


Alignment Monitoring and 1D-Hit Problems

Jim Pivarski, Dmitry Yakorev, Alexei Safonov

Texas A&M University

27 April, 2007

Overview of topics

AlignmentProducer Monitoring:

- Alignment/CommonAlignmentMonitor in CVS
- Structure and features

Database Geometry Monitoring:

- Beginnings of an offline tool

AlignableDetUnits in the muon system:

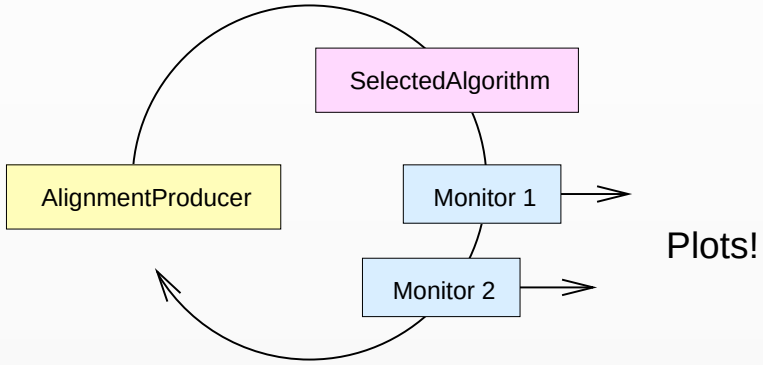
- DTs need Gero's fix

1D Treatment of DT Hits:

- The "other" 1D hits problem

Alignment/CommonAlignmentMonitor

Adds histograms, profile plots, and trees to AlignmentProducer through modular plugins, much like algorithm plugins



Lowers "potential barrier" to adding plots

- AlignmentMonitorBase manages the root file, iteration, and merging histograms from a distributed job
- Modularity allows separation of projects by group—e.g. CSC internal alignment doesn't affect tracker studies

Expected use:

- One module for each group, frequently modified in CVS
- One official module: AlignmentMonitorCSA07
- We'll rarely use the multiple-modules-in-one-job feature

Interface

```
replace AlignmentProducer.monitorConfig = {
   untracked vstring monitors = {"AlignmentMonitorHIP"}
   untracked PSet AlignmentMonitorHIP = {
      string outpath = "./"
      string outfile = "histograms.root"
      bool collectorActive = false
      int32 collectorNJobs = 0
      string collectorPath = "./"
   } }
```

Location:

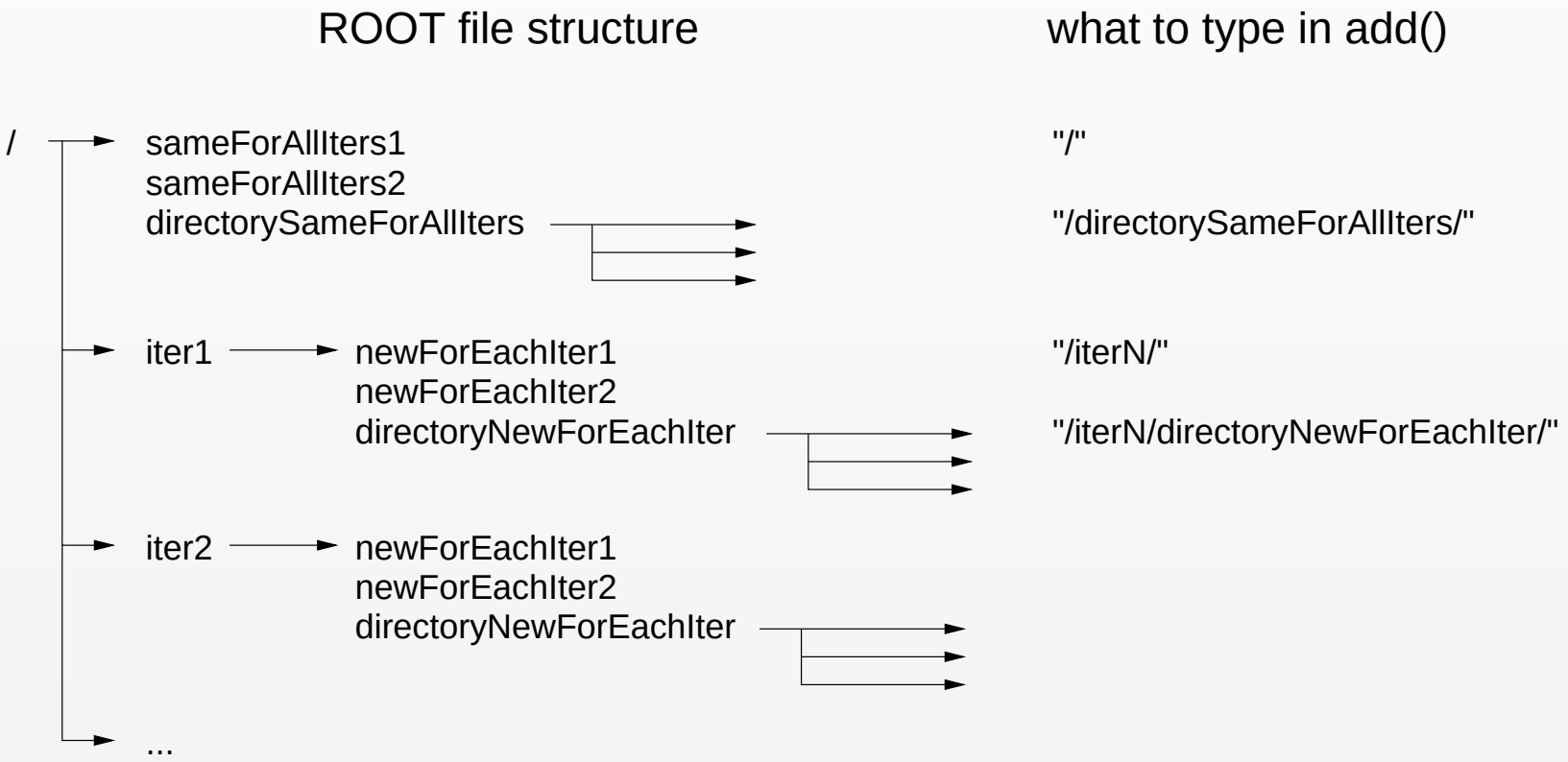
```
Alignment/CommonAlignmentMonitor/src/AlignmentMonitor*.cc
```

Subclasses reimplement:

```
book()                                          // beginning of iteration
event(EventSetup, TrajTrackPairCollection)      // event loop
afterAlignment(EventSetup)                      // after alignment
```

Structure of the ROOT file

```
m_sameForAllIters = (TH1F*)(add("/", new TH1F(...)))
m_newForEachIter = (TH1F*)(add("/iterN/", new TH1F(...)))
```



Nothing more is needed for collection jobs

Status

What works (tested with hundreds of events):

- Loading an arbirtary number of modules
- Arbitrarily-deep ROOT directory structure
- Iteration (via `AlignmentProducer.maxLoops` and/or multiple cmsRun invocations)
- Merging histograms/profiles from a distributed job
- Generating histograms from selected Alignables

What's next:

- Add lots of plots to a new module
- Use it for CSC internal alignment with MTCC data

