ITS reconstruction for HLT

J. Belikov, C.Cheshkov

OUTLINE

- Motivation
- Clusterer, vertexer and tracker
- Tracking performance: efficiency, resolution
- Timing Performance
- Conclusions and Outlook

MOTIVATION

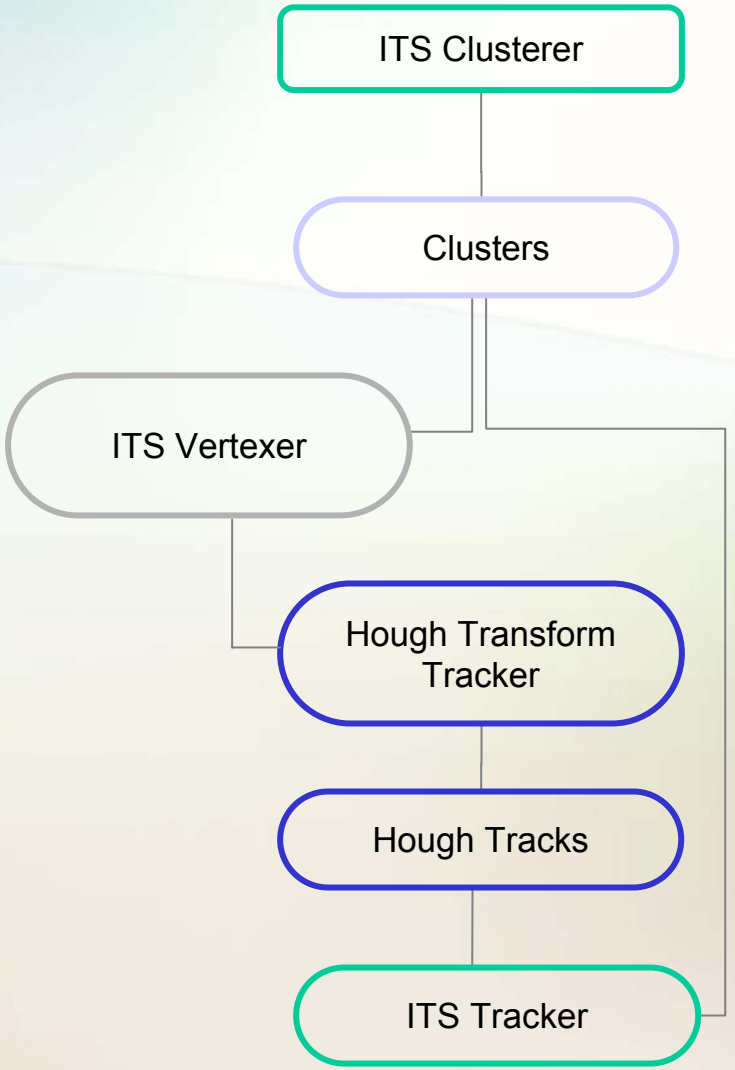
- So far HLT code only for TPC
- ITS reconstruction is:
  - Needed for open charm trigger
  - Desirable for jet analysis
  - Is required to provide the vertex to TPC Hough transform tracker

APPROACH

- We decided to use as a basis the off-line ITS reconstruction:
  - Already established algorithm
  - Robust and reliable
  - Need just to interface the HLT tracks from TPC and track them in ITS
- Is HLT tracks precision sufficient for ITS tracking?
- Which is the HLT TPC + ITS tracking performance?
- Timing performance?
- The results shown here are for the Hough Transform HLT tracks (with minor changes could be tried on Conformal Mapper tracks as well)

RECONSTRUCTION CHAIN

1. ITS clusterer
2. ITS Z vertexer
3. HLT TPC tracker (Hough Transform)
4. ITS tracker



ITS Clusterer

- Basically the off-line code with minor reorganizations:
  - Added an interface in case there is no RunLoader
  - Skipping of MC labels in case of raw data input
  - Avoided some unnecessary memory allocations (allocate the arrays once and only flush them before each new module)

ITS Vertexer

- The off-line code was modified so that:
  - Removed unnecessary intermediate step which goes from ITSclustersV2 -> ITSRecPoints
  - The clusters are split in bins of ϕ for faster access and filling into Z bins
  - The filling of Root histograms with Z bins is replaced by filling of an array of ints
  - Added interface in case of no RunLoader

ITS Vertexer

- As expected - no change in the vertex finding performance
- Resolution on Z position:

  70μm (dN/dy=2000) -> 60μm (dN/dy=8000)
- About 30 times faster compared to the off-line code (for dN/dy~4000)

