Quality assurance for TPC

Quality assurance

- Process:
    - Detect the problems
        - Define, what is the problem
    - What do we expect?
        - Defined in the TDR and in the PPR on the basis of simulation
    - Until which point the detector the information form the detector is reasonable?
        - How far we are from the expectation?
    - Define the limits of working conditions
    - Modify expectation

TPC Quality assurance (-1)

- TPC in lucky situation - TPC test in 2006
  - Answer to the part of the questions
- The base things – according expectation
  - Space point resolution, cluster shape, dEdx resolution
- Some not
  - The noise edge effect
  - Floating wires with higher gain
- Current “feeling” - from reconstruction point of view such effect should not affect the performance of the TPC

Quality assurance (0)

1) Quality assurance based on the statistical properties of the data
   1) Using data from calibration algorithm
   2) Calibration viewer adopted to the AliEve
2) Comparison of the MC data with the reconstructed data
   1) Code in PWG1
   2) Implemented as components
   3) Usage in the AliAnalysis not yet tested-implemented (Work in progress)
3) Low level monitoring – e.g counters to be implemented
4) Some low level histograms – e.g. mean amplitude vs z, vs x only qualitative information

d. **Time 0 calibration - Electronic calibration (pulser/data)**
e. **Time response function width (pulser/data)**

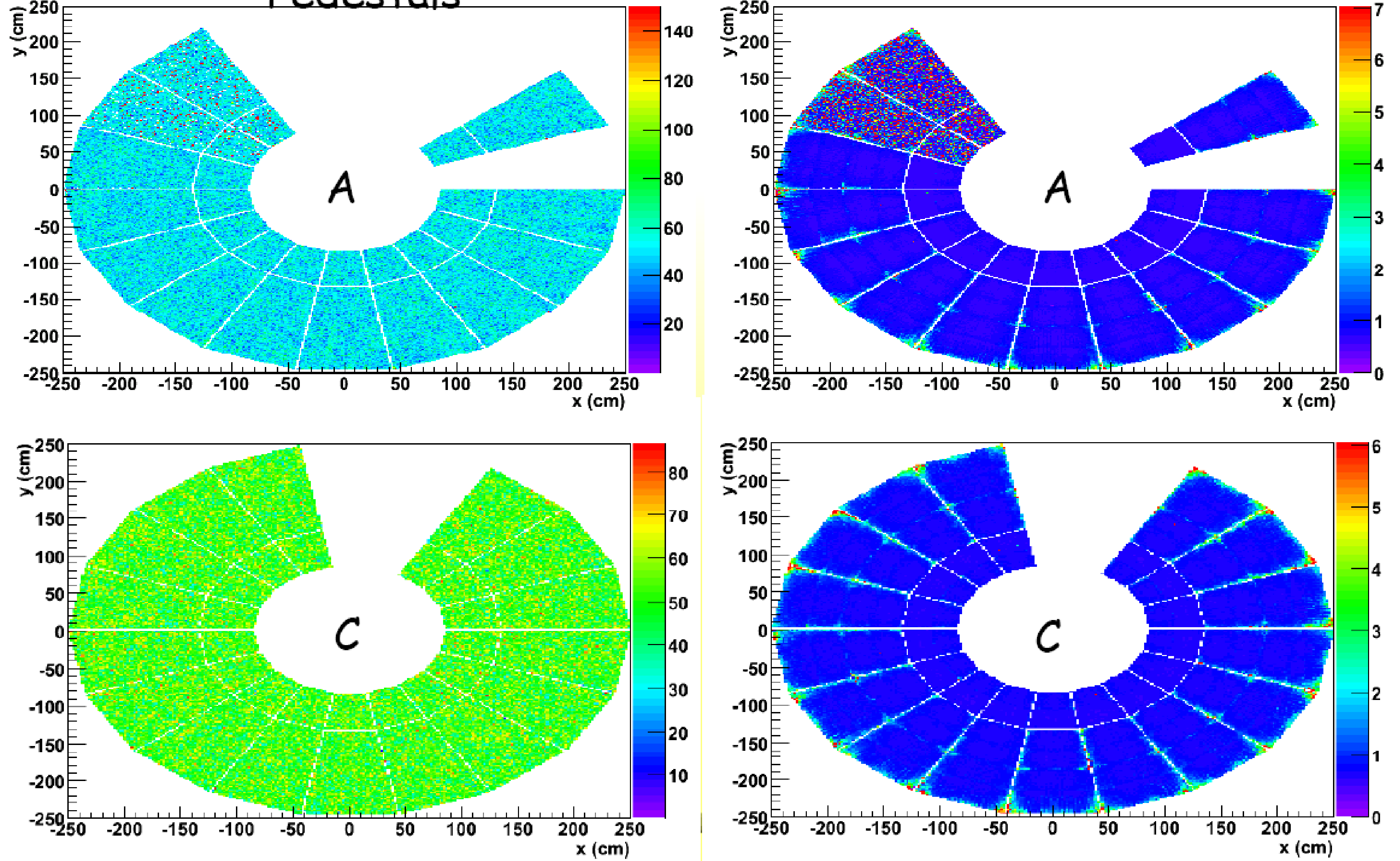
2. **Gain calibration**
   a. **Krypton gain calibration**
   b. **Gain calibration using cosmic (parameterization)**
   c. **Gain calibration using laser – central electrode plane (pad- by-pad fluctuation)**
   d. **Attenuation loss (cosmic)**
3. **Drift velocity calibration. –in relation with 3 c**
   a. **Laser system – tracks +CE signals (local drift velocity parameterization)**
4. **DCS values in OCDB.**
   a. **Corrections(p, T)**
   b. **Goofy (drift velocity, attenuation loss)**
   c. **Temperature map.**
5. **Space point resolution parameterization and cluster shape parameterization**
6. **Space point correction**
   a. **E distortions (laser) algorithm to be defined.**
   b. **ExB (B map + laser) algorithm to be defined.**
   c. **Drift velocity map – parameterization algorithm to be defined.**
7. Data quality monitoring based on calibration parameters –strongly related with points (1-6)
   a. Noise calibration – Detection of outliers (alarms), FFT spectra for outliers
   b. Electronic gain calibration – Detection of outliers (alarms)
   c. Time 0 calibration - Detection of outliers (alarms)
   d. Gain calibration using cosmic – Detection of outliers (alarms)
   e. Space point resolution parameterization and cluster shape parameterization – Pulls for sectors, pad-rows, detection of outliers (alarms)
8. **Central electrode plane (Unisochronity correction)**
9. **Ion tail characteristics and optimization of filter parameters (laser, cosmic)**
10. **Alignment**
   a. **TPC internal alignment –once per year.**
   b. **TPC global alignment –every magnetic field change.**

Quality assurance

- Data quality monitoring based on statistical properties of data - Extracting in calibration procedure
- Low level – digit level:
  - Noise and pedestal calibration
  - Electronic gain calibration
  - Time 0 calibration
- Direct answer
  - Number of dead channels
  - Percentage of suspicious channels
- Alarms:
  - ? To be defined ? - Consult with Hardware people

