Software Alignment Procedure in PANDA

An Example

Roman Klasen | roklasen@uni-mainz.de

Hardware vs. Software Alignment

Hardware Alignment

Measurements of component positions with:

- Laser Trackers
- Theodolites
- Rulers
- etc.

Software Alignment

Determination of component positions with:

- measurement data
- cosmic rays
- etc.

Both methods give actual positions of components. **This talk is not about that.**

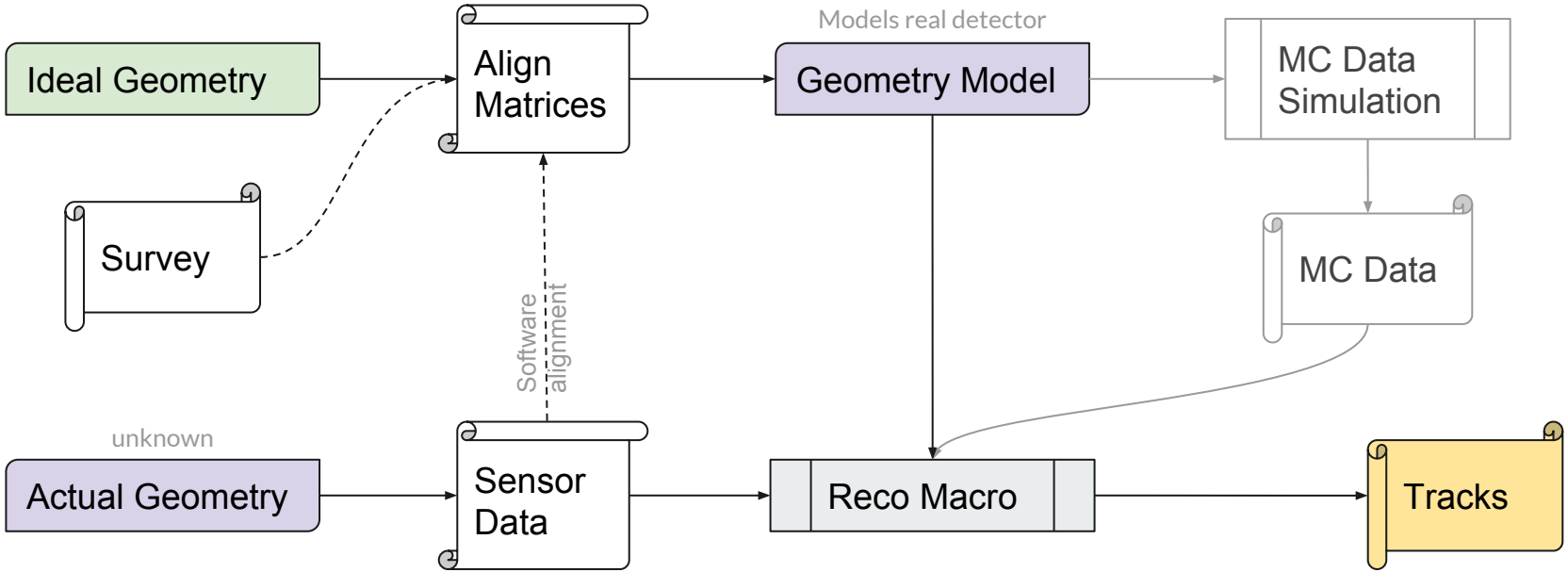
How we use the obtained Position Data

This talk is about how we use this measurement data in our reconstruction and simulation.

We can use this in two ways:

- **Shift Detector:** Shift the positions of our detector elements and run simulation macros
- **Shift Data:** Use MC data with ideal detector and shift MC data, Tracks, SDSHits etc.

