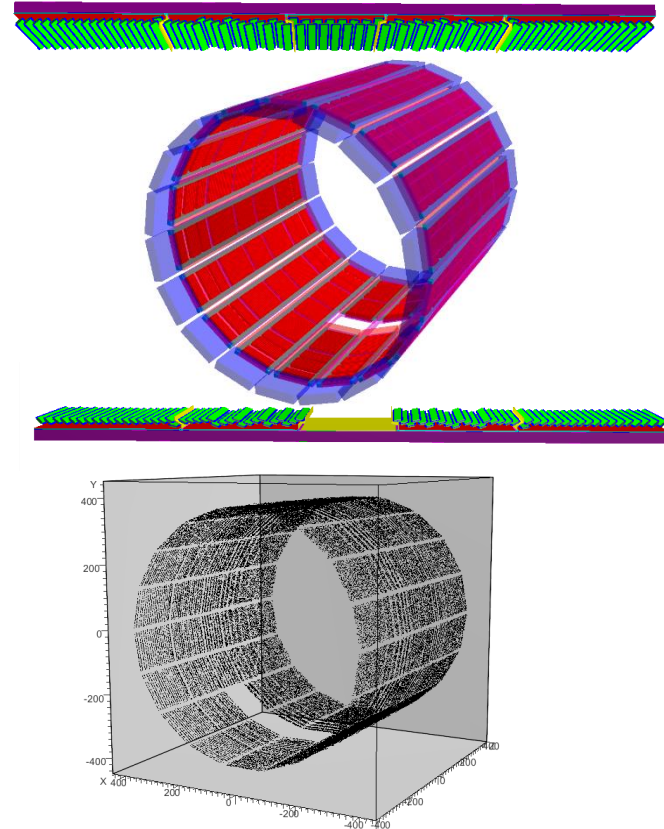# TOF in O2: status report

F. Noferini, C. Zampolli

Our Schedule (*simulation*)

Simulation (hits) -> Digitization (digits) -> Clusterization (clusters) -> Matching (track infos)

- TOF Geometry [done]
  - Definition of the volumes hosting TOF [done]
  - Definition of TOF volumes, materials, alignable object [done]

- TOF hit creation in simulation [done]
  - Hit data structure [done]
  - Propagation, hit generation [done]
  - hit merging [done]

- TOF digitization [done]
  - Digit data structure [done]
  - Signal parameterization [done]
  - Digit merging [done]
  - Continuous readout mode [done]
  - Digit task/macro [done]
  - Adaptation to the new work flow of O2 (Data Processing Layer) by S. Wenzel [done]

Our Schedule (*reconstruction and calibration*)

Simulation (hits) -> Digitization (digits) -> **Clusterization (clusters) -> Matching (track infos) → Calibration**

- TOF clusterization [done]
    - Cluster data structure [done]
    - Clusterization task/macro [done]
- TOF matching [done]
    - Loading ITS-TPC tracks [done]
    - Matching data structure info [done]
    - Matching algorithm [done]
    - Matching task/macro [done]
- **TOF Calibration objects [done]**
    - **LHC phase trending vs time [done]**
    - **Channel offsets, problematic channels and time slewing [done]**
- **TOF Calibration procedures [ongoing]**
    - **Calibration input written during matching algorithm [done]**
    - **CollectCalibInfoTOF class/macro [done]**
    - **CalibTOF class/macro [done]**
    - **Parallelization per sector [done]**
    - **Sector merging [done]**
    - **Definition of problematic channels [ongoing]**
    - **LHC phase vs time [done]**
    - **Channel calibrations [bunch crossing + within bunch crossing + time slewing]**

NEW wrt December

Next

As recently discussed (last week) we planned to work in the next months on:

- CCDB [to be done]
  - Waiting for general framework [to be done] – TOF can be one of the first use case of CCDB "standalone" ($\rightarrow$ outside DPL) usage both to read and write

- Integration of TOF matching and calibration accumulator in DPL [to be done]

- Use of FIT for determination of Bunch Crossing [to be done]

- Raw decoder/encoder [to be done]
  - Needs finalization of discussion on the raw data format (~spring 2019)

- Alignment [to be done]
  - Waiting for general framework [to be done]

- Estimate TOF reconstruction performances with HIJING events [to be done]

Typical TOF chain in simulation

- O2sim → simulation (at least ITS, TPC, TOF)

- digitizer-workflow → digitization (ITS,TPC,TOF) + TOF clust.

