



Universiteit Gent
Faculteit Wetenschappen
Vakgroep Fysica en Sterrenkunde

² No title yet

³ No sub-title neither, obviously...

⁴ 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

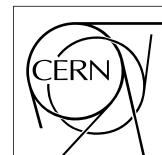
Promotoren: Dr. Michael Tytgat
Prof. Dr. Dirk Ryckbosch

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

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

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

Table of Contents

24	Acknowledgements	i
25	Nederlandse samenvatting	xvii
26	English summary	xix
27	1 Introduction	1-1
28	1.1 A story of High Energy Physics	1-1
29	1.2 Organisation of this study	1-1
30	2 Investigating the TeV scale	2-1
31	2.1 The Standard Model of Particle Physics	2-1
32	2.2 The Large Hadron Collider and the Compact Muon Solenoid	2-1
33	2.3 Muon Phase-II Upgrade	2-1
34	3 Amplification processes in gaseous detectors	3-1
35	3.1 Signal formation	3-1
36	3.2 Gas transport parameters	3-1
37	4 Resistive Plate Chambers	4-1
38	4.1 Principle	4-1
39	4.2 Rate capability of Resistive Plate Chambers	4-1
40	4.3 High time resolution	4-1
41	4.4 Resistive Plate Chambers at CMS	4-1
42	4.4.1 Overview	4-1
43	4.4.2 The present RPC system	4-2
44	4.4.3 Pulse processing of CMS RPCs	4-3
45	5 Longevity studies and Consolidation of the present CMS RPC subsystem	5-1
46	5.1 Testing detectors under extreme conditions	5-1
47	5.1.1 The Gamma Irradiation Facilities	5-3
48	5.1.1.1 GIF	5-3
49	5.1.1.2 GIF++	5-5
50	5.2 Preliminary tests at GIF	5-7
51	5.2.1 Resistive Plate Chamber test setup	5-7
52	5.2.2 Data Acquisition	5-9
53	5.2.3 Geometrical acceptance of the setup layout to cosmic muons	5-9
54	5.2.3.1 Description of the simulation layout	5-10
55	5.2.3.2 Simulation procedure	5-12
56	5.2.3.3 Results	5-13
57	5.2.4 Photon flux at GIF	5-13

58	5.2.4.1	Expectations from simulations	5-13
59	5.2.4.2	Dose measurements	5-18
60	5.2.5	Results and discussions	5-19
61	5.3	Longevity tests at GIF++	5-20
62	5.3.1	Description of the Data Acquisition	5-23
63	5.3.2	RPC current, environmental and operation parameter monitoring	5-24
64	5.3.3	Measurement procedure	5-25
65	5.3.4	Longevity studies results	5-25
66	6	Investigation on high rate RPCs	6-1
67	6.1	Rate limitations and ageing of RPCs	6-1
68	6.1.1	Low resistivity electrodes	6-1
69	6.1.2	Low noise front-end electronics	6-1
70	6.2	Construction of prototypes	6-1
71	6.3	Results and discussions	6-1
72	7	Conclusions and outlooks	7-1
73	7.1	Conclusions	7-1
74	7.2	Outlooks	7-1
75	A	A data acquisition software for CAEN VME TDCs	A-1
76	A.1	GIF++ DAQ file tree	A-1
77	A.2	Usage of the DAQ	A-2
78	A.3	Description of the readout setup	A-3
79	A.4	Data read-out	A-3
80	A.4.1	V1190A TDCs	A-4
81	A.4.2	DataReader	A-6
82	A.5	Communications	A-10
83	A.5.1	V1718 USB Bridge	A-10
84	A.5.2	Configuration file	A-11
85	A.5.3	WebDCS/DAQ intercommunication	A-15
86	A.5.4	Example of inter-process communication cycle	A-16
87	A.6	Software export	A-16
88	B	Details on the offline analysis package	B-1
89	B.1	GIF++ Offline Analysis file tree	B-1
90	B.2	Usage of the Offline Analysis	B-2
91	B.2.1	Output of the offline tool	B-3
92	B.2.1.1	ROOT file	B-3
93	B.2.1.2	CSV files	B-5
94	B.3	Analysis inputs and information handling	B-6
95	B.3.1	Dimensions file and IniFile parser	B-6
96	B.3.2	TDC to RPC link file and Mapping	B-7
97	B.4	Description of GIF++ setup within the Offline Analysis tool	B-8
98	B.4.1	RPC objects	B-9
99	B.4.2	Trolley objects	B-9
100	B.4.3	Infrastructure object	B-10
101	B.5	Handeling of data	B-12
102	B.5.1	RPC hits	B-12
103	B.5.2	Clusters of hits	B-14

104	C Structure of the hybrid simulation software	C-1
105	C.1 Introduction	C-1

List of Figures

107	2.1	Absorbed dose in the CMS cavern after an integrated luminosity of 3000 fb. R is the transverse distance from the beamline and Z is the distance along the beamline from the Interaction Point at Z=0.	2-2
108			
109			
110	2.2	A quadrant of the muon system, showing DTs (yellow), RPCs (light blue), and CSCs (green). The locations of new forward muon detectors for Phase-II are contained within the dashed box and indicated in red for GEM stations (ME0, GE1/1, and GE2/1) and dark blue for improved RPC (iRPC) stations (RE3/1 and RE4/1).	2-2
111			
112			
113			
114	2.3	RMS of the multiple scattering displacement as a function of muon p_T for the proposed forward muon stations. All of the electromagnetic processes such as bremsstrahlung and magnetic field effect are included in the simulation.	2-3
115			
116			
117	4.1	Signals from the RPC strips are shaped by the FEE described on Figure 4.1a. Output LVDS signals are then read-out by a TDC module connected to a computer or converted into NIM and sent to scalers. Figure 4.1b describes how these converted signals are put in coincidence with the trigger.	4-3
118			
119			
120			
121	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
122			
123			
124			
125			
126			
127			
128			
129			
130	5.1	(5.1a) Extrapolation from 2016 data of single hit rate per unit area in the barrel region. (5.1b) Extrapolation from 2016 data of single hit rate per unit area in the endcap region.	5-2
131			
132			
133	5.2	Background Fluka simulation compared to 2016 Data at $L = 10^{34} \text{ cm}^{-2} \cdot \text{s}^{-1}$ in the fourth endcap disk region. A mismatch in between simulation and data can be observed. [To be understood.]	5-3
134			
135			
136	5.3	Layout of the test beam zone called X5c GIF at CERN. Photons from the radioactive source produce a sustained high rate of random hits over the whole area. The zone is surrounded by 8 m high and 80 cm thick concrete walls. Access is possible through three entry points. Two access doors for personnel and one large gate for material. A crane allows installation of heavy equipment in the area.	5-4
137			
138			
139			
140			
141	5.4	^{137}Cs decays by β^- emission to the ground state of ^{137}Ba (BR = 5.64%) and via the 662 keV isomeric level of ^{137}Ba (BR = 94.36%) whose half-life is 2.55 min.	5-5
142			

143	5.5	Floor plan of the GIF++ facility. When the facility downstream of the GIF++ takes electron beam, a beam pipe is installed along the beam line (z-axis). The irradiator can be displaced laterally (its center moves from $x = 0.65$ m to 2.15 m), to increase the distance to the beam pipe.	5-5
144			
145			
146			
147	5.6	Simulated unattenuated current of photons in the xz plane (Figure 5.6a) and yz plane (Figure 5.6b) through the source at $x = 0.65$ m and $y = 0$ m. With angular correction filters, the current of 662 keV photons is made uniform in xy planes.	5-6
148			
149			
150	5.7	Description of the RPC setup. Dimensions are given in mm. A tent containing RPCs is placed at 1720 mm from the source container. The source is situated in the center of the container. RE-4-2-BARC-161 chamber is 160 mm inside the tent. This way, the distance between the source and the chambers plan is 2060 mm. Figure 5.7a provides a side view of the setup in the xz plane while Figure 5.7b shows a top view in the yz plane.	5-7
151			
152			
153			
154			
155			
156	5.8	RE-4-2-BARC-161 chamber is inside the tent as described in Figure 5.7. In the top right, the two scintillators used as trigger can be seen. This trigger system has an inclination of 10° relative to horizontal and is placed above half-partition B2 of the RPCs. PMT electronics are shielded thanks to lead blocks placed in order to protect them without stopping photons from going through the scintillators and the chamber.	5-8
157			
158			
159			
160			
161	5.9	Hit distributions over all 3 partitions of RE-4-2-BARC-161 chamber is showed on these plots. Top, middle and bottom figures respectively correspond to partitions A, B, and C. These plots show that some events still occur in other half-partitions than B2, which corresponds to strips 49 to 64, in front of which the trigger is placed, contributing to the inefficiency of detection of cosmic muons. In the case of partitions A and C, the very low amount of data can be interpreted as noise. On the other hand, it is clear that a little portion of muons reach the half-partition B1, corresponding to strips 33 to 48.	5-9
162			
163			
164			
165			
166			
167			
168			
169	5.10	Results are derived from data taken on half-partition B2 only. On the 18 th of June 2014, data has been taken on chamber RE-2-BARC-161 at building 904 (Prevessin Site) with cosmic muons providing us a reference efficiency plateau of $(97.54 \pm 0.15)\%$ represented by a black curve. A similar measurement has been done at GIF on the 21 st of July with the same chamber giving a plateau of $(78.52 \pm 0.94)\%$ represented by a red curve.	5-10
170			
171			
172			
173			
174			
175	5.11	Representation of the layout used for the simulations of the test setup. The RPC is represented as a yellow trapezoid while the two scintillators as blue cuboids looking at the sky. A green plane corresponds to the muon generation plane within the simulation. Figure 5.7a shows a global view of the simulated setup. Figure 5.7b shows a zommed view that allows to see the 2 scintillators as well as the full RPC plane.	5-11
176			
177			
178			
179			
180	5.12	γ flux $F(D)$ is plot using values from table 5.1. As expected, the plot shows similar attenuation behaviours with increasing distance for each absorption factors.	5-14
181			
182	5.13	Figure 5.13a shows the linear approximation fit done via formulae 5.7 on data from table 5.2. Figure 5.13b shows a comparison of this model with the simulated flux using a and b given in figure 5.13a in formulae 5.4 and the reference value $D_0 = 50\text{cm}$ and the associated flux for each absorption factor F_0^{ABS} from table 5.1	5-16
183			
184			
185			
186	5.14	Dose measurements has been done in a plane corresponding to the tents front side. This plan is 1900 mm away from the source. As explained in the first chapter, a lens-shaped lead filter provides a uniform photon flux in the vertical plan orthogonal to the beam direction. If the second line of measured fluxes is not taken into account because of lower values due to experimental equipments in the way between the source and the tent, the uniformity of the flux is well showed by the results.	5-18
187			
188			
189			
190			
191			

192	5.15	5-19
193	5.16 Evolution of the maximum efficiency for RE2 (5.16a) and RE4 (5.16b) chambers with increasing extrapolated γ rate per unit area at working point. Both irradiated (blue) and non irradiated (red) chambers are shown.	5-21
194		
195		
196	5.17 Evolution of the working point for RE2 (5.17a) and RE4 (5.17b) with increasing extrapolated γ rate per unit area at working point. Both irradiated (blue) and non irradiated (red) chambers are shown.	5-21
197		
198		
199	5.18 Evolution of the maximum efficiency at HL-LHC conditions, i.e. a background hit rate per unit area of 300 Hz/cm ² , with increasing integrated charge for RE2 (5.18a) and RE4 (5.18b) detectors. Both irradiated (blue) and non irradiated (red) chambers are shown. The integrated charge for non irradiated detectors is recorded during test beam periods and stays small with respect to the charge accumulated in irradiated chambers.	5-22
200		
201		
202		
203		
204		
205	5.19 Comparison of the efficiency sigmoid before (triangles) and after (circles) irradiation for RE2 (5.19a) and RE4 (5.19b) detectors. Both irradiated (blue) and non irradiated (red) chambers are shown.	5-22
206		
207		
208	5.20 Evolution of the Bakelite resistivity for RE2 (5.20a) and RE4 (5.20b) detectors. Both irradiated (blue) and non irradiated (red) chambers are shown.	5-23
209		
210	5.21 Evolution of the noise rate per unit area for the irradiated chamber RE2-2-BARC-9 only.	5-23
211		
212	A.1 (A.1a) View of the front panel of a V1190A TDC module [10]. (A.1b) View of the front panel of a V1718 Bridge module [11]. (A.1c) View of the front panel of a 6U 6021 VME crate [12].	A-3
213		
214		
215	A.2 Module V1190A <i>Trigger Matching Mode</i> timing diagram [10].	A-4
216		
217	A.3 Structure of the ROOT output file generated by the DAQ. The 5 branches (<code>EventNumber</code> , <code>number_of_hits</code> , <code>Quality_flag</code> , <code>TDC_channel</code> and <code>TDC_TimeStamp</code>) are visible on the left panel of the ROOT browser. On the right panel is visible the histogram cor- responding to the variable <code>nHits</code> . In this specific example, there were approximately 50k events recorded to measure the gamma irradiation rate on the detectors. Each event is stored as a single entry in the <code>TTree</code>	A-10
218		
219		
220		
221		
222	A.4 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 in- formation will be stored in the DAQ ROOT output. Are also given the minimal measurement time and waiting time after ramping up of the detectors is over before starting the data acquisition. Then, the list of HV points to scan and the number of triggers for each run of the scan are given in the table underneath.	A-12
223		
224		
225		
226		
227		
228		

List of Tables

229

230 5.1	Total photon flux ($E\gamma \leq 662$ keV) with statistical error predicted considering a 231 ^{137}Cs activity of 740 GBq at different values of the distance D to the source along 232 the x-axis of irradiation field [6].	5-13
233 5.2	Correction factor c is computed thanks to formulae 5.5 taking as reference $D_0 =$ 234 50 cm and the associated flux F_0^{ABS} for each absorption factor available in table 5.1.	5-15
235 5.3	The data at D_0 in 1997 is taken from [6]. In a second step, using Equations 5.8 236 and 5.9, the flux at D can be estimated in 1997. Then, taking into account the 237 attenuation of the source activity, the flux at D can be estimated at the time of the 238 tests in GIF in 2014. Finally, assuming a sensitivity of the RPC to γ $s = 2 \cdot 10^{-3}$, 239 an estimation of the hit rate per unit area is obtained.	5-17
240 A.1	Inter-process communication cycles in between the webDCS and the DAQ through 241 file string signals.	A-17

242

List of Acronyms

243

List of Acronyms

244

245

A

246

247

248 AFL Almost Full Level

249

250

B

251

252

253 BARC

Bhabha Atomic Research Centre

254 BLT

Block Transfer

255 BR

Branching Ratio

256

257

C

258

259

260 CAEN

Costruzioni Apparecchiature Elettroniche Nucleari S.p.A.

261 CERN

European Organization for Nuclear Research

262 CFD

Constant Fraction Discriminator

263 CMS

Compact Muon Solenoid

264 CSC

Cathode Strip Chamber

265

266

D

267

268

269 DAQ

Data Acquisition

270 DCS

Detector Control Software

271 DQM

Data Quality Monitoring

272 DT

Drift Tube

273

274

F

275

276

277	FEE	Front-End Electronics
278	FEB	Front-End Board
279		
280	G	
281		
283	GE-/-	Find a good description
284	GE1/1	Find a good description
285	GE2/1	Find a good description
286	GEANT	GEometry ANd Tracking - a series of software toolkit platforms developed by CERN
287		
288	GEM	Gas Electron Multiplier
289	GIF	Gamma Irradiation Facility
290	GIF++	new Gamma Irradiation Facility
291		
292	H	
293		
295	HL-LHC	High Luminosity LHC
296	HV	High Voltage
297		
298	I	
299		
301	iRPC	improved RPC
302	IRQ	Interrupt Request
303		
304	L	
305		
307	LHC	Large Hadron Collider
308	LS1	First Long Shutdown
309	LS3	Third Long Shutdown
310	LV	Low Voltage
311	LVDS	Low-Voltage Differential Signaling
312		
313	M	
314		
316	MC	Monte Carlo
317	MCNP	Monte Carlo N-Particle
318	ME-/-	Find good description
319	ME0	Find good description

320		
321	N	
322		
323		
324	NIM	Nuclear Instrumentation Module logic signals
325		
326	P	
327		
328		
329	PMT	PhotoMultiplier Tube
330		
331	R	
332		
333		
334	RE-/-	Find a good description
335	RE2/2	Find a good description
336	RE3/1	Find a good description
337	RE3/2	Find a good description
338	RE4/1	Find a good description
339	RE4/2	Find a good description
340	RE4/3	Find a good description
341	RMS	Root Mean Square
342	ROOT	a framework for data processing born at CERN
343	RPC	Resistive Plate Chamber
344		
345	S	
346		
347		
348	SPS	Super Proton Synchrotron
349		
350	T	
351		
352		
353	TDC	Time-to-Digital Converter
354		
355	W	
356		
357		
358	webDCS	Web Detector Control System

360

Nederlandse samenvatting –Summary in Dutch–

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

361

English summary

³⁶⁴ Le meme résume mais en Anglais (on commencera par la hein!).