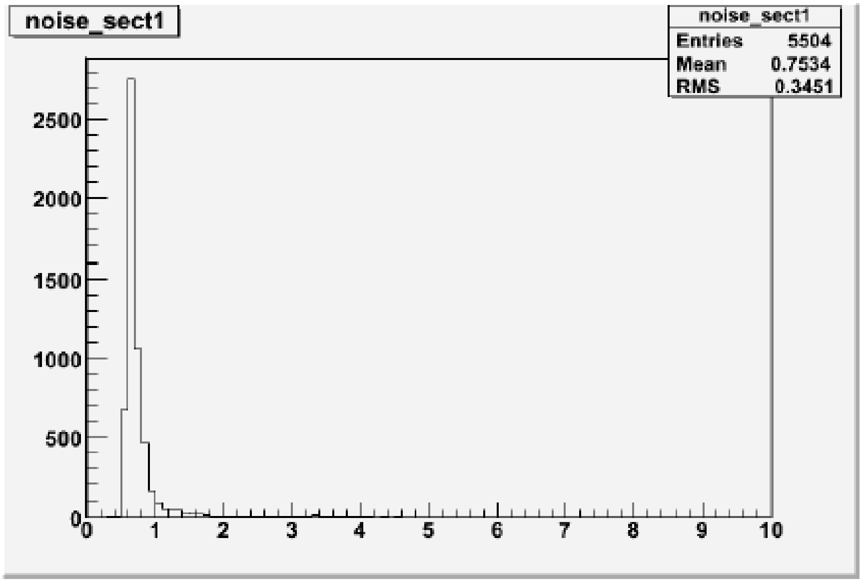
Examples - Output

Pedestals | Noise

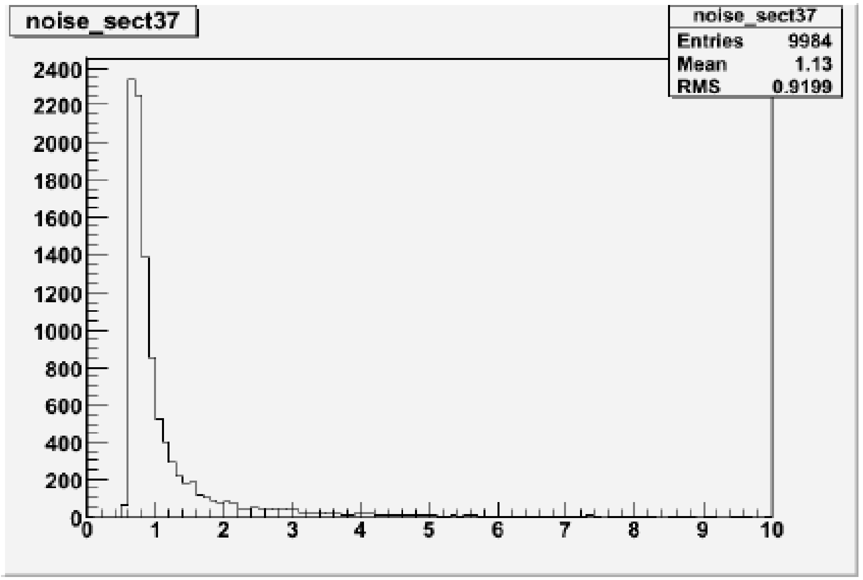


Noise distributions

- Noise levels about 1 ADC count
- Noise distributions, sector A01:

