# TPC calibration strategy

Several different frameworks will be involved in the TPC calibration, including DAQ, HLT, DCS and Offline. Several components inside these frameworks will be involved, among them Detector Algorithms (DA), automatic quality control (AMORE), Offline Calibration Data Base (OCDB). All calibrations will be based on common calibration classes, which are discussed below. These classes are common for all frameworks. Root files containing these classes are transported between frameworks according to the agreed protocols.

# 1 TPC Calibration classes

## 1.1 Calibration tasks:

1. Pedestal and noise calibration.

   a. Pedestal per time bin and pad

   b. Pedestal per pad

   Electronic calibration

   c. Electronics gain calibration (pulser)

   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.

## 1.2 Data base entries

**Existing:**

1. Pedestals

2. PadNoise

3. PadTime0

4. PadGainFactor

5. Parameters - Currently hardwired numbers - drift velocity, sampling frequency

6. Temperature

7. Pressure

   **To be added:**

8. ALTRO parameters (Frequency, acquisition window, moving average(on/off), zero suppression (on/off), Tail cancellation (on/off)

9. Drift velocity (Time Stamp), Attenuation loss ( TimeStamp)

10. Alignment

11. Laser tracks

### 1.3 Calibration entries

TPC calibration information will be generated by calibration classes running in DAQ and HLT Detector Algorithms. Each calibration class might generate several calibration objects, as outlined in figure 1 and table 1. Once all calibration objects are available, final calibration entries might be calculated based on the initial entries, as outlined in figure 2 and table 2. Calibration objects are generated in Detector Algorithms. Collection of calibrations and generation of final calibration entries will be performed by the shuttle preprocessor.

All calibration entries will be generated by calibration classes. A given calibration class may generate several calibration objects, see details in the table below. The naming convention of the calibration classes is AliTPCCalibXXX, where XXX gives the calibration task in question (Pedestal, Pulser, CE, Tracks, LaserTracks etc.) The calibration objects are correspondingly named tpccalibXXX.

## 2 Calibration procedure

All calibrations are calculated based on measured data using the standard TPC readout chain. Pedestals and noise are generated using special "black"

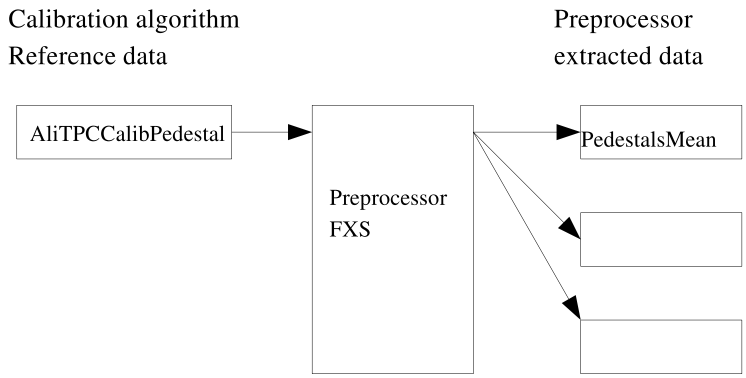


Figure 1: Preprocessor reference data

triggers, where a signal is generated in all readout pads. Such triggers are collected in special runs, identified by RunType == PEDESTAL. The pedestal/noise values are not expected to change during a physics run. The maximum frequency of pedestal runs is one such run before each physics run, once experience on the stability of the pedestal/noise measurements is obtained, it may be decided to reduce this frequency.

Pulser triggers are used to measure the performance of the readout electronics. A special pulse is given to the gating grid, causing readout from all pads. The performance of the electronics is not expected to change during the physics run, and pulser triggers are also taken in special runs, identified by RunType == PULSER.

The drift velocity of the TPC is monitored by measuring signals generated by laser pulses at the Central Electrode (CE). The drift velocity depends on environmental parameters (temperature, pressure etc.) and may change during the physics run. The laser triggers are therefore produced at fixed intervals during the physics run, identified as a LASER_EVENT in the trigger mask.

