Tracking Simulation Issues

Rob Kutschke, Fermilab
SiD Collaboration Meeting
Jan 29, 2008

Outline

- Aside: FNAL CD Resources
- Survey of codes that are alive, dead or undead.
- Thoughts on to get to there from here.
- I almost certainly have some details wrong or have missed something; please let me know so that I can correct this talk.

Aside: FNAL CD Resources

- Staff reassigned:
  - Hans Wenzel: Run II support.
  - Rob Kutschke: mostly Run II support.
    - Low duty cycle until April 1. After that ?
  - Lynn Garren: Install software; VO bookkeeping.
  - Adam Para: dual readout ECal.
- Computing hardware allocated to ILC remains:
  - O(2 TB) NAS disk each for SiD, ILD, 4th, accelerator.
    - Groups may pool space for common MC files.
  - FermiGrid slots (150, negotiable upwards).
  - ilcsim (4x1cores) and ilcsim2 (2x2 cores)
    - Short to medium jobs.
    - Submission to FermiGrid.

3 Paths for Tracking

- SLIC (G4 based MC)
  - Writes out MC truth intersections of particles with measurement surfaces: SimTrackerHit.
- In org.lcsim
  - FastMC
    - Parameterized cov matrix. Ignores hits (if present).
  - Cheating pattern recognition:
    - Still to come: gaussian smearing and fit.
  - Full tracking simulation:
    - Simulation of individual hit strips/pixels from SimTrackerHits.
      - Event/background overlaying done at SimTrackerHit level.
    - Form clusters.
    - Run pattern recognition and fitters on clusters.
  - Geometry system: many new features added recently.

What Exists?

- Lots of pieces exist in some form:
  - FastMC
  - Hit makers; virtual segmentation; hit bookkeeping.
  - Pattern recognition
    - Cheaters and real.
  - Track fitters
    - Weight matrix, TRF based Kalman filter and others.
  - Vertex fitting/finders.
  - Flavour tagging.
- Not all work; some are incomplete; the parts often do not work easily with each other.

FastMC

- LCDTRK:
  - Parameterize covariance matrix for tracks from the origin, as a function of PT and tan(lambda).
    - Not a function of vertex location.
  - May include beam constraint, or not.
- Inside org.lcsim job:
  - Loop over MCParticles (generated tracks).
    - Compute covariance matrix from parameterization.
    - Draw a random track from this covariance matrix.
- Glitches recently fixed:
  - sid01 used wrong parameterization.
  - Confusion about "reference point".

FastMC (2)

- Strengths:
  - It works and is being used.
  - Sufficient for some studies and sufficient for a first pass on many more.
- Weaknesses:
  - Incorrect error on curvature for pT < 1GeV
    - Uses a parabola, not a circle in bend view.
  - Incorrect covariance matrix for tracks from vertices outside of the beam pipe.
    - Impact on vertexing and flavor tagging.
  - Footprint on the calorimeter not computed correctly
    - Wrong centroid and wrong error ellipse.
    - Impact on PFA, particularly for low momentum tracks.
  - Only deals with cylinders + z-disks, not wafers.

Hit Making

1. MC Truth intersection of particle with a surface.
   - SimTrackerHit
2. MC truth pulse height on an individual hit pixel/strip, caused by a single particle. Real valued.
3. MC truth pulse height on a individual hit/pixel strip, including contributions from 1 or more particles, cross talk, other noise sources. Real valued.
4. Digitized version of (3). RawTrackerHit
   - Includes response of the chip. Integer valued.
   - Both Tim and Nick have codes for steps 1...4.
5. A cluster of (4). TrackerHit.
   - Measured centroid and error on centroid.
   - Does not yet have Lorentz force corrections.
   - Tim, Nick, Dima have cluster makers.

Hit Making (2)

6. A pair of crossed strip clusters (5).
   - Includes possible ghost solutions.
   - Tim and Dima have this.
7. A hit of type (5) or (6) attached to a track
   - Updated with all corrections.
   - Residual and error on residual.

- For some technologies, we need to be able to overlay events, shifted in time.
  - eg CCD’s.
- LCIO only knows about 3 of these 7 things.
  - Only these 3 may be persisted.