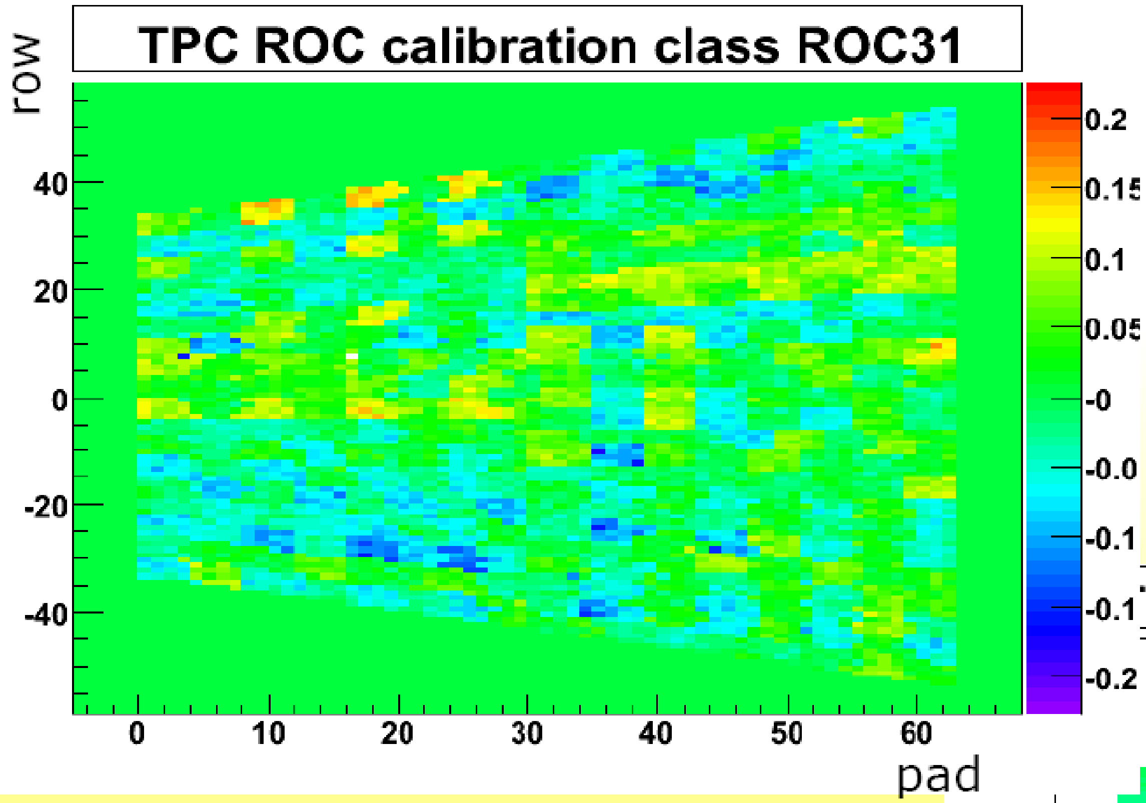


IROC



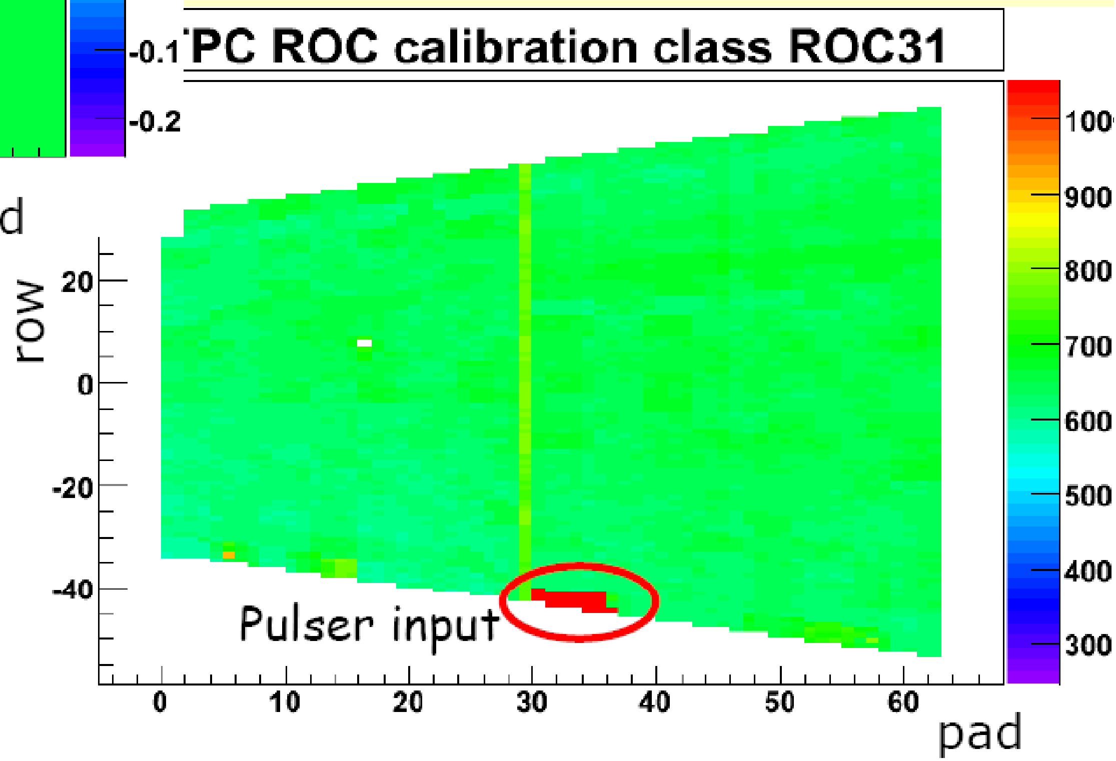
OROC

Examples - Output



Pad-Row T0 distribution

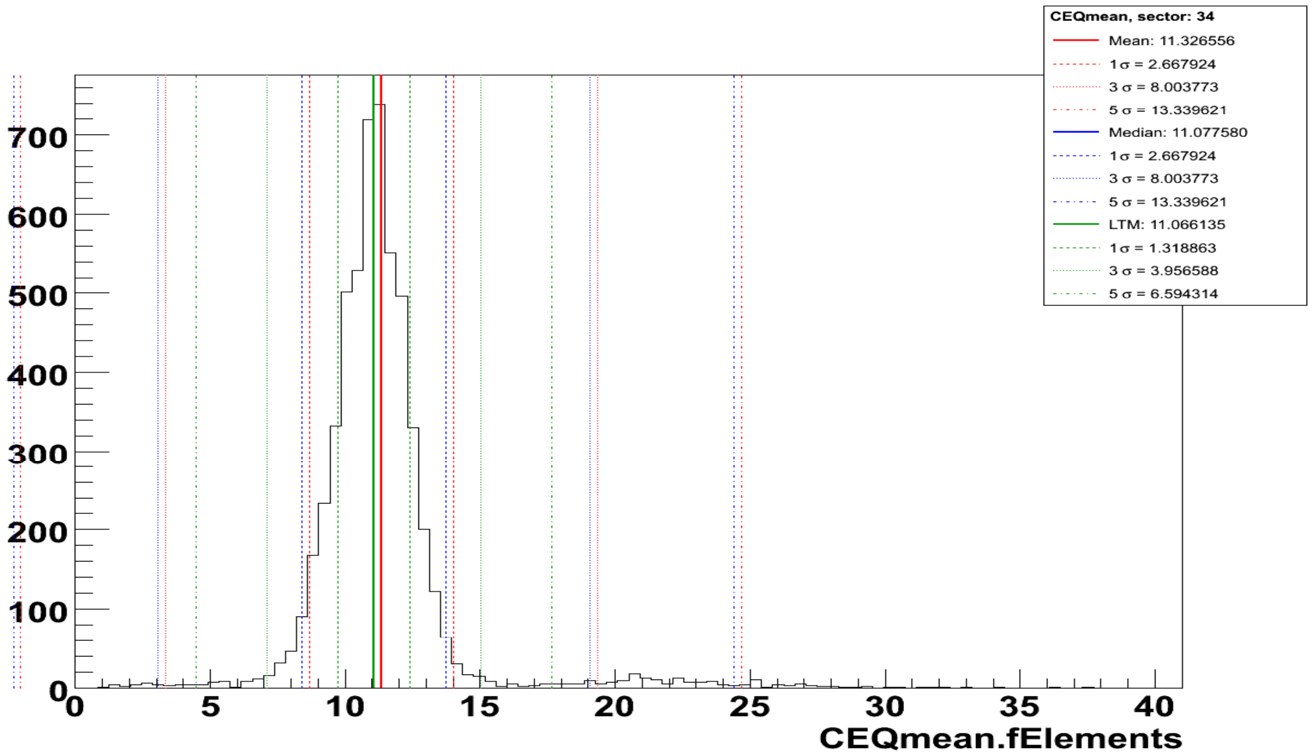
Pad-Row Q distribution



CalibViewer Functionality: DrawHisto1D

`DrawHisto1D:` Draws histogram for given sector with mean and different σ intervals