## 2.1 OCDB Calibration entries

Based on the calibration objects described above, final OCDB calibration entries will be generated by the TPC Shuttle preprocessor. The OCDB calibration entries will be used to correct TPC raw data for offline processing.

The Pedestal and PadNoise entries will be regnerated each calibration

| Calibration class AliTPCCalibXXX | System | Reference data | | OCDB entry | |
|---|---|---|---|---|---|
| | | name | size | names | size |
| Pedestal | DAQ, HLT | tpcCalibPedestal | 107.5 MB | pedestalMean<br>pedestalRMS | 2.2 MB<br>2.2 MB |
| Pulser | DAQ | tpcCalibPulser | 538.2 MB | pulserTmean<br>pulserTrms<br>pulserQmean | 2.2 MB<br>2.2 MB<br>2.2 MB |
| CE | DAQ | tpcCalibCE | 538.2 MB | CETmean<br>CETrms<br>CEQmean | 2.2 MB<br>2.2 MB<br>2.2 MB |
| Tracks | HLT, Offline | tpcCalibTracks<br>tpcCalibTracksGain<br><br>tpcCalibTracksAlign | ??<br>??<br><br>?? | ClusterParam<br>PadGainFactor<br>ClusterParam<br>TPCAlignment | small |
| LaserTracks | HLT, Offline | tpcCalibLaserTracks | ?? | TPCAlignment | small |
| PIDV0 | Offline | tpcCalibPIDV0 | ?? | ?? | small |
| DCS | | | | Temperature<br>Pressure<br>GasComposition<br>Voltages | 200 kB<br>1 kB<br>1 kB<br> |

Table 1: Preprocessor reference data

Table 2: Final OCDB entries

| OCDB entry | size | Reference data | |
|---|---|---|---|
| | | name | size |
| Pedestal | 2.2 MB | PedestalMean (AliTPCCalPad) | 2.2 MB |
| PadNoise | 2.2 MB | PedestalRMS (AliTPCCalPad) | 2.2 MB |
| PadTime0 | 2.2 MB | PulserTmean (AliTPCCalPad)<br>CETmean (AliTPCCalPad) | 2.2 MB<br>2.2 MB |
| PadGainFactor | | PulserQmean (AliTPCCalPad)<br>CEQmean (AliTPCCalPad)<br>TracksGain (AliTPCCalPad) | 2.2 MB<br>2.2 MB<br>2.2 MB |
| DriftVelocity | ?? | CETmean (AliTPCCalPad or TObjArray) | |
| Attenuation | ?? | | |
| Parameters | | | |
| Temperature | 200 kB | DCS (AliSplineFit) | |
| GasComposition | | DCS (AliSplineFit) | |
| HighVoltage | | DCS (AliSplineFit) | |

run, based on data from the AliTPCCalibPedestal calibration object. The PadTime0 entry will extract data both from AliTPCCalibPulser and AliTPCCalibCE. The combined entry will be regenerated during physics run (AliTPCCalibCE), and will use information from the previous pulser run, as available in the OCDB.

The PadGainFactor calibration will require several iterations, and will be carried out by a standalone calibration procedure, not being part of the DA/Shuttle framework. The resulting calibration entry will be valid for a long time frame, and the produced data base entry will be available for the quasi-online reconstruction.

# 3 Calibration in the AMORE framework

## 3.1 General overview

Calibration will run in DAQ and HLT Detector Algorithms. Each of these will produce a series of calibration classes. The calibration classes will contain functionality to produce histograms, trees and time-dependent graphs to be fed into AMORE. In general histograms will be used for the automatic monitoring, and trees will provide input for interactive expert monitoring.

