

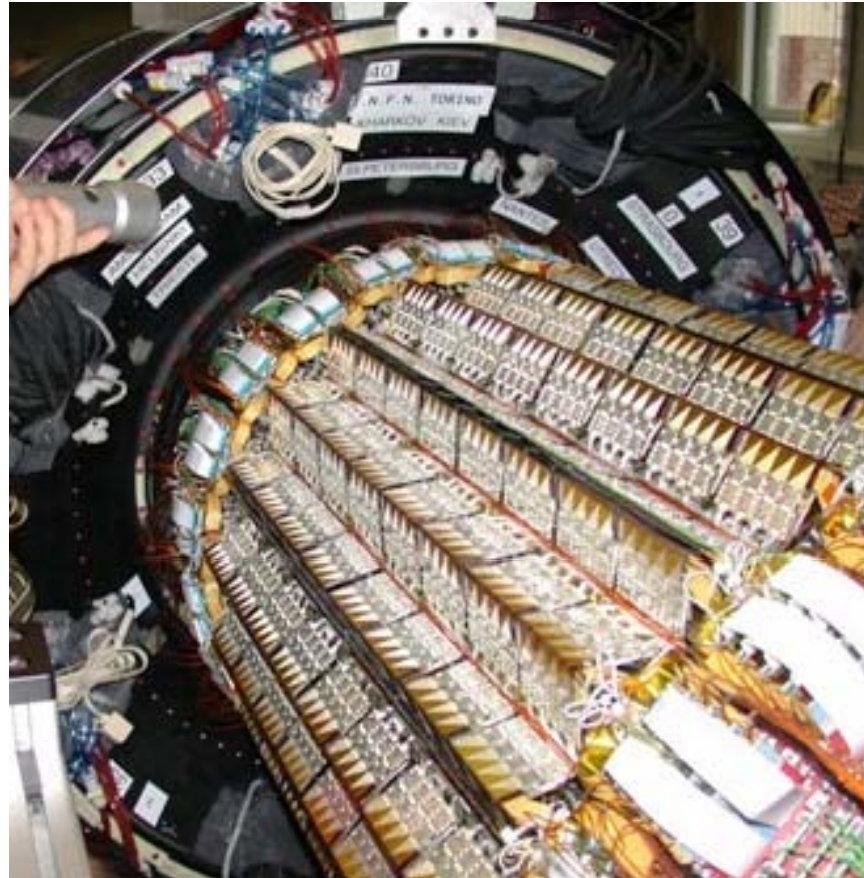
# ***SDD Offline Status***

Francesco Prino  
INFN Sezione di Torino

ALICE OFFLINE WEEK - October 2007

# Summary

- Raw data reader
- DAQ-DAs
  - ⇒ Code status
  - ⇒ Example plots
- Preprocessor
  - ⇒ Recent updates
- Calibration
  - ⇒ Residual maps
  - ⇒ Drift velocity
- Cluster Finder
- Next steps