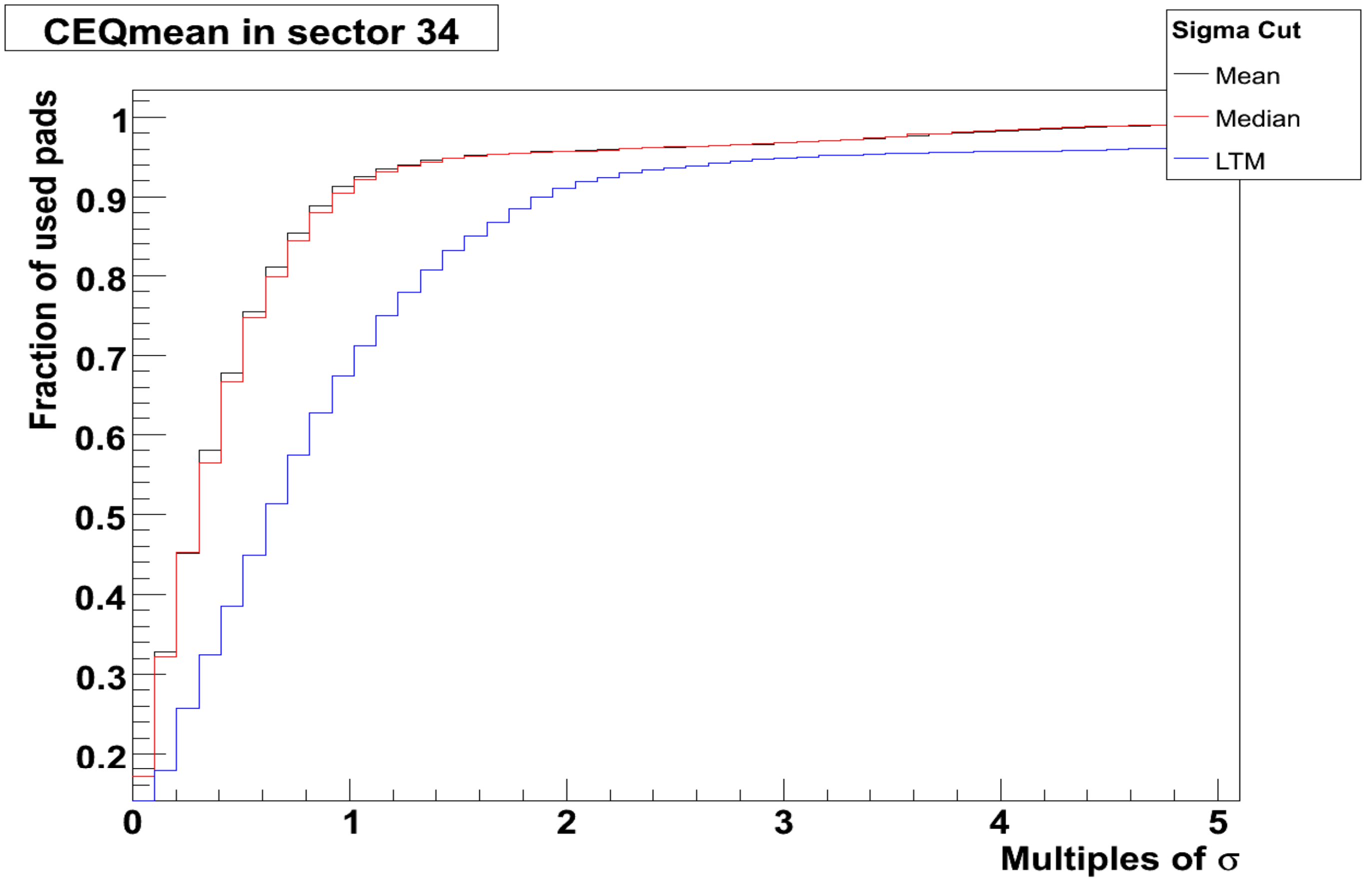
- `DrawHisto1D(char* type, Int_t sector, TVectorF& nsigma)`
- `TvectorF vec(3); vec[0]=1; ...`
- `DrawHisto1D("CEQmean", 34, vec)`



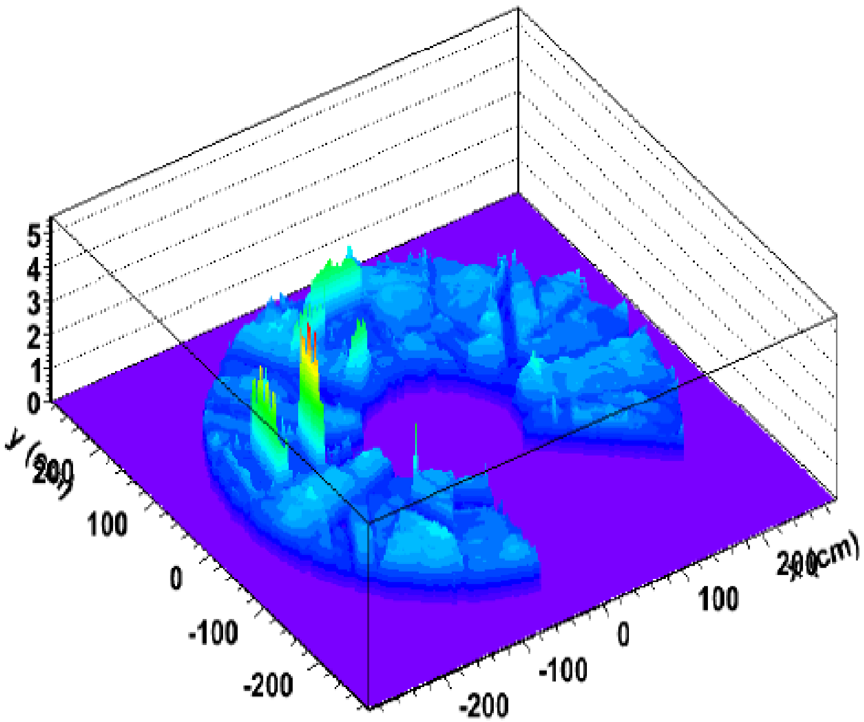
CalibViewer Functionality: SigmaCut

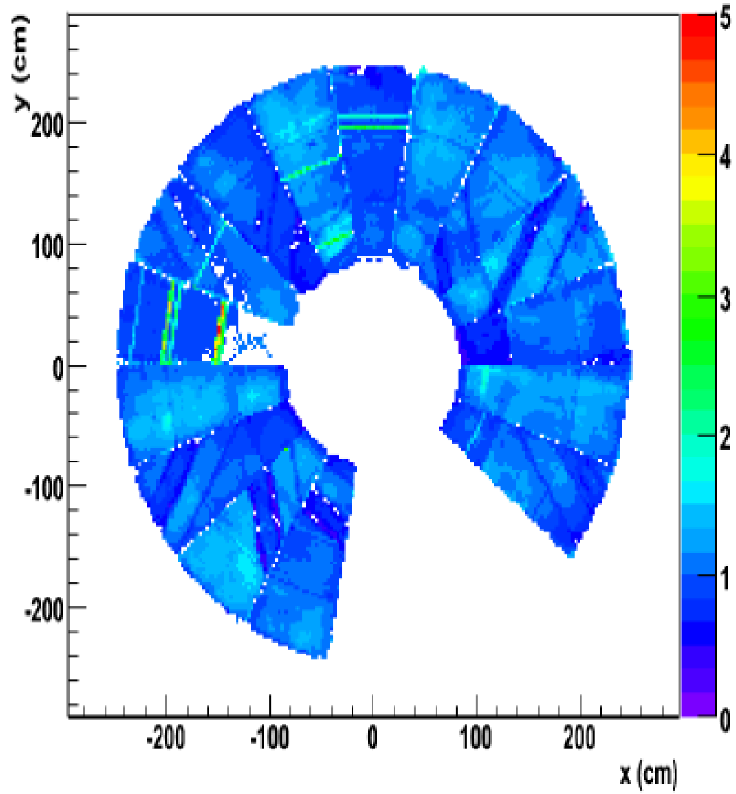
`SigmaCut:` **Shows fraction of rejected pads for different σ cuts**

```
- SigmaCut(char* type, Int_t sector,
      Float_t sigmaMax, Float_t sigmaStep)
- SigmaCut("CEQmean", 34, 5, 0.05)
```