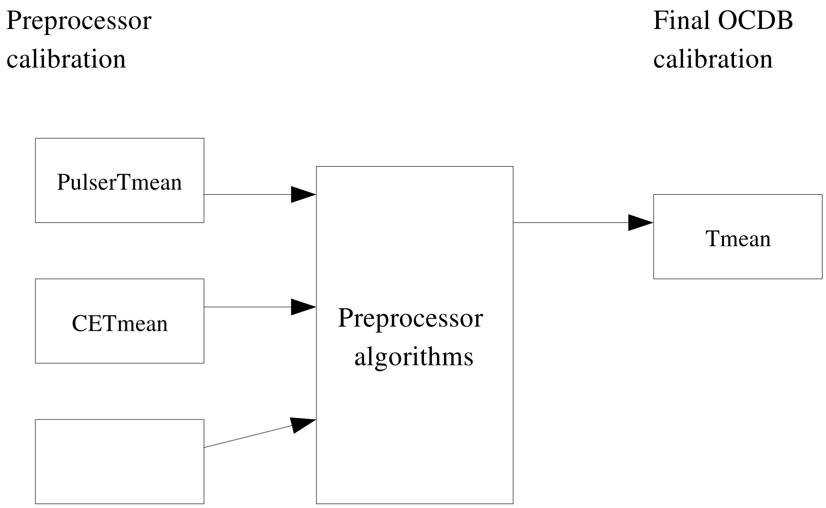


Figure 2: Final calibration data

Both histograms and trees will be wrapped in monitoring objects[1] before being submitted to AMORE.

The AMORE framework will take care of collecting subtrees from various DAs to produce a full tree for the expert monitoring. It will be decided later whether this collection will take place continuosly or only when triggered by a request from the Expert monitor. Reference trees and the most recent tree will be available at any time, but previous trees will normally not be stored. It might be of interest to allow AMORE alarms to trigger storage of the current tree to some intermediate storage. The data flow to AMORE is illustrated in the figure 3.

## 3.2 Histograms

We plan to generate 7 sets of histograms, for pedestals, noise, gain, t0, width, gain with Central Electrode and drift velocity from Central Electrode signal.

[1]Encapsulating calibration objects originating from the HLT DAs generates problematic dependencies in the current setup. It will be necessary to find a scheme to communicate the relevant objects with a minimum of induced overhead, for instance by including lightweight classes to generate these objects in AliRoot STEER.

Each of these signals will be histogrammed sector by sector (each sector is divided in two read-out chambers, so altogether this makes 72 2-dimensional histograms). For each signal there will be three set of histograms: pad-by-pad histograms (2-dim), 1-dim histograms of values (mean value, median, LTM, fraction of outliers) and 2-dim profile histograms (mean value per pad row), altogether 2*72*7 2-dimensional histograms and 4*72*7 1-dimensional histograms. These histograms should be automatically monitored/compared to reference histograms, and "status" or "quality" histograms could be generated based on these comparisons.

We would also like to monitor the phase of each readout partition, i.e. one histogram per event with 216 bins corresponding to the readout partitions. Other variables to be monitored are the drift velocity and gain parameters as function of time. Here we will have 72 (chambers)x3(fit parameters)x2(drift, gain) graphs value versus time.

All histograms can be generated from TPC calibration classes (AliTPCCalPad). To get Median, LTM , Mean, RMS, outliers list, we have functionality inside TPC calibration classes. Selected histograms will be accumulated as histograms of differences with respect to reference trees.

In addition to this we will have histograms based on cluster and track information. These histograms will be generated by the HLT, and submitted to DAQ through the HLT/DAQ interface (HLT appears as a "subdetector" in DAQ).

## 3.3 Expert monitor

The expert monitor will read full trees to be collected in AMORE, based on subtrees generated by the calibration classes obtained from DAQ and HLT DAs. The expert monitor will give interactive access to the full calibration information to allow for efficient detector problem resolution.