1

Introduction

365

366

³⁶⁷ **1.1 A story of High Energy Physics**

³⁶⁸ **1.2 Organisation of this study**

2

369

370

Investigating the TeV scale

371 2.1 The Standard Model of Particle Physics

372 2.2 The Large Hadron Collider and the Compact Muon Solenoid

373 2.3 Muon Phase-II Upgrade

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

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

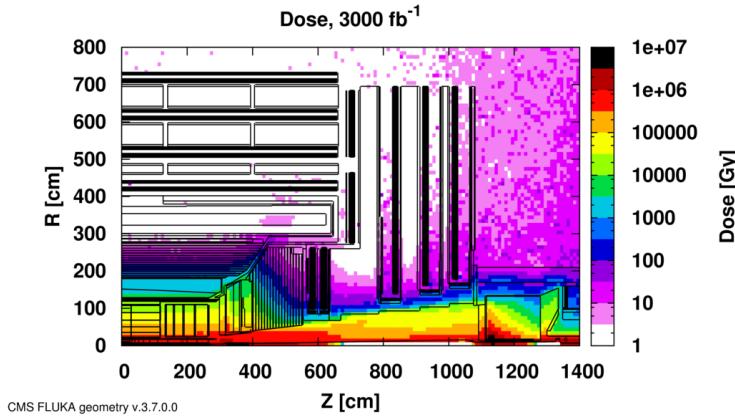


Figure 2.1: Absorbed dose in the CMS cavern after an integrated luminosity of 3000 fb^{-1} . R is the transverse distance from the beamline and Z is the distance along the beamline from the Interaction Point at $Z=0$.

388 The measurement of small production cross-section and/or decay branching ratio processes, such
 389 as the Higgs boson coupling to charge leptons or the $B_s \rightarrow \mu^+ \mu^-$ decay, is of major interest and
 390 specific upgrades in the forward regions of the detector will be required to maximize the physics
 391 acceptance on the largest possible solid angle. To ensure proper trigger performance within the
 392 present coverage, the muon system will be completed with new chambers. In figure 2.2 one can
 393 see that the existing Cathode Strip Chambers (CSCs) will be completed by Gas Electron Multipliers
 394 (GEMs) and Resistive Plate Chambers (RPCs) in the pseudo-rapidity region $1.6 < |\eta| < 2.4$ to
 395 complete its redundancy as originally scheduled in the CMS Technical Proposal [3].

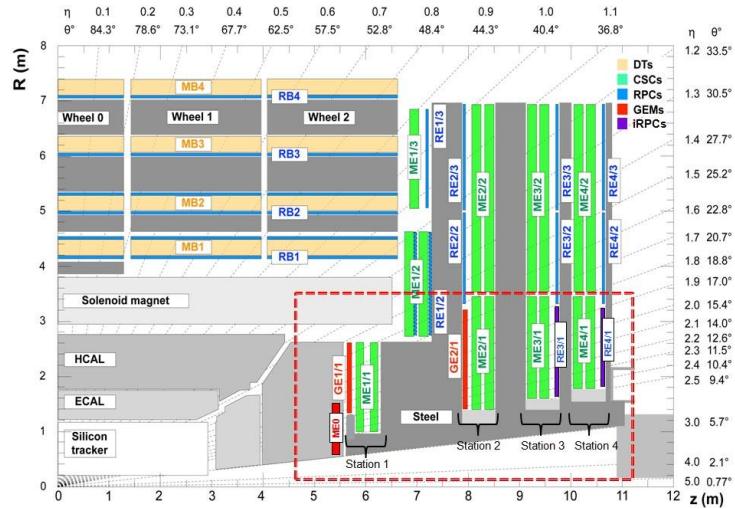


Figure 2.2: A quadrant of the muon system, showing DTs (yellow), RPCs (light blue), and CSCs (green). The locations of new forward muon detectors for Phase-II are contained within the dashed box and indicated in red for GEM stations (ME0, GE1/1, and GE2/1) and dark blue for improved RPC (iRPC) stations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

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

397 good bunch crossing identification can be obtained with the present CMS RPC system, given their
 398 fast response of the order of 1 ns. In order to contribute to the precision of muon momentum mea-
 399 surements, muon chambers should have a spatial resolution less or comparable to the contribution
 400 of multiple scattering [1]. Most of the plausible physics is covered only considering muons with
 401 $p_T < 100$ GeV thus, in order to match CMS requirements, a spatial resolution of $\mathcal{O}(\text{few mm})$ the
 402 proposed new RPC stations, as shown by the simulation in figure 2.3. According to preliminary
 403 designs, RE3/1 and RE4/1 readout pitch will be comprised between 3 and 6 mm and 5 η -partitions
 404 could be considered.

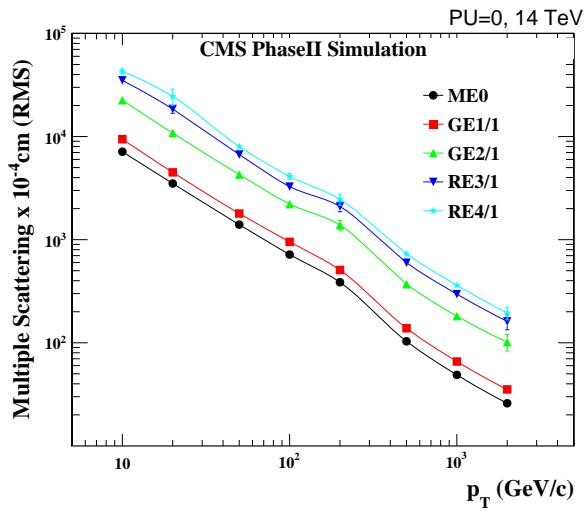


Figure 2.3: RMS of the multiple scattering displacement as a function of muon p_T for the proposed forward muon stations. All of the electromagnetic processes such as bremsstrahlung and magnetic field effect are included in the simulation.

3

405

406

Amplification processes in gaseous detectors

407

3.1 Signal formation

408

3.2 Gas transport parameters

4

409

410

Resistive Plate Chambers

411 4.1 Principle

412 4.2 Rate capability of Resistive Plate Chambers

413 4.3 High time resolution

414 4.4 Resistive Plate Chambers at CMS

415 4.4.1 Overview

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

419

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

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

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

439 **4.4.2 The present RPC system**

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

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

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

461 The Link Board has 96 input channels (one channel corresponds to one RPC strip). The input
 462 signals are the ~ 100 ns binary pulses which are synchronous to the RPC hits, but not to the LHC
 463 clock (which drives the entire CMS electronics). Thus the first step of the FEB signals processing
 464 is synchronization, i.e. assignment of the signals to the BXes (25 ns periods). Then the data are
 465 compressed with a simple zero-suppressing algorithm (the input channels are grouped into 8 bit
 466 partitions, only the partitions with at least one nonzero bit are selected for each BX). Next, the non-
 467 empty partitions are time-multiplexed i.e. if there are more than one such partition in a given BX,
 468 they are sent one-by-one in consecutive BXes. The data from 3 neighbouring LBs are concentrated
 469 by the middle LB which contains the optical transmitter for sending them to the USC over a fiber at
 470 1.6 Gbps.

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

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

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

483 4.4.3 Pulse processing of CMS RPCs

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

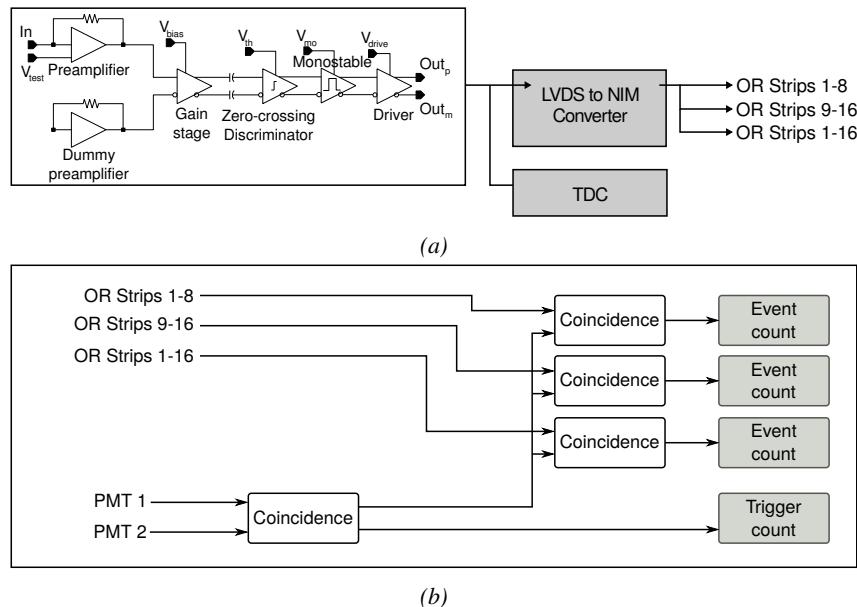


Figure 4.1: Signals from the RPC strips are shaped by the FEE described on Figure 4.1a. Output LVDS signals are then read-out by a TDC module connected to a computer or converted into NIM and sent to scalers. Figure 4.1b describes how these converted signals are put in coincidence with the trigger.

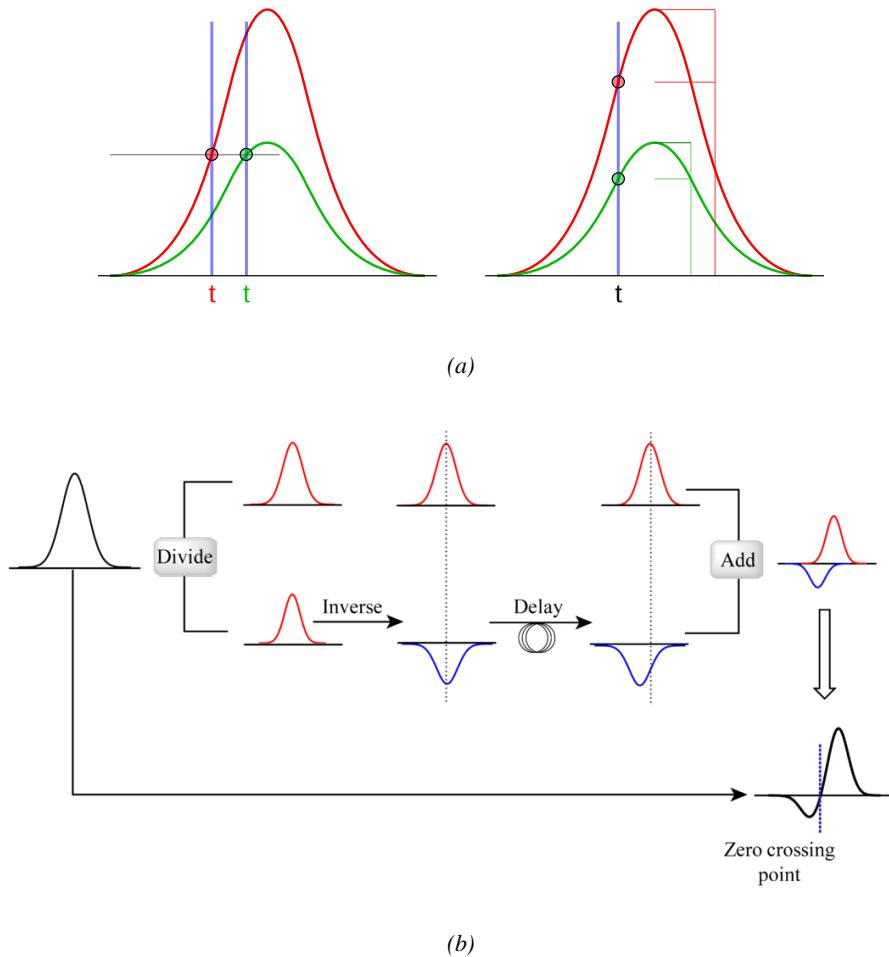


Figure 4.2: Description of the principle of a CFD. A comparison of threshold triggering (left) and constant fraction triggering (right) is shown in Figure 4.2a. Constant fraction triggering is obtained thanks to zero-crossing technique as explained in Figure 4.2b. The signal arriving at the input of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

5

492

493

494

Longevity studies and Consolidation of the present CMS RPC subsystem

495

5.1 Testing detectors under extreme conditions

496

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

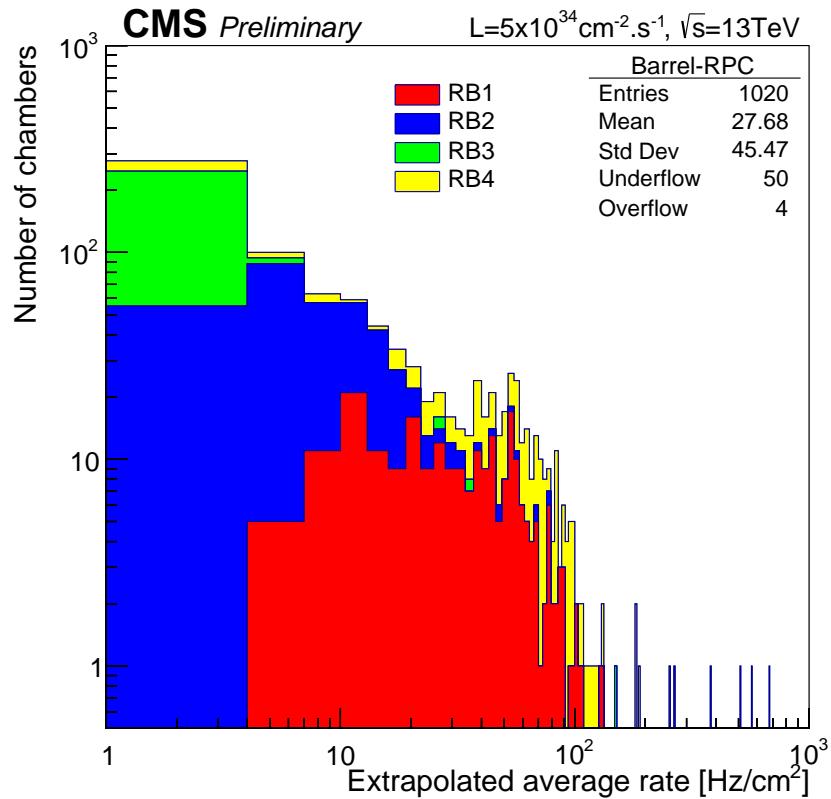
[To update.]

503

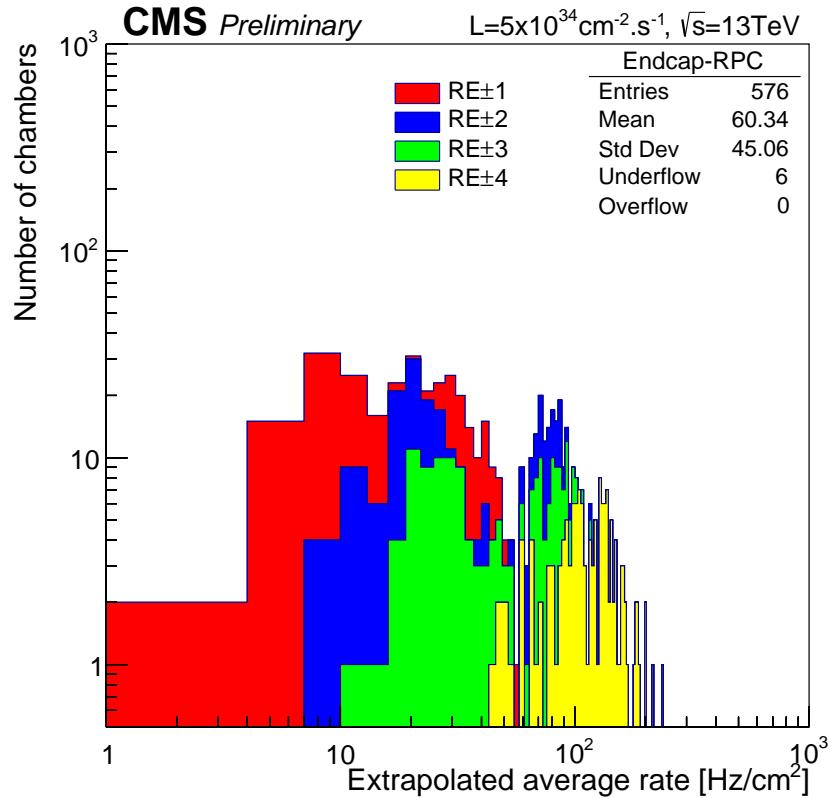
504

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

512



(a)



(b)

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

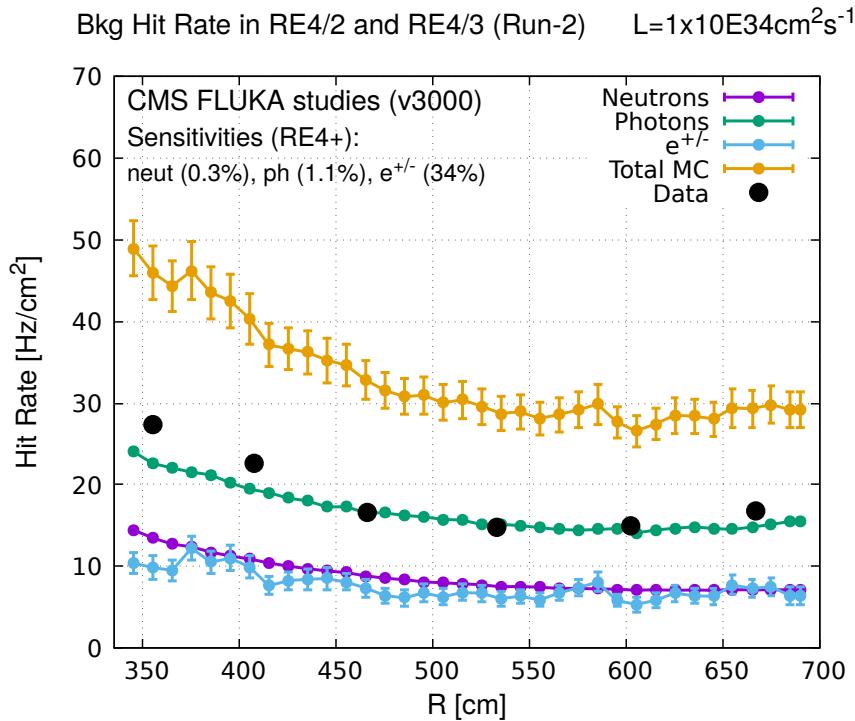


Figure 5.2: Background Fluka simulation compared to 2016 Data at $L = 10^{34}\text{cm}^{-2}\cdot\text{s}^{-1}$ in the fourth endcap disk region. A mismatch in between simulation and data can be observed. [\[To be understood.\]](#)

