Problems with Frame Conversions in RadialStripTopology

Johannes Hauk, DESY

Hamburg CMS Tracker Alignment Meeting

23.02.2010

Issue

- Numerical calculations of frame conversions for position errors in wedge shaped sensors (TID/TEC) found to be problematic
  - Measurement Frame:
    Position and error of RecHit in terms of strips (strip 273.3 ± 3.2)
  - Local Frame:
    Position and error of RecHit in Cartesian coordinates (x|y) in [cm]
    (This is stored in event)
- In local reconstruction of RecHits only Measurement→Local (M→L)
- We (TkAl) need also L→M for (x'|y') residuals in TID/TEC
- Code in

  `Geometry/CommonTopologies/src/RadialStripTopology.cc`
  - `LocalError localError(const MeasurementPoint&, const MeasurementError&);`
  - `MeasurementError measurementError(const LocalPoint&, const LocalError&);`

Spotted Problem

- No detailed performance studies here
- Evidence for inadequate conversions
  - Sqared errors in Measurement Frame sometimes negative

    `measurementError.uu() < 0.`
- Spotted some time ago
  - See HN

    `https://hypernews.cern.ch/HyperNews/CMS/get/tracker-performance/818.html`
- After performance tests and implementation, three revisions with speed improvements introduced bad numerical calculations
- Going back to Rev. 1.14 solves problem mostly, but not completely

- Problem not visible when using standard StripCPE WithTrackAngle (Cluster Parameter Estimator) (errors(x) ≥ 15 μm)
- Spotted when testing new StripCPE Geometric (sometimes very small errors)

Performed tests

- Use current implementation
- Change back calculation methods for errors to “good” revision 1.14
- Use old implementation (rev. 1.14 directly)

- Use rev. 1.14 and additionally change in error calculation methods all internal float→double
- Change all internal float→double in all methods

- 8 mio. cosmic events from CRAFT ’09, same setup
- Only tracks with at least one valid and useful hit in TID/TEC
  - If squared error is negative, hit is declared to be not useable

Results

Version of RadialStripTopology	# tracks TID/TEC	# hits good TID	TEC	TID/TEC	# hits bad	fraction of bad hits (permille)
Present Implementation	5307766	3168423	11269386	14437809	509191	34,07
L -> M back to rev. 1.14	5307766	3211510	11734913	14946423	577	0,04
M -> L back to rev. 1.14	5307766	3168423	11269386	14437809	509191	34,07
L -> M and M -> L back to rev. 1.14	5307766	3211510	11734913	14946423	577	0,04
Revision 1.14	5307745	3211495	11734888	14946383	571	0,04
L -> M in double precision	5307745	3211696	11734499	14946195	268	0,02
M -> L in double precision	5307749	3211569	11734842	14946411	549	0,04
M -> L and L -> M in double precision	5307749	3211748	11734942	14946690	270	0,02
Everything in double precision	5307749	3211706	11734965	14946671	288	0,02

- All tests change results, except "L→M back to rev. 1.14"
- Differences in # tracks with good hit in TID/TEC due to
  - Other input for Refitter because of other hit parameters? (Refitter always looses some hits and/or tracks)
  - Different hits have negative squared errors calculated?
- Differences in # hits (good and bad) due to
  - Other tracks used by Refitter
  - Different set of hits with negative errors calculated

Remaining questions

- How many hits have wrong parametrisation?
  - Guesstimate: about same no. of hits with completely bad parametrisation for negative and positive squared errors (perhaps under certain threshold s_1, depending on individual module geometry)
  - Transition region with systematic bias on parameters (between threshold values s_1 and s_2)
  - Adequate results for values above s_2
- Is effect also problematic for values retrieved from present standard CPE (WithTrackAngle), or only for new parametrisation?
- How to proceed?
  - First, only problem with conversion back L→M spotted, only important for TkAl
  - Now, also evidence for problems coming from other parts of code
  - Need to convince experts to have closer look and solve problem

Summary

- Study show that frame conversions for wedge shaped modules can be inadequate, but cannot give criteria or quantitative hints
- Shows also that not only conversion back from Cartesian Local Frame to Measurement Frame (L→M) is problematic, but is the gravest (important for reconstruction)
- Problem only visible using new developed StripCPE Geometric
- Old revision 1.14 is much better, should be quickly reset
  - Still not numerically ideal
  - Changing to double precision reduces bad hits to ~half
- Dedicated studies on frame conversions from experts needed