## 3.4 Histogram sizes

The size of 2D histograms is on the level of 2 MBy per TPC side. 7x2 MBy 14 MBy. The size of other histograms is negligible in comparison with this one. Total size will be around 20MBy. If we want to compare results with reference histograms we should multiply it by 2 40 MBy.

## 3.5 Refresh frequency

Entries for the calibration histograms should be generated whenever a laser trigger or a calibration pulse trigger occurs. The frequency of these triggers have not yet been decided, but they will not exceed 1 Hz and 0.01 Hz respectively. Depending on the signal, 10-100 such triggers are necessary to obtain a reliable histogram which could be compared to reference histograms. Comparison should happen continuously once these numbers of calibration triggers are reached. This would lead to the following maximum repetition times: gain, t0, width: 0.001Hz. Pedestal and noise information will be collected from special triggers taken at the beginning and the end of each run. At the periods these triggers are activated, they will occur with a frequency of 0.1 Hz. Comparison to reference histograms will be made at each end-of-run.

The parameters drift velocity and gain will be calculated from special laser triggers, with a frequency of about 0.01 Hz. These calibrations will be handled by HLT. The drift velocity calibration might be updated for each such trigger, whereas the gain calibration will be updated when about 1000 tracks are recorded (which will correspond to several times per run).

## 3.6 Time dependence

Graphs will be generated containing amplitude vs. time, drift velocity vs. time. These will be based on pedestal/noise measurements done at the end of each run. We will also record baseline and noise vs. time.

## 3.7 Global events

We need access to global events to monitor drift velocity and gain using CE. We assume that the signal coming from the CE and readout by the different readout partitions (RCUs) is synchronous with either the sampling clock or the 40 Mhz clock. In the latter case, we need to know the phase of the trigger with respect to the sampling clock. Other calibrations can be done patch by patch, i.e. on the LDCs. Calibration tasks run on HLT will need access to ITS and TRD information, but this communication will be handled internally in the HLT system.

## 3.8 Event reconstruction

All calibration tasks that need access to reconstruction will be performed by the HLT, and use HLT reconstruction.

### 3.9 External access

We need access to reference histograms and trees. We will also need access to temperature and pressure graphs from DCS. These graphs will be accumulated using HLT, and will be forwarded to AMORE as part of the HLT calibration classes.

### 3.10 Critical errors

Discovery of wrong phase, intensity of laser below fixed limit, missing patches/sectors would severely affect the quality of the data, and should trigger alarms.

## 4 Quality assurance

Data quality monitoring is based on monitoring of statistical properties of the data. A big fraction of the properties to be monitored are extracted during the calibration procedure. The QA procedure will evolve in time together with the further development and tuning of the calibration algorithm.

We consider two modes of the QA algorithms:

- Tuning phase (mainly expert mode)
- Standard operation

The expert mode of QA will be particularly important during the commissioning phase of ALICE. The main additional functionality implemented in the expert mode is the possibility to generate statistical graphs using correlation with other variables and make custom selections. These will help us to better understand the processes in the detector, and make the transition to the standard operational mode faster.

### 4.1 Tuning phase

Detect the problems. Define, what is the problem

What do we expect? Defined in the TPC TDR and in the PPR on the basis of simulation How far we are from the expectation? Modify expectation.

Until which point the information from the detector is reasonable? Define the limits of working conditions. Up to which point the physics performance will not be influenced. What impact the observed imperfection could make on physics performance.

In the following we put focus on these topics.