513 In the past, extensive long-term tests were carried out at several gamma and neutron facilities
 514 certifying the detector performance. Both full size and small prototype RPCs have been irradiated
 515 with photons up to an integrated charge of $\sim 0.05\text{C}/\text{cm}^2$ and $\sim 0.4\text{C}/\text{cm}^2$, respectively [4, 5].
 516 During Run-I, the RPC system provided stable operation and excellent performance and did not
 517 show any aging effects for integrated charge of the order of $0.01\text{C}/\text{cm}^2$. Projections on currents
 518 from 2016 Data, has allowed to determine that the total integrated charge, by the end of HL-LHC,
 519 would be of the order of $1\text{C}/\text{cm}^2$ (including a safety factor 3). [\[Corresponding figure needed.\]](#)

520

521 5.1.1 The Gamma Irradiation Facilities

522 5.1.1.1 GIF

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

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

537

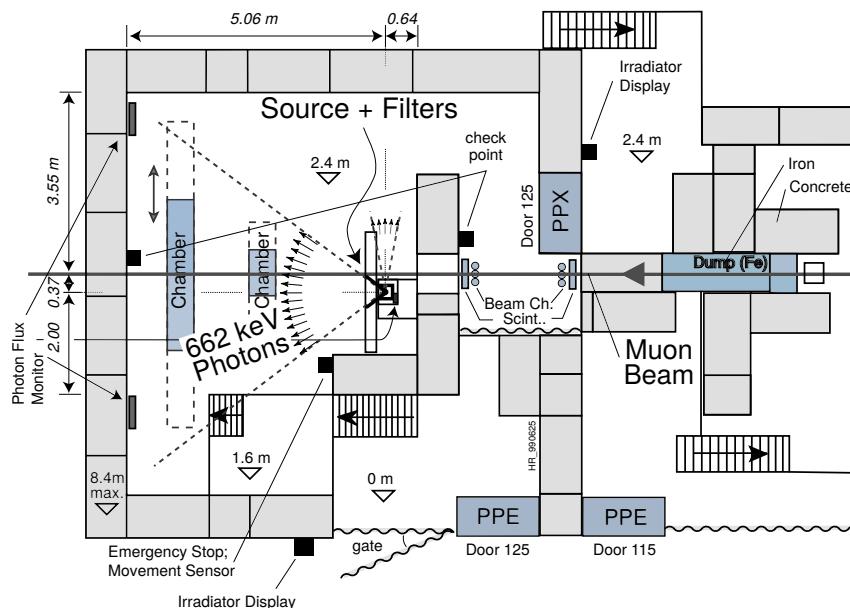


Figure 5.3: Layout of the test beam zone called X5c GIF at CERN. Photons from the radioactive source produce a sustained high rate of random hits over the whole area. The zone is surrounded by 8 m high and 80 cm thick concrete walls. Access is possible through three entry points. Two access doors for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

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

544

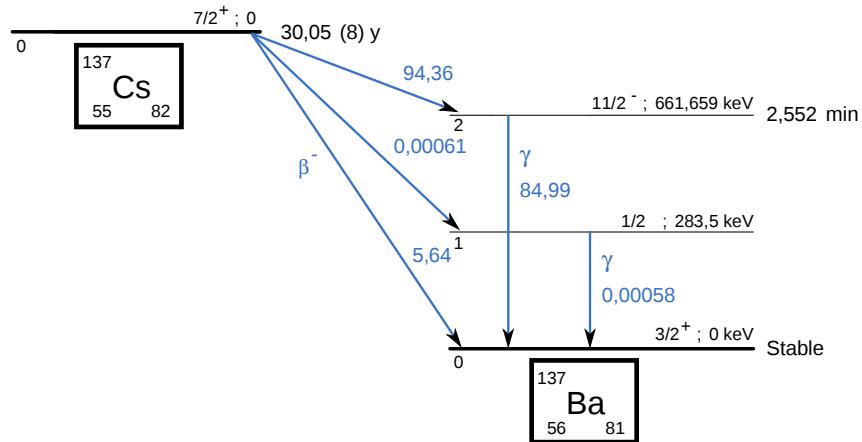


Figure 5.4: ^{137}Cs decays by β^- emission to the ground state of ^{137}Ba ($BR = 5.64\%$) and via the 662 keV isomeric level of ^{137}Ba ($BR = 94.36\%$) whose half-life is 2.55 min.

5.1.1.2 GIF++

The new Gamma Irradiation Facility (GIF++), located in the SPS North Area at the downstream end of the H4 test beam, has replaced its predecessor during LS1 and has been operational since spring 2015 [7]. Like GIF, GIF++ features a ^{137}Cs source of 662 keV gamma photons, their fluence being controlled with a set of filters of various attenuation factors. The source provides two separated large irradiation areas for testing several full-size muon detectors with continuous homogeneous irradiation, as presented in Figure 5.5.

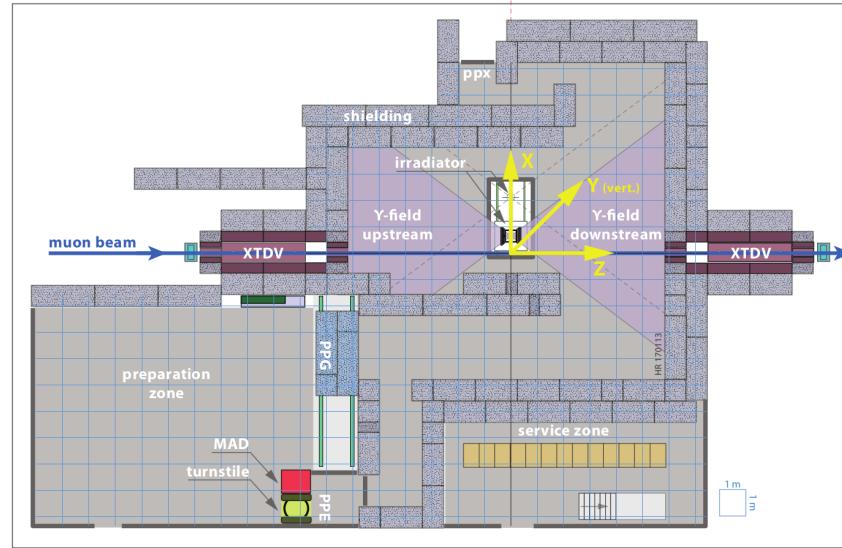


Figure 5.5: Floor plan of the GIF++ facility. When the facility downstream of the GIF++ takes electron beam, a beam pipe is installed along the beam line (z -axis). The irradiator can be displaced laterally (its center moves from $x = 0.65$ m to 2.15 m), to increase the distance to the beam pipe.

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

556

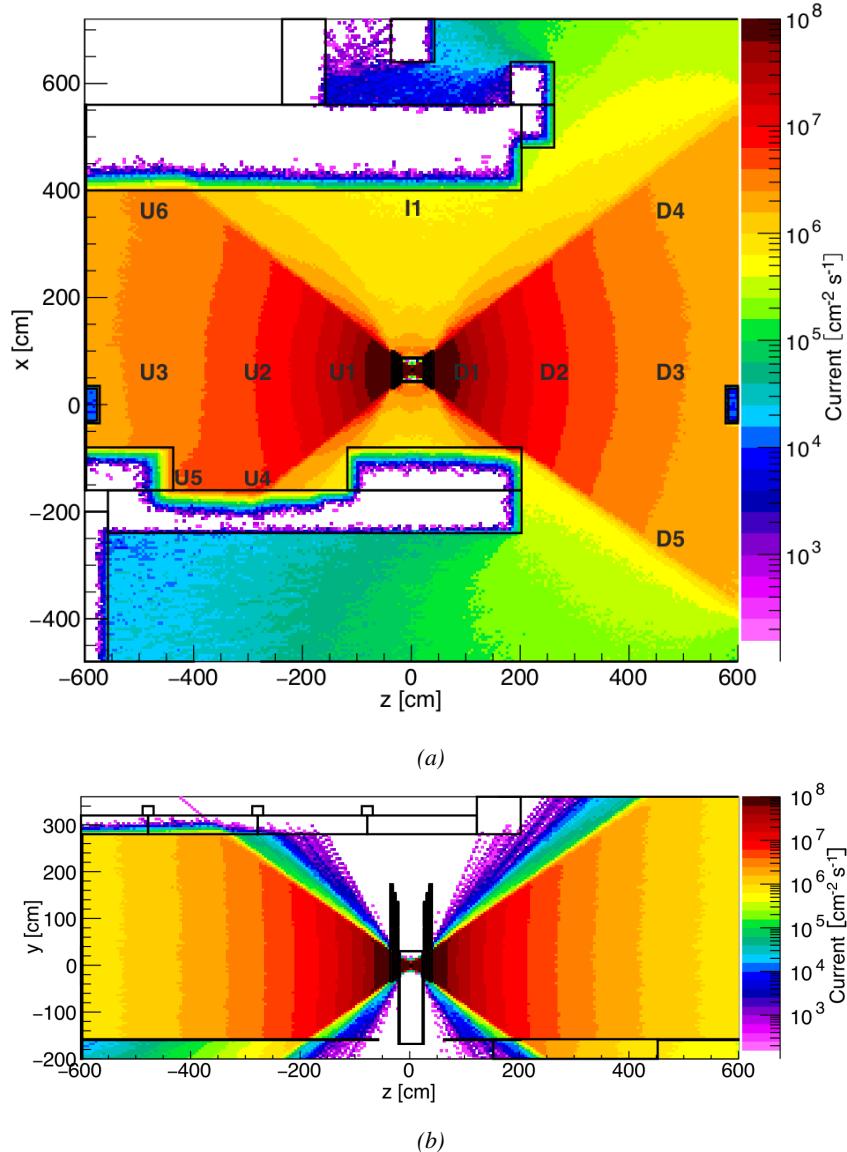


Figure 5.6: Simulated unattenuated current of photons in the xz plane (Figure 5.6a) and yz plane (Figure 5.6b) through the source at $x = 0.65$ m and $y = 0$ m. With angular correction filters, the current of 662 keV photons is made uniform in xy planes.

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

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

563

564 5.2 Preliminary tests at GIF

565 5.2.1 Resistive Plate Chamber test setup

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

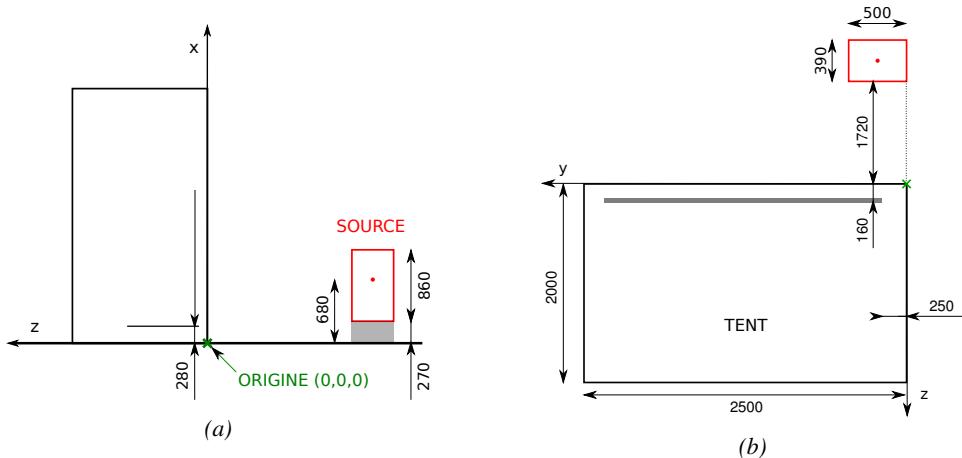


Figure 5.7: Description of the RPC setup. Dimensions are given in mm. A tent containing RPCs is placed at 1720 mm from the source container. The source is situated in the center of the container. RE-4-2-BARC-161 chamber is 160 mm inside the tent. This way, the distance between the source and the chambers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



Figure 5.8: RE-4-2-BARC-161 chamber is inside the tent as described in Figure 5.7. In the top right, the two scintillators used as trigger can be seen. This trigger system has an inclination of 10° relative to horizontal and is placed above half-partition B2 of the RPCs. PMT electronics are shielded thanks to lead blocks placed in order to protect them without stopping photons from going through the scintillators and the chamber.

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

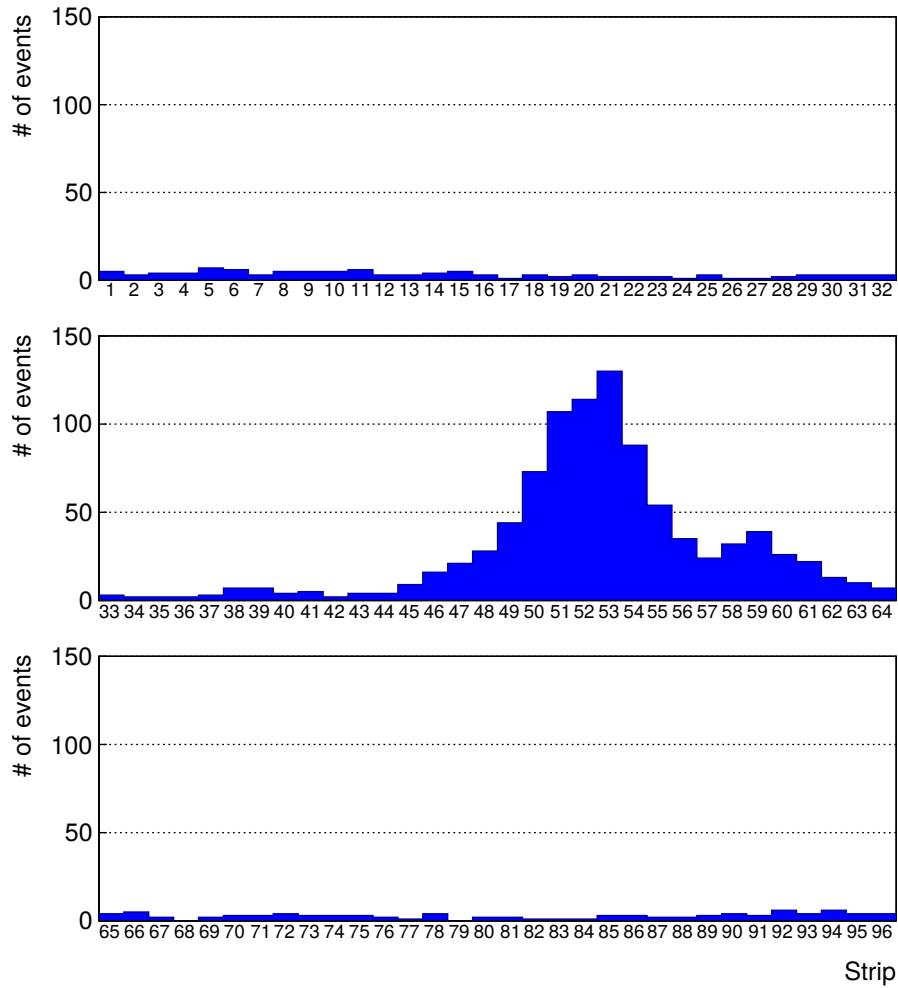


Figure 5.9: Hit distributions over all 3 partitions of RE-4-2-BARC-161 chamber is showed on these plots. Top, middle and bottom figures respectively correspond to partitions A, B, and C. These plots show that some events still occur in other half-partitions than B2, which corresponds to strips 49 to 64, in front of which the trigger is placed, contributing to the inefficiency of detection of cosmic muons. In the case of partitions A and C, the very low amount of data can be interpreted as noise. On the other hand, it is clear that a little portion of muons reach the half-partition B1, corresponding to strips 33 to 48.

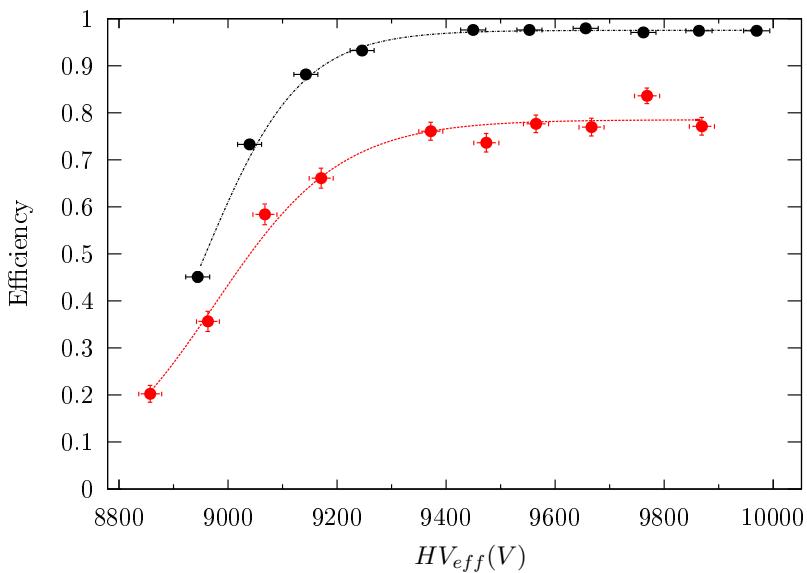
581 5.2.2 Data Acquisition

582 5.2.3 Geometrical acceptance of the setup layout to cosmic muons

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

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

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



598 *Figure 5.10: Results are derived from data taken on half-partition B2 only. On the 18th of June 2014, data
 599 has been taken on chamber RE-2-BARC-161 at building 904 (Prevessin Site) with cosmic muons providing us a
 600 reference efficiency plateau of $(97.54 \pm 0.15)\%$ represented by a black curve. A similar measurement has been
 601 done at GIF on the 21st of July with the same chamber giving a plateau of $(78.52 \pm 0.94)\%$ represented by a
 602 red curve.*

598 **5.2.3.1 Description of the simulation layout**

600 The layout of GIF setup has been reproduced and incorporated into a Monte Carlo (MC) simulation
 601 to study the influence of the disposition of the telescope on the final distribution measured by the
 602 RPC. A 3D view of the simulated layout is given into Figure 5.11. Muons are generated randomly
 603 in a horizontal plane located at a height corresponding to the lowest point of the PMTs. This way,
 604 the needed size of the plane in order to simulate events happening at very big azimuthal angles (i.e.
 605 $\theta \approx \pi$) can be kept relatively small. The muon flux is designed to follow the usual $\cos^2\theta$ distribution
 606 for cosmic particle. The goal of the simulation is to look at muons that pass through the muon
 607 telescope composed of the two scintillators and define their distribution onto the RPC plane. During
 608 the reconstruction, the RPC plane is then divided into its strips and each muon track is assigned to a
 609 strip.

