Tracking and Vertexing

Norman Graf (SLAC)

HPS Collaboration Meeting @ JLab
November 16, 2016

Topics

- Monte Carlo Simulations
  - Brad and Rafo
- Event Reconstruction
  - Track Finding
  - Track Fitting
- Detector Alignment
  - Alessandra
- Vertexing
  - Sho and Holly
- Analysis
  - Everyone

Meetings & Results Pages

- Weekly meetings to discuss both hardware and software tracking issues
  - Subscribe to mailing list
- 2015 Performance Studies
  - Confluence page with task list and performance plots related to analysis of the 2015 data
- 2016 Performance Studies
  - Confluence page with task list and performance plots related to analysis of the 2015 data

The Global View

- Glass 65% full:
  - The tracking and vertexing works…
    - Omar's Thesis
    - Sho's Thesis
    - Performance and analysis plots will be shown during the rest of this meeting

The Global View

- Glass half full:
  - The tracking and vertexing works…
    - Omar's Thesis
    - Sho's Thesis
    - Performance and analysis plots will be shown during the rest of this meeting
- Glass half empty:
  - We could be doing (much) better
    - Better Simulation
    - Faster Reconstruction
    - Low momentum tracking efficiency
    - Track and vertexing resolutions

Alignment

- Currently use GBL track fits fed to millipede II
  - Hold some layers constant, float others to minimize χ^2
  - Provides internal alignment of Si sensors
  - See Alessandra's talk for details
- Need to tie this both to the ECal and Lab coordinate system
- Need to constrain weak modes

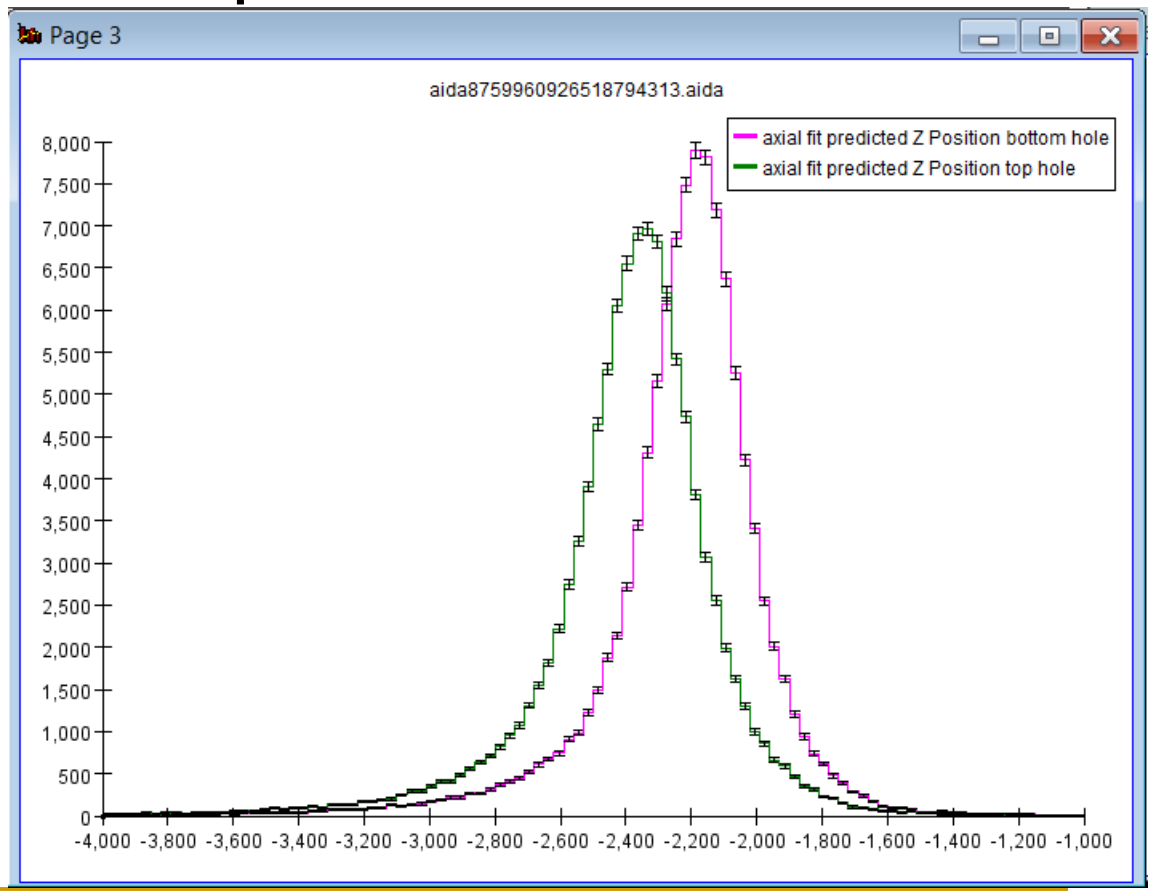
Alignment Moving Forward

- Include beam spot (and ECal?) into alignment procedure for single-track events
- Include vertex constraint for multiple track events
- Include vertex and mass constraint for Møller events