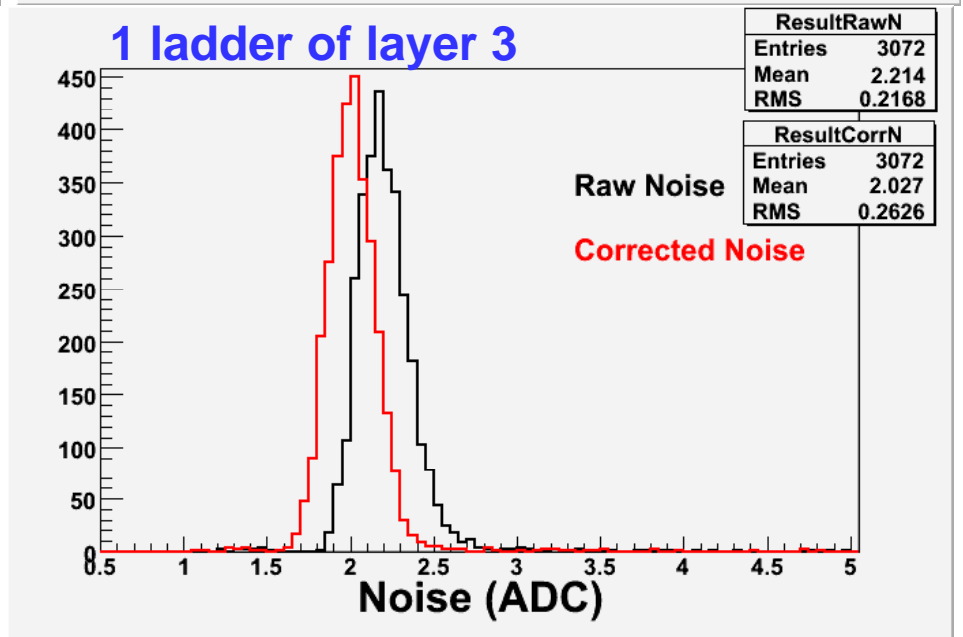
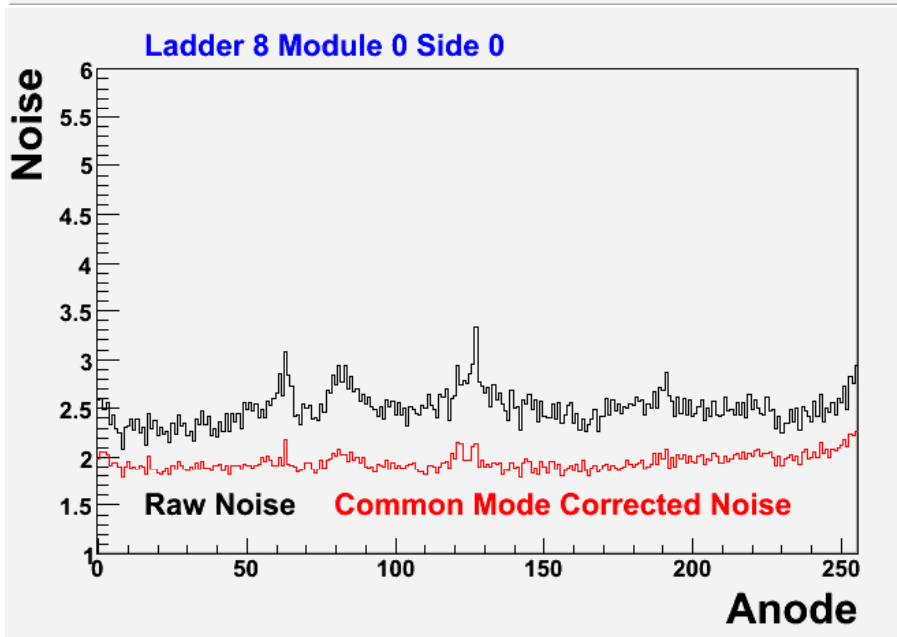
