# Realistic flavour tagging in Marlin

Kristian Harder, RAL

14th September 2007
SiD tracking meeting

# aims of this presentation

**my intentions are good!**

- not trying to convert you to Marlin
- not trying to show off the great stuff we have

**instead,**

- trying to survey missing pieces (in both US+European software)
- trying to encourage collaboration

# general remarks

- many ILC physics analyses will require b-/c-tagging, $q/\bar{q}$ id
- b-/c-tagging best done using secondary vertex reconstruction
- vertexing requires precise track finding+fitting
  with proper covariance matrices
- realistic studies rely on realistic detector simulation

➡ I will review the status of these components in Marlin (+Mokka)
and compare to org.lcsim

(with very limited knowledge of org.lcsim; thanks to Dmitry Onoprienko for input!)

# detector simulation

GEANT4 simulation code Mokka describes various LDC models

- modifying models not easy
  - some simple operations doable on the fly
  - others require setting up SQL database server
- vertex barrels: simple cylinders and actual ladder simulation available
  - people mostly use cylinders so far
  - testing of LCFIVertex with ladders started this week
- geometry output interface (GEAR) now complete
  - could we somehow convert GEAR and LCDD xml files?
  - are our LCIO output formats compatible?

Both Mokka and SLIC seem to be in similar stage

# digitisation

current default for vertex studies is quite simplistic:

- Gaussian smearing for barrel and forward silicon hits
- outer barrel and forward layers should be strips, but treated as pixels
- TPC: drift length dependent resolution
- forward chambers behind TPC endcaps: not digitised at all!
  (a step backwards with respect to 2001 BRAHMS!)

proper vertex detector digitisation available (A. Raspereza, DEPFETs)

- tests with LCFIVertex underway

➡ we especially need to catch up on strip digitisation!
(again, a step backwards with respect to 2001 BRAHMS!)

# tracking options

several levels of sophistication available:

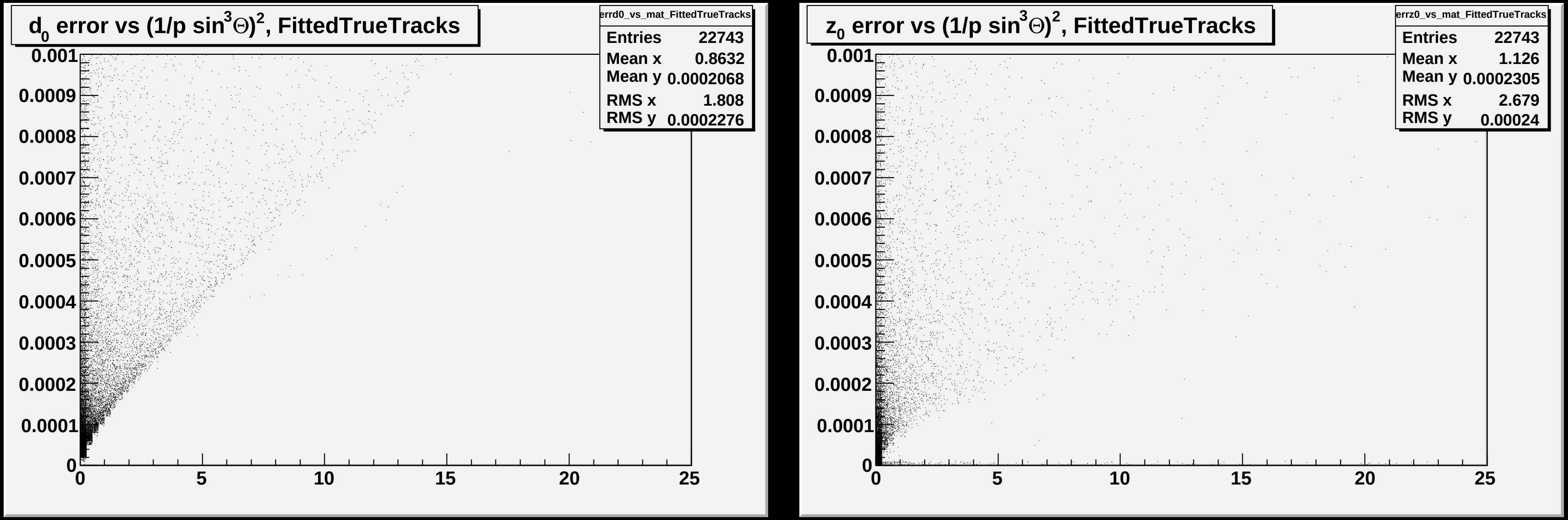
- cheater without fit (i.e. true track parameters. no covariance matrix → not for vertexing)
- cheater with track fit (currently default)
- realistic tracking:
  - — standalone TPC tracking
  - — standalone silicon tracking (two packages)
  - — global tracking

Both track cheater and realistic tracking give excellent performance, but vertexing with realistic tracking not well tested yet

Still using BRAHMS Fortran tracking code! We should get rid of that soon.

# covariance matrices

Problems with track covariance matrices
caused trouble with vertexing performance last year!
**This is something to watch out for.**



above plots: MarlinReco HEAD 13 Sep 2007 — looking mostly (!) good

# primary vertex fit

LCFIvertex provides primary vertex fitter (tear down algorithm):

- slow
- not optimised
- no beam spot averaging
- but works!

improved per event version currently being tested;
beam spot averaging still in early experimental stage

# secondary vertices

Implementation of SLD ZVTOP (ZVRES+ZVKIN) available in LCFIVertex

see talks by Hillert/Jeffery/Devetak/Grimes for all the details

It took enormous effort to make this code work!
We should share and improve it rather than write additional vertex packages!

detector dependence mostly absorbed in ReconstructedParticle objects

➡ this code should work with SiD events too!

★ Oxford group successfully fed SiD LCIO files into Marlin+LCFIVertex

★ Norman and Sonja investigate LCFIVertex interface to org.lcsim

# flavour tag

LCFIVertex provides Neural Net based b/c/bc tag.
Based on TESLA study by Richard Hawkings.

Problem:
Need about 200k events for training, better more (a million?)
for each detector variant to be studied

— done with fast simulation so far
— want to switch to Mokka now
— this will require GRID processing
(has been done before, now do it more systematically)

# performance analysis

next release of LCFIVertex will contain new plot processor (Victoria Martin):

- covering flavour tag inputs and outputs, efficiency+purity
- AIDA-based
- proven to work with JAIDA+AIDAJNI+JAS3
- proven not to work with RAIDA+ROOT
  due to missing parts of RAIDA implementation
  - ➡ completion of RAIDA currently under discussion

code for detailed analysis of track properties exists (KH), but:

- not publicly available yet
- links directly to ROOT rather than AIDA

➡ will need work

# current activities summary

My knowledge about current Marlin-related activity:

- switching from cylindrical geometries to actual VTX ladders
- mass produce Mokka events for flavour tag neural network retraining
- develop proper digitization
- improve IP treatment
- test vertexing with realistic tracking
- improve diagnostics for tracking and vertexing
- tagging performance comparison of different detector geometries
- physics studies

# summary

## Mokka+Marlin:

- all components in place to do flavour tagging
- everything is interoperable (integrated in Marlin framework)
- some weak points
  (digitisation, neural net training samples, primary vertex fit, ...)
- very active developer community

## US code

is catching up and in some places exceeding Marlin capabilities

We should make sure we share more of the less detector dependent code:

- run Marlin+LCFIVertex on org.lcsim output files?
- interface LCFIVertex to org.lcsim?