Misalignment Parameters - Lumi Sim and Reco



Shift Detector vs. Shift Data

Shift Detector

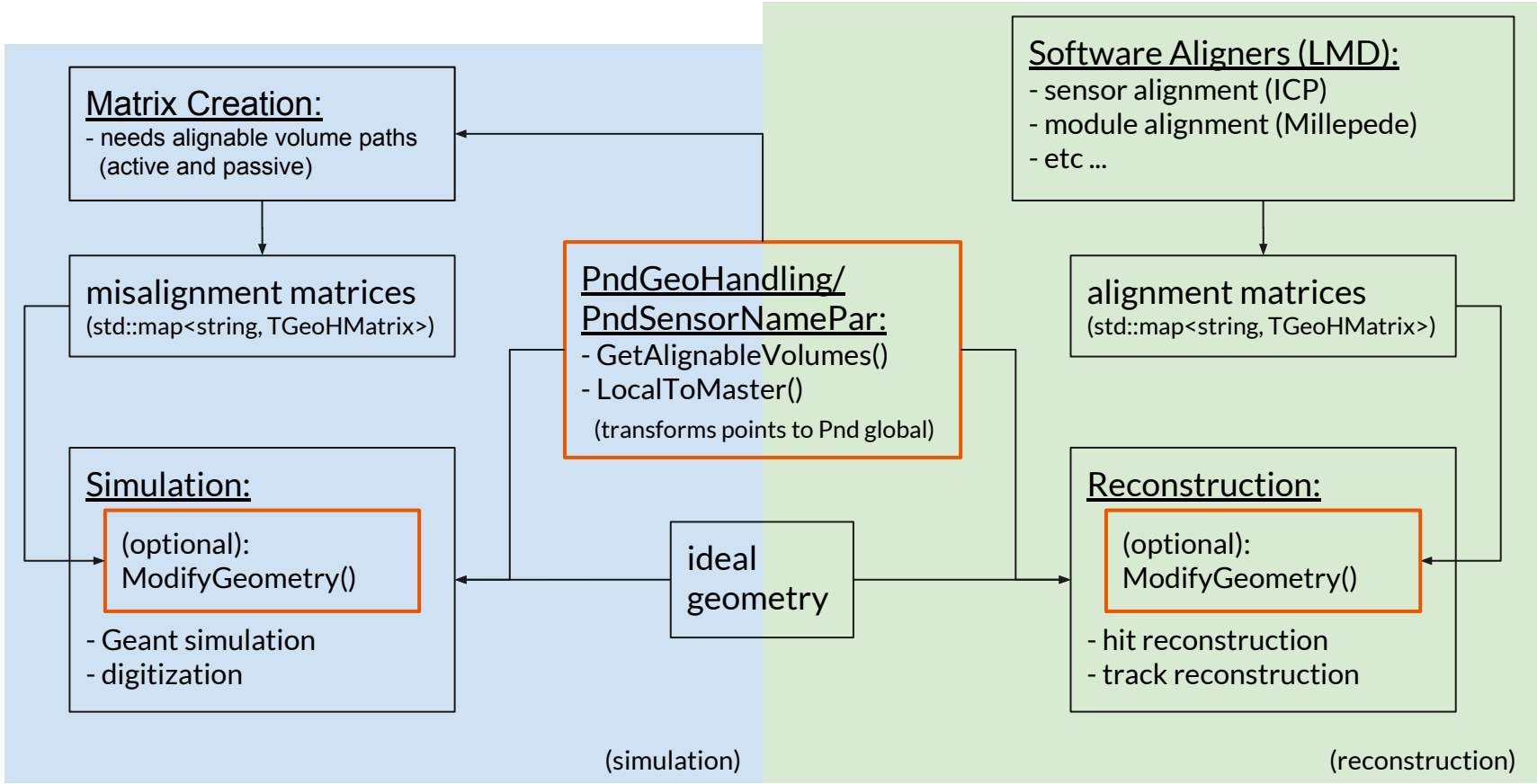
- Realistic Detector Acceptance
- Realistic scenario for Track Finder, Fitter etc.
- Reco Macro need to only load Geometry from gGeoManager (which handles Align.)
- But need to generate MC Data again (esp. If you want multiple misaligned geometries)

Shift Data

- Can use existing MC data
- Wrong detector acceptance may lead to implausible tracks:
- Don't see some tracks that should be there
- See tracks that can't be there
- Reco Macros must account for Misalignment