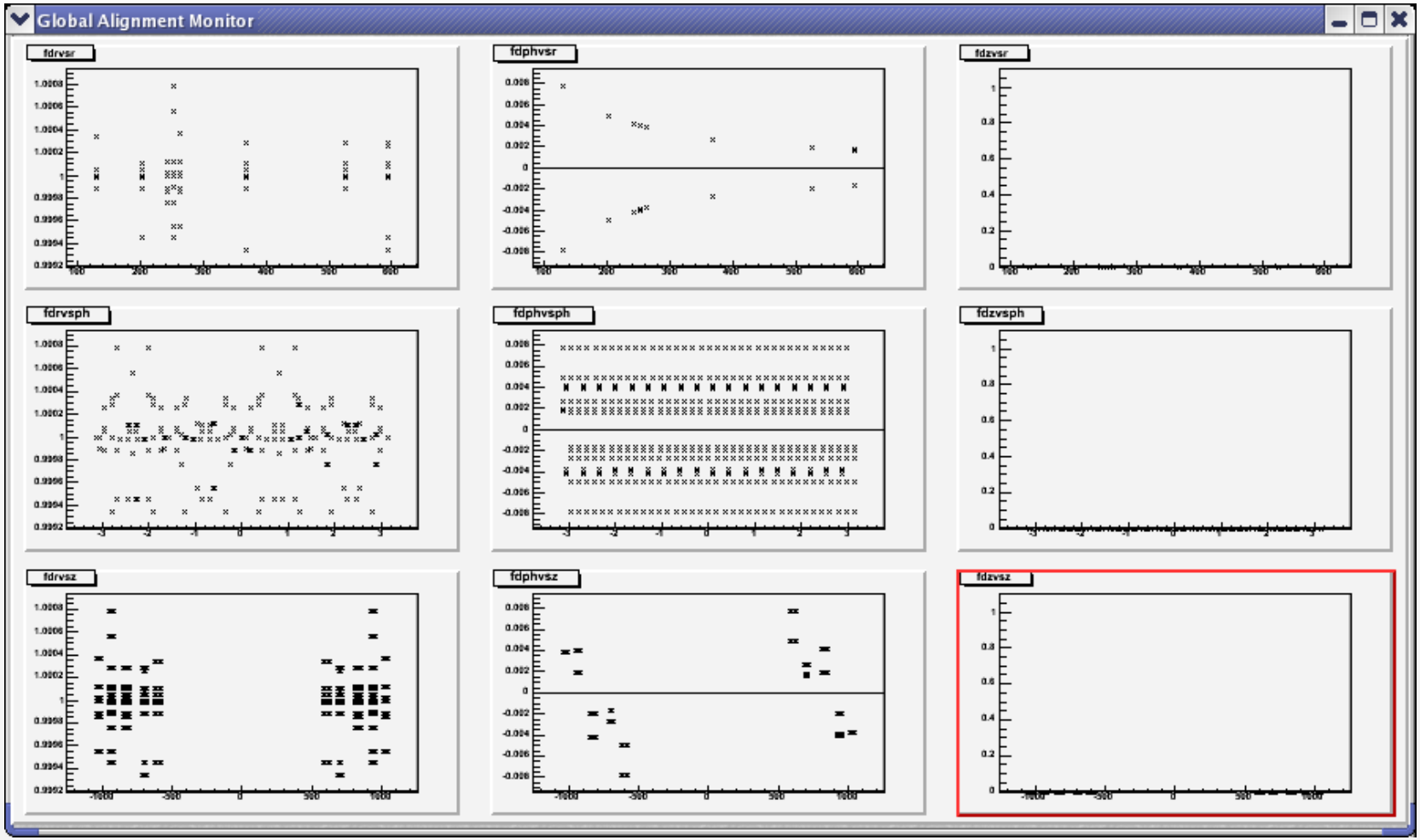
— New Topic —

Monitoring differences in geometry between alignments

vs. R

vs. ϕ

vs. Z



$(\vec{x_1} - \vec{x_2})$ $(\vec{x_1} - \vec{x_2}) \cdot \hat{\phi}$ $(\vec{x_1} - \vec{x_2}) \cdot \hat{Z}$

Status

Basic structure:

- Compiled C++ ROOT GUI
- Forks `cmsRun` processes which read SQLite files/the database

What works:

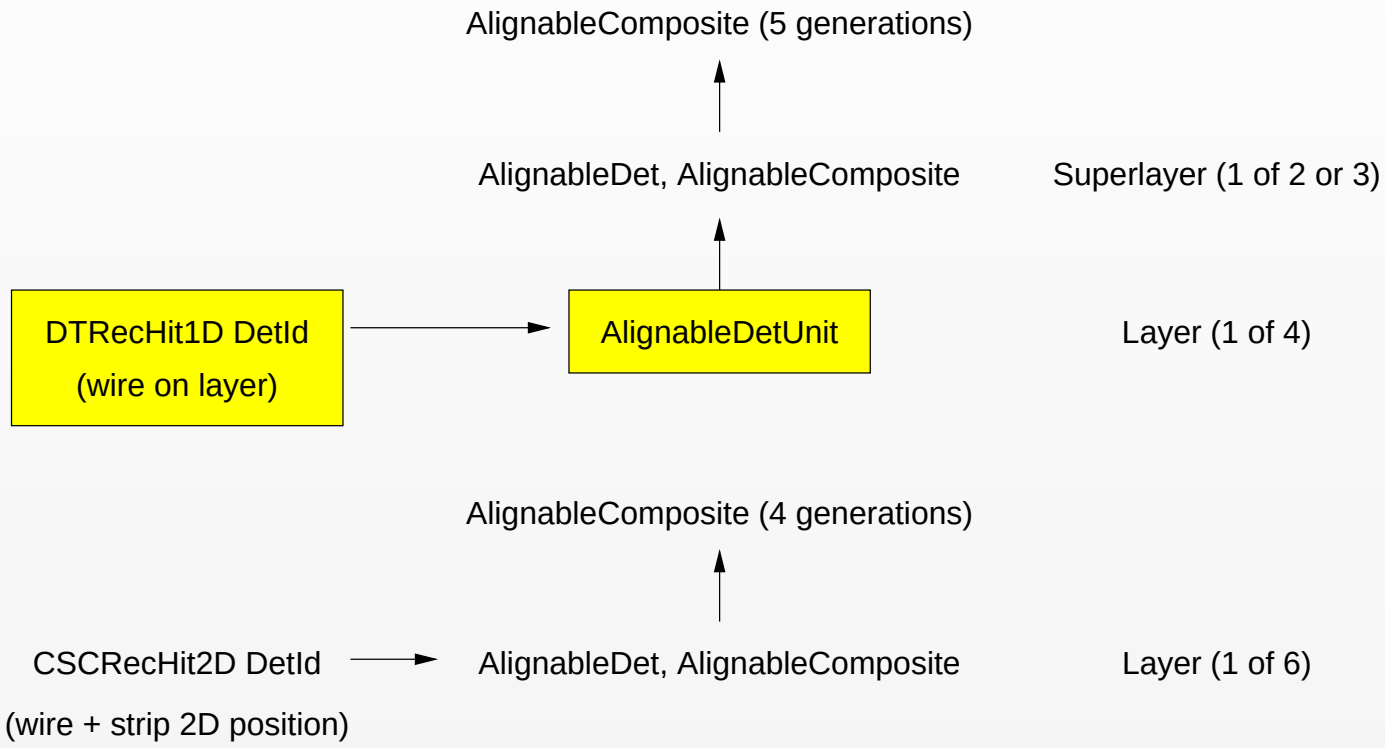
- Loading two geometry files
- Calculating and displaying translation differences
- Tabs to switch between plots

What's next:

- Read from multiple databases— plot versus time
- Represent differences in rotation angles
- A different framework? DQM? Iguana? Compile database access into ROOT GUI?