Tim Nelson's Hit Making Code

- Detailed simulation of pulse heights on individual hit strips/pixels.
  - Includes chip response.
  - Works now on strips; will add pixel support.
  - Includes a cluster maker.
- Works on wafers; any polyhedral shape.
  - Both barrel and endcap.
  - Not on the cylinder + z-disk detector models (sid01).
- Example that loops over clusters, looks back to strips, exercises many features, makes histograms:
  - sandbox.RobKutschke.TKNHits.ReadTKNClusterDriverV1.java
  - Utilities in: org.lcsim.contrib.RobKutschke.TKNHits.
  - May be a little behind the hit making code.
- See: Tim's talk in the Sim/PFA parallel session

Nick Sinev’s Hit Making Code

- Simulation of pulse height on pixels. Not strips.
- Old Version.
  - CCD’s only. Cylinders, z-disks, wafers.
  - No detailed model of charge collection. Just sigma.
  - org.lcsim.mc.CCDSim;
- New Version.
  - Detailed model of charge collection. All pixel types.
  - Works for wafers, cylinders, zplanes.
  - Slow but OK. Very slow for non-depleted sensors.
  - Still under development.
  - lcsim.sandbox.NickSinev.PixSim;
- Link to: Nick's talk in the tracking parallel session.

Dima Onoprienko: Virtual Segmentation

- Strips + pixels; Cylinders + z-disks + wafers.
  - Allows more tiling patterns of strips and pixels than does Tim's code.
- Implements all of the hit bookkeeping steps mentioned a few pages ago.
- Includes bookkeeping for arbitration of clusters assigned to multiple tracks.
- Trivial model of pulse height (one strip gets it all).
  - Copy Tim's and/or Nick's pulse heights.
- See:
  - His talk at ALCPG07
  - https://confluence.slac.stanford.edu/display/ilc/Monday + attachments.
  - https://confluence.slac.stanford.edu/display/ilc/Thursday
  - org.lcsim.contrib.onoprien.tracking.**

Advertised Pattern Recognition Plan

- Find tracks first in the pixel vertex detector
  - Extrapolate into tracker
  - Because vertexer is 3D, tracker not.
- This will miss tracks that originated "far" from beamline: outside layer 2 of pixel detector.
  - Need codes targeted to find these tracks.
  - Might want to first clean up the event by excluding hits found in the first pass.
- May also want an outside-in pass and other cleanup passes.

Pattern Recognition Codes

- Cheaters:
  - Finds all hits created by each track. No fit.
- Part Cheater part real:
  - Vertex detector assisted.
- Real:
  - For tracks from close to beamline
    - Nick's hep.lcd code.
    - seedtracker (includes ZSegmentFit).
  - For tracks with large displacement from beamline:
    - Calorimeter assisted tracking.
    - Tracker standalone
      - AxialBarrelTrackFinder* and UCSC.
    - seedtracker can work here too.

Cheaters

- Use MC truth to find hits that belong on each track
  - No fit done.
- Officially supported cheater:
  - org.lcsim.recon.cheater.ReconCheater.java
    - Calls: org.lcsim.recon.ztracking.cheater.TrackingCheater.java
  - Example: on JAS3 examples menu.
  - Used by Nick's weight matrix fitter.
  - I am aware of one bug that sometimes puts wrong hits on a few tracks. Minor physics impact.
- Code used by Calorimeter Assisted tracking
  - org.lcsim.recon.mcTrackFinder.MCTrackFinder.java
- Others?

Vertex Assisted Pattern Recognition

- I have been told that this code exists but I have not seen it and I am not sure if anyone is using it.
  - Maybe some variant of the UCSC effort?
- Cheating part:
  - Use cheater to find tracks in pixel vertex detector.
  - Select tracks with enough hits, typically 4 or 5.
  - Fit these tracks.
- Non-cheating part:
  - Extrapolate tracks into the tracker and add hits.