ITS Tracker

- Make use of the off-line ITStrackerV2 with several modifications:
  - Clusters are sorted not only in Z, but also in ϕ
  - In Kalman filter, Root TMatrixD was replaced by explicit calculations
  - As a results the timing performance was improved by a factor of 5-10
- The tracking is done in 2 passes. One with and one without vertex constraint
- Note: Vertex constraint is applied only for cluster search and not for track params

ITS Tracking Procedure

- HLT Hough tracks contained in the ESD are transformed into AliITStrackV2 and passed to the ITS tracker
  - No dE/dx info -> all tracks assumed to be pions
  - By definition the input params are constrained to the vertex
  - The covariant matrix is filled with the averaged sigmas for the diagonal elements while all the correlations are set to 0
- As a last step in the tracking, tracks are propagated to the beam pipe and then to the vertex
- After the tracking is finished the AliITStrackV2 are transformed back to the HLT ESD tracks and stored (the vertex constraint is removed)

ITS tracking procedure

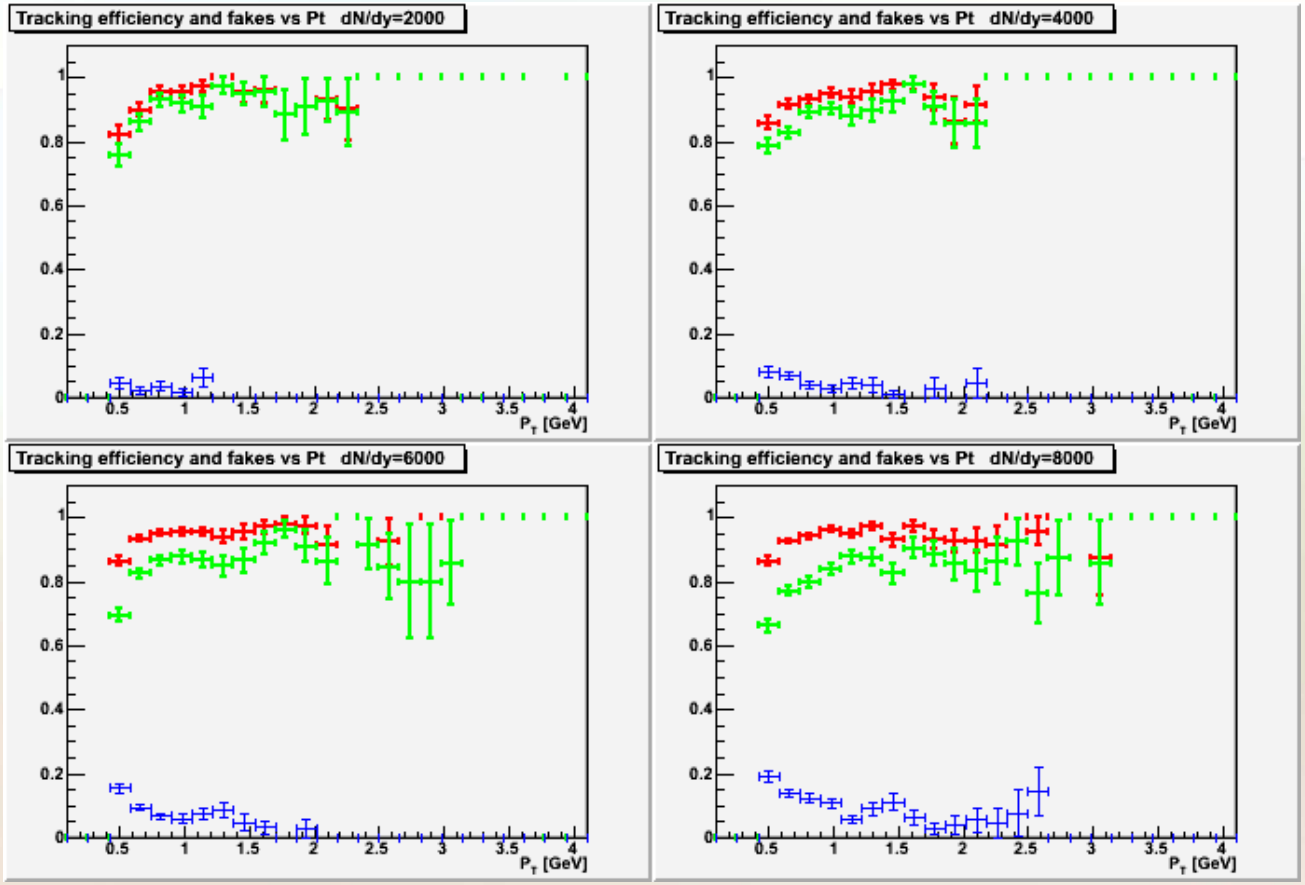
- The problem related to the covariance matrix for Hough Transform tracks:
  - So far only diagonal elements (filled with average errors taken from comparison results)
  - To do:
    - From Hough Transform we have the size of the track peaks both in Rϕ and η directions
    - Take these sizes (or fractions of them) as errors and covert them in order to fill the full cov.matrix
- It seems that even the present simplified solution works fine