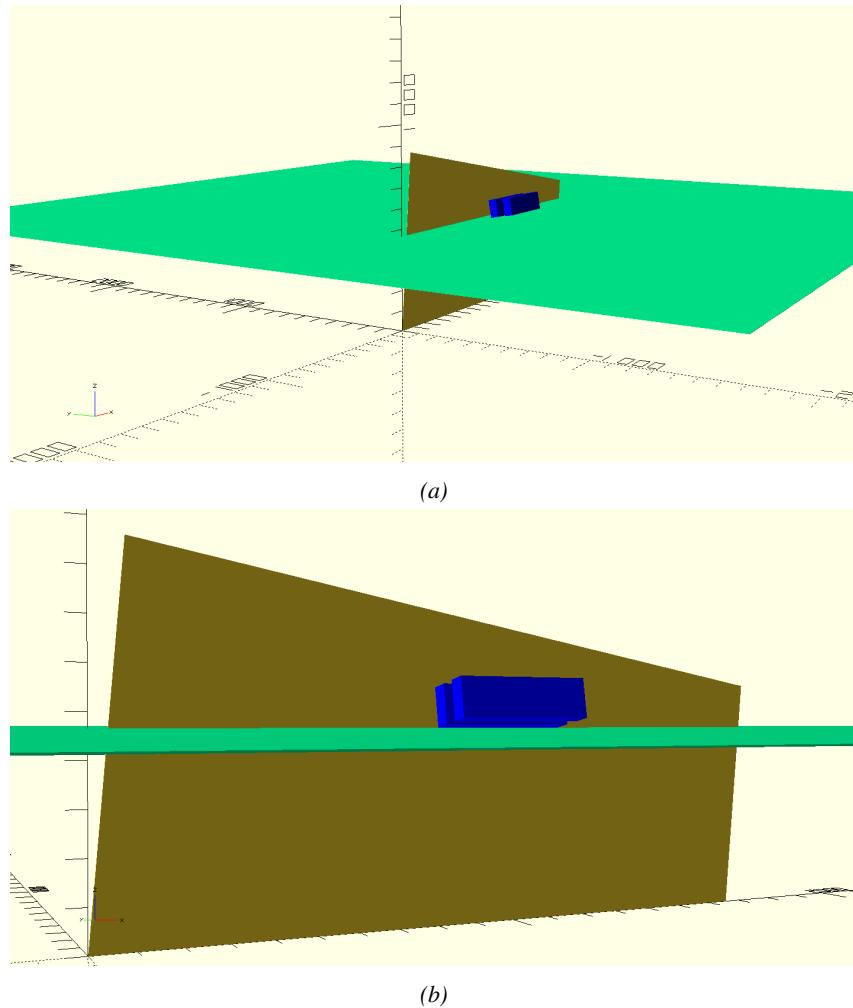


Figure 5.11: Representation of the layout used for the simulations of the test setup. The RPC is represented as a yellow trapezoid while the two scintillators as blue cuboids looking at the sky. A green plane corresponds to the muon generation plane within the simulation. Figure 5.7a shows a global view of the simulated setup. Figure 5.7b shows a zoomed view that allows to see the 2 scintillators as well as the full RPC plane.

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

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

623 **5.2.3.2 Simulation procedure**

624 The simulation software has been designed using C++ and the output data is saved into ROOT
 625 histograms. Simulations start for a threshold T_{scint} varying in a range from 0 to 45 mm in steps
 626 of 5 mm, where $T_{scint} = 0$ mm corresponds to the case where there isn't any threshold apply on
 627 the input signal while $T_{scint} = 45$ mm, which is the scintillator thickness, is the case where muons
 628 cannot arrive orthogonally onto the scintillator surface. For a given T_{scint} , a set of RPC thresholds
 629 are considered. The RPC threshold, T_{RPC} varies from 2 mm, the thickness of the gas volume, to
 630 3 mm in steps of 0.25 mm. For each $(T_{scint}; T_{RPC})$ pair, $N_\mu = 10^8$ muons are randomly generated
 631 inside the muon plane described in the previous paragraph with an azimuthal angle θ chosen to follow
 632 a $\cos^2\theta$ distribution.

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

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

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

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

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

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

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

5.2.3.3 Results

Influence of T_{scint} on the muon distribution

Influence of T_{RPC} on the muon distribution

Influence of the telescope inclination on the muon distribution

Comparison to data taken at GIF without irradiation

5.2.4 Photon flux at GIF

5.2.4.1 Expectations from simulations

In order to understand and evaluate the γ flux in the GIF area, simulations had been conducted in 1999 and published by S. Agosteo et al [6]. Table 5.1 presented in this article gives us the γ flux for different distances D to the source. This simulation was done using GEANT and a Monte Carlo N-Particle (MCNP) transport code, and the flux F is given in number of γ per unit area and unit time along with the estimated error from these packages expressed in %.

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

Table 5.1: Total photon flux ($E\gamma \leq 662$ keV) with statistical error predicted considering a ^{137}Cs activity of 740 GBq at different values of the distance D to the source along the x -axis of irradiation field [6].

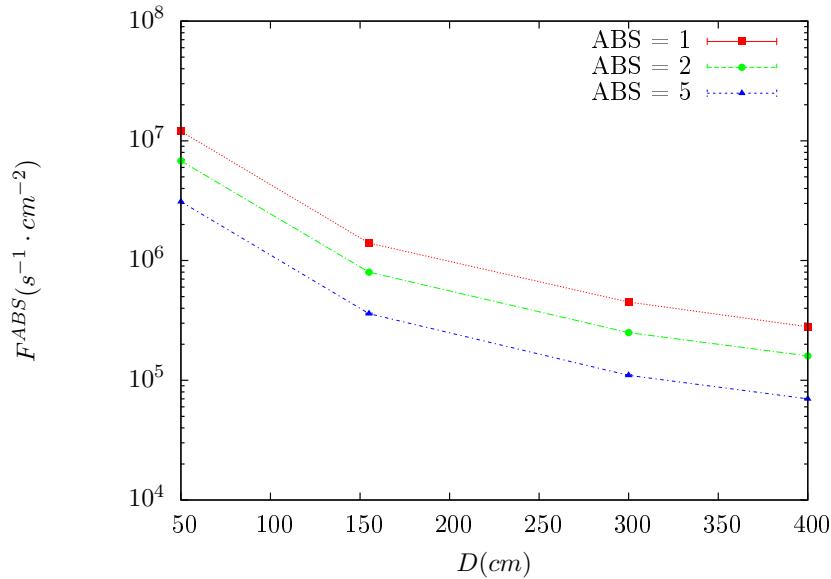


Figure 5.12: γ flux $F(D)$ is plot using values from table 5.1. As expected, the plot shows similar attenuation behaviours with increasing distance for each absorption factors.

The simulation doesn't directly 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 5.12 that contains the data from Table 5.1. In the case of a pointlike source emitting isotropic and homogeneous gamma radiations, the gamma flux F at a distance D to the source with respect to a reference point situated at D_0 where a known flux F_0 is measured will be expressed like in Equation 5.4, assuming that the flux decreases as $1/D^2$, where c is a fitting factor.

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

By rewriting Equation 5.4, it comes that :

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

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

Finally, using Equation 5.5 and the data in Table 5.1 with $D_0 = 50$ cm as reference point, we can build Table 5.2. It is interesting to note that c for each value of D doesn't depend on the absorption factor.

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

Table 5.2: Correction factor c is computed thanks to formulae 5.5 taking as reference $D_0 = 50$ cm and the associated flux F_0^{ABS} for each absorption factor available in table 5.1.

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

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

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

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

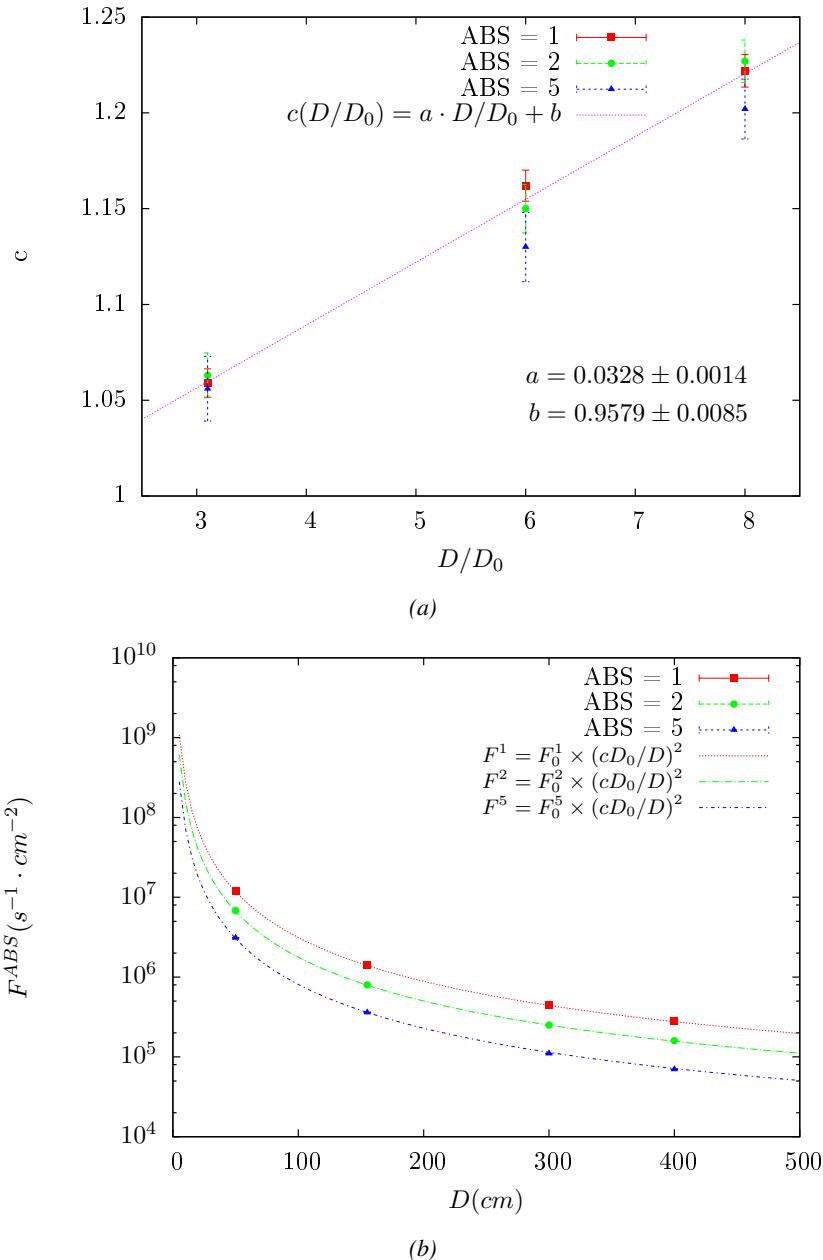


Figure 5.13: Figure 5.13a shows the linear approximation fit done via formulae 5.7 on data from table 5.2. Figure 5.13b shows a comparison of this model with the simulated flux using a and b given in figure 5.13a in formulae 5.4 and the reference value $D_0 = 50\text{cm}$ and the associated flux for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

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

694 tests where done in between the 20th and the 31st of August 2014, i.e. at a time $t = (17.47 \pm 0.02)$ y
 695 resulting in an attenuation of the activity from 740 GBq in 1997 to 494 GBq in 2014. All the needed
 696 information to extrapolate the flux through our detector in 2014 has now been assembled, leading
 697 to the Table 5.3. It is interesting to note that for a common RPC sensitivity to γ of $2 \cdot 10^{-3}$, the
 698 order of magnitude of the estimated hit rate per unit area is of the order of the kHz for the fully
 699 opened source. Moreover, taking profit of the two working absorbers, it will be possible to scan
 700 background rates at 0 Hz, ~ 300 Hz as well as ~ 600 Hz. Without source, a good estimate of the
 701 intrinsic performance will be available. Then at 300 Hz, the goal will be to show that the detectors
 702 fulfill the performance certification of CMS RPCs. Then a first idea of the performance of the
 703 detectors at higher background will be provided with absorption factors 2 (~ 600 Hz) and 1 (no
 704 absorption). *[Here I will also put a reference to the plot showing the estimated background rate at
 705 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 $D_0^{1997} = 206$ 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 ± 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 ± 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 ± 9

Table 5.3: The data at D_0 in 1997 is taken from [6]. In a second step, using Equations 5.8 and 5.9, the flux at D can be estimated in 1997. Then, taking into account the attenuation of the source activity, the flux at D can be estimated at the time of the tests in GIF in 2014. Finally, assuming a sensitivity of the RPC to γ $s = 2 \cdot 10^{-3}$, an estimation of the hit rate per unit area is obtained.

⁷⁰⁶ **5.2.4.2 Dose measurements**

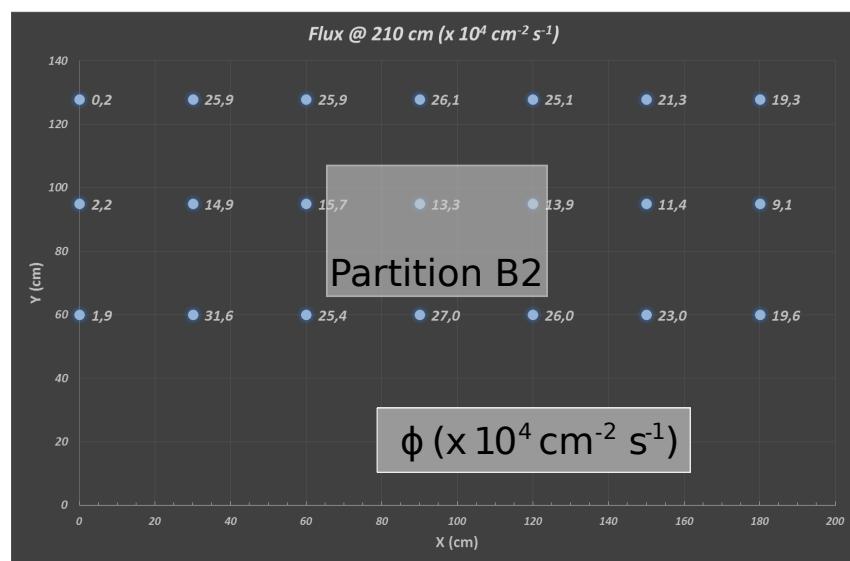


Figure 5.14: Dose measurements has been done in a plane corresponding to the tents front side. This plan is 1900 mm away from the source. As explained in the first chapter, a lens-shaped lead filter provides a uniform photon flux in the vertical plan orthogonal to the beam direction. If the second line of measured fluxes is not taken into account because of lower values due to experimental equipments in the way between the source and the tent, the uniformity of the flux is well showed by the results.

⁷⁰⁷ **5.2.5 Results and discussions**

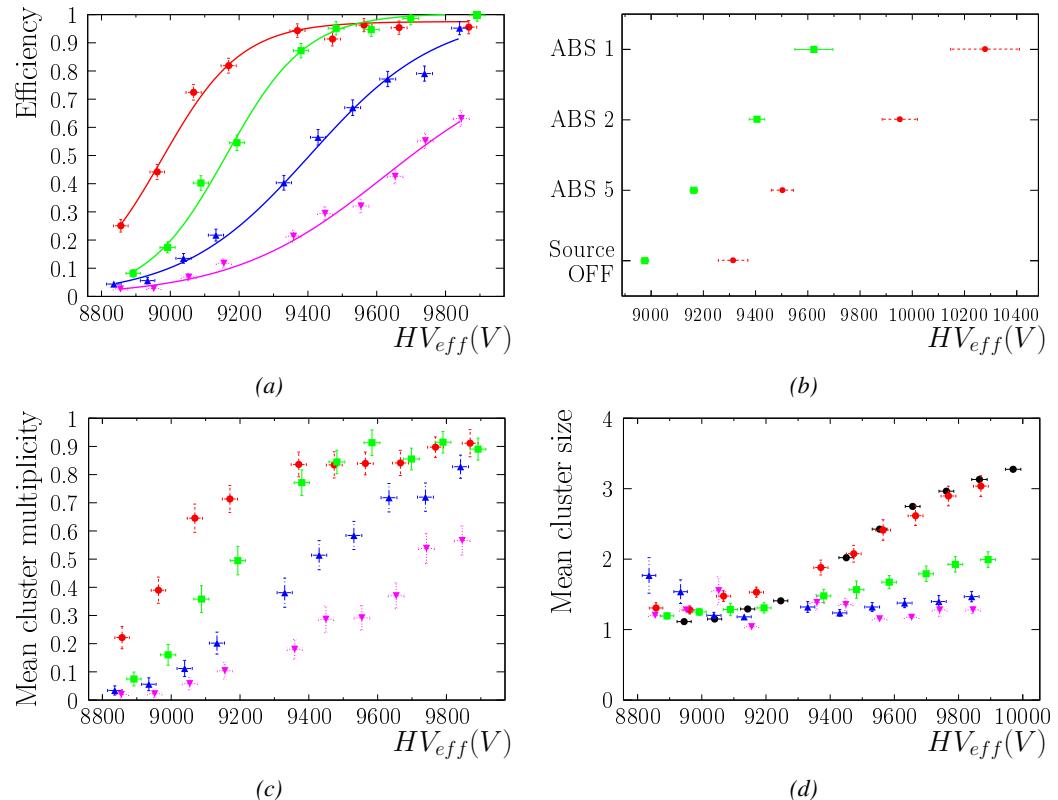


Figure 5.15

708 5.3 Longevity tests at GIF++

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

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

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

730 Figures 5.16 and 5.17 give us for different test beam periods, and thus for increasing integrated
731 charge through time, a comparison of the maximum efficiency, obtained using a sigmoid-like func-
732 tion, and of the working point of both irradiated and non irradiated chambers [8]. No aging is yet to
733 see from this data, the shifts in γ rate per unit area in between irradiated and non irradiated detec-
734 tors and RE2 and RE4 types being easily explained by a difference of sensitivity due to the various
735 Bakelite resistivities of the HPL electrodes used for the electrode production.

