SVT Tracking (& Alignment)

SW Review 1.27.2014

Per Hansson Adrian

U.S. DEPARTMENT OF ENERGY
Office of Science

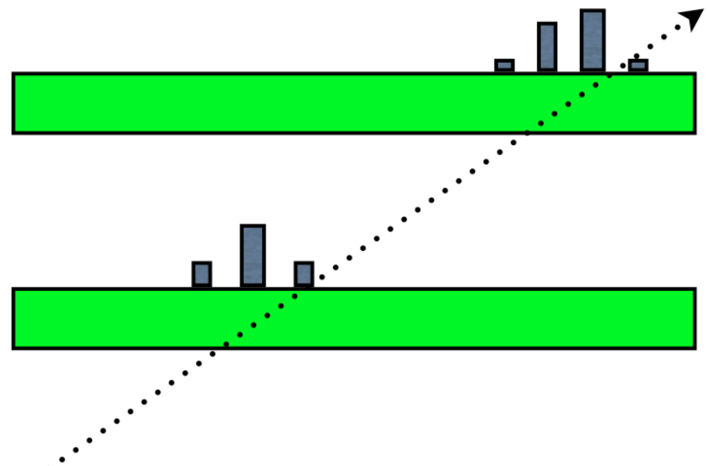
SLAC NATIONAL ACCELERATOR LABORATORY

Outline

- Hit reconstruction
- Track finding
- Track fitting
- Alignment
- Magnetic field
- Special runs
- Performance analysis

(Sensor) Hit Reconstruction

Build clusters from sensor strips

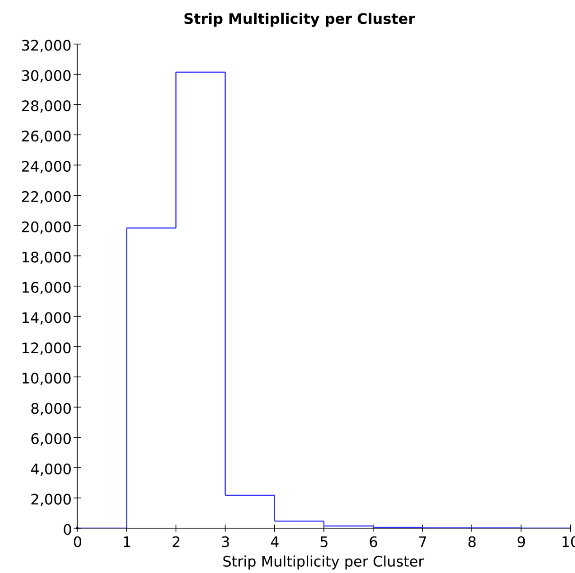


Nearest neighbor algorithm (1D)

1. Find seed strip ($S>4\times\sigma_{noise}$)
2. Add neighbors with $S>3\times\sigma_{noise}$ until strip found with $S<3\times\sigma_{noise}$
3. Repeat 1,2 until no seed strips found
4. Reject clusters with $S<4\times\sigma_{noise}$

Output strip clusters contains

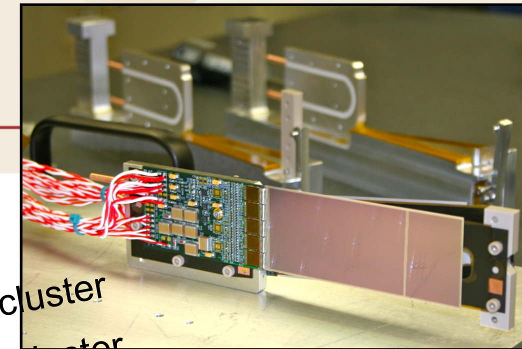
- Position (pulse height weighted mean)
- Cluster time (pulse height weighted mean)



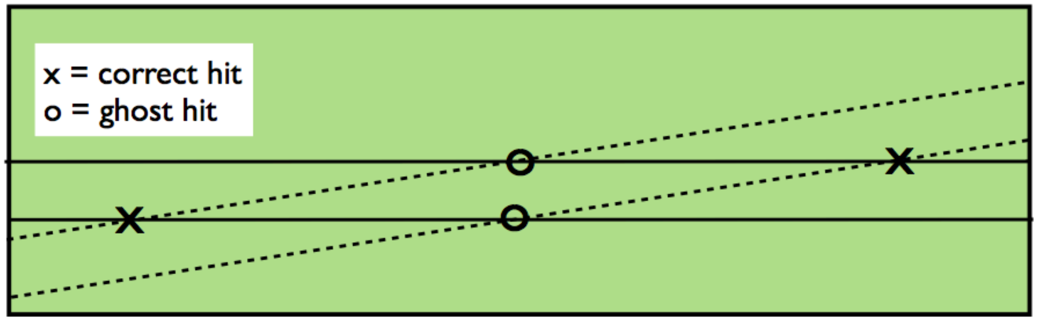
⇒ Need to worry about overlapping clusters
⇒ Currently, offline track selection on distance to neighbor