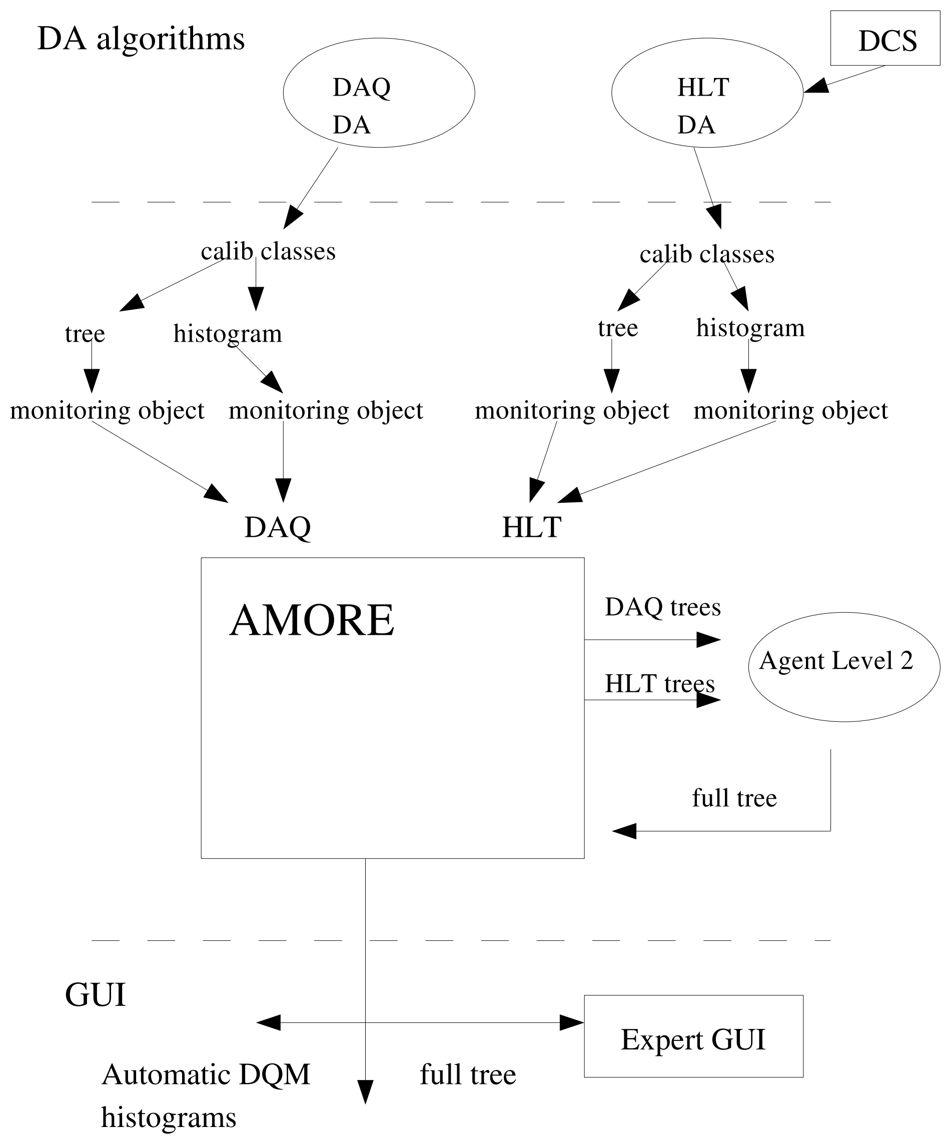


Figure 3: AMORE calibration flow for the TPC

| noise(ADC) | 2 | 3 | 4 |
|---|---|---|---|
| space resolution worsening factor | 1.05 | 1.15 | 1.3 |

### 4.1.1 Quality assurance - pedestal and noise

According to the TPC TDR the electronic noise of Alice TPC was designed to be on the level of 1000 e, which correspond to 1 ADC channel (TPC TDR). The most probable signal to noise ratio was about 20 in inner chambers and $\sim$ 30 in the outer chambers. In such a set-up the space resolution and the dEdx resolution is determined by other stochastic processes like diffusion and angular effects. E.g for space resolution the diffusion component is of the order of $\sim$ 0.7 mm while the noise component $\sim$ 0.2 mm. The spatial resolution can be affected by lowering the gain or increasing the noise level. E.g increasing the noise by factor of 3, the mean space resolution will be worsened by a factor $\sim$ 15%.