Tracking Efficiency

- The overall efficiency is quite satisfactory
- ITS tracking almost completely “kills” double found Hough tracks
- Good tracks list from AliITSComparisonV2 macro
- Found tracks definition: >= 4 clusters in ITS

TPC only (HT)
ITS+TPC
Fakes

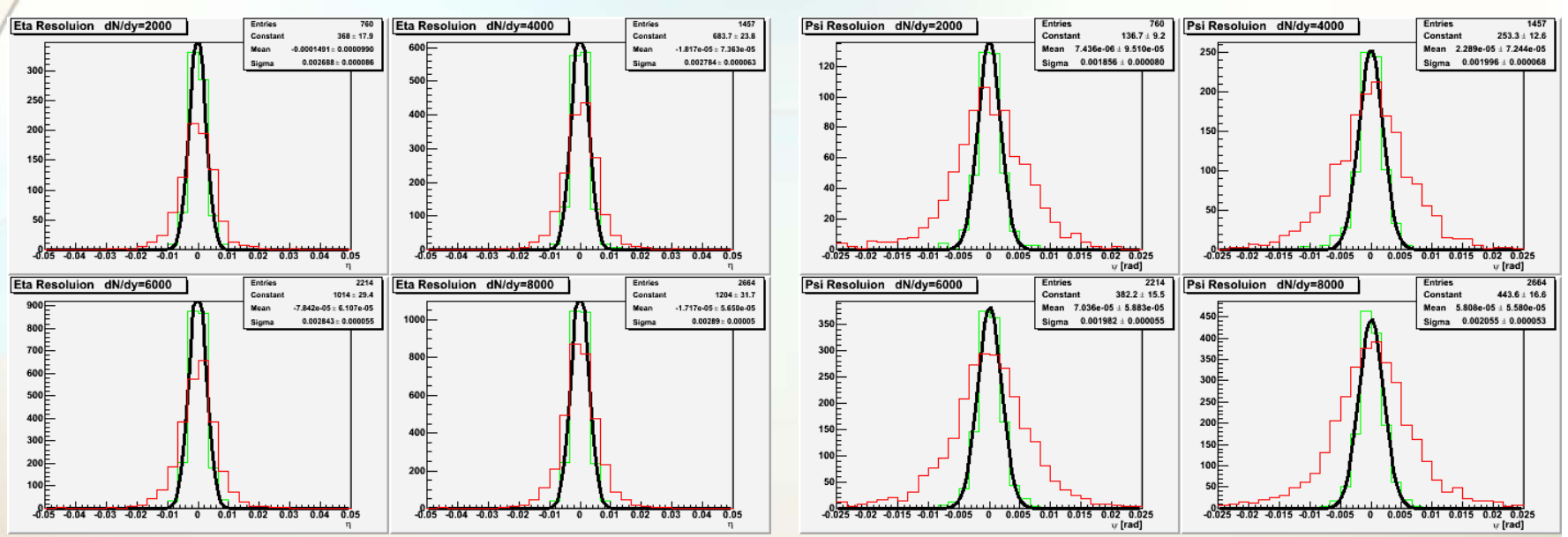
B=0.5T



Note: Definitions of Hough and ITS tracking eff are quite different. Hough eff ploted only to guide the eye.