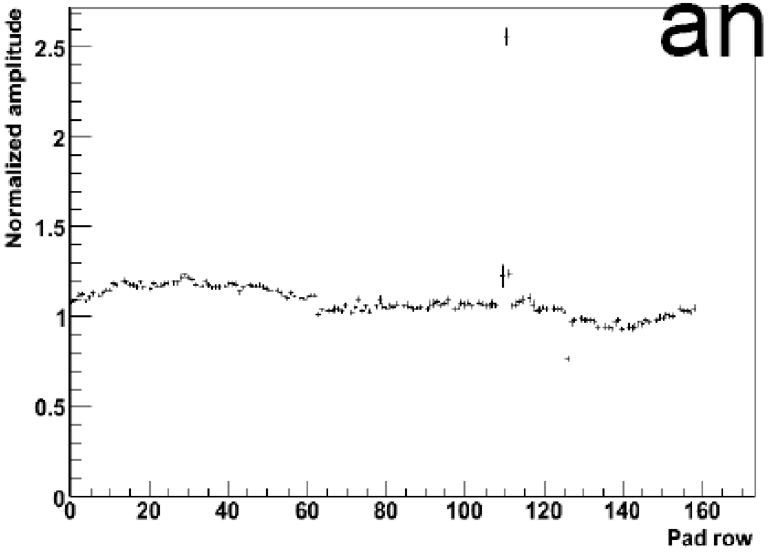
Central electrode amplitude analysis

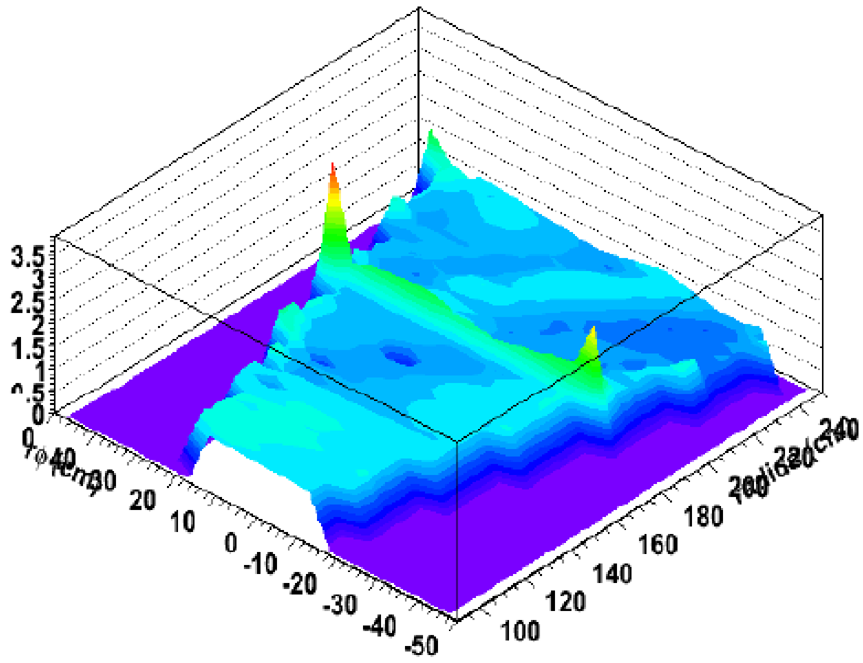


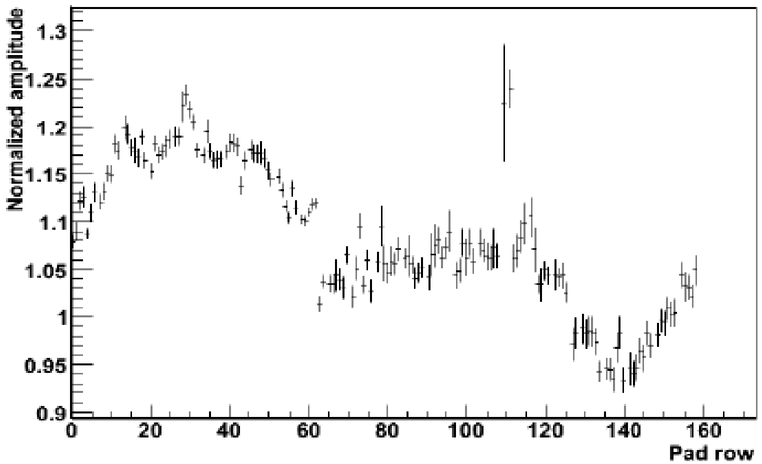


Marian Ivanov, Jens Wiechula

Central electrode amplitude analysis







Jens Wiechula

Quality assurance(2)

- Data quality monitoring based on statistical properties of data - Extracting in calibration procedure
- High level (track- cluster level):
  - ***Time 0 calibration - Detection of outliers (alarms)***
- ***Gain calibration using cosmic – Detection of outliers (alarms)***
- ***Space point resolution parametrization and cluster shape parametrization – Pulls for sectors, pad-rows, detection of outliers (alarms)***
- ***DCA resolution, theta – fi dependence***

The detector response (A(x,t)) to the particle deposit is a random variable. This variable is position and time dependent. The response can be written as product of the energy deposit (E_d) and gain function (G(x,t)).

Averaging the detector response over the many particles we can get the mean gain function at given position at some time interval. The systematic uncertainty of such estimate is given by the uncertanty of energy deposit function. The importance of the energy deposit influence can be demonstrated in figure 1.

$$A(x,t) = E_d(x)G(x,t) \quad (1)$$

$$< A(x,t) >=< E_d(x) >< G(x,t) > \quad (2)$$

$$< G(x,t) >= \frac{< A(x,t) >}{< E_d(x) >} \quad (3)$$

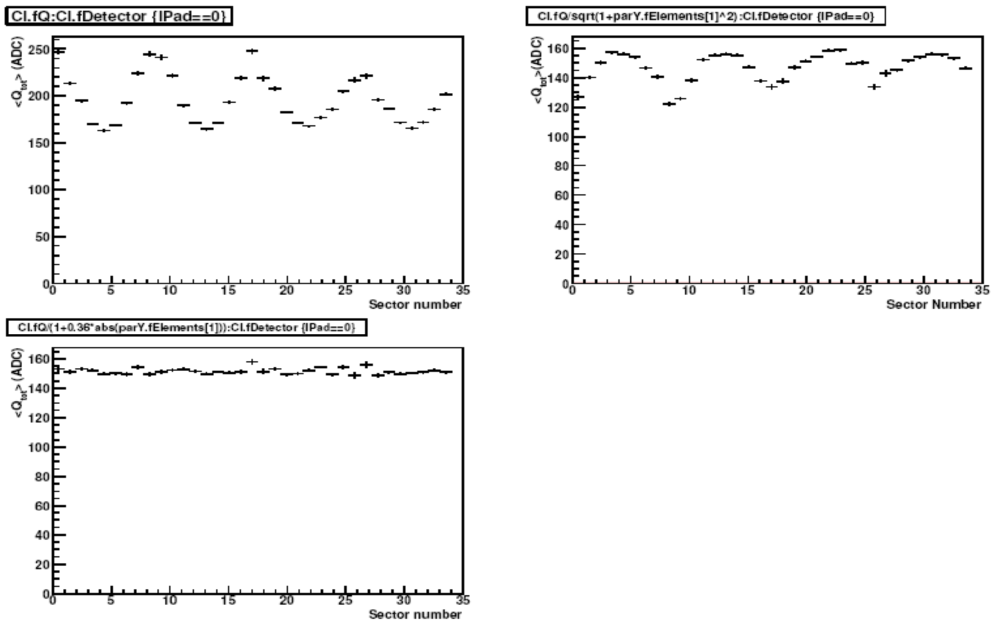


Fig. 1. Cosmic simulation. The mean amplitude at different sectors of the TPC. The gain factor is the same in all sectors. a.) Raw mean amplitude b.) Amplitude normalized to the tracke length c.) Amplitude normalized to the angle (simple linear parameterization)