— New Topic —

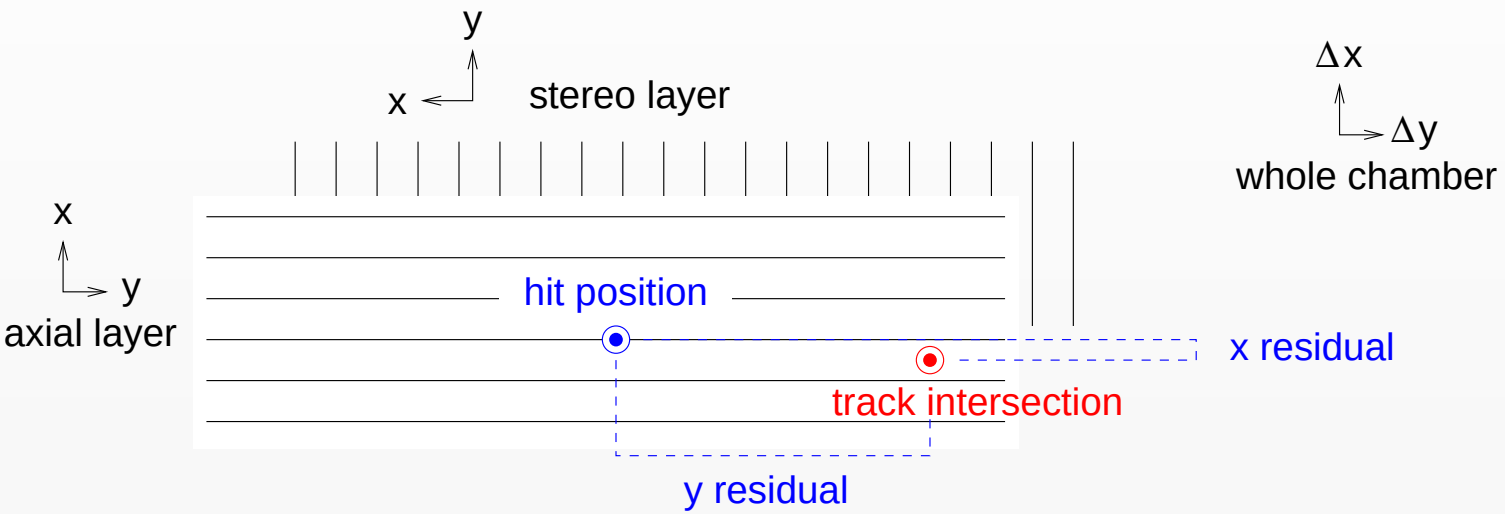
AlignableDets and AlignableDetUnits in the Muon System



- Without Gero's fix, we calculate DT corrections on layer plane, then apply to superlayer plane