Nick's hep.lcd Pattern Recognition Code

- Used in the past to show that we have high pattern recognition efficiency, even in the core of jets.
  - Worked on cylinders + z-disks.
  - Does not work on wafers.
  - Tracker modeled with a z resolution of 5 mm!
    - Very different from our current baseline!
- Last worked in hep.lcd.
  - Never ported to org.lcsim

seedtrack: Rich Partridge

- Replacement for and generalization of Nick's hep.lcd code. All new code.
  - Pick 3 magic layers. Find all triplets and form helices.
    - Can use any 3 layers: barrel/endcap and tracker/vertexer.
    - Even 3 axial tracker layers: ZSegmentFit!
    - Can ask for track to be close to IP; or not.
  - Confirm with a 4th layer
  - Follow road and add hits.
  - Cylinder + z-disks
    - Needs mods to work with waferized layers.
- Can repeat with many different strategies:
  - Strategy = 3 magic layers + confirm layer + cuts.
- Has internal toy fitter. Needs a proper fitter.
- org.lcsim.contrib.seedtracker;
- Still under development.

ZSegmentFit

- Rich Partridge.
- Developed as adjunct for seedtracker.
- For tracks with only axial silicon strip tracker hits, find allowed region in z-tan(lambda) space.
  - Polygon shaped.
  - Report centroid of allowed region as fitted value.
  - Cov computed assuming all points equally probable.
- Can also deal with a single good z measurement plus one or more axial silicon strip measurements.
- Code:
  - org.lcsim.fit.zsegment.ZSegmentFitter
- Talk at ALCPG Sim/Reco meeting Dec 11, 2007

Calorimeter Assisted Tracking

- Dima Onoprienko and Erkhard von Toerne.
- Ignore hits attached to previously found tracks.
  - eg. use cheater to find tracks from beam line.
  - Ignore MIP stubs from these tracks.
- Find MIP stubs not attached to known tracks.
- Extrapolate MIP stubs into tracker and vertex detector; add hits.
  - Arbitrate hits assigned to multiple tracks.
- See:
  - Dima's talk at LCWS07
  - Example: org.lcsim.recon.cat.ExampleDriver;
    - Previously: org.lcsim.contrib.garfield

AxialBarrelTrackFinder*

- Standalone track finder for axial strip tracker.
  - Uses z extent of wafers as crude z measurement.
    - Uses ZSegmentFit from seedtracker for this.
  - org.lcsim.contrib.tracking.AxialBarrelTrackFinder*;
- Originally developed by Tim Nelson.
  - Extended by Bruce Schumm + UCSC students:
    - Lori Stevens, Tyler Rice, Chris Meyer
- Lori Stevens:
  - Talk at ALCPG Sim/Reco meeting Aug. 21, 2007
- Combining with Calorimeter Assisted Tracking
  - Tyler Rice, Chris Meyer.
  - Talk at ALCPG Sim/Reco meeting Aug. 21, 2007

Weight Matrix Fitter (Nick Sinev)

- Cheating pattern recognition.
  - Gaussian smearing of SimTrackerHits
- Cylinders and z-disks; pixels and strips.
  - No detectors made of wafers
- Reports track parameters valid at PCA z-axis.
  - This is what we want for tracks from vertices inside the beampipe.
  - We want more for tracks from vertices outside the beampipe.

TRF Based Kalman Filter

- Rob Kutschke.
- Work plan:
  1. Self test: internally generated tracks + hits.
  2. Cheating pattern recognition; gaussian hit smearing.
     - First for cylinders + z disks; later with wafers.
  3. Write out track parameters valid at:
     - PCA z-axis
     - Innermost hit.
     - Outermost hit and/or footprint on ECal.
     - Other locations for displaced tracks?
  4. First release for public use.
  5. Provide interfaces to accept hits from different pattern recognition programs.
- Initial success with first step for most cases.
  - Talk at ALCPG Sim/Reco Weekly Meeting Oct 9, 2007

TRF Based Kalman Filter (2)

- Encountered a numerical precision problem:
  - Only occurs on inward fit:
    - 5 tracker barrel axial strip layers
    - pixel z disk
    - Only for a narrow range of dip angles.
  - Error matrix on the predicted XY position at z-disk:
    - XY correlation coefficient is +/-0.999999992
    - Both X and Y depend on the unknown z position and dip angle of the track.
- A surprisingly large amount of work to understand that this is a true precision issue and not an algebra or programming error and to understand the source and scope of the problem.