The requirements for pedestal knowledge are determined mainly by dEdx measurement. The relative dEdx resolution in the Alice TPC is on the level of 5%. In order to know the pedestal on the level of 1 % of the most probable signal (  10-20 ADC channel), the required precision and stability of the pedestal should be on level of 0.1 ADC. The pedestal and noise will be measured before each physics run. The experience from TPC tests in 2006 indicates that such a frequency is sufficient.

The TPC test in 2006 showed that the mean observed noise in the TPC is even better than the original requirement ($\sim$ 0.7 ADC in IROC). Problems were encountered at the edges of the chambers, which are more sensitive to noise induction. As a consequence the noise can be much higher at the edges and the noise distribution can be highly non-Gaussian. Therefore, some robust statistic should be used for its estimation.

Input data:

- TPC raw data without zero suppression preprocessed by the calibration algorithm AliTPCCalibPedestal.
- AliTPCCalibPedestal produce the noise and pedestal maps.
- The noise maps (AliTPCCalPad) - current and reference.
- The pedestal maps (AliTPCCalPad) - current and reference.

Histograms and graphs to be monitored:

- Noise and pedestal distribution for each sector.

- Cumulative noise distribution.
- Distribution of the differences between pedestal and noise from reference runs.
- The median of the noise distribution as a function of sector
- The median of the noise distribution as a function of time

All these can be generated by AliTPCCalPad. Moreover in the expert mode using the trees, selections can be be made applying user defined cuts .

Observables to be checked:

- Mean and median of the noise distribution - Alarms on median
- The fraction - p0 of "non usable" channels - The noise bigger than a threshold th0 (e.g 4 ADC)
- The fraction - p1 of "suspicious" channels - The noise bigger than a threshold th1 (e.g 2 ADC)

#### 4.1.2 Quality assurance - not responding channels

The Alice TPC consists of 159 pad rows. Signals from these pad rows are used in order to extract the properties of the tracks. The resolution of the variables are scaling as square root of the number of used measurement points. By simple scaling, the absence of 10 % of the channels lead to 5 % deterioration of the performance.

There are following reasons:

- Electronic problems. (e.g missing contacts)
- Single event upset.
- Data corruption during data readout.

The fraction and maps due to the latter two reasons can change on a time scale smaller than one run.

Otherwise we consider changes on the run level. The special pulser will be used to generate the dead map channel maps.

To eliminate or reduce the fraction of not responding channels due to data corruption, the decoding algorithm should be made robust enough, and minimal amount of channels (digits) should be skipped in case of the detection of data corruption.

Missing channels due to the single event upset should be on negligible level (much below % level)

The results for the TPC test 2006 indicates the amount of the dead - not responding - channels on the level below 1 per mile.

To produce the dead channel map the output of the pulser calibration can (will) be used (AliTPCCalibPulser).

Input data:

- Raw data sets with pulser ==> AliTPCCalibPulser
- The amplitude maps (AliTPCCalPads)
- The time maps (AliTPCCalPads)

The typical dispersion of the electronics gain is on the percent level. Results from the TPC test in 2006 show some fraction of the outliers with significantly higher response. Such outliers are grouped close to the pulser connectors. There were no observation of outliers in other directions, except of the pads not responding at all.

Histograms and graphs to be monitored:

- The amplitude distribution for each chamber and each pad geometry.
- Graphs of the median of the amplitude distribution.

Observables to be checked:

- The fraction of the pads with signal below p1 ratio of the median (for given pad type).

# 5 HTML Documentation

**AliTPCCalibPedestal**
http://aliceinfo.cern.ch/static/aliroot-new/html/roothtml/AliTPCCalibPedestal.html

**AliTPCCalibPulser**
http://aliceinfo.cern.ch/static/aliroot-new/html/roothtml/AliTPCCalibPulser.html

**AliTPCCalibCE**
http://aliceinfo.cern.ch/static/aliroot-new/html/roothtml/AliTPCCalibCE.html