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

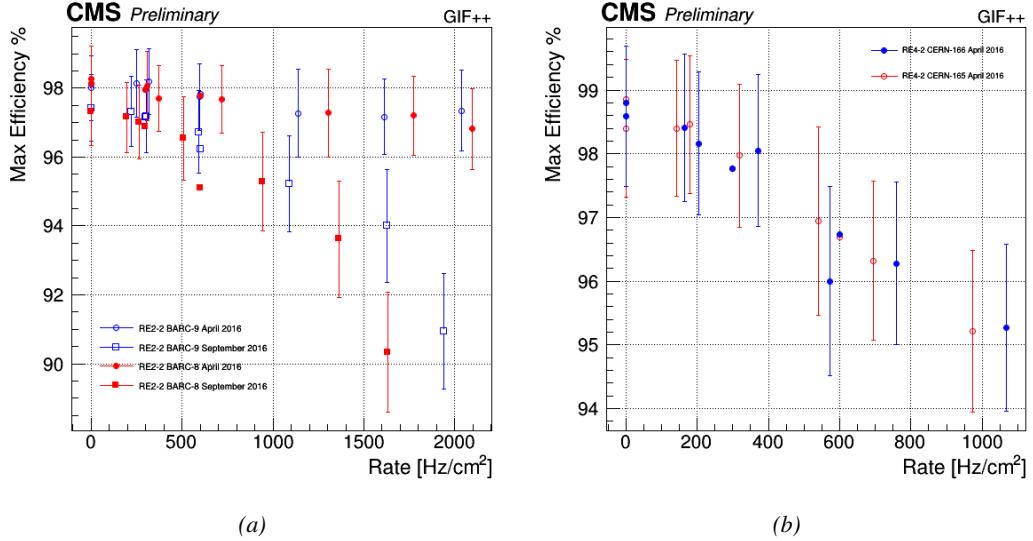


Figure 5.16: Evolution of the maximum efficiency for RE2 (5.16a) and RE4 (5.16b) chambers with increasing extrapolated γ rate per unit area at working point. Both irradiated (blue) and non irradiated (red) chambers are shown.

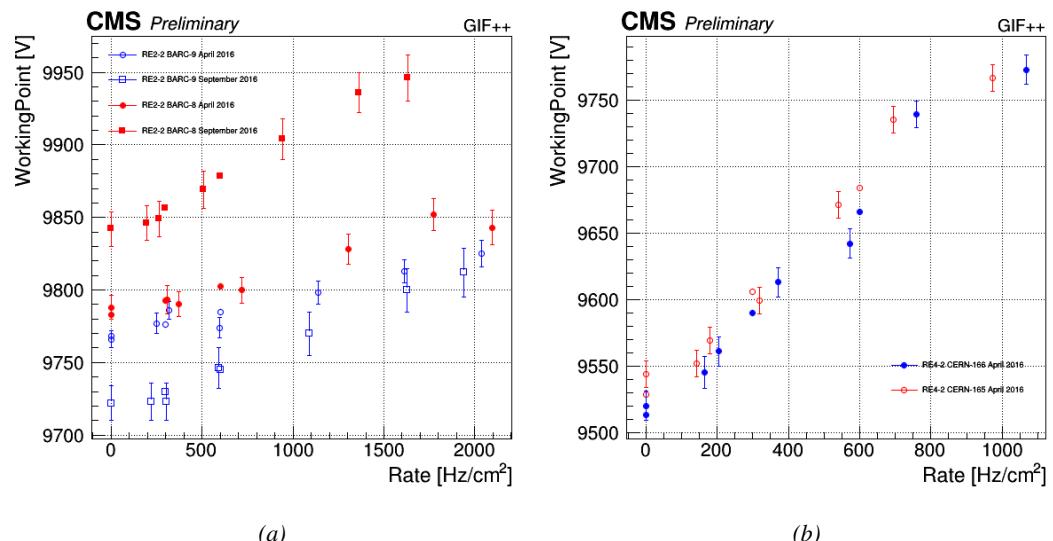


Figure 5.17: Evolution of the working point for RE2 (5.17a) and RE4 (5.17b) with increasing extrapolated γ rate per unit area at working point. Both irradiated (blue) and non irradiated (red) chambers are shown.

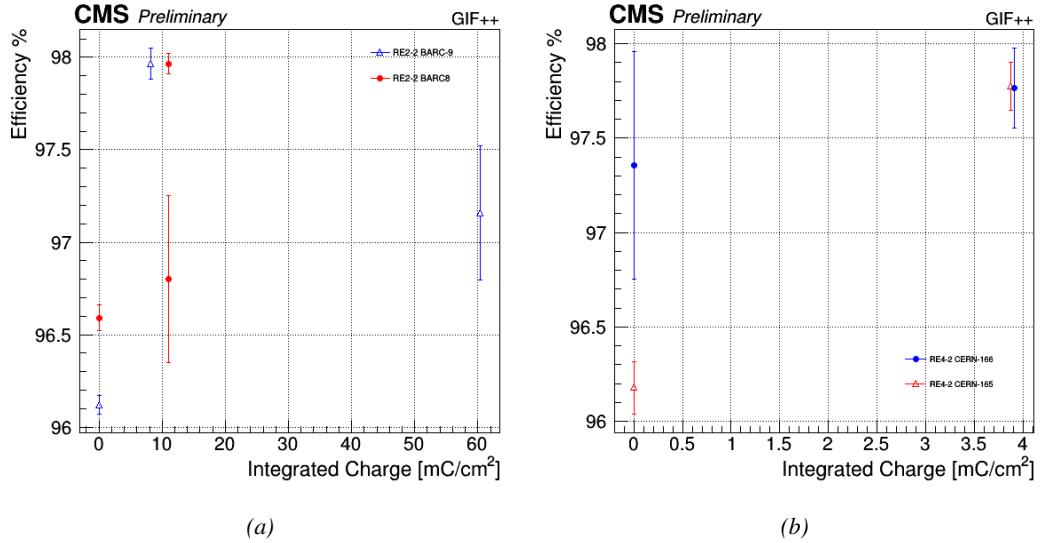


Figure 5.18: Evolution of the maximum efficiency at HL-LHC conditions, i.e. a background hit rate per unit area of $300 \text{ Hz}/\text{cm}^2$, with increasing integrated charge for RE2 (5.18a) and RE4 (5.18b) detectors. Both irradiated (blue) and non irradiated (red) chambers are shown. The integrated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

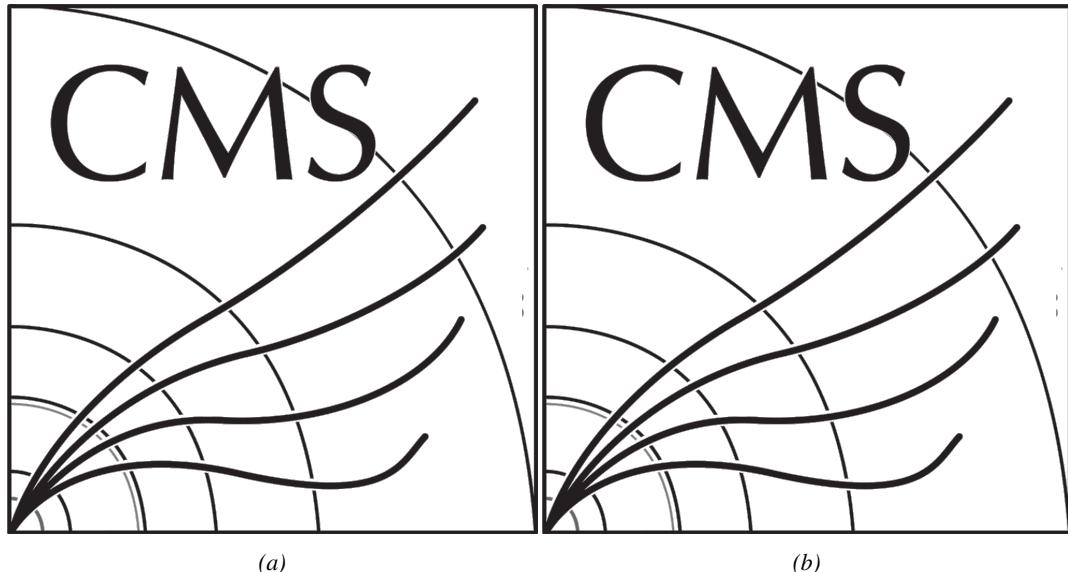


Figure 5.19: Comparison of the efficiency sigmoid before (triangles) and after (circles) irradiation for RE2 (5.19a) and RE4 (5.19b) detectors. Both irradiated (blue) and non irradiated (red) chambers are shown.

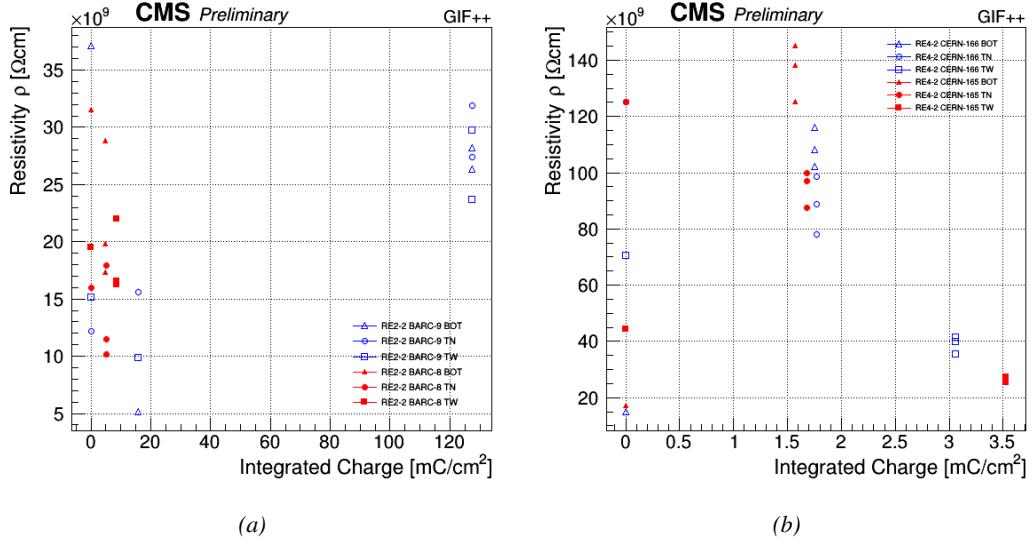


Figure 5.20: Evolution of the Bakelite resistivity for RE2 (5.20a) and RE4 (5.20b) detectors. Both irradiated (blue) and non-irradiated (red) chambers are shown.

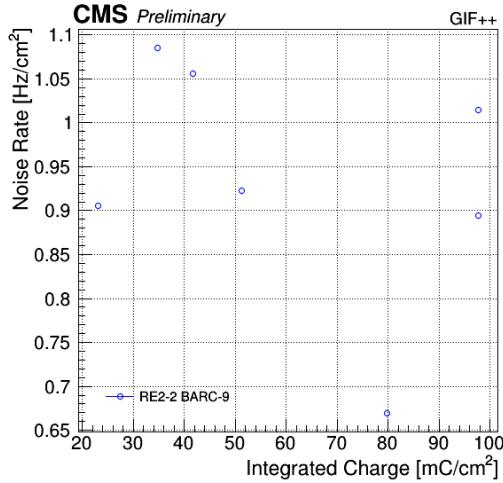


Figure 5.21: Evolution of the noise rate per unit area for the irradiated chamber RE2-2-BARC-9 only.

5.3.1 Description of the Data Acquisition

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

755 moved outside the beam line and is placed in a distance of 5.2 m to the source, which irradiates the
 756 bunker using an attenuation filter of 2.2 which corresponds to a fluence of 10^7 gamma/cm^2 .

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

765 Two different types of tests are conducted on RPCs via the DAQ. Indeed, the performance of the
 766 detectors is measured periodically during dedicated test beam periods using the H4 muon beam. In
 767 between these test beam periods, when the beam is not available, the chambers are irradiated by the
 768 ^{137}Cs in order to accumulate deposited charge and the gamma background is measured.

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

775 In the case of performance test, the trigger signal used for data acquisition is generated by the
 776 coincidence of three scintillators. A first one is placed upstream outside of the bunker, a second one
 777 is placed downstream outside of the bunker, while a third one is placed in front of the trolley, close by
 778 the chambers. Every time a trigger is sent to the TDCs, i.e. every time a muon is detected, knowing
 779 the time delay in between the trigger and the RPC signals, signals located in the right time window
 780 are extracted from the buffers and saved for later analysis. Signals are taken in a time window of
 781 400 ns centered on the muon peak (here we could show a time spectrum). On the other hand, in the
 782 case of background rate measurement, the trigger signal needs to be "random" not to measure muons
 783 but to look at gamma background. A trigger pulse is continuously generated at a rate of 300 Hz using
 784 a dual timer. To integrate an as great as possible time, all signals contained within a time window of
 785 10us prior to the random trigger signal are extracted form the buffers and saved for further analysis
 786 (here another time spectrum to illustrate could be useful, maybe even place both spectrum together
 787 as a single Figure).

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

793 5.3.2 RPC current, environmental and operation parameter monitoring

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

$$796 HV_{eff} = HV_{app} \times \left(0.2 + 0.8 \cdot \frac{P_0}{P} \times \frac{T}{T_0} \right) \quad (5.10)$$

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

797 **5.3.3 Measurement procedure**

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

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

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

801 **5.3.4 Longevity studies results**

6

802

803

Investigation on high rate RPCs

804 **6.1 Rate limitations and ageing of RPCs**

805 **6.1.1 Low resistivity electrodes**

806 **6.1.2 Low noise front-end electronics**

807 **6.2 Construction of prototypes**

808 **6.3 Results and discussions**

7

809

810

Conclusions and outlooks

⁸¹¹ **7.1 Conclusions**

⁸¹² **7.2 Outlooks**

References

813

- 814 [1] CERN. Geneva. LHC Experiments Committee. *The CMS muon project : Technical Design*
815 *Report*. Tech. rep. CERN-LHCC-97-032. CMS Collaboration, 1997.
- 816 [2] CERN. Geneva. LHC Experiments Committee. *Technical Proposal for the Phase-II Upgrade*
817 *of the CMS Detector*. Tech. rep. CERN-LHCC-2015-010. CMS Collaboration, 2015.
- 818 [3] CERN. Geneva. LHC Experiments Committee. *CMS, the Compact Muon Solenoid : technical*
819 *proposal*. Tech. rep. CERN-LHCC-94-38. CMS Collaboration, 1994.
- 820 [4] M. Abbrescia et al. “Study of long-term performance of CMS RPC under irradiation at the
821 CERN GIF”. In: *NIMA* 533 (2004), pp. 102–106.
- 822 [5] H.C. Kim et al. “Quantitative aging study with intense irradiation tests for the CMS forward
823 RPCs”. In: *NIMA* 602 (2009), pp. 771–774.
- 824 [6] S. Agosteo et al. “A facility for the test of large-area muon chambers at high rates”. In: *NIMA*
825 452 (2000), pp. 94–104.
- 826 [7] PoS, ed. *CERN GIF ++ : A new irradiation facility to test large-area particle detectors for*
827 *the high-luminosity LHC program*. Vol. TIPP2014. 2014, pp. 102–109.
- 828 [8] M. Abbrescia et al. “Cosmic ray tests of double-gap resistive plate chambers for the CMS
829 experiment”. In: *NIMA* 550 (2005), pp. 116–126.
- 830 [9] A. Fagot. *GIF++ DAQ v4.0*. 2017. URL: https://github.com/afagot/GIF_DAQ.
- 831 [10] CAEN. *Mod. V1190-VX1190 A/B, 128/64 Ch Multihit TDC*. 14th ed. 2016.
- 832 [11] CAEN. *Mod. V1718 VME USB Bridge*. 9th ed. 2009.
- 833 [12] W-Ie-Ne-R. *VME 6021-23 VXI*. 5th ed. 2016.
- 834 [13] Wikipedia. *INI file*. 2017. URL: https://en.wikipedia.org/wiki/INI_file.
- 835 [14] S. Carrillo A. Fagot. *GIF++ Offline Analysis v6*. 2017. URL: https://github.com/afagot/GIF_OfflineAnalysis.

A

837

838

839

A data acquisition software for CAEN VME TDCs

840 Certifying detectors in the perspective of HL-LHC required to develop tools for the GIF++ ex-
841 periment. Among them was the C++ Data Acquisition (DAQ) software that allows to make the com-
842 munications in between a computer and TDC modules in order to retrieve the RPC data [9]. In this
843 appendix, details about this software, as of how the software was written, how it functions and how
844 it can be exported to another similar setup, will be given.