The mean amplitude at different x,y,z position is not proportional to the gain

The unknown energy deposit make a big influence

The influence doucmented on the cosmic MC data

The gain function is product of the gas gain function (G_g(x,t)), coupling function (Pad Response function (PRF)), electronic gain (G_{chip}) and attenuation

loss (A_{att}).

$$G(x,t) = A_{att}(x,t)G_g(x,t)G_{coupling}(x)G_{chip}(x) \quad (4)$$

In the ALICE TPC, the gas gain function $G_g(x,t)$ and coupling function ($G_{coupling}$) are usually smooth function of the position which are given mainly by distance between the pads and wires. The time dependence of $G_g(x,t)$ is given mainly by the time variation of the pressure. The other variables will be controlled on the level $\leq 1\%$.

The attenuation function is given mainly be electron attachment, it is drift length dependent - exponential decay. The decay length depends on the admixture of the electronegative molecules in the gas. The concentration of these admixtures is time dependent, therefore it decay length should be regularly controlled.

The electronic gain (G_{chip}) is determined by the tolerance (...). In ALICE TPC the variation is on the level of $\sigma \approx 1\%$. The grouping on the chip level is obsereved. (need picture 1D and 2D).

The energy deposit (E_d) is given by the type of the particle (mass), their local properties :momenta, pad length, tracklet orientation (ϕ and θ).

$$E_d(x) = E_d(m, p(x), l(x)) \approx k\frac{dE}{dx}(\beta\gamma)L(padlength, \phi, \theta) \quad (5)$$

The mean energy deposit $< E_d(x) >$ can strongly differ for different physical cases (cosmic, beam-beam interaction).

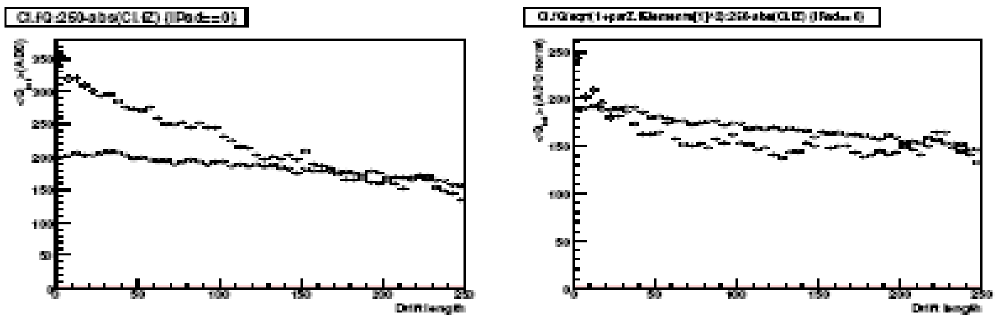


Fig. 2. Mean total charge as function of z for cosmic topology and for pp events. Before and after normalization to the local inclination angle.

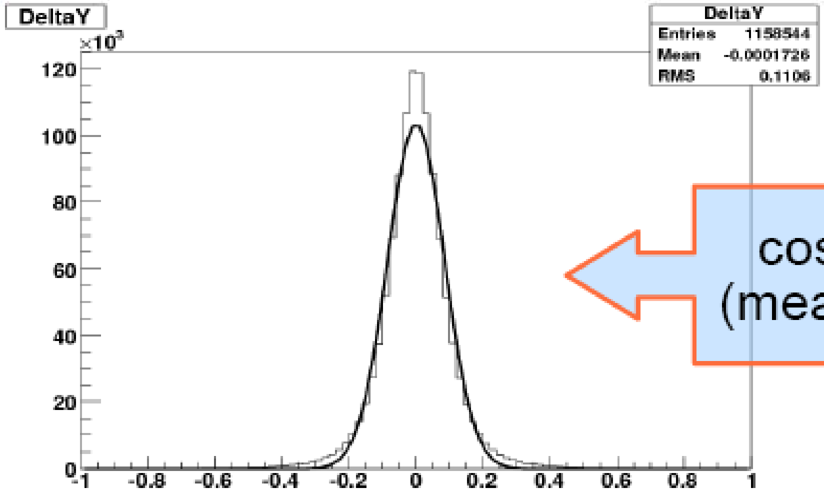
The energy deposit is particle and angular dependent

Example, mean aplitude as function of z - comparison of the pp event and cosmic (MC) data

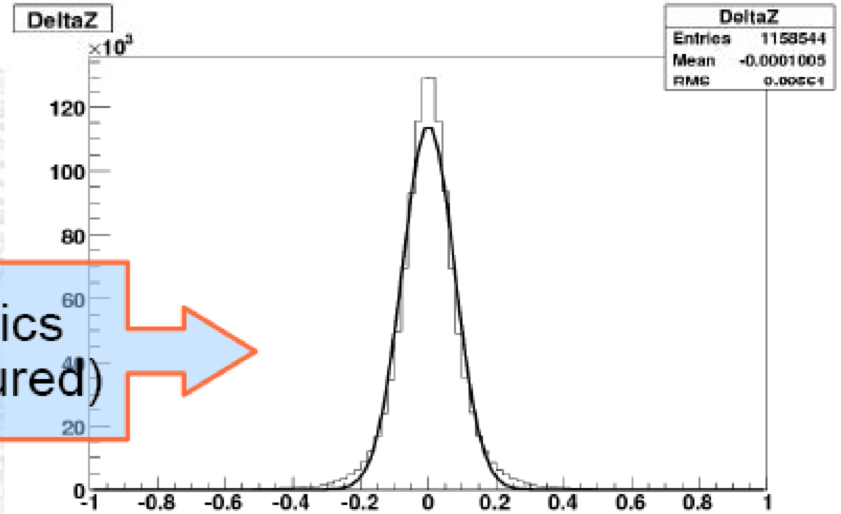
TECHNISCHE UNIVERSITÄT DARMSTADT

GSI

σ Histograms

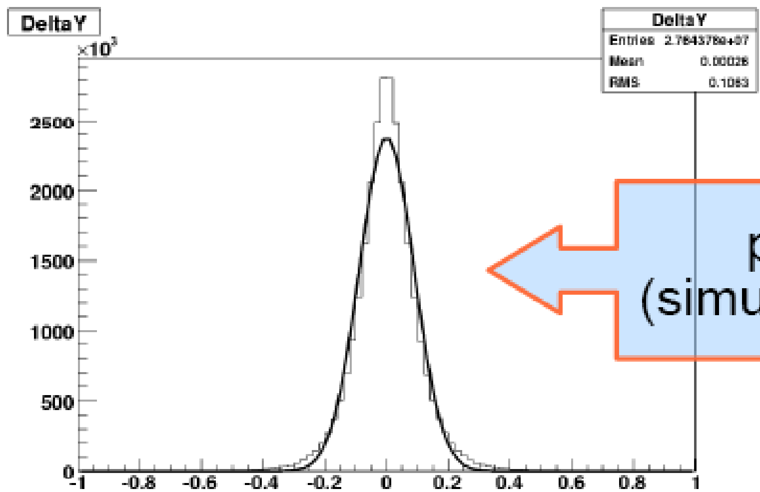


(RMS = 0.1106)

