50 pb^{-1} Misalignment Scenario

Jim Pivarski

Texas A&M University

18 August, 2009

Motivation

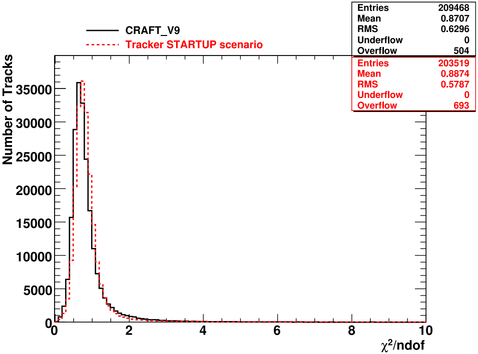
- STARTUP scenarios are based on CRAFT experience
  - tracker STARTUP misalignment: constructed from earlier scenarios to yield data-driven resolutions seen in CRAFT
  - muon STARTUP misalignment: as much as possible, the same distributions as CRAFT (e.g. no alignment in wheel ±2)
- It would be pessimistic to assume that alignment will not improve with collisions in 2010, especially "unaligned" chambers

Definition of 50 pb^{-1} scenario

- tracker misalignment: same as STARTUP
- muon misalignment: result of MC exercise, using tracker misalignment and latest MuonAlignmentFromReference

STARTUP tracker misalignment

- Definition:
  - TIB and TOB: same as 100 pb^{-1} scenario
  - Pixel endcap: same as SurveyLASOnly (i.e. no tracks)
  - everything else: same as 10 pb^{-1} scenario
- Randomly-generated with hierarchial errors
- Roughly ϕ-symmetric: unlike cosmic ray alignment but like early collisions alignment



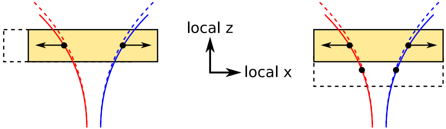
Jula Draeger, Gero Fluke

Real-data CRAFT alignment yields better χ^2 *(in modules sampled by vertical cosmic rays)*

Similar picture from residuals, DMR, and Nhan's split cosmics tests

$50\ \mathrm{pb}^{-1}$ muon misalignment

- Procedure (using MuonAlignmentFromReference on simulated data)
  1. start with no-tracks random misalignment
  2. align as many chambers as possible (mostly central barrel) with $p_T > 100$ GeV cosmic rays in 6 degrees of freedom
  3. re-align all chambers with $p_T > 20$ GeV collisions with δ_z fixed
- Why?
  - we use average of separate μ^+ and μ^- alignments ("two-bin method") to reduce sensitivity to dE/dx and $\vec{B}(\vec{x})$ errors
  - this method works for all parameters except δ_z:



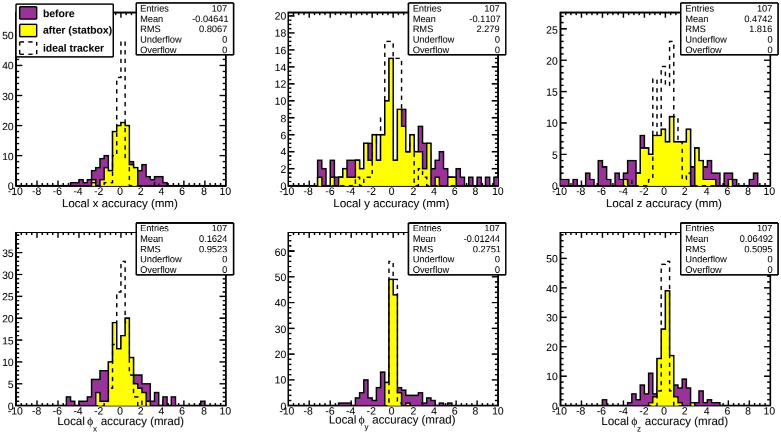
oppositely-charged tracks contribute to x residuals with opposite signs: two-bin method cancels their effect

oppositely-charged tracks contribute to "z residuals" with the same sign: two-bin method does not solve it

  - alternative is to align at very high energies ($p_T > 100$ GeV), where dE/dx and $\vec{B}(\vec{x})$ don't matter
  - such high-energy muons are only available in cosmic rays