845 A.1 GIF++ DAQ file tree

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

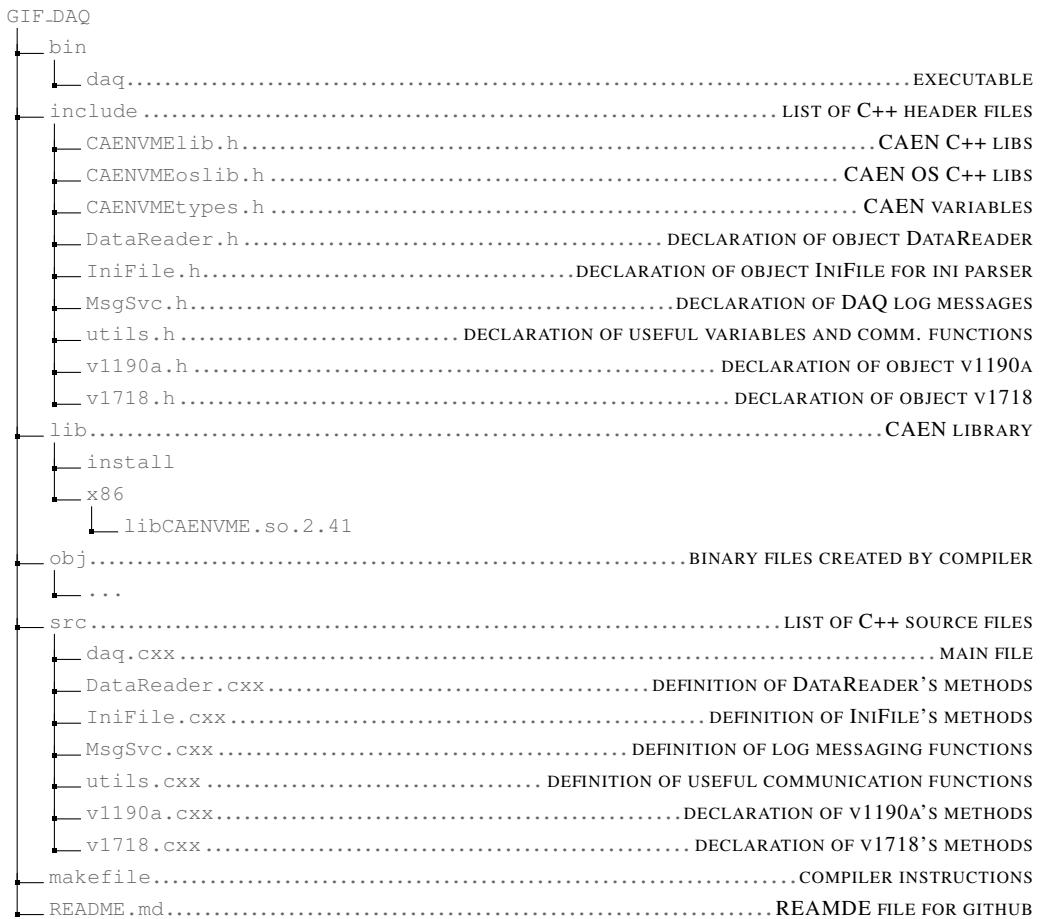
- 848 • CAEN USB Driver, to mount the VME hardware,
849 • CAEN VME Library, to communicate with the VME hardware, and
850 • ROOT, to organize the collected data into a TTree.

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

853
854 make

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

858



859 A.2 Usage of the DAQ

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

866
 867 bin/daq /path/to/the/log/file/in/the/output/data/folder

868 where /path/to/the/log/file/in/the/output/data/folder is the only argument required. This
 869 log file is important for the webDCS as this file contains all the content of the communication of the
 870 webDCS and the different systems monitored by the webDCS. Its content is constantly displayed
 871 during data taking for the users to be able to follow the operations. The communication messages
 872 are normally sent to the webDCS log file via the functions declared in file MsgSvc.h, typically
 873 MSG_INFO(string message).

874

875 A.3 Description of the readout setup

876 The CMS RPC setup at GIF++ counts 5 V1190A Time-to-Digital Converter (TDC) manufactured
 877 by CAEN [10]. V1190A are VME units accepting 128 independent Multi-Hit/Multi-Event TDC
 878 channels whose signals are treated by 4 100 ps high performance TDC chips developed by CERN
 879 / ECP-MIC Division. The communication between the computer and the TDCs to transfer data is
 880 done via a V1718 VME master module also manufactured by CAEN and operated from a USB
 881 port [11]. These VME modules are all hosted into a 6U VME 6021 powered crate manufactured by
 882 W-Ie-Ne-R than can accommodate up to 21 VME bus cards [12]. These 3 components of the DAQ
 883 setup are shown in Figure A.1.

884

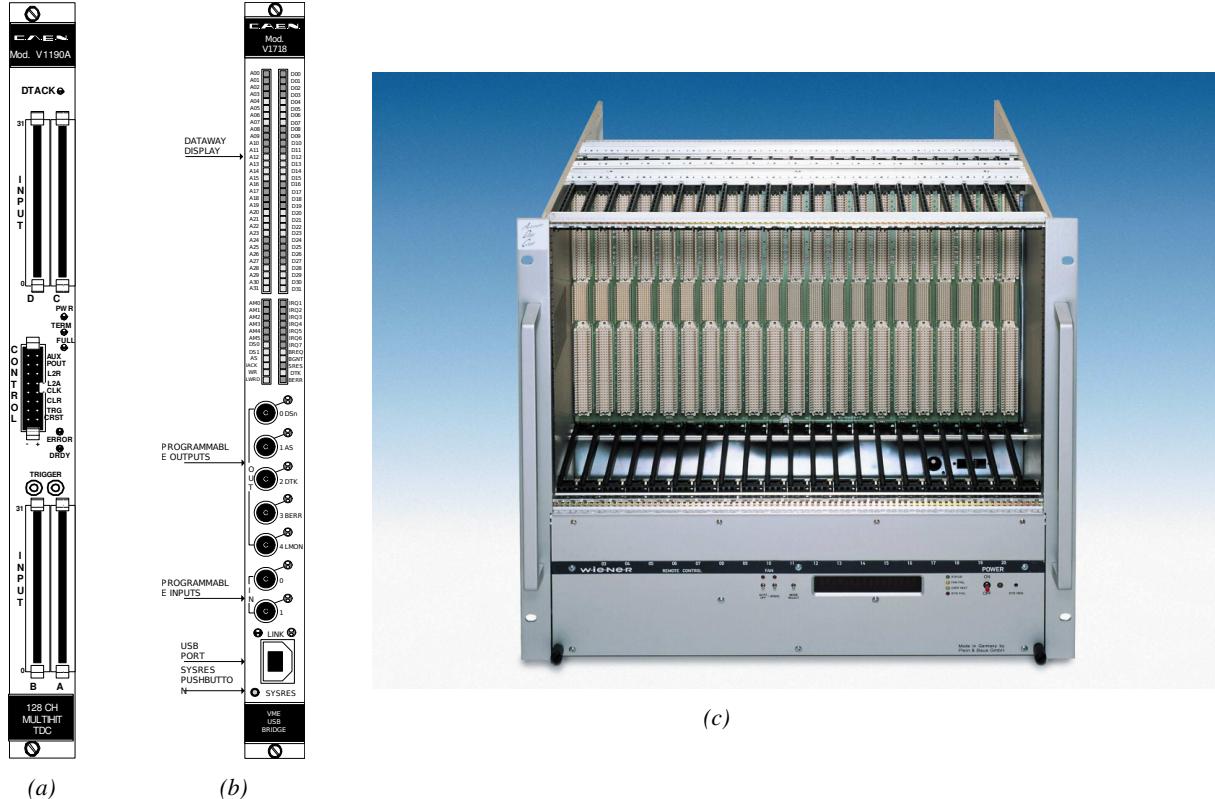


Figure A.1: (A.1a) View of the front panel of a V1190A TDC module [10]. (A.1b) View of the front panel of a V1718 Bridge module [11]. (A.1c) View of the front panel of a 6U 6021 VME crate [12].

885

A.4 Data read-out

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

888 that comes as an input of the DAQ software.

889

890 A.4.1 V1190A TDCs

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

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

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

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

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

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

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

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

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

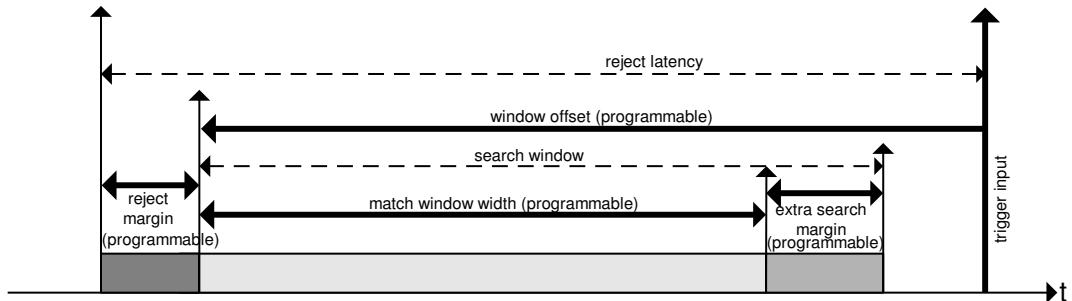


Figure A.2: Module V1190A Trigger Matching Mode timing diagram [10].

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

- 912 • **1:** the match window is entirely contained after the trigger signal,

- 913 • **2:** the match window overlaps the trigger signal, or

- 914 • **3:** the match window is entirely contained before the trigger signal as displayed on Figure A.2.

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

```

937
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);
};

938

```

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

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

944

945 A.4.2 DataReader

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

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

- 954 • a **global header** providing information of the event number since the beginning of the data
 acquisition,
- 956 • a **TDC header**,
- 957 • the **TDC data** (*if any*), 1 for each hit recorded during the event, providing the channel and the
 time stamp associated to the hit,
- 959 • a **TDC error** providing error flags,
- 960 • a **TDC trailer**,
- 961 • a **global trigger time tag** that provides the absolute trigger time relatively to the last reset,
 and
- 963 • a **global trailer** providing the total word count in the event.

964 As previously described in Section 4.4.3, CMS RPC FEEs provide us with 100 ns long LVDS
 965 output signals that are injected into the TDCs' input. Any avalanche signal that gives a signal above
 966 the FEEs threshold is thus recorded by the TDCs as a hit within the match window. Each hit is
 967 assigned to a specific TDC channel with a time stamp, with a precision of 100 ps. The reference
 968 time, $t_0 = 0$, is provided by the beginning of the match window. Thus for each trigger, coming from
 969 a scintillator coïncidence or the pulse generator, a list of hits is stored into the TDCs' buffers and
 970 will then be transferred into a ROOT Tree.

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

980 Due to irradiation, an event in GIF++ can count up to 300 words per TDC. A limit of 4096 words
 981 (12 bits) has been set to generate IRQ which represent from 14 to almost 700 events depending on
 982 the average of hits collected per event. Then the block size has been set to 100 events with enabled
 983 bus errors. When an AFL is reached for one of the TDCs, the VME bridge stops the acquisition by
 985 sending a BUSY signal.

986

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

```
988
989 struct RAWData{
990     vector<int>           *EventList;
991     vector<int>           *NHitsList;
992     vector<int>           *QFlagList;
993     vector<vector<int>>   *Channellist;
994     vector<vector<float>>  *TimeStampList;
995 };

```

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

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

```
996
997 class DataReader
998 {
999     private:
1000     bool      StopFlag;
1001     IniFile  *iniFile;
1002     Data32    MaxTriggers;
1003     v1718    *VME;
1004     int       nTDCs;
1005     v1190a   *TDCs;
1006     RAWData  TDCData;

1007     public:
1008     DataReader();
1009     virtual ~DataReader();
1010     void      SetIniFile(string inifilename);
1011     void      SetMaxTriggers();
1012     Data32    GetMaxTriggers();
1013     void      SetVME();
1014     void      SetTDC();
1015     int       GetQFlag(Uint it);
1016     void      Init(string inifilename);
1017     void      FlushBuffer();
1018     void      Update();
1019     string   GetFileName();
1020     void      WriteRunRegistry(string filename);
1021     void      Run();
1022 };


```

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

1000 Each event is transferred from `TDCData` and saved into branches of a ROOT `TTree` as 3 integers
1001 that represent the event ID (`EventCount`), the number of hits read from the TDCs (`nHits`), and the
1002 quality flag that provides information for any problem in the data transfer (`qflag`), and 2 lists of
1003 `nHits` elements containing the fired TDC channels (`TDCCh`) and their respective time stamps (`TDCTS`),
1004 as presented in Source Code A.4. The ROOT file file is named using information contained into
1005 the configuration file, presented in section A.5.2. The needed information is extracted using method
1006 `DataReader::GetFileName()` and allow to build the output filename format `ScanXXXXXX_HVX_DAQ.root`

1006 where ScanXXXXXX is a 6 digit number representing the scan number into GIFT++ database and HVX
 1007 the HV step within the scan that can be more than a single digit. An example of ROOT data file is
 1008 provided with Figure A.3.

```
1009
1010 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);

1011 //...
//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();
}
```

1011 *Source Code A.4: Highlight of the data transfer and organisation within DataReader::Run() after the data has been collected into TDCData.*

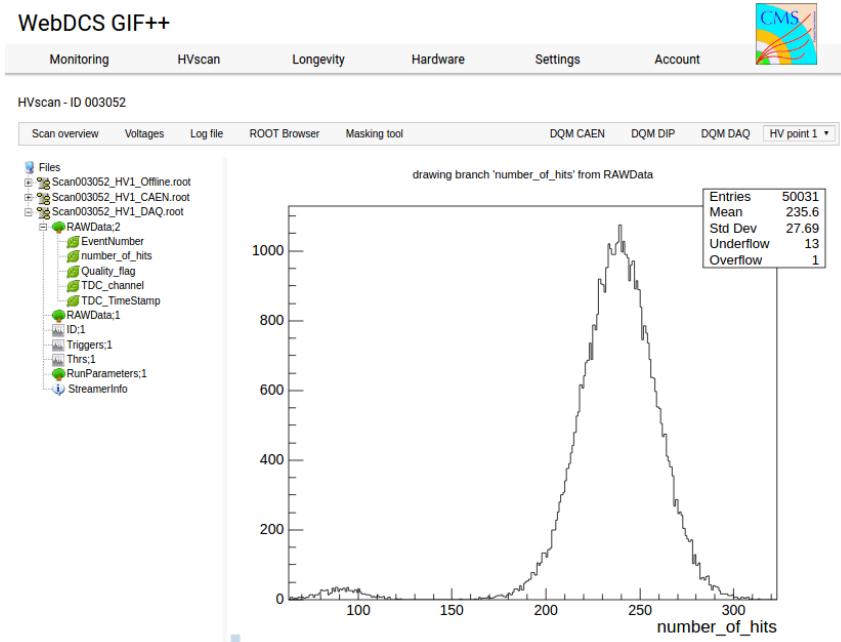


Figure A.3: Structure of the ROOT output file generated by the DAQ. The 5 branches (EventNumber, number_of_hits, Quality_flag, TDC_channel and TDC_TimeStamp) are visible on the left panel of the ROOT browser. On the right panel is visible the histogram corresponding to the variable nHits. In this specific 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.

1012 A.5 Communications

1013 To ensure data readout and dialog in between the machine and the TDCs or in between the webDCS
 1014 and the DAQ, different communication solutions were used. First of all, it is important to have a
 1015 module to allow the communication in between the TDCs and the computer from which the DAQ
 1016 operates. When this communication is effective, shifters using the webDCS to control data taking
 1017 can thus send instructions to the DAQ.
 1018

1019 A.5.1 V1718 USB Bridge

1020 In the previous section, the data transfer has been discussed. The importance of the `v1718` object
 1021 (Source Code A.5), used as private member of `DataReader`, was not explicated. VME master
 1022 modules are used for communication purposes as they host the USB port that connects the pow-
 1023 ered crate buffer to the computer where the DAQ is installed. From the source code point of view,
 1024 this object is used to control the communication status, by reading the returned error codes with
 1025 `v1718::CheckStatus()`, or to check for IRQs coming from the TDCs through `v1718::CheckIRQ()`.
 1026 Finally, to ensure that triggers are blocked at the hardware level, a NIM pulse is sent out of one of the
 1027 5 programmable outputs (`v1718::SendBUSY()`) to the VETO of the coincidence module where the
 1028 trigger signals originate from. As long as this signal is ON, no trigger can reach the TDCs anymore.

```

1029
class v1718{
    private:
        int             Handle;
        Data32          Data;           // Data
        CVIRQLevels     Level;         // Interrupt level
        CVAddressModifier AM;          // Addressing Mode
        CVDataWidth     dataSize;      // Data Format
        Data32          BaseAddress;   // Base Address

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

1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040

```

Source Code A.5: Description of C++ object v1718.

A.5.2 Configuration file

The DAQ software takes as input a configuration file written using INI standard [13]. This file is partly filled with the information provided by the shifters when starting data acquisition using the webDCS, as shown by Figure A.4. This information is written in section [**General**] and will later be stored in the ROOT file that contains the DAQ data as can be seen from Figure A.3. Indeed, another TTree called `RunParameters` as well as the 2 histograms `ID`, containing the scan number, start and stop time stamps, and `Triggers`, containing the number of triggers requested by the shifter, are available in the data files. Moreover, `ScanID` and `HV` are then used to construct the file name thanks to the method `DataReader::GetFileName()`.

