HelicalTrackFitter Improvements

Lori Stevens (UCSC/SLAC)

SiD Tracking Meeting
July 27, 2007

Outline

- Fitting TrackerHits
- Reordering hits
- Fitting barrel hits
- Bug in HelixSwimmer

HelicalTrackFitter.java

- First CircleFitter fits a circle to x-y coordinates
- If successful, SlopeInterceptLineFitter fits a line to s-z coordinates
- If successful, stores parameters and creates symmetric covariance matrix, assuming no correlation between circle and line fits.

Fitting TrackerHits

- HelicalTrackFitter currently takes Cartesian coordinate array arguments
- Overload fit method to take TrackerHit List argument

Reordering hits

- HelicalTrackFitter only handles successive hits that are separated by no more than π.

Put points in order:

- Order hits according to z coordinate
- Look at endpoints to find closest point to origin
- Reverse order if necessary

Fitting barrel hits

- HelicalTrackFitter fits vertex hits only

Let's include barrel hits…

- Look at Type for TrackerHit
  - Type 0: vertex hit
  - Type 1: barrel hit
- Assigning Type for Cartesian arrays
  - Type 0: if dz>0
  - Type 1: if dz<0
- Barrel hits: 2D hits, use CircleFitter

Found along the way: bug in HelixSwimmer

- When track originated outside first vertex layer, HelicalTrackFitterTest failed
- Traced to problem in HelixSwimmer
- Is the swimmer swimming the wrong way?

Code taken from from HelicalTrackFitterTest.java

```
//defining detector
double[] radii = {1.2, 1.4, 1.6, 1.8, 2.0};
double[] lengths = {10., 10., 10., 10., 10.};

// swimmer to create hits
HelixSwimmer swimmer = new HelixSwimmer(bField);

// define track
//starting point of track (radius of point: 1.196)
Hep3Vector pos = new BasicHep3Vector(.7,.97,0);
double[] momentum = {0.2, 0.1, 0.1};
Hep3Vector mom = new BasicHep3Vector(momentum);
int charge = 1;

swimmer.setTrack(mom, pos, charge);

// swim the track and make some hits...
for(int i=0; i<nmeas; ++i){
   double s=swimmer.getDistanceToCylinder(radii[i],lengths[i]);
   Hep3Vector point = swimmer.getPointAtDistance(s);

   x[i] = point.x();
   y[i] = point.y();
   z[i] = point.z();
```

```
//defining detector
double[] radii = {1.2, 1.4, 1.6, 1.8, 2.0};
double[] lengths = {10., 10., 10., 10., 10.};

// swimmer to create hits
HelixSwimmer swimmer = new HelixSwimmer(bField);

// define track
//starting point of track (radius of point: 1.204)
Hep3Vector pos = new BasicHep3Vector(.7,.98,0);
double[] momentum = {0.2, 0.1, 0.1};
Hep3Vector mom = new BasicHep3Vector(momentum);
int charge = 1;

swimmer.setTrack(mom, pos, charge);

// swim the track and make some hits...
for(int i=0; i<nmeas; ++i){
   double s=swimmer.getDistanceToCylinder(radii[i],lengths[i]);
   Hep3Vector point = swimmer.getPointAtDistance(s);

   x[i] = point.x();
   y[i] = point.y();
   z[i] = point.z();
```

What happens

- After the previous code, a println statement gives the following data:
- When starting point is set at r<1.2
  - Hit1: x:0.7 y:1.0 z:0.0 r:1.2
- When starting point is set at r>1.2
  - Hit1: x:21.4 y:9.4 z:10.0 r:23.4

Possible culprit

- In HelixSwimmer:
  - in getDistanceToRadius(double r):

trajectory.getDistanceToInfiniteCylinder(r)

- Println statement gives 0.2 (r<1.2) and 1026.7 (r>1.2)
- Have only investigated this far

For the future

- Continue working on HelicalTrackFitter
- Test changes:
  - apply to SeedTracker algorithm
- Investigate bug in HelixSwimmer