— New Topic —

1D treatment of DT hits

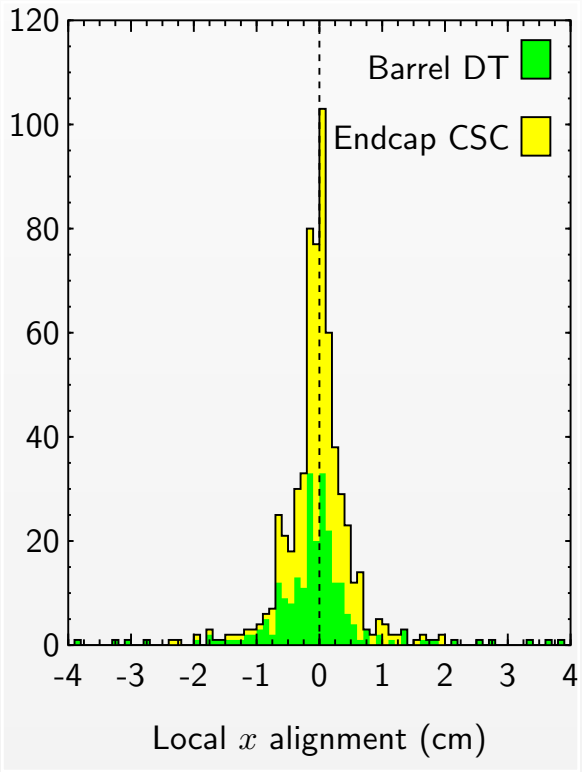


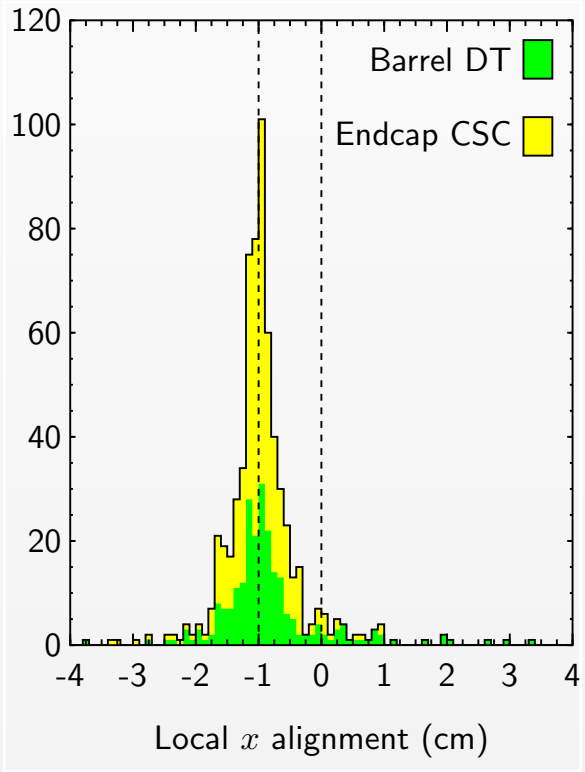
- Local x, y residuals are correctly transformed into Δx, Δy alignment corrections
- Axial y and stereo y contain no alignment info
- ParameterSelector can turn off Δy corrections, not y residuals
- In HIP, I want to add: `if(DT) {residInverse.yy = 0;}`
- How should we express "`if(DT)`"?

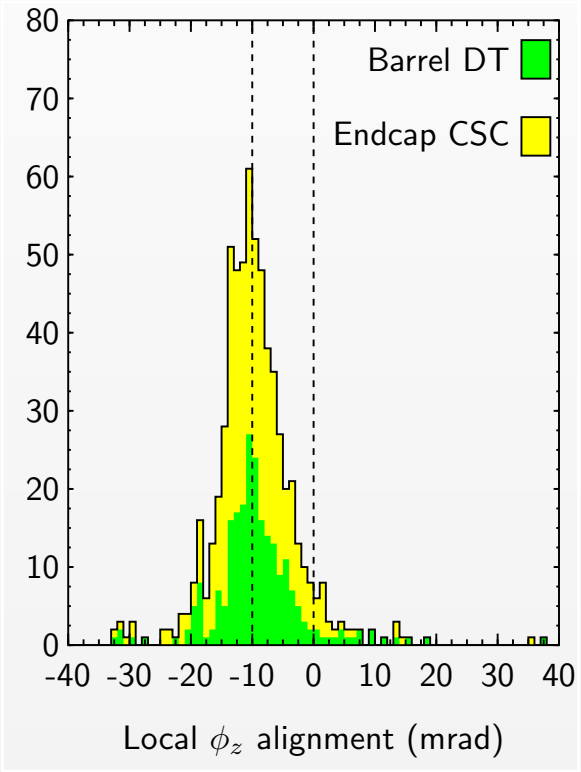
This has significant consequences!

Without `residInverse.yy = 0`, DT corrections are $\mathcal{O}(1\text{ m})$!

With `residInverse.yy = 0`, they are as expected:







Summary

- Monitor plugins are in CVS:
  Alignment/CommonAlignmentMonitor V00-00-00
  Alignment/CommonAlignmentProducer V00-15-00
- Work has begun on a database monitoring tool
- Muon DTs are a use case for Gero's AlignableDet / AlignableDetUnit fix
- We also need to ignore alignment corrections coming from DT y residuals