 (2004), pp. 102–106.
- 2611 [54] H.C. Kim et al. “Quantitative aging study with intense irradiation tests for the CMS forward  
2612 RPCs”. In: *NIMA* 602 (2009), pp. 771–774.
- 2613 [55] S. Agosteo et al. “A facility for the test of large-area muon chambers at high rates”. In: *NIMA*  
2614 452 (2000), pp. 94–104.
- 2615 [56] PoS, ed. *CERN GIF ++ : A new irradiation facility to test large-area particle detectors for*  
2616 *the high-luminosity LHC program.* Vol. TIPP2014. 2014, pp. 102–109.
- 2617 [57] A. Fagot. *GIF++ DAQ v4.0*. 2017. URL: [https://github.com/afagot/GIF\\_DAQ](https://github.com/afagot/GIF_DAQ).
- 2618 [58] CAEN. *Mod. V1190-VX1190 A/B, 128/64 Ch Multihit TDC*. 14th ed. 2016.
- 2619 [59] CAEN. *Mod. V1718 VME USB Bridge*. 9th ed. 2009.
- 2620 [60] W-Ie-Ne-R. *VME 6021-23 VXI*. 5th ed. 2016.
- 2621 [61] Wikipedia. *INI file*. 2017. URL: [https://en.wikipedia.org/wiki/INI\\_file](https://en.wikipedia.org/wiki/INI_file).
- 2622 [62] S. Carrillo A. Fagot. *GIF++ Offline Analysis v6*. 2017. URL: [https://github.com/afagot/GIF\\_OfflineAnalysis](https://github.com/afagot/GIF_OfflineAnalysis).