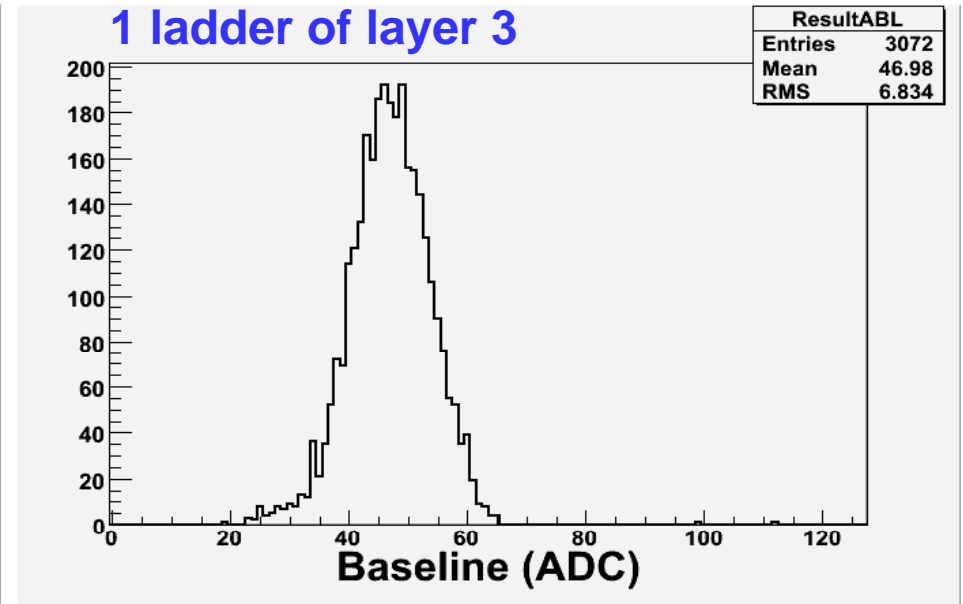
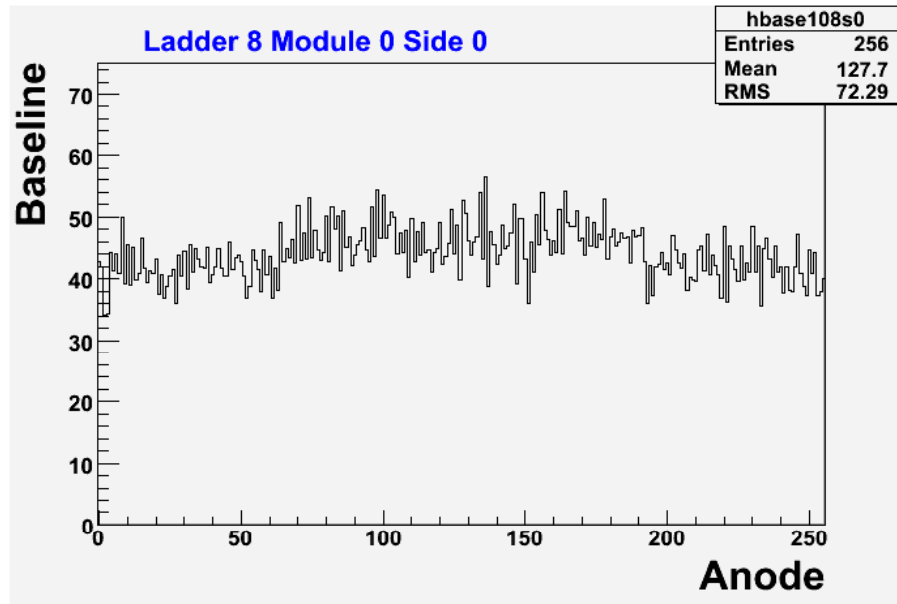
# *AliITSRawStreamSDD*