- tpc-reco-workflow

- root run_track_itscooked_z0.C

- root run_match_TPCITS.C

Preparing tracks for matching with TOF

- root run_match_tof.C → matching TOF/tracks **(+ calib input)**

- **root run_collect_calib_tof.C → TOF accumulator for calib**

- **root run_calib_tof.C → TOF calibration objects**

New wrt last report

Calibration objects

Two **new** classes in *DataFormats/Reconstruction/include/ReconstructionDataFormats:*

CalibLHCphaseTOF → LHC phase vs time

- std::vector<std::pair<int, float>> mLHCphase; ///< <timestamp, LHCphase>

CalibTimeSlewingParamTOF → channel calibration (similar to what exists in AliRoot), storing at the same time the timeSlewing, the offset calibration, and the information to declare a channel "problematic"

Time slewing + offset

- int mChannelStart[NSECTORS=18][NCHANNELXSECTOR=8736];   ///< array with the index of the first element of a channel in the time slewing vector (per sector)

- std::vector<std::pair<float, float>>* mTimeSlewing[18]; //! pointers to the sector vectors

- std::vector<std::pair<float, float>> mTimeSlewingSec00; ///< timeslewing correction <tot,time> sector 0

- … (for all 18 sectors) → sectors in different vectors to allow **parallelization**

To define problematic

- float mFractionUnderPeak[NSECTORS][NCHANNELXSECTOR];   ///< array with the fraction of entries below the peak

- float mSigmaPeak[NSECTORS][NCHANNELXSECTOR];   ///< array with the sigma of the peak

- float mSigmaErrPeak[NSECTORS][NCHANNELXSECTOR];   ///< array with the sigma error

CalibTOF tasks

- Matching produces the input for the calibration (tree)
  - *o2calib_tof.root (from run_match_tof)*
- Then we collect all the calibration infos ordered channel by channel (TOF accumulator task → CollectCalibInfoTOF)
  - *o2calibration_tof.root (from run_collect_tof)*
- Then we run the calibration (TOF calibration task → CalibTOF) (includes: LHCphase, TOF channel offset, TOF channel timeslewing, TOF problematic channels)
 - *o2_calparams_tof.root (from run_calib_tof)*

CalibTOF
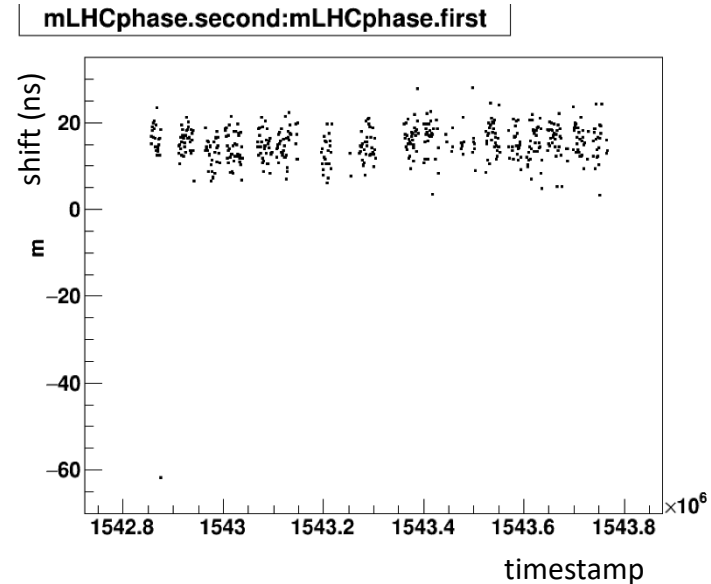
CalibTOF allows parallelization per sector

- *run* method requires sector as argument
- A forked process can run on many sectors and write its own output (i.e. o2calparams_tof_fork0.root, o2calparams_tof_fork1.root, …)
- Output can be quickly merged in o2calparams_tof.root
- Debug flag allowed (=1 or =2, =1 suggested!)

… see "macro/run_calib_tof.C"
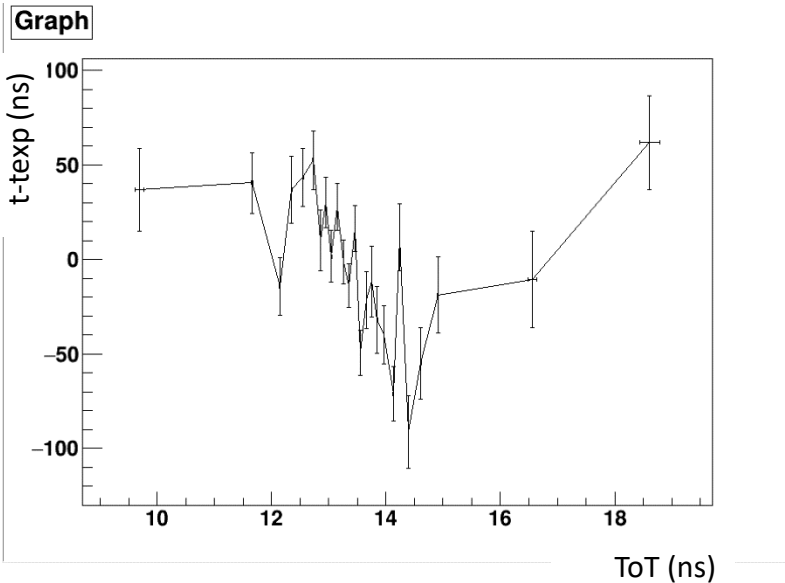
Calibration output
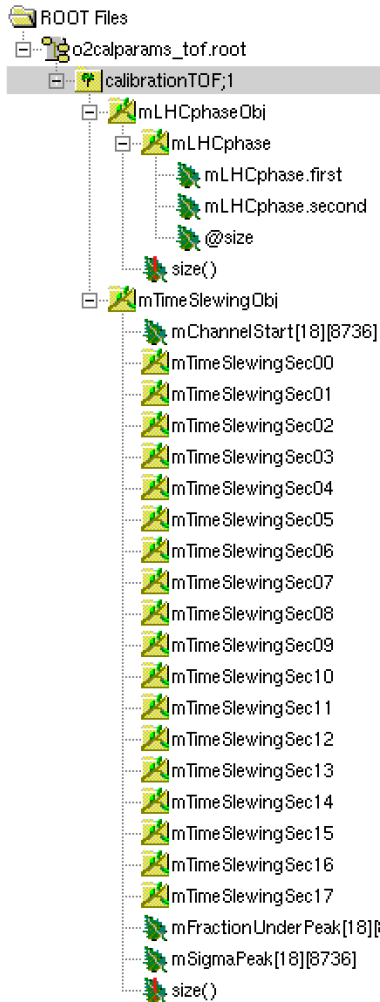
LHC phase vs time

