

Дефектоскопия детекторов по возникающим в них переходным процессам

подготовил: Воробьев Г.

Научный руководитель: Тетерин П.

НИЯУ МИФИ, Б17-102, 2020

New Small Wheel (HL-LHC, ATLAS, CERN)



Для начала остановимся на дефектоскопии sTGC.

NSW для модернизации эксперимента ATLAS состоят из двух типов детекторов – sTGC и MicroMegas. Перед установкой на “колесо” они в последний раз тестируются на дефекты и собираются в сектора.

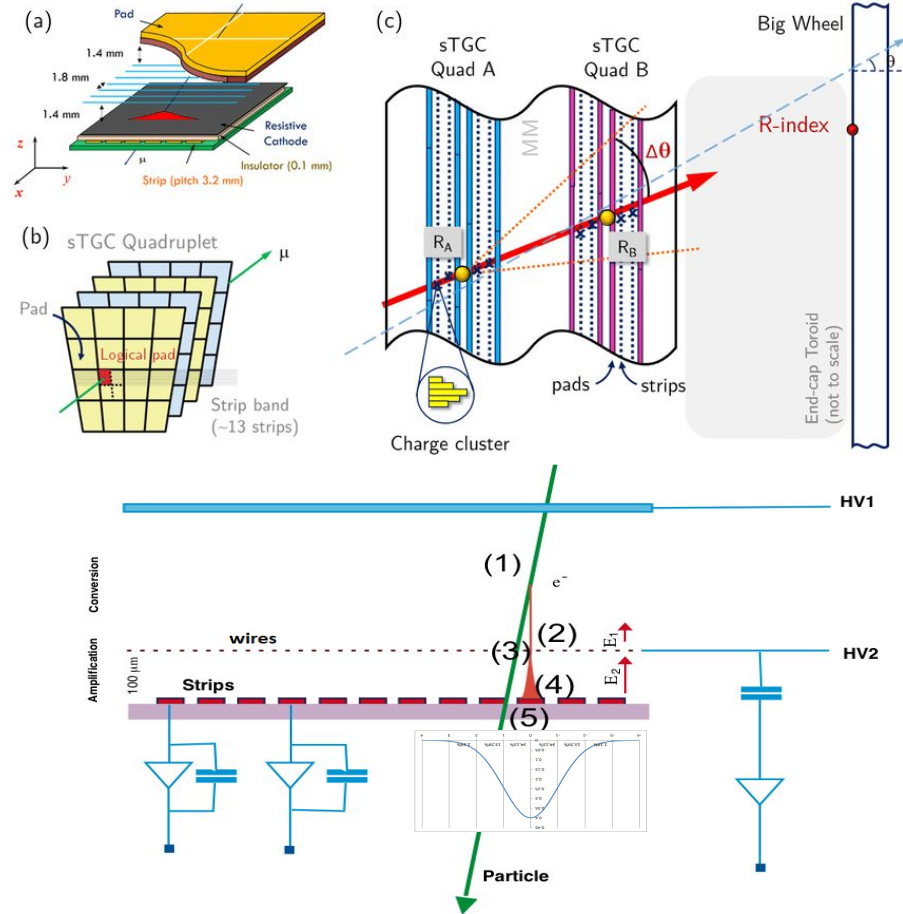
Основная цель этого колеса - детектирование мюонов.

sTGC.

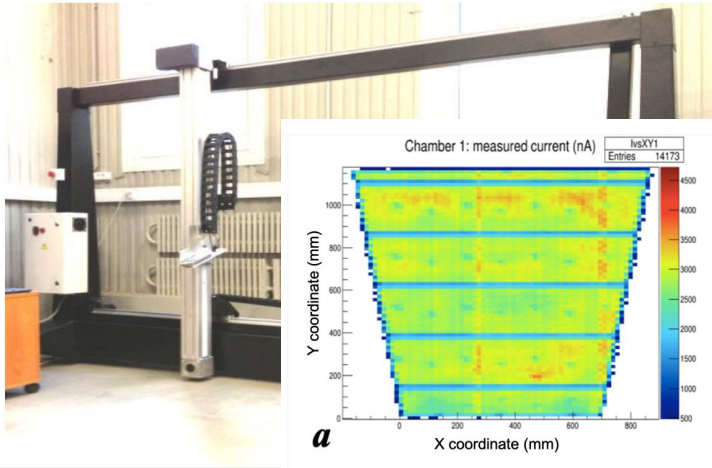
Принцип работы:

- 1) Ионизация газа пролетающей частицей
- 2) Движение рождённых зарядов (ион-электрон) под действием поля
- 3) Ускорение электронов (сильное поле)
- 4) Ускоренные электроны порождают новые ион-электрон пары - лавинный эффект
- 5) Заряды попадают на стрипы и становятся measurable электрическим сигналом