TRF Based Kalman Filter (3)

- First attempt at a work around:
  - Kalman circle fit in tracker.
  - Kalman helix fit starting at first z plane.
- This fails for:
  - Stereo strips in tracker barrel.
  - Charge division in tracker barrel.
  - sigma(z) = 10 cm/sqrt(12) in tracker barrel.
- One more idea to try:
  - Replace pixels with equivalent pair of strips.
  - If this fails, I will give up trying to fix it and move on.
- What's next:
  - With my reduced availability it will be hard to have a public release to step 4 much before mid April 2007.

Other Tracking Code

- Fred Blanc and Steve Wagner
  - SODTracker: Silicon Outer Detector;
    - Track finding in tracker alone?
  - KFFitter
    - A port of BaBar Kalman filter; fits SODTracks.
    - Some geometry: hard coded magic numbers.
  - Neither under active development or used by others.
- Caroline Milstene's kalman filter
  - Developed for her muon work.
  - Can swim through thick materials.
- Not aware of any code for charge division in tracker.
- FTF: TRF based track finder.
  - Not aware that anyone has exercised this in org.lcsim.

Vertex Finding/Fitting

- Vertex packages in org.lcsim
  - org.lcsim.recon.vertexing.zvtop4;
  - org.lcsim.recon.vertexing.billoir;
  - org.lcsim.contrib.JanStrube.zvtop
  - org.lcsim.contrib.JanStrube.vtxFitter
  - org.lcsim.recon.cat.kshort;
    - K0s finder code for Calorimeter Assisted Tracking
- I am have not used any of these.
- I am not sure of the status of the first 4; rumors have it that the zvtop ports are incomplete.

Linear Collider Flavor Identification (LCFI)

- Vertex finding, fitting, and flavor tagging package.
  - Contact: Ben Jeffery, grad student at Oxford.
  - An evolution of zvtop.
  - Appears much more advanced than what we have.
  - Native in Marlin: C++
  - Flavour tagging is 10 NN variables, including vertex topology and vertex mass; no leptons in jets.
  - Improper treatment of vertices outside beampipe.
- I believe our best bet is to use this code. Options:
  - Call C++ from java.
    - Not yet done.
  - Run LCFI in Marlin, communicate via LCIO.
    - Tim Barklow + Tomasz Lastovicka.

Summary: My Personal View

- We do not have working code for our primary pattern recognition algorithm: vertex detector first.
  - We have lots of codes for the secondary algorithms.
  - seedtrack is advertised as the solution. Not here yet.
- FastMC works and is being used.
  - User must understand reference point convention.
- Geometry system is converging.
  - Do we have enough flexibility for the description of endcaps and forward planes?
- Hit making machinery is finally converging.
  - Dima's code has the flexibility to explore many different tiling patterns, including those not supported by main geometry system.

My Interpretation (2)

- Kalman filter long overdue.
- LCFI looks very promising.
  - Need to test drive it.
    - Andrei N. says that Tim Barklow plans to do this.
  - Is it convenient enough to use? Or do we need a native java fitter for "everyday" use?
- Lots of abandoned code.
  - Is anything worth saving?

Critical Path Items

- High Risk:
  - Primary pattern recognition code.
    - Is seedtrack the answer?
    - Even for waferized detectors?
- Medium Risk:
  - Kalman filter.
  - Secondary pattern recognition codes.
    - Port to waferized detectors.
  - Integration over multiple bunch crossings.
    - The issue is making sure we have it all.
- Low Risk:
  - Vertexing + flavour tagging code.
  - Hit machinery for a single event.
  - Define a baseline waferized detector model.

Things We Can Do Now

- Benchmark analyses using FastMC.
- Learn to use LCFI, using FastMC for inputs.
  - Decide if it is indeed the right answer.
- Studies of segmentation/occupancy in endcap and forward regions.
- First iteration of baseline waferized detector model.
- Adapt LCIO to us (or us to LCIO?).
- Understand time scales for critical path items.
  - Add effort if it makes sense.

Backup Slides

Precision Problem

- Kalman filter is intrinsically directional.
  - Current track parameters + cov express the information from all previous hits.
- Repeating step of KF:
  - Computed weighted average of previous information (track parameters) with the new information (hit).
    - Weight proportional to inverse of error matrix.
- Must start KF with a set of seed track parameters and a covariance matrix that is diagonal with large numbers on the diagonal.
  - If too small then fit is biased towards seed parameters.
  - If too large then precision problems.