- Code significantly upgraded in the last 2 months by P. Cerello, F. Prino and M. Siciliano
  - ⇒ Adapted to evolution of raw data format
  - ⇒ Few bugs (mostly related with "rare events") fixed
  - ⇒ Code cleaned and now in "readable" shape
  - ⇒ Presently being used as the default reader for the analysis of test data which are being collected at Point2
- Simulated raw data format updated accordingly
- Main concern:
  - ⇒ The acquisition presently writes "in parallel" the data of the 12 modules of each DDL to reduce dead times
    - ✓ *Writes N words with data from a given module, then N' from another module and so on*
  - ⇒ BUT the ClusterFinder can start to find the clusters on a given module when the reading of its data is completed
  - ⇒ SO: with the present raw data format it is necessary to keep in memory the data and the calibration constants of 12 modules at the same time
    - ✓ *Acquisition experts have been triggered ...*

# DAQ-DA for SDD

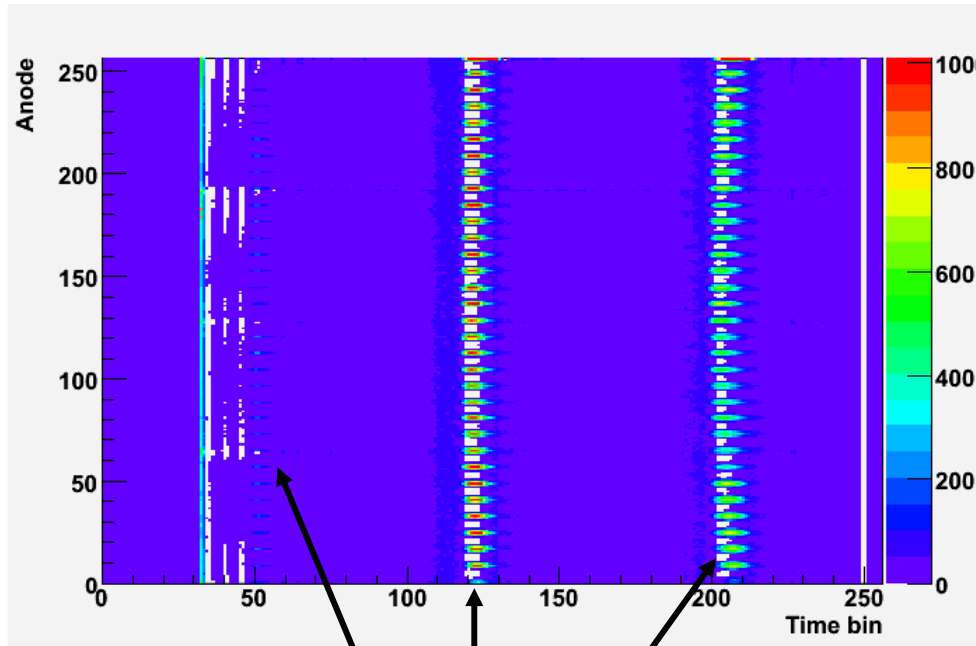
- 3 DAs to extract SDD calibration parameters
  - ⇒ Presently being used by W. Ferrarese and M. Siciliano on SDD test data
  - ⇒ Several upgrades aimed at easy graphical output for QA
- ITSSDDBASda.cxx
  - ⇒ Analyzes special SDD calibration runs taken without zero suppression during LHC fill periods (every  $\approx 24$ h)
  - ⇒ Provides: **Baselines, Noise, Common Mode Corrected Noise, Noisy anodes**
- ITSSDDGAINda.cxx
  - ⇒ Analyzes special SDD calibration runs taken with Test Pulse signal to front-end electronics during LHC fill periods (every  $\approx 24$ h)
  - ⇒ Provides: **Anode gain, Dead anodes**
- ITSSDDINJda.cxx
  - ⇒ Analyzes injector events collected every  $\approx 10$  min. during physics runs
  - ⇒ Provides **Drift Velocity** (anode dependent)

# SDD DAQ-DA: Example plots (I)



# SDD DAQ-DA: Example plots (II)

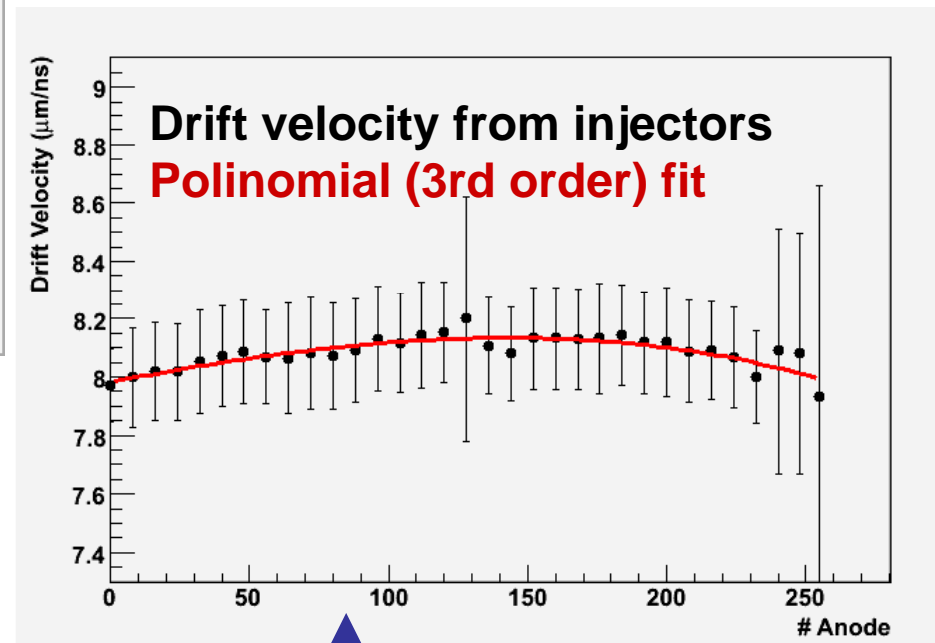
## • Injectors



3 lines of 33 MOS injectors  
integrated on the SDD sensor at 3  
different distances from anodes