WebDCS GIF++

Monitoring HVscan Longevity Hardware Settings Account

DAQ High Voltage Scan

Type scan: Rate Scan Comments:

Source configuration: Source OFF U 333 D 333 HV after scan: Turn off

Beam configuration: Beam OFF

Waiting time: 1 (min)

Trigger mode: External Internal Random

Minimal measure time: 10 (min)

Chamber	RE2-2-NPD-BARC-8	RE2-2-CERN-105	RE2-2-NPD-BARC-9	RE4-2-CERN-105	RE4-2-KODEL-1-4	Max triggers
HV _{eff} 1	8600	8500	8600	8500	6500	<input type="text"/>
HV _{eff} 2	8700	8600	8700	8600	6600	<input type="text"/>
HV _{eff} 3	8800	8700	8800	8700	6700	<input type="text"/>
HV _{eff} 4	8900	8800	8900	8800	6800	<input type="text"/>
HV _{eff} 5	9000	8900	9000	8900	6900	<input type="text"/>
HV _{eff} 6	9100	9000	9100	9000	7000	<input type="text"/>
HV _{eff} 7	9200	9100	9200	9100	7100	<input type="text"/>
HV _{eff} 8	9300	9200	9300	9200	7200	<input type="text"/>
HV _{eff} 9	9400	9300	9400	9300	7300	<input type="text"/>
HV _{eff} 10	9500	9400	9500	9400	7400	<input type="text"/>

Start HV scan

Figure A.4: 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.

1041 The rest of the information is written beforehand in the configuration file template, as explicated
 1042 in Source Code A.6, and contains the hardware addresses to the different VME modules in the
 1043 setup as well as settings for the TDCs. As the TDC settings available in the configuration file are not
 1044 supposed to be modified, an improvement would be to remove them from the configuration file and
 1045 to hardcode them inside of the DAQ code itself or to place them into a different INI file that would
 1046 host only the TDC settings to lower the probability for a bad manipulation of the configuration file
 1047 that can be modified from one of webDCS' menus.

1048

```
[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
```

1049

Source Code A.6: 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.

1051 In order to retrieve the information of the configuration file, the object `IniFile` has been developed
 1052 to provide an INI parser, presented in Source Code A.7. It contains private methods returning a
 1053 boolean to check the type of line written in the file, whether a comment, a group header or a key line
 1054 (`IniFile::CheckIfComment()`, `IniFile::CheckIfGroup()` and `IniFile::CheckIfToken()`). The
 1055 key may sometimes be referred to as *token* in the source code. Moreover, the private element
 1056 `FileData` is a map of `const string` to `string` that allows to store the data contained inside the
 1057 configuration file via the public method `IniFile::GetFileData()` following the formatting (see
 1058 method `IniFile::Read()`):

```
1059
  1060     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;
```

1061 More methods have been written to translate the different keys into the right variable format
 1062 when used by the DAQ. For example, to get a `float` value out of the configuration file data, knowing
 1063 the group and the key needed, the method `IniFile::floatType()` can be used. It takes 3 arguments
 1064 being the group name and key name (both `string`), and a default `float` value used as exception in
 1065 the case the expected combination of group and key cannot be found in the configuration file. This
 1066 default value is then used and the DAQ continues on working after sending an alert in the log file for
 1067 further debugging.

```

1068 typedef map< const string, string > IniFileData;
1069
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();
};


```

1070 *Source Code A.7: Description of C++ object `IniFile` used as a parser for INI file format.*

1071 A.5.3 WebDCS/DAQ intercommunication

1072 When shifters send instructions to the DAQ via the configuration file, it is the webDCS itself that
1073 gives the start command to the DAQ and then the 2 softwares use inter-process communication
1074 through file to synchronise themselves. This communication file is represented by the variable **const**
1075 string __runstatuspath.

1076 On one side, the webDCS sends commands or status that are readout by the DAQ:

- 1077 • INIT, status sent when launching a scan and read via function `CtrlRunStatus(...)`,
- 1078 • START, command to start data taking and read via function `CheckSTART()`,
- 1079 • STOP, command to stop data taking at the end of the scan and read via function `CheckSTOP()`,
1080 and
- 1081 • KILL, command to kill data taking sent by user and read via function `CheckKILL()`

1082 and on the other, the DAQ sends status that are controled by the webDCS:

- 1083 ● `DAQ_RDY`, sent with `SendDAQReady()` to signify that the DAQ is ready to receive commands
1084 from the webDCS,
- 1085 ● `RUNNING`, sent with `SendDAQRunning()` to signify that the DAQ is taking data,
- 1086 ● `DAQ_ERR`, sent with `SendDAQError()` to signify that the DAQ didn't receive the expected com-
1087 mand from the webDCS or that the launch command didn't have the right number of argu-
1088 ments,
- 1089 ● `RD_ERR`, sent when the DAQ wasn't able to read the communication file, and
- 1090 ● `WR_ERR`, sent when the DAQ wasn't able to write into the communication file.

1091 **A.5.4 Example of inter-process communication cycle**

1092 Under normal conditions, the webDCS and the DAQ processes exchange commands and status via
1093 the file hosted at the address `__runstatuspath`, as explained in subsection A.5.3. An example of
1094 cycle is given in Table A.1. In this example, the steps 3 to 5 are repeated as long as the webDCS tells
1095 the DAQ to take data. A data taking cycle is the equivalent as what is called a *Scan* in GIFT++ jargon,
1096 referring to a set a runs with several HV steps. Each repetition of steps 3 to 5 is then equivalent to a
1097 single *Run*.

1098

1099 At any moment during the data taking, for any reason, the shifter can decide that the data taking
1100 needs to be stopped before it reached the end of the scheduled cycle. Thus at any moment on the
1101 cycle, the content of the inter-process communication file will be changed to `KILL` and the DAQ will
1102 shut down right away. The DAQ checks for `KILL` signals every 5s after the TDCs configuration is
1103 over. So far, the function `CheckKILL()` has been used only inside of the data taking loop of method
1104 `DataReader::Run()` and thus, if the shifter decides to KILL the data taking during the TDC con-
1105 figuration phase or the HV ramping in between 2 HV steps, the DAQ will not be stopped smoothly
1106 and a *force kill* command will be sent to stop the DAQ process that is still awake on the computer.
1107 Improvements can be brought on this part of the software to make sure that the DAQ can safely
1108 shutdown at any moment.

1109

1110 **A.6 Software export**

1111 In section A.2 was discussed the fact that the DAQ as written in its last version is not a standalone
1112 software. It is possible to make it a standalone program that could be adapted to any VME setup
1113 using V1190A and V1718 modules by creating a GUI for the software or by printing the log mes-
1114 sages that are normally printed in the webDCS through the log file, directly into the terminal. This
1115 method was used by the DAQ up to version 3.0 moment where the webDCS was completed. Also, it
1116 is possible to check branches of DAQ v2.X to have example of communication through a terminal.

1117

1118 DAQ v2.X is nonetheless limited in it's possibilities and requires a lot of offline manual interven-
1119 tions from the users. Indeed, there is no communication of the software with the detectors' power
1120 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.

1121 and loop over to take data without any further manual intervention. In v2.X, the data is taken for a
1122 single detector setting and at the end of each run, the softwares asks the user if he intends on taking
1123 more runs. If so, the software invites the user to set the operating voltages accordingly to what is
1124 necessary and to manual update the configuration file in consequence. This working mode can be a
1125 very first approach before an evolution and has been successfully used by colleagues from different
1126 collaborations.

1127

1128 For a more robust operation, it is recommended to develop a GUI or a web application to inter-
1129 face the DAQ. Moreover, to limit the amount of manual interventions, and thus the probability to
1130 make mistakes, it is also recommended to add an extra feature into the DAQ by installing the HV
1131 Wrapper library provided by CAEN of which an example of use in a similar DAQ software devel-
1132 opped by a master student of UGent, and called TinyDAQ, is provided on UGent's github. Then, this
1133 HV Wrapper will help you communicating with and give instructions to a CAEN HV powered crate
1134 and can be added into the DAQ at the same level where the communication with the user was made
1135 in DAQ v2.X. In case you are using another kind of power system for your detectors, it is stringly
1136 adviced to use HV modules or crates that can be remotely controled via a using C++ libraries.

1137

B

1138

1139

Details on the offline analysis package

1140 The data collected in GIF++ thanks to the DAQ described in Appendix A is difficult to interpret by
1141 a human user that doesn't have a clear idea of the raw data architecture of the ROOT data files. In
1142 order to render the data human readable, a C++ offline analysis tool was designed to provide users
1143 with detector by detector histograms that give a clear overview of the parameters monitored during
1144 the data acquisition [14]. In this appendix, details about this software in the context of GIF++, as of
1145 how the software was written and how it functions will be given.

1146 B.1 GIF++ Offline Analysis file tree

1147 GIF++ Offline Analysis source code is fully available on github at https://github.com/afagot/GIF_OfflineAnalysis. The software requires ROOT as non-optionnal dependency
1148 as it takes ROOT files in input and write an output ROOT file containing histograms. To compile the
1149 GIF++ Offline Analysis project is compiled with cmake. To compile, first a build/ directory must
1150 be created to compile from there:

```
1152 mkdir build  
1153 cd build  
1154 cmake ..  
1155 make  
1156 make install
```

1154 To clean the directory and create a new build directory, the bash script cleandir.sh can be used:

```
1155  
1156 ./cleandir.sh
```

1157 The source code tree is provided below along with comments to give an overview of the files' con-
1158 tent. The different objects created for this project (`Infrastructure`, `Trolley`, `RPC`, `Mapping`, `RPCHit`,
1159 `RPCCluster` and `Inifile`) will be described in details in the following sections.

1160

```

GIF_OfflineAnalysis
├── bin
│   └── offlineanalysis ..... EXECUTABLE
├── build..... CMAKE COMPILATION DIRECTORY
└── ...
    ├── include..... LIST OF C++ HEADER FILES
    │   ├── Cluster.h..... DECLARATION OF OBJECT RPCCLUSTER
    │   ├── Current.h..... DECLARATION OF GETCURRENT ANALYSIS MACRO
    │   ├── GIFTrolley.h..... DECLARATION OF OBJECT TROLLEY
    │   ├── Infrastructure.h..... DECLARATION OF OBJECT INFRASTRUCTURE
    │   ├── IniFile.h..... DECLARATION OF OBJECT INI FILE FORINI PARSER
    │   ├── Mapping.h..... DECLARATION OF OBJECT MAPPING
    │   ├── MsgSvc.h..... DECLARATION OF OFFLINE LOG MESSAGES
    │   ├── OfflineAnalysis.h..... DECLARATION OF DATA ANALYSIS MACRO
    │   ├── RPCTracker.h..... DECLARATION OF OBJECT RPC
    │   ├── RPCHit.h..... DECLARATION OF OBJECT RPCHIT
    │   ├── types.h..... DEFINITION OF USEFUL VARIABLE TYPES
    │   └── utils.h..... DECLARATION OF USEFUL FUNCTIONS
    ├── obj..... BINARY FILES CREATED BY COMPILER
    └── ...
        ├── src..... LIST OF C++ SOURCE FILES
        │   ├── Cluster.cc..... DEFINITION OF OBJECT RPCCLUSTER
        │   ├── Current.cc..... DEFINITION OF GETCURRENT ANALYSIS MACRO
        │   ├── GIFTrolley.cc..... DEFINITION OF OBJECT TROLLEY
        │   ├── Infrastructure.cc..... DEFINITION OF OBJECT INFRASTRUCTURE
        │   ├── IniFile.cc..... DEFINITION OF OBJECT INI FILE FORINI PARSER
        │   ├── main.cc..... MAIN FILE
        │   ├── Mapping.cc..... DEFINITION OF OBJECT MAPPING
        │   ├── MsgSvc.cc..... DEFINITION OF OFFLINE LOG MESSAGES
        │   ├── OfflineAnalysis.cc..... DEFINITION OF DATA ANALYSIS MACRO
        │   ├── RPCTracker.cc..... DEFINITION OF OBJECT RPC
        │   ├── RPCHit.cc..... DEFINITION OF OBJECT RPCHIT
        │   └── utils.cc..... DEFINITION OF USEFUL FUNCTIONS
        ├── cleandir.sh..... BASH SCRIPT TO CLEAN BUILD DIRECTORY
        ├── CMakeLists.txt..... SET OF INSTRUCTIONS FOR CMAKE
        ├── config.h.in..... DEFINITION OF VERSION NUMBER
        └── README.md..... README FILE FOR GITHUB

```

1161

B.2 Usage of the Offline Analysis

1162

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:

1164

1165

```
Scan00XXXX_HVY
```

1166

1167

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

1168 of the GIF++ webDCS. Usually, the offline analysis tool is automatically called by the webDCS at
 1169 the end of data taking or by a user from the webDCS panel if an update of the tool was brought.
 1170 Nonetheless, an expert can locally launch the analysis for tests on the GIF++ computer, or a user can
 1171 get the code on its local machine from github and download data from the webDCS for its own anal-
 1172 ysis. To launch the code, the following command can be used from the `GIF_OfflineAnalysis` folder:

1173
 1174 `bin/offlineanalysis /path/to/Scan00XXXX_HVY`

1175 where, `/path/to/Scan00XXXX_HVY` refers to the local data files. Then, the offline tool will by itself
 1176 take care of finding all available ROOT data files present in the folder, as listed below:

- 1177
 - 1178 ● `Scan00XXXX_HVY_DAQ.root` containing the TDC data as described in Appendix ?? (events, hit
 and timestamp lists), and
 - 1179 ● `Scan00XXXX_HVY_CAEN.root` containing the CAEN mainframe data recorded by the monitor-
 ing tool webDCS during data taking (HVs and currents of every HV channels). This file is
 created independently of the DAQ.

1182 **B.2.1 Output of the offline tool**

1183 **B.2.1.1 ROOT file**

1184 The analysis gives in output ROOT datafiles that are saved into the data folder and called using the
 1185 naming convention `Scan00XXXX_HVY_Offline.root`. Inside those, a list of `TH1` histograms can be
 1186 found. Its size will vary as a function of the number of detectors in the setup as each set of histograms
 1187 is produced detector by detector. For each partition of each chamber, can be found:

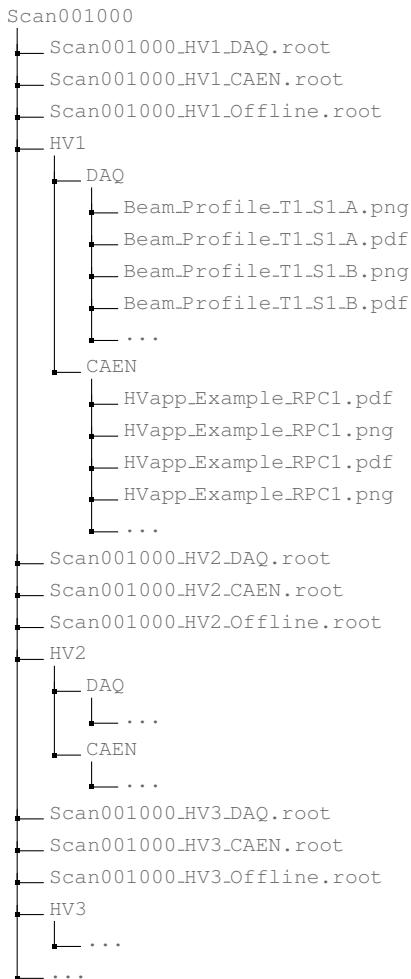
- 1188
 - 1189 ● `Time_Profile_Tt_Sc_p` shows the time profile of all recorded events (number of events per
 time bin),
 - 1190 ● `Hit_Profile_Tt_Sc_p` shows the hit profile of all recorded events (number of events per chan-
 nel),
 - 1192 ● `Hit_Multiplicity_Tt_Sc_p` shows the hit multiplicity (number of hits per event) of all recorded
 events (number of occurrences per multiplicity bin),
 - 1194 ● `Strip_Mean_Noise_Tt_Sc_p` shows noise/gamma rate per unit area for each strip in a se-
 lected time range. After filters are applied on `Time_Profile_Tt_Sc_p`, the filtered version
 of `Hit_Profile_Tt_Sc_p` is normalised to the total integrated time and active detection area
 of a single channel,
 - 1198 ● `Strip_Activity_Tt_Sc_p` shows noise/gamma activity for each strip (normalised version of
 previous histogram - strip activity = strip rate / average partition rate),
 - 1200 ● `Strip_Homogeneity_Tt_Sc_p` shows the *homogeneity* of a given partition ($\text{homogeneity} = \exp(-\text{strip rates standard deviation(strip rates in partition/average partition rate)})$),
 - 1202 ● `mask_Strip_Mean_Noise_Tt_Sc_p` shows noise/gamma rate per unit area for each masked
 strip in a selected time range. Offline, the user can control the noise/gamma rate and decide to
 mask the strips that are judged to be noisy or dead. This is done via the *Masking Tool* provided
 by the webDCS,

- 1206 ● `mask_Strip_Activity_Tt_Sc_p` shows noise/gamma activity per unit area for each masked
1207 strip with respect to the average rate of active strips,
- 1208 ● `NoiseCSize_H_Tt_Sc_p` shows noise/gamma cluster size, a cluster being constructed out of
1209 adjacent strips giving a signal at the *same time* (hits within a time window of 25 ns),
- 1210 ● `NoiseCMult_H_Tt_Sc_p` shows noise/gamma cluster multiplicity (number of reconstructed
1211 clusters per event),
- 1212 ● `Chip_Mean_Noise_Tt_Sc_p` shows the same information than `Strip_Mean_Noise_Tt_Scp` us-
1213 ing a different binning (1 chip corresponds to 8 strips),
- 1214 ● `Chip_Activity_Tt_Sc_p` shows the same information than `Strip_Activity_Tt_Scp` using
1215 chip binning,
- 1216 ● `Chip_Homogeneity_Tt_Sc_p` shows the homogeneity of a given partition using chip binning,
- 1217 ● `Beam_Profile_Tt_Sc_p` shows the estimated beam profile when taking efficiency scan. This
1218 is obtained by filtering `Time_Profile_Tt_Sc_p` to only consider the muon peak where the
1219 noise/gamma background has been subtracted. The resulting hit profile corresponds to the
1220 beam profile on the detector channels,
- 1221 ● `L0_Efficiency_Tt_Sc_p` shows the level 0 efficiency that was estimated **without** muon track-
1222 ing,
- 1223 ● `MuonCSize_H_Tt_Sc_p` shows the level 0 muon cluster size that was estimated **without** muon
1224 tracking, and
- 1225 ● `MuonCMult_H_Tt_Sc_p` shows the level 0 muon cluster multiplicity that was estimated **without**
1226 muon tracking.