Channel calibration parameters

The object will then be stored in CCDB

Quantities to define good/problematic channels
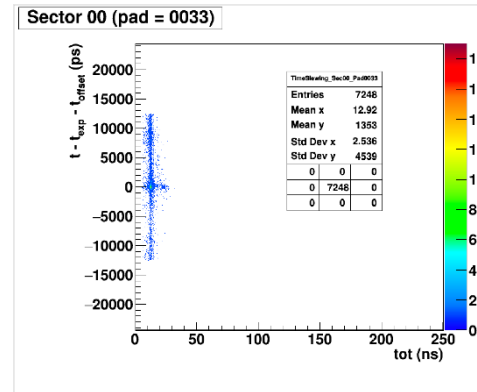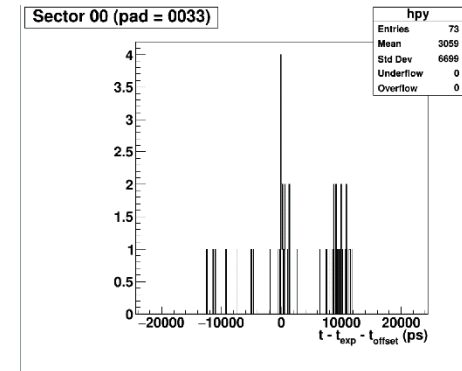
Debug mode

1. timeslewingTOFXXXXXX.root created with time slewing correction channel-by-channel

Example from:
timeslewingTOF000143.root, pad_01_56_06

2. Problematic fits stored (heavy output since they are 2D histograms)

Example from:
TOFDBG_TimeSlewing_Sec00_Pad0033_TotBins0001_0001.root

One caveat

Channel offsets are due to several effects. The contribution given by cables can be larger than a single bunch crossing (> 25 ns). We are not able at this stage to know in which bunch we were when the collision occurred (and we don't have any information on the event time) → ***will be fixed when we can use the FIT information***

At present: if we split our offsets in two contributions, the total offset is:

TOF offset = 25 ns x #bunch_offset + fine_offset

Currently we are computing only fine_offset (re-aligning all times to [-12.5, 12.5] ns) which is, in any case, the most CPU consuming component

#bunch_offset is just an integer (our idea is to compute it by looking at the fill scheme)

Validation test

We are currently using LHC18r to test the code.

In order to do that we converted our AliRoot calibration objects in an o2::tof calibration tree (11 GB)

- Single CPU test → 1.5 hours to calibrate the full LHC18r period (160k channels) from scratch in our laptops (not so bad!)

- 4 CPU test → it works but we still need to check the perfromances

We expect some optimization in the next months.

PID	User	Command	Swap	USS	**PSS**	RSS
559	noferini	/software/alice/sw/slc7_x86	0	58752	**77530**	152612
585	noferini	/software/alice/sw/slc7_x86	0	194348	**214551**	294320
584	noferini	/software/alice/sw/slc7_x86	0	194552	**215208**	295748
586	noferini	/software/alice/sw/slc7_x86	0	201960	**221877**	300856
583	noferini	/software/alice/sw/slc7_x86	0	224804	**245279**	325496
				874416	**974445 MB in total**	

WP12 - summary

D12.5	TOF simulation in O2 F. Noferini & C. Zampolli	
D12.5.1	General geometry and base classes FN&CZ	DONE
D12.5.2	Hits creation and digitization FN&CZ	DONE
D12.5.3	Simulated data compatible with Timeframe FN&CZ	DONE
D12.5.4	TOF trigger simulation FN&CZ	OBSOLETE

WP13 - summary

D13.5	TOF	F.Noferini & C.Zampolli
D13.5.1	Porting of TOF geometry to O2 FN&CZ	DONE
D13.5.2	Porting of clusterization FN&CZ	DONE
D13.5.3	TOF data decoder FN&CZ	We just started to work on this
D13.5.4	Porting of TOF matching algorithm FN&CZ	DONE
D13.5.5	Porting of calibration code and CCDB objects FN&CZ	CCDB missing (work in progress)