Фитируя по Гауссу получаем координату частицы

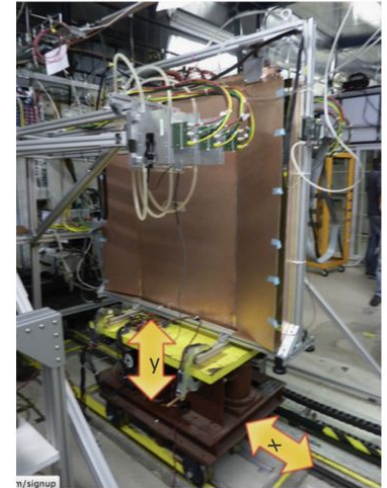


Методы дефектоскопии



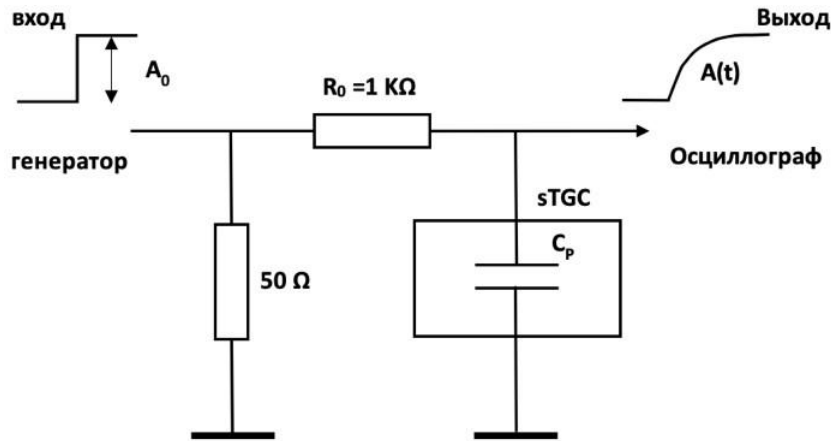
- 1) Высоковольтный тест
- 2) Рентгеновское сканирование
- 3) Космический тест
- 4) Тест гамма-облучения на источнике GIF++ (gamma irradiation facility) в CERN

На пучках тестируют обычно прототипы. Были попытки измерять ёмкости катодных плат (стрипов и падов).



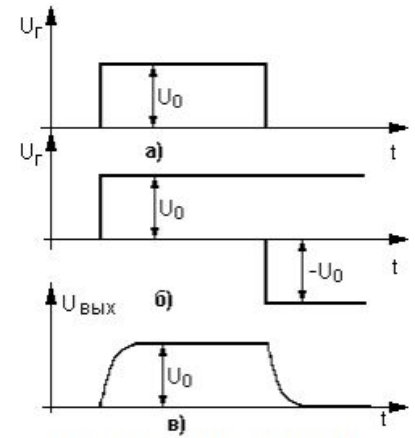
Общий недостаток - Неполное покрытие дефектов, а, в случае 3-4 - обязательное наличие считывающей электроники на детекторах(доп. оборудование). Помимо этого полное или частичное **отсутствие мобильности**. (Необходимо совершать множество транспортировочных и демонтажных действий, чтоб детектировать поломку. А также, сложные и большие устройства дефектоскопии.)

Переходные процессы в электрических цепях. Метод исследования отклика



Переходные процессы - следствие законов коммутации (запрет на скачкообразное изменение энергий реактивных компонентов)

← схема замещения интегрирующей RC-цепи →





Поставим цель:

Автоматизировать метод исследования отклика, посредством написания программы считывания и анализа данных осциллографа.

(спойлер: не LabVIEW)



Python-реализация программы считывания и анализа данных отклика

Преимущества python языка над языком "G":

- 1) Он один из самых **распространённых**
- 2) **Удобство, скорость** работы и написания, закреплённые многолетним использованием, причём в самых разных задачах
- 3) Большинство решений python - **open source**, в то время как LabVIEW - сама по себе нет
- 4) Прямая поддержка библиотеки ROOT (**pyROOT**)

PyVISA и PyTektronixScope



PyTektronixScope 0.2

```
pip install PyTektronixScope
```

```
Finished processing dependencies for PyTektronixScope
```

```
D:\Users\gree-\Desktop\PyTektronixScope-0.2>
```

```
Microsoft Windows [Version 10.0.18363.836]
(c) Корпорация Майкрософт (Microsoft Corporation),
C:\Users\gree->pip install -U pyvisa
Collecting pyvisa
  Downloading PyVISA-1.10.1.tar.gz (6.8 MB)
     |-----| 6.8 MB 1.3 MB/s
Installing collected packages: pyvisa
  Running setup.py install for pyvisa ... done
Successfully installed pyvisa-1.10.1
```





Считывание данных осциллографа Tektronix MDO3024

всего в 3-и строчки:

```
1 from PyTektronixScope import PyTektronixScope
2
3 scope = TektronixScope(instrument_resource_name)
4 X,Y = scope.read_data_one_channel('CH2', t0 = 0, DeltaT = 1E-6,
  ↔ x_axis_out=True)
```