Determination of drift velocity in 33  
different positions along anode coordinate



# *AliITSPreprocessorSDD*

- AliITSPreprocessorSDD updated on September 28th to read:
  - ⇒ 520 ASCII files with baselines, noise and gain produced by ITSSDDGAINda.cxx
  - ⇒ 520 ASCII files with coefficients of polynomial fit to drift velocity vs. anode number produced by ITSSDDINJda.cxx
  - ⇒ A set of simulated files grouped in 8 tarballs is available on /afs/cern.ch/user/f/frprino/public/CalibSDD/FXS
- New version included in SHUTTLE tests since October 1st → runs successfully
- To be done:
  - ⇒ Read parameters from DCS (used only for reference)

# *AliITSresponseSDD*

- **Added Time-Offset constant**

- ⇒ Parameter to account for delays in electronic chain

- ⇒ It was hard-coded in AliITSClusterFinderV2SDD

- ⇒ New data members

```
Float_t fTimeOffset;  
static const Float_t fgkTimeOffsetDefault;
```

- **Added charge conversion constant**

- ⇒ Convert charge signal from ADC to keV

- ⇒ It was hard-coded in AliITSClusterFinderV2SDD

- ⇒ New data members

```
Float_t fADC2keV;  
static const Float_t fgkADC2keV;
```



# AliITSCalibrationSDD (I)

- Residual maps

- ⇒ Matrices with systematic deviations on coordinates of reconstructed points

- ✓ *Systematic effects due to non-constant drift field or dopant inhomogeneities*

- ✓ *Measured for all SDD modules by means of laser scans*

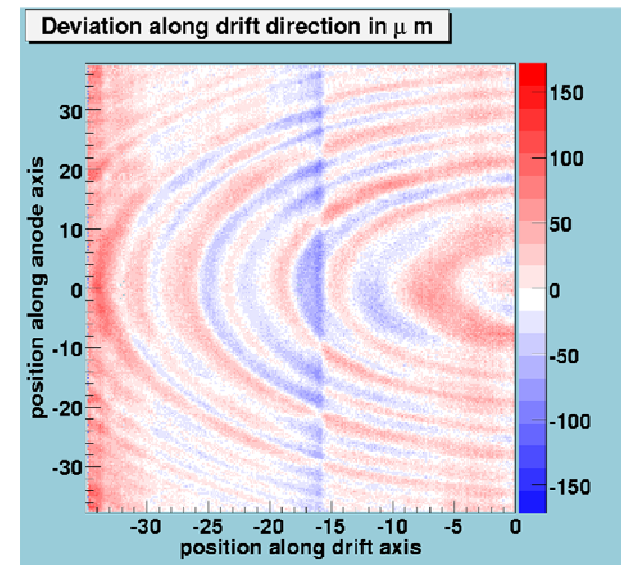
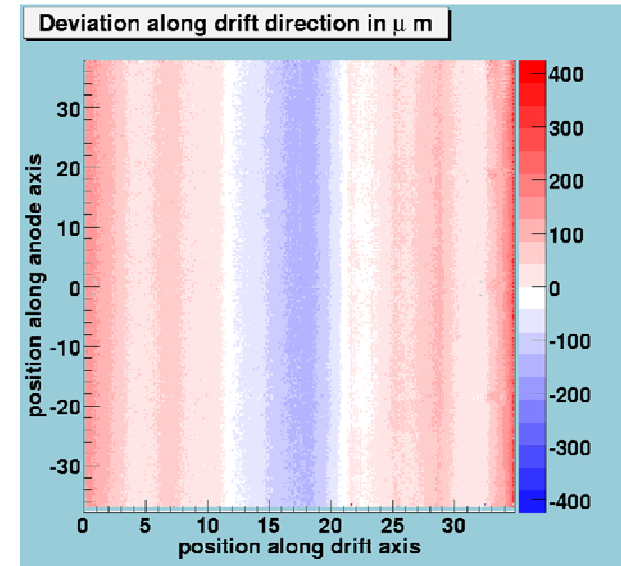
- ⇒ 4 Maps (2 sides x 2 coordinates) for each module