Angular resolution

TPC only (HT)
ITS+TPC

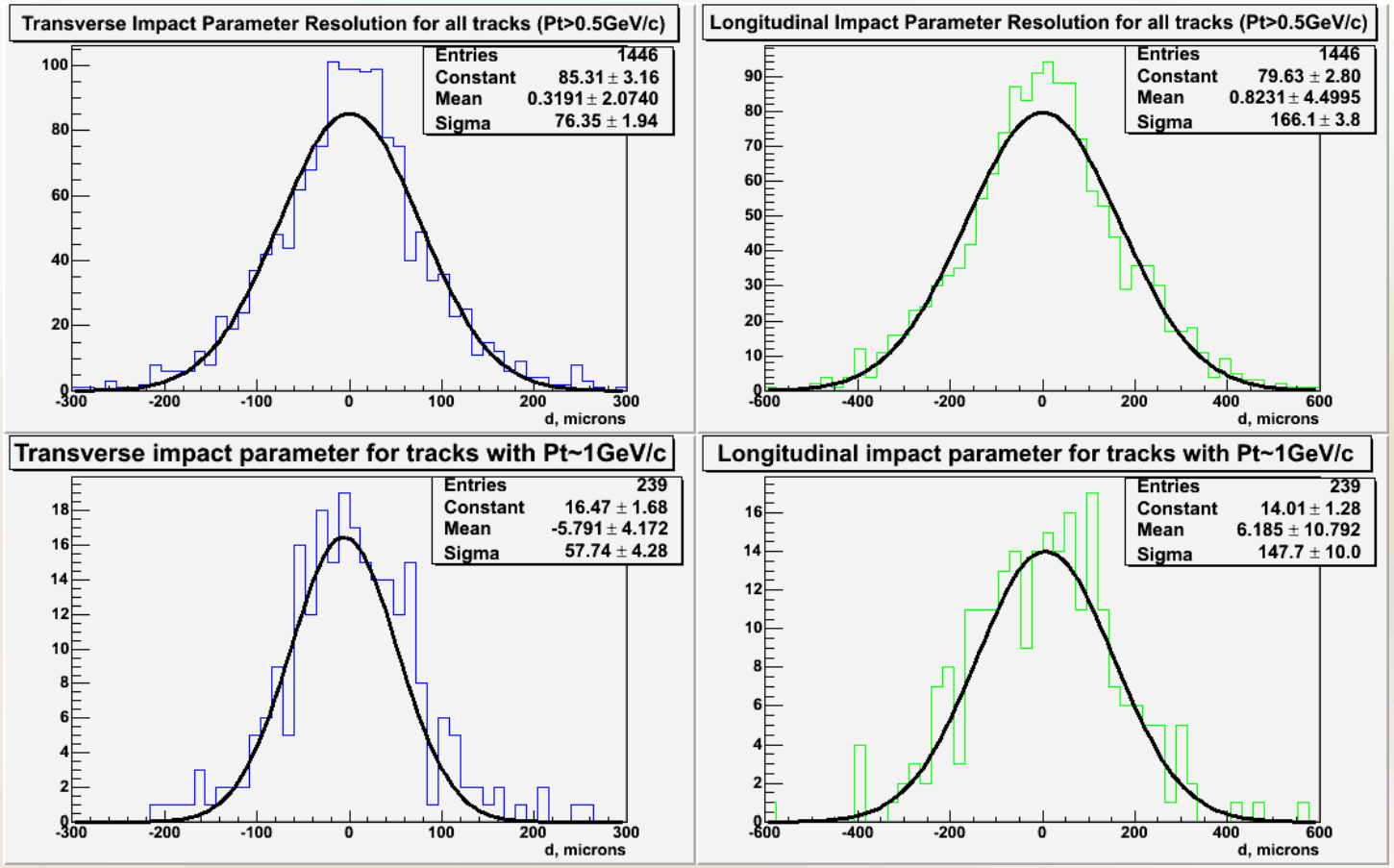


- The resolution on η and the emission angle Ψ are improved by a **factor 2-2.5**
- Resolution is dominated by ITS

$\Rightarrow$ **Very close to the off-line**

Impact Parameter Resolution

- The impact params resolution is completely dominated by SPD and therefore we get the "off-line" quality
- Example: for 1GeV/c track, the trans. impact param resolution is 60 microns.



Timing Performance

dN/dy	Clusterer	Vertexer	Tracker
~0	0.5s	20ms	0.15s
2000	1.3s	45ms	0.45s
4000	1.5s	85ms	0.95s
6000	1.75s	150ms	1.70s
8000	2.0s	210ms	2.70s

- Tests done on Intel Itanium II mashines (~1300 SpecInt's)
- Timings for the case of raw data input (no RunLoader, no MC labels)
- Still some "overheads" in the clusterer due to:
  - Filling of the clusters tree
  - Loops over all bins. Can be replaced by "jumping" method using dynamic arrays of pointers (similar to Hough space filling)

The code

- Everything is already in the HLT module inside AliRoot repository:
- One can try it by running RunHLTITS.C macro
- The macro will use an already produced AliESDs.root and will create AliESDits.root with updated by ITS tracking Hough tracks

Conclusions and Outlook

- Both tracking and timing performances of the presented ITS tracking (+Hough TPC tracking) are quite satisfactory

  $\Rightarrow$ **Fully acceptable for HLT**

- To do:
  - Further optimization of the clusterer
  - Correct cov.matrix from HT
  - Check the performance on physics channels: D0->Kπ; Λ, Ξ decays; effect on jet resolution and efficiency