Подготовка ПО для анализа. 1

```
>>> import ROOT
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "C:\Program Files\root\bin\ROOT.py", line 24, in <module>
    import cppy
  File "C:\Program Files\root\bin\cppyu.py", line 61, in <module>
    import libPyROOT as _backend
ImportError: DLL load failed: Не найден указанный модуль.
>>>
```

PyROOT версий python 3+ и ROOT 6+ - не работает на Windows

Известные удачные попытки были на python 2.6 и ROOT 5, что, вообще говоря, не рационально



ROOT
Data Analysis Framework

Подготовка ПО для анализа. 2

Решение - перейти на операционную систему Linux (@ui02.lxfarm.mephi.ru)



```
cal b17_01@ui02:~
```

```
[b17_01@ui02 ~]$ bash
(base) [b17_01@ui02 ~]$ conda install -c conda-forge root
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

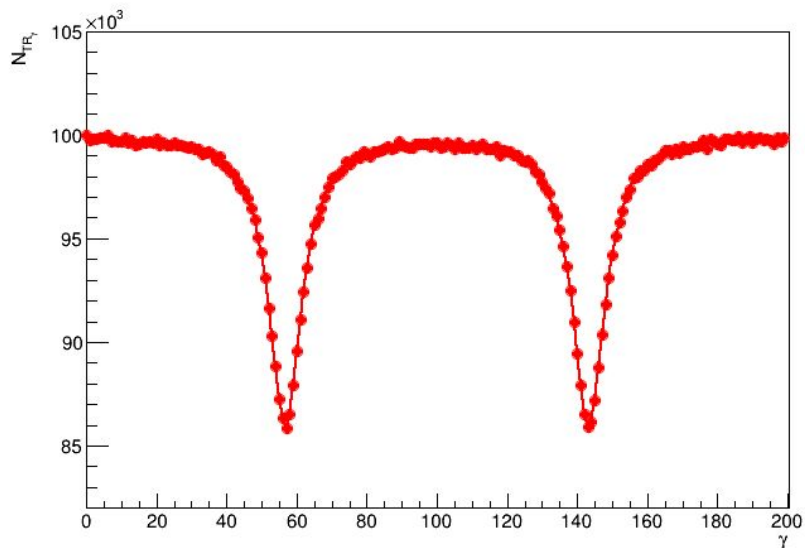
Среди списка пакетов на Anaconda Cloud был найден официальный пакет ROOT, поддерживаемый операционной системой Linux. После чего ROOT был успешно установлен и pyROOT был успешно протестирован в среде:

```
cal b17_01@ui02:~
```

```
(base) [b17_01@ui02 ~]$ python3.7
Python 3.7.6 | packaged by conda-forge | (default, Mar 23 2020, 23:03:20)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import ROOT
>>> exit()
```

Подключение было реализовано с помощью SSH-клиента PuTTY. Была поставлена программа Miniconda3 4.8.2 (в целях настроить удобную схему управления пакетами и версиями языка Python).

Анализ данных отклика



результат программы

пример txt файла

```
96439
95878
95026
94317
93080
91640
90300
88815
87234
86339
85853
86540
87945
89561
91106
92409
93605
94740
95635
95945
96429
96995
```

скрипт

```
from ROOT import TCanvas, TGraph

n = 200
x, y = array( 'd' ), array( 'd' )
file = open("N2.txt", "r")
list1=file.readlines()
for i in range( n ):
    x.append( i )
    y.append( float(list1[i]))
    print(' i %i %f %f ' % (i,x[i],y[i]))

g = TGraph(200,x,y)
MyC = TCanvas("MyC", "MyC", 1)
g.SetMarkerStyle(8)
g.SetMarkerColor(2)
g.SetMarkerSize(1)
g.SetLineColor(2)
g.SetLineWidth(2)
g.GetYaxis().SetTitle("Y")
g.GetXaxis().SetTitle("X")
g.GetYaxis().SetRangeUser(82000,105000)
g.GetXaxis().SetRangeUser(0,200)
g.SetTitle(" ")
g.Draw("ALP")
MyC.Draw()
```



Заключение

Мы рассмотрели различные **методики дефектоскопии** и рассмотрели один из самых перспективных - **метод исследования отклика**, возникающего при переходном процессе, вызванного внешним сигналом (сигналом генератора). Этот метод эффективен и удобен для проверки качества **на всём протяжении жизненного цикла детектора и цикла производства** (Отдельно отмечу, что метод способен оценить влияние каждого этапа сборки на дефекты). Причём, потенциально, данный метод можно использовать и на **других детекторах** (помимо sTGC), например на MicroMegas или других пропорциональных камерах. В целях развития метода был **предложен код** для считывания отклика, а также предоставлена вся необходимая информация для подготовки оборудования к исследованию такого рода сигнала и пример кода анализа данных с помощью пакета библиотек ROOT (а конкретнее с использованием pyROOT).

Развитие написанного софта **с помощью технологий машинного обучения** станет новым толчком в понимании дефектообразования и поможет отрасли найти новые решения в детекторостроении.

Ну и были **получены навыки** (python, anaconda cloud, pyROOT)