- ⇒ Long-lived calibration objects (validity = infinite)

- New in OCDB: maps separated from other (short-lived) objects (on CVS since September 5th)

- ⇒ New class AliITSMAPSDD

- ⇒ Four pointers to AliITSMAPSDD as data members of AliITSCalibrationSDD



# AliITSCalibrationSDD (II)

- **Drift Velocity:**

- ⇒ Anode dependent drift velocity added in OCDB files

- ⇒ New data members

```
Float_t fDriftVelParW0[4];
```

```
Float_t fDriftVelParW1[4];
```

with the coefficients of the 3rd order polynomial fit to drift velocity vs. anode for the 2 detector sides

- ⇒ New method

```
Float_t GetDriftSpeedAtAnode(Float_t nAnode)
```

to compute drift velocity using the polynomial function

- ⇒ On CVS since September 28th

- **In progress (= code ready, being tested):**

- ⇒ Use of anode dependent drift velocity in simulation and reconstruction

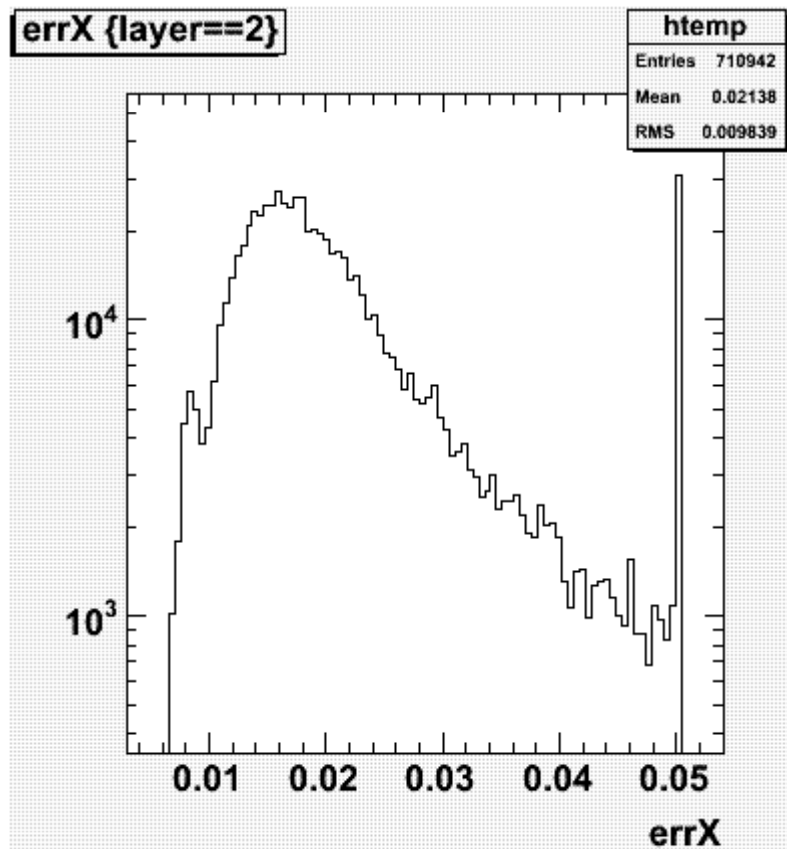
- ✓ *Modifications in cluster finders and in AliITSSimulationSDD*