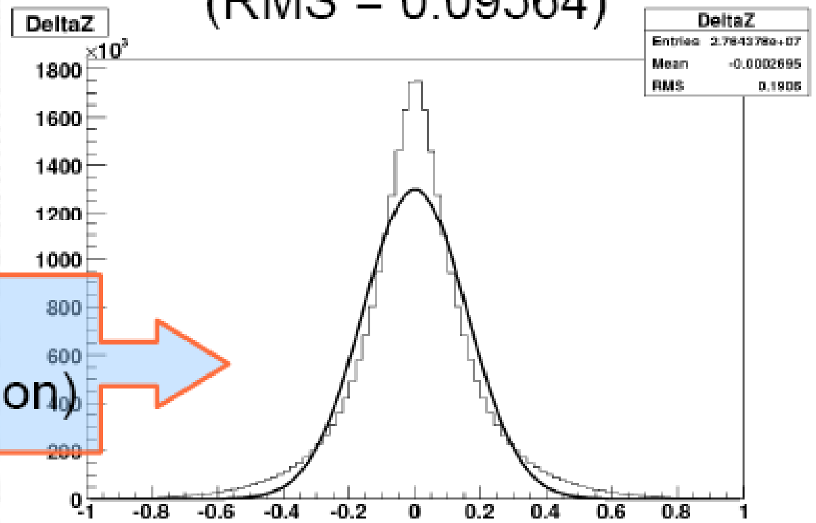


(RMS = 0.09564)

cosmics (measured)



(RMS = 0.1063)

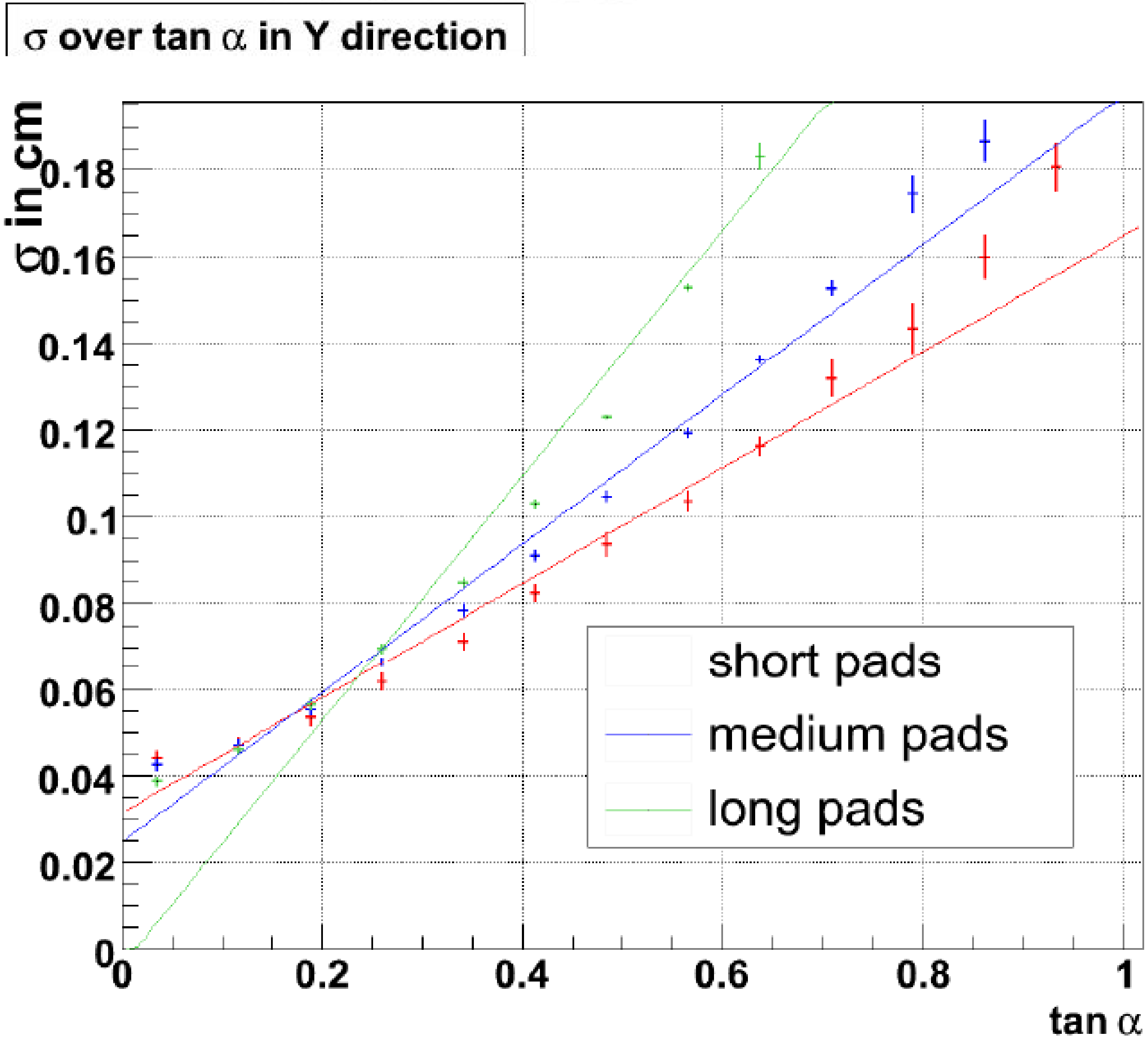


(RMS = 0.1906)

pp (simulation)

Marian Ivanov
Lars Bozyk
29 August 2007

σ vs. tan α



(|Qmean|<0 && AngleC>1 && Zm>50 && Zm<80)