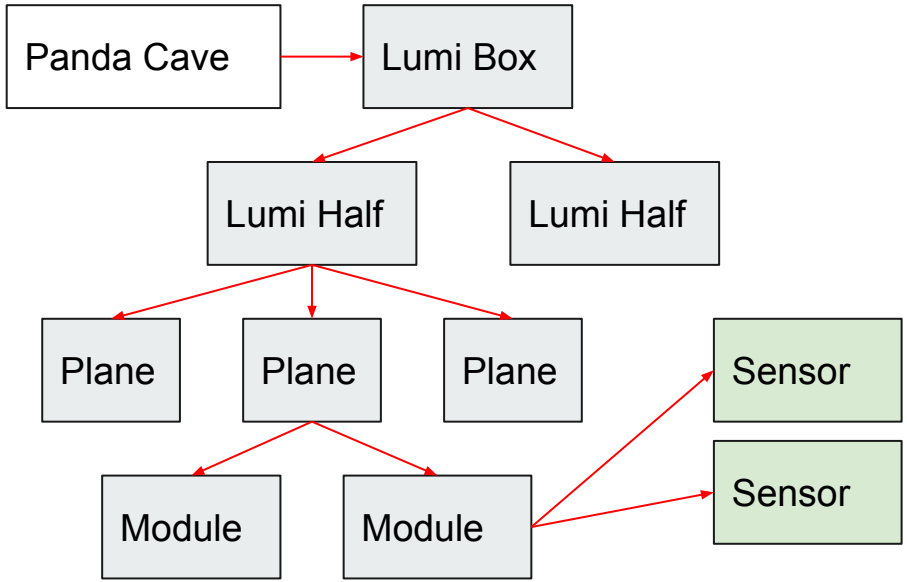
We will shift the detector components and generate new MC data here.

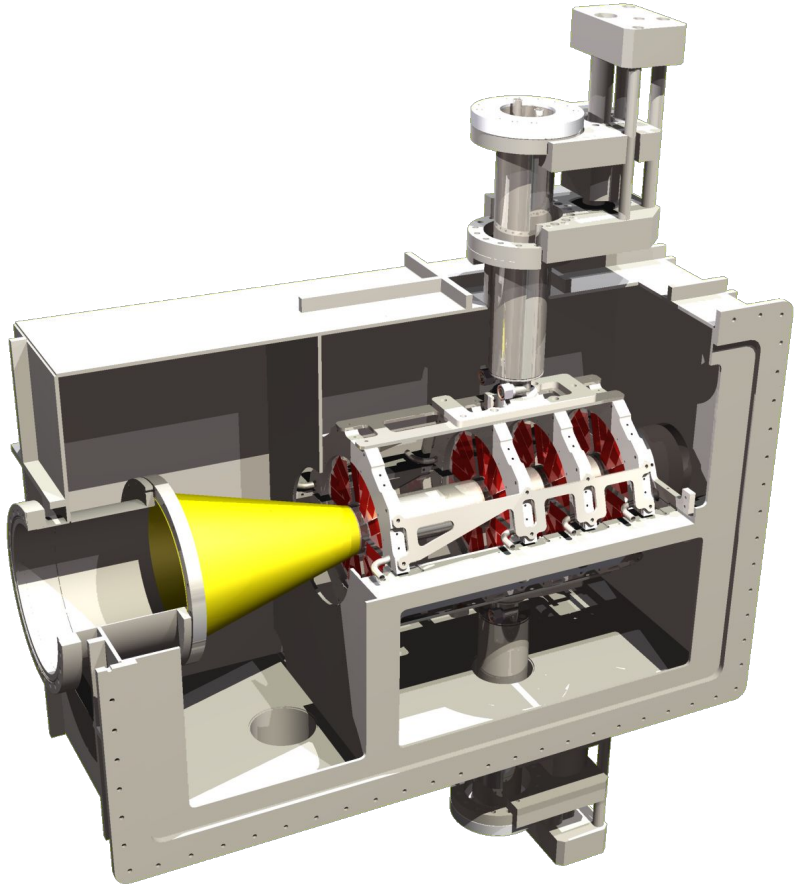
Alignment Concept - Lumi



Lumi Geometry

Alignable Components





Outline

- Get misalignment parameters from survey or create random misalignment matrices
- Get all paths to alignable objects
- Create `std::map<std::string, TGeoHMatrix> matrices`
- In MCdata generation macro, apply matrices to detector
- Init and Run FairRunAna

Current caveats:

- You must apply the matrices **before** calling ***fRun->Init()***
- You can't load geometry data from **gGeoManager** before calling ***fRun->Init()***
- That means: two macros, one for matrix creation, one for application

Code Example - Matrix Creation

```
// We shift in x-y-plane and rotate about z axis
TGeoHMatrix createRandomMatrix(double angleSigma, double shiftSigma) {
  TRandom3 *PRNG = new TRandom3(seed);
  double mean = 0; double sigmaZ; double shift[3];

  sigmaZ = PRNG->Gaus(mean, angleSigma);
  shift[0]=PRNG->Gaus(mean, shiftSigma); shift[1]=PRNG->Gaus(mean, shiftSigma); shift[2]=0;

  TGeoHMatrix result;
  result.RotateZ(sigmaZ);
  result.SetTranslation(shift);

  return result;
}
```

Code Example - Component Path and Matrix

```
PndLmdGeometryHelper &helper = PndLmdGeometryHelper::getInstance();

vector<string> paths = helper.getAllAlignPaths();      // we need this in PndGeoHandling!
// align path looks like this:  "/cave_1/lmd_root_0/half_0/plane_0/module_0/sensor_0/"

std::map < std::string, TGeoHMatrix > matrices;
for (auto &i : paths) {
  double shift = 100e-4;                               // 100 microns in Centimeters
  double rot = TMath::RadToDeg() * 1000e-6;            // 1 milirad in Degrees
  TGeoHMatrix tempMat = createRandomMatrix(rot, shift);// wrapping function
  matrices[i] = tempMat;
}
// matrices object now contains everything needed
```

Code Example - MC generation

```
// excerpt from runLumiPixel0SimBox.C
FairRunSim *fRun = new FairRunSim();
PndLmdDetector *Lum = new PndLmdDetector("LUM", kTRUE);
// load all components, materials, magnetic fields etc.
string matricesFile = "misalignMatrices.root";
TFile *misalignmentMatrixRootfile =new TFile(matricesFile.c_str(),"READ");
if (misalignmentMatrixRootfile->IsOpen()) {
  std::map < std::string, TGeoHMatrix > *matrices;
  gDirectory->GetObject("PndLmdMisalignMatrices", matrices);       // read matrices from file
  misalignmentMatrixRootfile->Close();
  Lum->SetMisalignmentMatrices(*matrices);                         // needed in common interface
}
else { // handle errors }
fRun->Init();
// run rest of macro
```

Code Example - Geometry Class

This is needed in a common interface!

```
// get's called sometime after Lum->SetMisalignmentMatrices(*matrices);
void PndLmdDetector::ModifyGeometryByFullPath() {
  TString volPath;
  for (auto const& entry : fAlignmentMatrices) {
    volPath = entry.first;
    gGeoManager->cd(volPath);
    TGeoNode* n3 = gGeoManager->GetCurrentNode();
    TGeoMatrix* l3 = n3->GetMatrix();            // new matrix, representing real position
    TGeoHMatrix nlocal = *l3 * entry.second;     // from new local mis RS to the global one
    TGeoHMatrix* nl3 = new TGeoHMatrix(nlocal);
    TGeoPhysicalNode* pn3 = gGeoManager->MakePhysicalNode(volPath);
    pn3->Align(nl3);
  }
}
```

Reference: https://github.com/FairRootGroup/FairRoot/tree/master/examples/simulation/Tutorial4

See our Code

You can see an example of the complete process at: **`pandaroot/macro/lmd`**

Matrix creation:

- `PandaRoot/macro/lmd/geo/createPndLmdMisalignmentMatrices.C`

Set misalignment matrices in MC data generation:

- `PandaRoot/macro/lmd/runLumiPixel0SimBox.C`

Misalign Detector Geometry:

- `PandaRoot/lmd/LmdMC/PndLmdDetector.cxx`

Design considerations

Most components adhere to the same rules. We don’t need copies of this code in every component class.

- Introduce a class that implements `ModifyGeometryByFullPath()` that all detector classes can inherit from. There is no need to copy this code to every detector class.
- Alternatively: implement this code in current class like `FairDetector` or `FairModule` that everyone inherits from already
- Move getAllAlignPaths() to PndGeoHandling for the same reasons.

This way, all alignment-relevant code is already available for simple use.

Thank you for your attention!