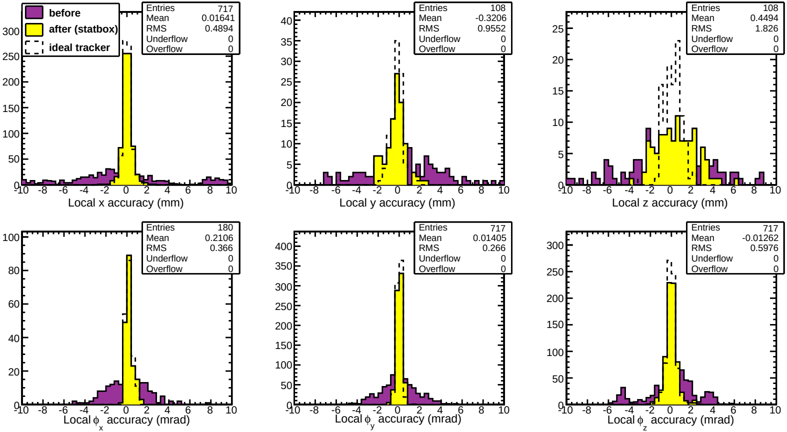
Results: cosmic ray pass

- Only showing aligned chambers
- Final product includes tracker misalignment (yellow)



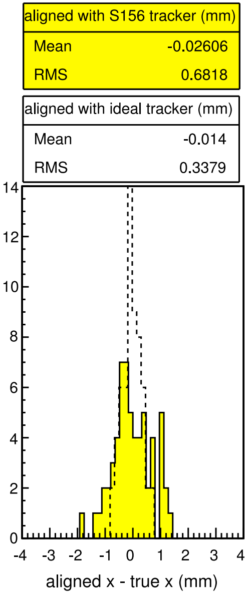
Results: collisions pass

- Showing all chambers (except one fit failure), only aligned parameters
  - barrel stations 1–3: all but δ_z (but showing inherited δ_z)
  - barrel station 4: only δ_x, δ_{ϕ_y}, δ_{ϕ_z} (no Δy)
  - CSCs: only δ_x, δ_{ϕ_y}, δ_{ϕ_z} (poor Δy)
- Note 490 μm δ_x resolution and low sensitivity to tracker misalignment

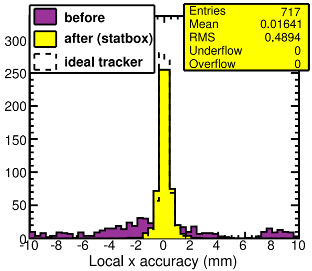


Dependence on tracker

- Right: CSA08 station 1 (best) resolution, with and without S156 tracker misal
- Below: 50 pb^{-1} resolution for all chambers, with and without STARTUP tracker misal
- Caution! apples and oranges:
  - S156 is the result of an algorithm, different correlations than random
  - S156 was a 10 pb^{-1} exercise, but achieved unexpectedly high resolution



- However, new muon alignment algorithm seems to be less sensitive



Unaligned degrees of freedom

- Station 4 DTs and all CSCs have little sensitivity to global z for basic read-out or geometrical reasons
  - global z component of their APEs $\rightarrow$ 1000 cm
- In addition, ME1/3 chambers were fixed to initial misaligned positions due to errors (many fit failures)
  - all components of their APEs $\rightarrow$ 1000 cm
  - (we should investigate, but it may be an old problem in the 22X MC)
- All other APEs $\rightarrow$ 0
- Important to use these AlignmentErrorRcds to mask out unaligned parameters

Concluding remarks

- “Intermediate” (between STARTUP and DESIGN) scenario useful for early physics analyses that depend on muon alignment

- Running the alignment algorithm itself matches muon chamber positions to a given tracker misalignment scenario

- Where to find the muon alignment (4 records, including APEs):

  `/afs/cern.ch/user/p/pivarski/public/MuonAlignmentRcd_cosmics-50pb-1-noME13_3XY_v1.db`

  `/afs/cern.ch/user/p/pivarski/public/MuonAlignmentRcd_cosmics-50pb-1-noME13_22X_v1.db`

  `/afs/cern.ch/user/p/pivarski/public/MuonAlignmentRcd_cosmics-50pb-1-noME13_v1.xml`