1227 In the histogram labels, t stands for the trolley number (1 or 3), c for the chamber slot label in
1228 trolley t and p for the partition label (A, B, C or D depending on the chamber layout) as explained
1229 in Chapter 5.3.

1230 In the context of GIF++, an extra script called by the webDCS is called to extract the histograms
1231 from the ROOT files. The histograms are then stored in PNG and PDF formats into the correspond-
1232 ing folder (a single folder per HV step, so per ROOT file). the goal is to then display the histograms
1233 on the Data Quality Monitoring (DQM) page of the webDCS in order for the users to control the
1234 quality of the data taking at the end of data taking. An example of histogram organisation is given
1235 below:

1236



1238 *Here can put some screens from the webDCS to show the DQM and the plots available to users.*
 1239

1240 **B.2.1.2 CSV files**

1241 Moreover, up to 3 CSV files can be created depending on which ones of the 3 input files were in the
 1242 data folder:

- 1243 • `Offline-Rate.csv` : contains the summary of the noise/gamma hit and cluster rates for each
 1244 chamber partitions,
- 1245 • `Offline-Current.csv` : contains the summary of the currents and voltages applied on each
 1246 RPC HV channel, and
- 1247 • `Offline-L0-EffCl.csv` : contains the summary of the level 0 efficiency and muon cluster
 1248 information **without** tracking.

1249 Note that these 3 CSV files are created along with their *headers* (`Offline-[...]-Header.csv`
 1250 containing the names of each data columns) and are automatically merged together when the offline

1251 analysis tool is called from the webDCS, contrary to the case where the tool is runned locally from
 1252 the terminal as the merging bash script is then not called. Thus, the resulting files, used to make
 1253 official plots, are:

- 1254 ● Rate.csv ,
- 1255 ● Current.csv ,
- 1256 ● L0-EffCl.csv .

1257 **B.3 Analysis inputs and information handling**

1258 The usage of the Offline Analysis tool as well as its output have been presented in the previous sec-
 1259 tion. It is now important to dig further and start looking at the source code and the inputs necessary
 1260 for the tool to work. Indeed, other than the raw ROOT data files that are analysed, more information
 1261 needs to be imported inside of the program to perform the analysis such as the description of the
 1262 setup inside of GIF++ at the time of data taking (number of trolleys, of RPCs, dimensions of the
 1263 detectors, etc...) or the mapping that links the TDC channels to the coresponding RPC channels in
 1264 order to translate the TDC information into human readable data. 2 files are used to transmit all this
 1265 information:

1266

- 1267 ● Dimensions.ini, that provides the necessary setup and RPC information, and
- 1268 ● ChannelsMapping.csv, that gives the link between the TDC and RPC channels as well as the
 mask for each channel (masked or not?).

1270 **B.3.1 Dimensions file and IniFile parser**

1271 This input file, present in every data folder, allows the analysis tool to know of the number of ac-
 1272 tive trolleys, the number of active RPCs in those trolleys, and the details about each RPCs such as
 1273 the number of RPC gaps, the number of pseudo-rapidity partitions (for CMS-like prototypes), the
 1274 number of strips per partion or the dimensions. To do so, there are 3 types of groups in the INI file
 1275 architecture. A first general group, appearing only once at the head of the document, gives informa-
 1276 tion about the number of active trolleys as well as their IDs, as presented in Source Code B.1. For
 1277 each active trolley, a group similar to Source Code B.2 can be found containing information about
 1278 the number of active detectors in the trolley and their IDs. Each trolley group as a `Tt` name format,
 1279 where `t` is the trolley ID. Finally, for each detector stored in slots of an active trolley, there is a group
 1280 providing information about their names and dimensions, as showed in Source Code B.3. Each slot
 1281 group as a `TtSs` name format, where `s` is the slot ID of trolley `t` where the active RPC is hosted.

1282

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

1284 *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 signle string as "13" and then read character by character by the program.*

```
1285 [T1]
nSlots=4
SlotsID=1234
```

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.

```
1286 [T1S1]
Name=RE2-2-NPD-BARC-8
Partitions=3
Gaps=3
Gap1=BOT
Gap2=TN
Gap3=TW
1287 AreaGap1=11694.25
AreaGap2=6432
AreaGap3=4582.82
Strips=32
ActiveArea-A=157.8
ActiveArea-B=121.69
ActiveArea-C=93.03
```

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.

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

1292 B.3.2 TDC to RPC link file and Mapping

1293 The same way the INI dimension file information is stored using `map`, the channel mapping and mask
 1294 information is stored and accessed through `map`. First of all, the mapping CSV file is organised into
 1295 3 columns separated by tabulations (and not by commas, as expected for CSV files as it is easier using
 1296 streams to read tab or space separated data using C++):

```
1297
1298   RPC_channel      TDC_channel      mask
```

1299 using as formatting for each field:

```
1300
1301   TSCCC      TCCC      M
```

1302 `TSCCC` is a 5-digit integer where `T` is the trolley ID, `s` the slot ID in which the RPC is held insite
 1303 the trolley `T` and `ccc` is the RPC channel number, or *strip* number, that can take values up to
 1304 3-digits depending on the detector,

1305 `TCCC` is a 4 digit integer where `T` is the TDC ID, `ccc` is the TDC channel number that can take values
 1306 in between 0 and 127, and

1307 `M` is a 1-digit integer indicating if the channel should be considered (`M = 1`) or discarded (`M = 0`)
 1308 during analysis.

1309 This mapping and masking information is readout and stored thanks to the object `Mapping`, pre-
 1310 sented in Source Code B.4. Similarly to `IniFile` objects, this class has private methods. The first
 1311 one, `Mapping::CheckIfNewLine()` is used to find the newline character '`\n`' or return character
 1312 '`\r`' (depending on which kind of operating system interacted with the file). This is used for the
 1313 simple reason that the masking information has been introduced only during the year 2017 but the
 1314 channel mapping files exist since 2015 and the very beginning of data taking at GIFT++. This means
 1315 that in the older data folders, before the upgrade, the channel mapping file only had 2 columns, the
 1316 RPC channel and the TDC channel. For compatibility reasons, this method helps controlling the
 1317 character following the readout of the 2 first fields of a line. In case any end of line character is
 1318 found, no mask information is present in the file and the default `M = 1` is used. On the contrary, if
 1319 the next character was a tabulation or a space, the mask information is present.

1320 Once the 3 fields have been readout, the second private method `Mapping::CheckIfTDCCh()` is
 1321 used to control that the TDC channel is an existing TDC channel. Finally, the information is stored
 1322 into 3 different maps (`Link`, `ReverseLink` and `Mask`) thanks to the public method `Mapping::Read()`.
 1323 `Link` allows to get the RPC channel by knowing the TDC channel while `ReverseLink` does the op-
 1324 posite by returning the TDC channel by knowing the RPC channel. Finally, `Mask` returns the mask
 1325 associated to a given RPC channel.

```
1326
typedef map<UInt, UInt> MappingData;

class Mapping {
  private:
    bool          CheckIfNewLine(char next);
    bool          CheckIfTDCCh(UInt channel);
    string        FileName;
    MappingData  Link;
    MappingData  ReverseLink;
    MappingData  Mask;
    int           Error;

1327
  public:
    Mapping();
    Mapping(string baseName);
    ~Mapping();

    void SetFileName(const string filename);
    int  Read();
    UInt GetLink(UInt tdcchannel);
    UInt GetReverse(UInt rpcchannel);
    UInt GetMask(UInt rpcchannel);
};
```

1328 *Source Code B.4: Description of C++ object `Mapping` used as a parser for the channel mapping and mask file.*

1329 **B.4 Description of GIFT++ setup within the Offline Analysis tool**

1330 In the previous section, the tool input files have been discussed. The dimension file information is
 1331 stored in a map hosted by the `IniFile` object. But this information is then used to create a series of

1332 new objects that helps defining the Gif++ infrastructure directly into the Offline Analysis. Indeed,
 1333 from the `RPC`, to the more general `Infrastructure`, every element of the Gif++ infrastrucutre is
 1334 recreated for each data analysis based on the information provided in input. All this information
 1335 about the infrastructure will be used to assign each hit signal to a specific strip channel of a specific
 1336 detector, and having a specific active area. This way, rate per unit area calculation is possible.
 1337

1338 B.4.1 RPC objects

1339 `RPC` objects have been developped to represent physical active detectors in Gif++ at the moment
 1340 of data taking. Thus, there are as many `RPC` objects created during the analysis than there were
 1341 active RPCs tested during a run. Each `RPC` hosts the information present in the corresponding INI
 1342 slot group, as shown in B.3, and organises it using a similar architecture. This can be seen from
 1343 Source Code B.5.

1344 To make the object more compact, the lists of gap labels, of gap active areas and strip active
 1345 areas are stored into `vector` dynamical containers. `RPC` objects are always contructed thanks to the
 1346 dimension file information stored into the `IniFILE` and their ID, using the format `TtSs`. Using the
 1347 `RPC` ID, the constructor calls the methods of `IniFile` to initialise the `RPC`. The other constructors
 1348 are not used but exist in case of need. Finally, some getters have been written to access the different
 1349 private parameters storing the detector information.

1350

```
1351 class RPC{
1352     private:
1353         string           name;          //RPC name as in webDCS database
1354         Uint             nGaps;        //Number of gaps in the RPC
1355         Uint             nPartitions; //Number of partitions in the RPC
1356         Uint             nStrips;      //Number of strips per partition
1357         vector<string> gaps;         //List of gap labels (BOT, TOP, etc...)
1358         vector<float>  gapGeo;       //List of gap active areas
1359         vector<float>  stripGeo;    //List of strip active areas
1360
1361     public:
1362         RPC();
1363         RPC(string ID, IniFile* geofile);
1364         RPC(const RPC& other);
1365         ~RPC();
1366         RPC& operator=(const RPC& other);
1367
1368         string GetName();
1369         Uint   GetNGaps();
1370         Uint   GetNPartitions();
1371         Uint   GetNStrips();
1372         string GetGap(Uint g);
1373         float  GetGapGeo(Uint g);
1374         float  GetStripGeo(Uint p);
1375     };
1376 }
```

1352 *Source Code B.5: Description of C++ objects `RPC` that describe each active detectors used during data taking.*

1353 B.4.2 Trolley objects

1354 `Trolley` objects have been developped to represent physical active trolleys in Gif++ at the moment

1355 of data taking. Thus, there are as many trolley objects created during the analysis than there were
 1356 active trolleys hosting tested RPCs during a run. Each `Trolley` hosts the information present in the
 1357 corresponding INI trolley group, as shown in B.2, and organises it using a similar architecture. In
 1358 addition to the information hosted in the INI file, these object have a dynamical container of `RPC`
 1359 objects, representing the active detectors the active trolley was hosting at the time of data taking.
 1360 This can been seen from Source Code B.6.

1361 `Trolley` objects are always contructed thanks to the dimension file information stored into the
 1362 `IniFILE` and their ID, using the format `Tt`. Using the Trolley ID, the constructor calls the methods
 1363 of `IniFile` to initialise the `Trolley`. Retrieving the information of the RPC IDs via `SlotsID`, a new
 1364 `RPC` is constructed and added to the container `RPCs` for each character in the ID string. The other
 1365 constructors are not used but exist in case of need. Finally, some getters have been written to access
 1366 the different private parameters storing the trolley and detectors information.

```
1367
1368 class Trolley{
1369     private:
1370         Uint          nSlots; //Number of active RPCs in the considered trolley
1371         string        SlotsID; //Active RPC IDs written into a string
1372         vector<RPC*> RPCs;   //List of active RPCs
1373
1374     public:
1375         //Constructors, destructor and operator =
1376         Trolley();
1377         Trolley(string ID, IniFile* geofile);
1378         Trolley(const Trolley& other);
1379         ~Trolley();
1380         Trolley& operator=(const Trolley& other);
1381
1382         //Get GIFTrolley members
1383         Uint    GetNSlots();
1384         string  GetSlotsID();
1385         Uint    GetSlotID(Uint s);
1386
1387         //Manage RPC list
1388         RPC*   GetRPC(Uint r);
1389         void   DeleteRPC(Uint r);
1390
1391         //Methods to get members of RPC objects stored in RPCs
1392         string  GetName(Uint r);
1393         Uint    GetNGaps(Uint r);
1394         Uint    GetNPartitions(Uint r);
1395         Uint    GetNStrips(Uint r);
1396         string  GetGap(Uint r, Uint g);
1397         float  GetGapGeo(Uint r, Uint g);
1398         float  GetStripGeo(Uint r, Uint p);
1399     };

```

1369 *Source Code B.6: Description of C++ objects `Trolley` that describe each active trolley used during data taking.*

1370 B.4.3 Infrastructure object

1371 The `Infrastructure` object has been developped to represent the GIFT++ bunker area dedicated to
 1372 CMS RPC experiments. With this very specific object, all the information about the CMS RPC
 1373 setup within GIFT++ at the moment of data taking is stored. It hosts the information present in the

1374 corresponding INI general group, as shown in B.1, and organises it using a similar architecture. In
 1375 addition to the information hosted in the INI file, this object have a dynamical container of `Trolley`
 1376 objects, representing the active tolleys in GIFT++ area. This can been seen from Source Code B.7.

1377 The `Infrastructure` object is always contructed thanks to the dimension file information stored
 1378 into the `IniFILE`. Retrieving the information of the trolley IDs via `TrolleysID`, a new `Trolley` is
 1379 constructed and added to the container `Trolleys` for each character in the ID string. By extension,
 1380 it is easy to understand that the process described in Section B.4.2 for the construction of RPCs
 1381 takes place when a trolley is constructed. The other constructors are not used but exist in case of
 1382 need. Finally, some getters have been written to access the different private parameters storing the
 1383 infrastructure, tolleys and detectors information.

1384

```

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);

    1385      //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);
    };

```

1386 *Source Code B.7: Description of C++ object `Infrastructure` that contains the full information about CMS
 RPC experiment in GIFT++.*

1387 B.5 Handeling of data

1388 As discussed in Appendix A.4.2, the raw data as a `TTree` architecture where every entry is related to
 1389 a trigger signal provided by a muon or a random pulse, whether the goal of the data taking was to
 1390 measure the performance of the detector or the noise/gamma background respectively. Each of these
 1391 entries, referred also as events, contain a more or less full list of hits in the TDC channels to which
 1392 the detectors are connected. To this list of hits corresponds a list of time stamps, marking the arrival
 1393 of the hits within the TDC channel.

1394 The infrastructure of the CMS RPC experiment within `GIF++` being defined, combining the
 1395 information about the raw data with the information provided by both the mapping/mask file and the
 1396 dimension file allows to build new physical objects that will help in computing efficiency or rates.

1397 B.5.1 RPC hits

1398 The raw data stored in the ROOT file as output of the `GIF++` DAQ, is readout by the analysis tool
 1399 using the structure `RAWData` presented in Source Code B.9 that differs from the structure presented
 1400 in Appendix A.4.2 as it is not meant to hold all of the data contained in the ROOT file. In this sense,
 1401 this structure is in the case of the offline analysis tool not a dynamical object and will only be storing
 1402 a single event contained in a single entry of the `TTree`.

1403

```

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

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

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

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

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

```

1405

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

```

1406   struct RAWData{
1407     int           iEvent;      //Event i
1408     int           TDCNHits;    //Number of hits in event i
1409     int           QFlag;       //Quality flag list (1 flag digit per TDC)
1410     vector<Uint> *TDCCh;     //List of channels giving hits per event
1411     vector<float> *TDCTS;     //List of the corresponding time stamps
1412   };

```

1407 *Source Code B.9: Description of C++ structure RAWData.*

1408 Each member of the structure is then linked to the corresponding branch of the ROOT data tree,
1409 as shown in the example of Source Code B.10, and using the method `GetEntry(int i)` of the ROOT
1410 class `TTree` will update the state of the members of `RAWData`.

```

1411   TTree* dataTree = (TTree*)dataFile.Get("RAWData");
1412   RAWData data;
1413
1414   dataTree->SetBranchAddress("EventNumber", &data.iEvent);
1415   dataTree->SetBranchAddress("number_of_hits", &data.TDCNHits);
1416   dataTree->SetBranchAddress("Quality_flag", &data.QFlag);
1417   dataTree->SetBranchAddress("TDC_channel", &data.TDCCh);
1418   dataTree->SetBranchAddress("TDC_TimeStamp", &data.TDCTS);

```

1413 *Source Code B.10: Example of link in between RAWData and TTree.*

1414 The data is then analysed entry by entry and to each element in the TDC channel list, a `RPCHit` is
1415 constructed and stored into 3D dynamical list called `GIFHitList` (Source Code B.9). The 3 dimensions
1416 refer to the 3 layers of the readout in `GIF++`: in the bunker there are *trolleys* holding detectors
1417 in *slots* and each detector readout is divided into 1 or more pseudo-rapidity *partitions*. Using these
1418 3 information allows to assign an address to each readout partition and this address will point to a
1419 specific hit list.

1420 B.5.2 Clusters of hits

```

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);

```

1422

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

C

1423

1424

Structure of the hybrid simulation software

1425

C.1 Introduction

1426

insert text here...