- Ties SVT coordinate system to HPS lab system
- Couples top and bottom halves of detector
- Constrains weak (momentum) mode
- (Sho had introduced some ad-hoc corrections to deal with some of these issues)

Field-Off Straight Tracks

- Project tracks found in both top and bottom SVT back to intersection with Y-Z plane (y=0)
- Plot value of Z at intercept
- Top z = -2346
- Bottom z = -2180
- $\sigma \sim 150$



Tracking Software

- Track finding and fitting were adapted from software developed for generic collider detectors
- Adoption of this software allowed rapid development during the design phase of HPS but required a few compromises
  - Use of a generic geometry definition and pattern-recognition system.
    - Fast for development, not optimized for production.
  - Rotation of our coordinate system to spoof a solenoidal field
  - Use of track parameters not natural for a fixed-target geometry.

Tracking Simulation

- Definition of the SVT support structures could (should?) be improved
- Geometry description has gotten very complicated (just ask Matt Solt)
  - Can this be simplified/improved?
- Will need vastly more MC events for the 2018 running.
- Can we be smarter in our production?
  - Faster?
  - Better?
  - Can we get away with no-pileup MC?

Fast Monte Carlo

- Takashi has a simple detector simulation which has proven to be very useful in quickly answering a number of questions related to acceptance and occupancies
  - Geant3-based
  - Uses Z scoring planes to record particle positions at rough location of SVT sensors and ECal
- It would be useful to incorporate this functionality more closely into our software/analysis framework
  - Tighter coupling of existing framework with HPS geometry and EDM
  - Reproduce functionality within hps-java

Software CPU Performance

- Our tracking software is SLOW!
  - Not currently an issue, but will definitely become critical during long 2018 run
  - Maurik has shown timing breakdown, but there has been no appreciable action to-date
  - Overall CPU budget dominated by tracking, primarily track-finding/fitting, followed by raw hit-fitting
    - Fix what we have
    - Start over
- Fitting readout samples to determine hit time and pulse height
  - Currently using generic minuit fit
  - Need to evaluate possible gains from a dedicated fitter

Pattern Recognition

- Currently begin by creating 3D spacepoints from stereo pairs and finding tracks *ab initio* via strategies.
- Revisit our track-finding strategies to incorporate new L0 in the upgrade
- Support “4-layer” tracking, both to identify WABs and recoil electron tracks

Pattern Recognition

- Possible improvements:
  - Improved axial/stereo matching (L4-L6)
  - Improved and/or more strategies using 3D points
  - Cluster-seeded tracking
    - ECal cluster position and energy define a trajectory which originates from the beam-spot (HPS Note 2015-006).
    - Find tracks consistent with that hypothesis.
  - Implement pattern recognition based on 1D strip hits.
    - No “ghost” hits, or parallax issues
    - Could see increased efficiency by not requiring hits in both axial and stereo layers per station.

Track Fitting

- Track fit quality is not chi-squared distributed
- Discrepancy between data and MC
- Resolution of issues complicated by:
  - Strip cluster position
  - Module position (alignment)
  - Track extrapolation (non-uniform field)
  - Multiple scattering and energy loss
- GBL refit could benefit from external review
- Whole chain needs better documentation
  - Javadoc on what the code is expected to be doing
  - Documentation on procedure, algorithm, math

Vertex Fitting

- Currently using Billoir's implementation of a Kalman fit using perigee parameters.
  - Presupposes that track states have been defined "close" to the actual vertex.
    - Our track states defined at z=0
  - Fitting assumes Gaussian uncertainties with correct covariance matrices
    - Current track parameter pulls not normally distributed
    - ∴ no surprise that vertex resolution is off
    - ∴ no surprise that vertex pulls not normally distributed
- Implementing another algorithm would be very useful as a cross-check.
- Documentation needed

Unit and Integrated Tests

- Our software management system includes a testing mechanism supporting both unit and integrated test suites
- In principle, this provides some protection against software changes affecting performance
- In practice, we only have sketchy test coverage and many (most?) of the tests do not throw assertions on (un)expected behavior
- Will need to assure this system is robust before engaging in any major code refactoring/rewriting

Manpower

- Many of the principal developers of the tracking/vertexing software have moved on
- Opportunities abound for individuals or institutions to contribute, either improving existing software or developing/implementing new code.

Summary

- Current code and algorithms are working, but…
- Improvements to the Tracking and Vertexing feed directly into improvements in the bump-hunt and vertex analyses and our discovery reach
- Major changes are unlikely for the analysis of data already taken, but will be necessary for physics run in 2018.
  - We have a little over a year to get ready.
    - + Much can be done in that time
    - - Much needs to get done in that time
- Great opportunities for new contributors