- See backup for _cfg.py 22X snippet

How to use it in 22X

```
process.load("Configuration.StandardSequences.FrontierConditions_GlobalTag_noesprefer_cff")
process.GlobalTag.globaltag = cms.string("CRAFT_ALL_V12::All")
process.es_prefer_cscBadChambers = cms.ESPrefer("PoolDBESSource", "cscBadChambers")
del process.DTFakeVDriftESProducer

from CondCore.DBCommon.CondDBSetup_cfi import *
process.MuonAlignmentInputSQLite = cms.ESSource("PoolDBESSource",
         CondDBSetup,
         connect = cms.string(
"sqlite_file:/afs/cern.ch/user/p/pivarski/public/MuonAlignmentRcd_cosmics-50pb-1-noME13_22X_v1.db"),
             toGet = cms.VPSet(cms.PSet(record = cms.string("DTAlignmentRcd"),
                      tag = cms.string("DTAlignmentRcd")),
                  cms.PSet(record = cms.string("DTAlignmentErrorRcd"),
                      tag = cms.string("DTAlignmentErrorRcd")),
                  cms.PSet(record = cms.string("CSCAlignmentRcd"),
                      tag = cms.string("CSCAlignmentRcd")),
                  cms.PSet(record = cms.string("CSCAlignmentErrorRcd"),
                      tag = cms.string("CSCAlignmentErrorRcd"))))
process.TrackerAlignmentInputDB = cms.ESSource("PoolDBESSource",
         CondDBSetup,
         connect = cms.string("frontier://FrontierProd/CMS_COND_21X_ALIGNMENT"),
             toGet = cms.VPSet(cms.PSet(record = cms.string("TrackerAlignmentRcd"),
                      tag = cms.string("TrackerCRAFTScenario22X_v3_mc")),
                  cms.PSet(record = cms.string("TrackerAlignmentErrorRcd"),
                      tag = cms.string("Tracker_GeometryErr_v3_offline"))))
process.es_prefer_MuonAlignmentInputSQLite = cms.ESPrefer("PoolDBESSource", "MuonAlignmentInputSQLite")

process.es_prefer_TrackerAlignmentInputDB = cms.ESPrefer("PoolDBESSource", "TrackerAlignmentInputDB")
```

Alternative

The following has been reported to work better

```
process.load("Configuration.StandardSequences.FrontierConditions_GlobalTag_cff")
process.GlobalTag.globaltag = cms.string("CRAFT_ALL_V12::All")
del process.es_prefer_GlobalTag
del process.DTFakeVDriftESProducer

from CondCore.DBCommon.CondDBSetup_cfi import *
process.MuonAlignmentInputSQLite = cms.ESSource("PoolDBESSource",
         CondDBSetup,
         connect = cms.string(
"sqlite_file:/afs/cern.ch/user/p/pivarski/public/MuonAlignmentRcd_cosmics-50pb-1-noME13_22X_v1.db"),
             toGet = cms.VPSet(cms.PSet(record = cms.string("DTAlignmentRcd"),
                      tag = cms.string("DTAlignmentRcd")),
                  cms.PSet(record = cms.string("DTAlignmentErrorRcd"),
                      tag = cms.string("DTAlignmentErrorRcd")),
                  cms.PSet(record = cms.string("CSCAlignmentRcd"),
                      tag = cms.string("CSCAlignmentRcd")),
                  cms.PSet(record = cms.string("CSCAlignmentErrorRcd"),
                      tag = cms.string("CSCAlignmentErrorRcd"))))
process.TrackerAlignmentInputDB = cms.ESSource("PoolDBESSource",
         CondDBSetup,
         connect = cms.string("frontier://FrontierProd/CMS_COND_21X_ALIGNMENT"),
             toGet = cms.VPSet(cms.PSet(record = cms.string("TrackerAlignmentRcd"),
                      tag = cms.string("TrackerCRAFTScenario22X_v3_mc")),
                  cms.PSet(record = cms.string("TrackerAlignmentErrorRcd"),
                      tag = cms.string("Tracker_GeometryErr_v3_offline"))))
process.es_prefer_MuonAlignmentInputSQLite = cms.ESPrefer("PoolDBESSource", "MuonAlignmentInputSQLite")

process.es_prefer_TrackerAlignmentInputDB = cms.ESPrefer("PoolDBESSource", "TrackerAlignmentInputDB")
```