# AliITSClusterFinderV2SDD (I)

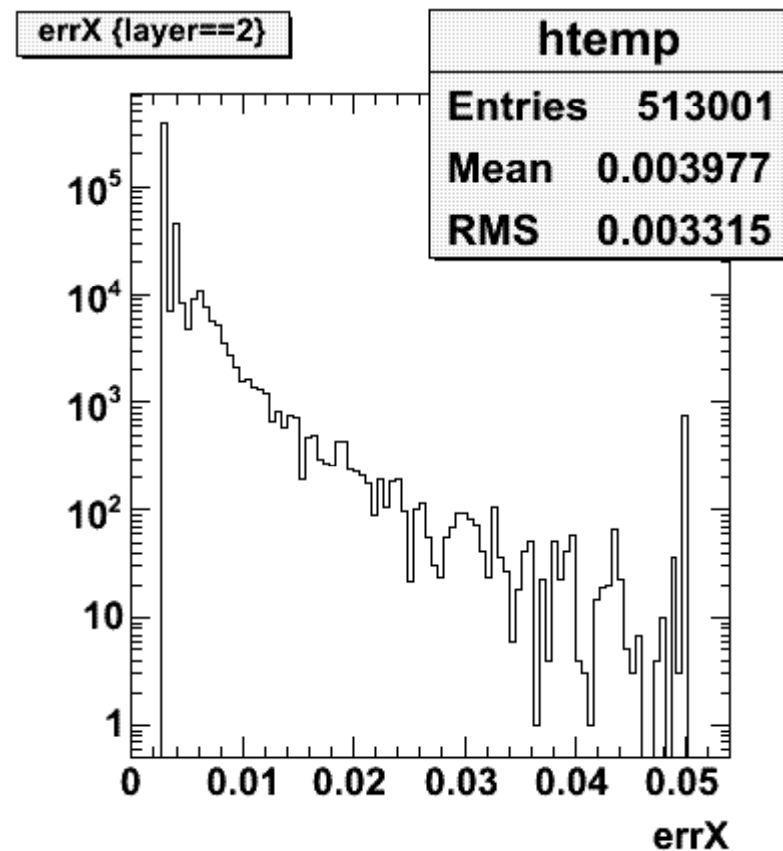
- Problem with errors on SDD clusters discovered by Andrea Dainese when debugging ITS tracking
  - ⇒ Errors on SDD clusters given by `AliITSTrackerMI::GetError()` larger than the nominal SDD resolution by a factor  $\approx 10$
- Origin(s) of the problem:
  - ⇒ SDD charge converted from ADC to keV on April 3rd 2007 without modifying the corresponding cut values (hard coded) in `AliITSTrackerMI`
    - ✓ *Charge in SDD clusters no longer matching TPC track charge ⇒ SDD cluster errors enlarged (by a factor  $\approx 3$ ) by the tracker*
  - ⇒ Bug in the calculation of the cluster size along anodes introduced when removing hard coded numbers in `AliITSClusterFinderSDD` (June 1st 2007)
    - ✓ *SDD cluster size very large (always = 5 anodes) ⇒ SDD cluster errors enlarged (again by a factor  $\approx 3$ ) by the tracker*
- Problems now solved (on CVS since September 27th)

# AliITSClusterFinderV2SDD (II)

- Distribution of errors on SDD clusters given by AliITStrackeMI::GetError() (from A. Dainese)