Marian Ivanov
Lars Bozyk

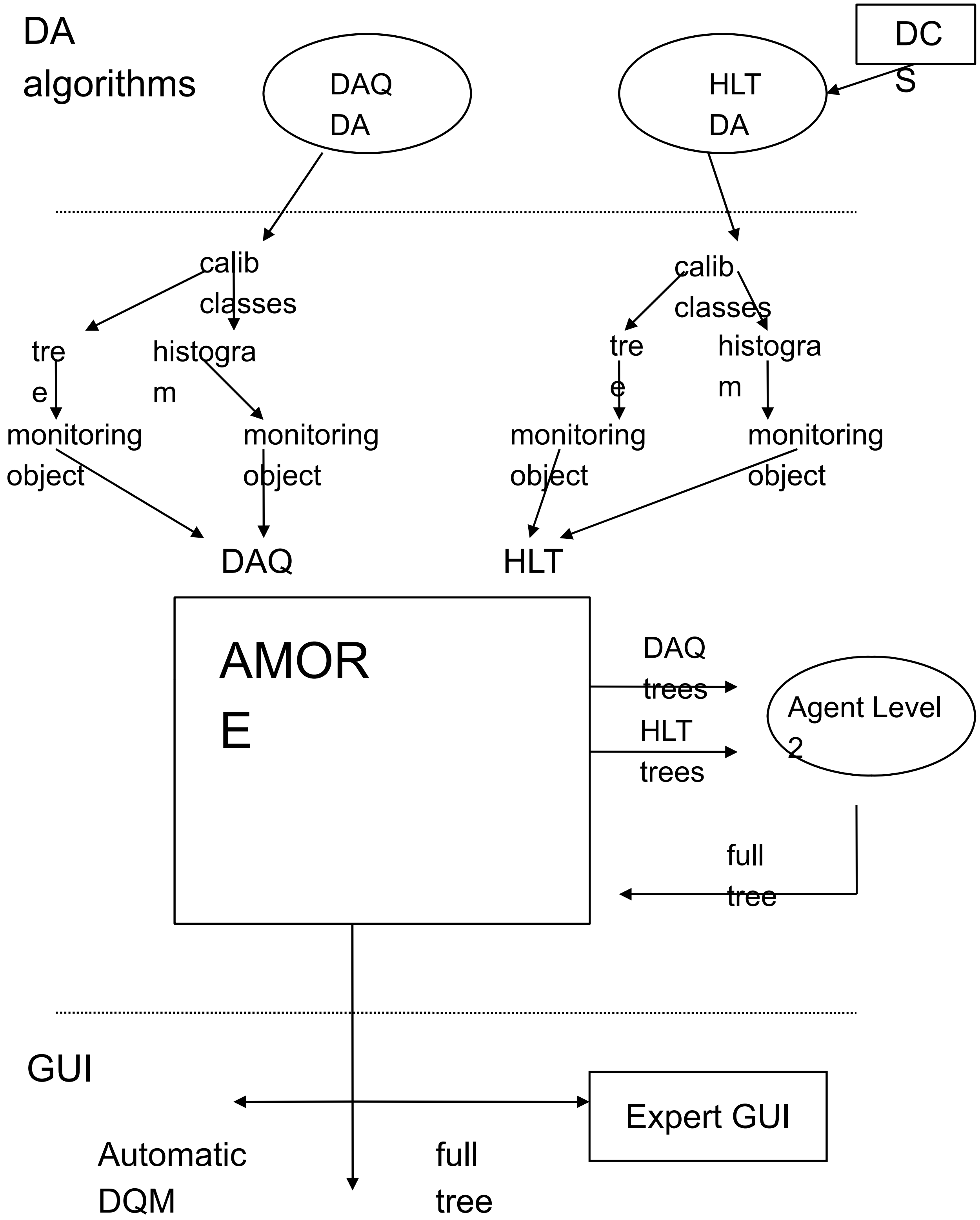
29 August 2007

Calibration algorithm

- Component model
  - The input and output clearly defined
  - External event loop (Possibility to plugin component to the DAQ DA, HLT DA, OFFLINE code, selectors, AliAnalisys*)
- The data source - Input
  - Raw data
  - Set of clusters belonging to the track
- Output
  - Calibration component
  - Calibration components provides the functionality to generate different histograms, views, graphs
- The calibration constants can be grouped to the database (Ttree) with simple Query mechanism
- **BASE feature to make simple and powerful data visualization**

Usage of the Calibration algorithm in QA

- 2 Options:
  - Use the CA component to generate predefined views, histograms
    - Problem: Duplication of the CPU requirements
  - Access the calibration data during the data taking
    - Problem: Not possible in all of the cases (Restricted access)
- **More details summarized in the amore.doc document (H.Helstrup)**


DA algorithms
DAQ DA
HLT DA
DCS
calib classes
calib classes
tree
histogram
tree
histogram
monitoring object
monitoring object
monitoring object
monitoring object
DAQ
HLT
AMORE
DAQ trees
HLT trees
Agent Level 2
full tree
GUI
Automatic DQM
full tree
Expert GUI

TPC Calibration Viewer GUI

This demo can easily be opened with the following command at the aliroot prompt:(http://indico.cern.ch/conferenceDisplay.py?confId=17348)

`AliTPCCalibViewerGUI::ShowGUI("CalibTree.root")`

(`CalibTree.root` is a file containing calibration data generated with `MakeTree`)

GUI realized as a `TGCompositeFrame`

- possibility to open different views in several tabs
- allows integration into existing frameworks (AliEve, AMORE, MOOD, HLT)

AliTPCCalibViewerGUI demo choreography

Action	Explanation
start AliROOT .x UliStyle.C AliTPCCalibViewerGUI::ShowGUI("CalibTree.root")	
➔ select CETmean	1D histogram of CETmean
➔ select CEQmean ➔ 2D	change to 2D plot
➔ Side A ➔ Side C ➔ Sector ➔ slide through sectors using up and down buttons ➔ jump to sectors 34 and 70 by entering number	show effect of different cuts (to plot sides or sectors)
	scale changes with every new picture, annoying for comparison
➔ Side C ➔ fix scale: Enter 50 for Maximum and press Enter Enter 0 for Minimum and press Enter ➔ Side A ➔ slide through some sectors	now the scale is fixed, better if you want to compare different sectors/plots/whatever
➔ Sector 57 ➔ use "Drag and Draw" on the color scale in the diagram to set the maximum to ~ 15, minimum to ~ 0 ➔ press "Get scale from plot" button	shows how to set the minimum/maximum values according to what is shown in the diagram
➔ select "divide by" from normalization combo box ➔ click "Normalized" radio button	because the scale is fixed now you only see a roughly uniform color
➔ uncheck "Get Min + Max auto." ➔ uncheck "Set fixed max." and "Set fixed min." ➔ check/uncheck "Set fixed max." for demonstration	now the scale is set automatically, but manually entered Max-value is still available
➔ press "Get scale from plot" button ➔ check "Set fixed max." and "Set fixed min." ➔ select different normalizations (_Mean, _Median, _LTM, LFitIntern_4_8.fElements)	not much difference for whole sector statistic values, but local fit adapts local variations very well (except at gluing boundaries)
➔ check "Draw options" ➔ select "proflego2" from draw options drop down box	show fancy 3D plot possibility
	that's all for the basic functionalities, now let's go to the "expert mode"
➔ uncheck "Draw options" ➔ Side A ➔ check "additional cuts" ➔ write "CEOutliers~==0" in text box below ➔ press Enter	show effect of user specified cut (here: cut on outliers) Note the abbreviation: ~ == .fElements
➔ write "CEOutliers~>0" in text box ➔ select "CEOutliers~==0" again from the drop down list	history functionality for custom text boxes
➔ uncheck "Set fixed max." and "Set fixed min." ➔ click "Raw" radio button ➔ select "CETmean" from Predefined-list ➔ write "lx~ ++ ly~" in custom fit text box ➔ press Enter	now a linear fit is made through the displayed data (as you see it on screen, with applied cuts); the fit function is written to the aliroot console

TPC Calibration Viewer GUI

`AliTPCCalibViewerGUI` provides a graphical interface for visualization of calibration information. It utilises the `AliTPCCalibViewer` class for generating the diagrams.