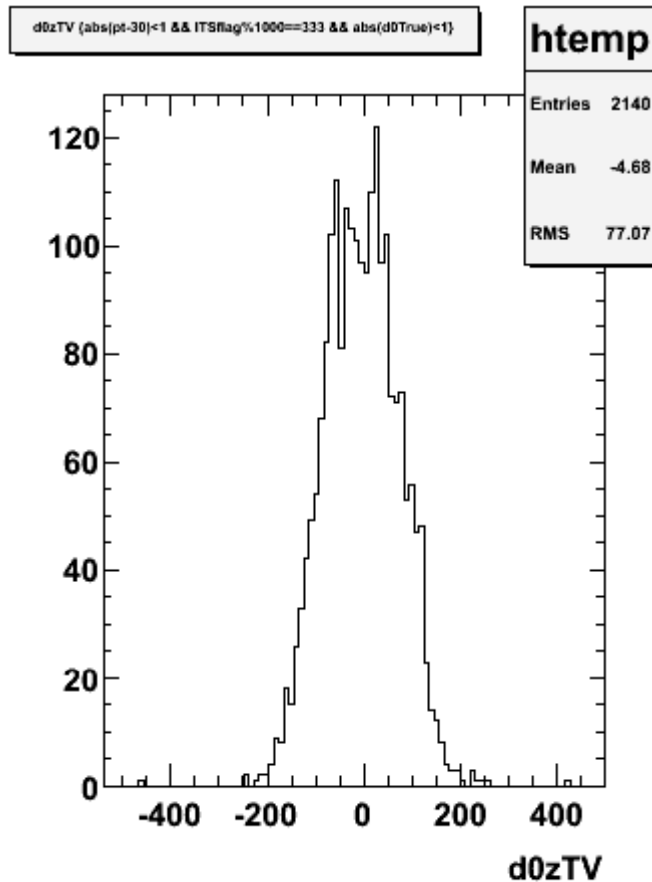
*AliRoot HEAD August 2nd*



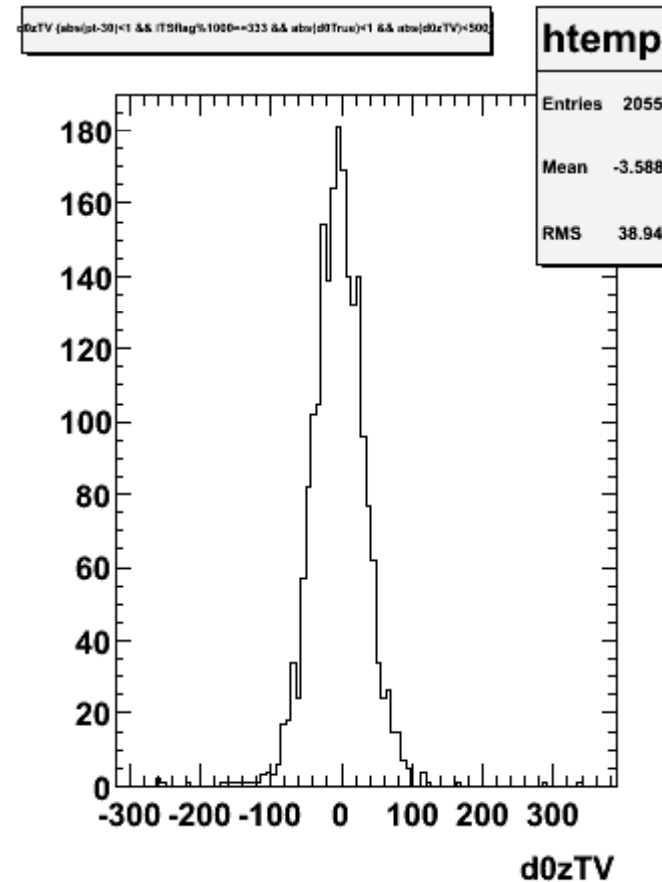
*AliRoot HEAD September 27th* 12

# AliITSClusterFinderV2SDD (III)

- Distribution of track impact parameter along z (from A. Dainese)



*AliRoot HEAD August 2nd*



*AliRoot HEAD September 27th* 13

# AliITSClusterFinderV2SDD (IV)

- In progress (= code ready, being tested):
  - ⇒ Reconstruction of events simulated with anode dependent drift velocity
- To be done:
  - ⇒ Check memory consumption with reconstruction of real data
    - ✓ *Possible necessity to keep in memory data and calibration constants of 12 modules at the same time*
    - ✓ *Possible reduction of memory used by correction maps: for several modules a 1D correction histogram (instead of a 2D map) is enough*

# Next steps

- Request from Raffaele Grosso to remove the call to OCDB in geometry instantiation ( `gAlice->Init()` )
  - ⇒ Presently not possible for SDD because `AliITSsegmentationSDD` needs the calibration to get the drift velocity
  - ⇒ Possible strategy:
    - ✓ *Remove the drift velocity in `AliITSsegmentationSDD`*
    - ✓ *Modify methods `GetPadCxz`, `GetPadIxz`, `LocaToDet`, `DetToLocal` ... so that for SDD they get/return one space coordinate and one time coordinate (instead of two space coordinates)*
  - ⇒ Code implementation will start soon, but requires significant modifications in a few classes and deep testing (expected in  $\approx 2$  weeks)
- Devise a strategy for drift velocity de-calibration
  - ⇒ Possible solution: use a different set of calibration files in simulation and reconstruction