(Stereo) Hit Reconstruction

Test run stereo pair module

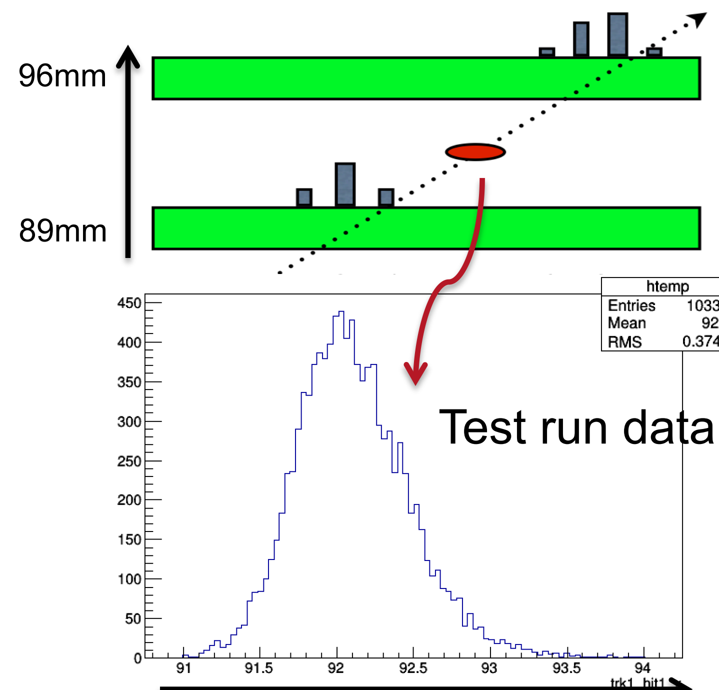


Build 3D hits from (2D) strip clusters



Take all combinations of clusters in adjacent stereo pair sensors to build “stereo hits”

- Starting 3D hit position is taken as midway between clusters
- Reject *very* bad combinations (not pointing to target)
- Stereo hit positions are updated with track direction in track finding/fitting



Track Finding

Inherited from linear collider simulation (lcsim “seed tracker”)

- Seed-confirm-extend philosophy
- Very fast: test often, reject early
- Based entirely on stereo hits

Track finding is governed using a “Strategy”

```
<Strategy name="HelicalTrackHit Strategy">
    <!--Cutoffs-->

    <MinPT>0.050</MinPT>
    <MinHits>4</MinHits>
    <MinConfirm>1</MinConfirm>

    <MaxDCA>80.0</MaxDCA>
    <MaxZ0>80.0</MaxZ0>

    <MaxChisq>25.0</MaxChisq>
    <BadHitChisq>10.0</BadHitChisq>

    <!--Layers-->
    <Layers>
        <Layer type="Seed" layer_number="1" detector_name="Tracker" be_flag="BARREL" />
        <Layer type="Seed" layer_number="3" detector_name="Tracker" be_flag="BARREL" />
        <Layer type="Seed" layer_number="5" detector_name="Tracker" be_flag="BARREL" />
        <Layer type="Confirm" layer_number="7" detector_name="Tracker" be_flag="BARREL" />
        <Layer type="Extend" layer_number="9" detector_name="Tracker" be_flag="BARREL" />
    </Layers>
</Strategy>
```

1. Fit a 3-hit track seed using stereo hits
2. Reject if failing strategy cuts
3. Add hits from confirm layers
4. Reject if failing strategy cuts
5. Add hit from extend layers, reject if worse chi2
6. Reject if failing chi2 and # hits

Remove overlapping tracks (shared hits <=1)

V. Karimaki, Computer Physics Communications 69 (1992) 133—141

Track Fitting

SLAC

Fit track in two independent views (const. magnetic field)

- Circle fit in the “bend plane”
- Straight line fit in non-bend plane

Both are fast non-iterative fit algorithms

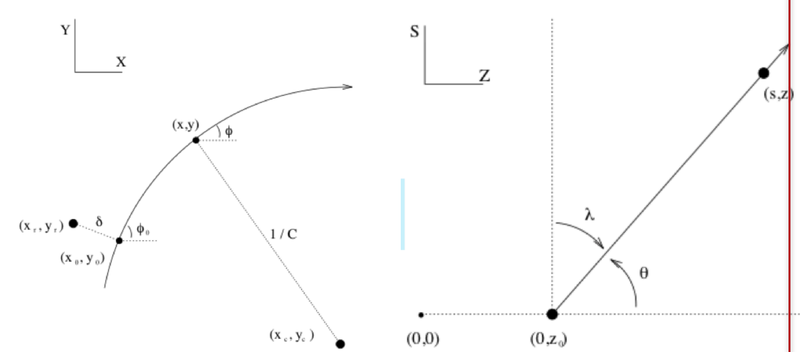
- Parameter estimations
- Covariance matrix
- (Seed)Track finding uses these algorithms at each step

⇒ Merge final fit into a “helix” track object together with the hits of the track

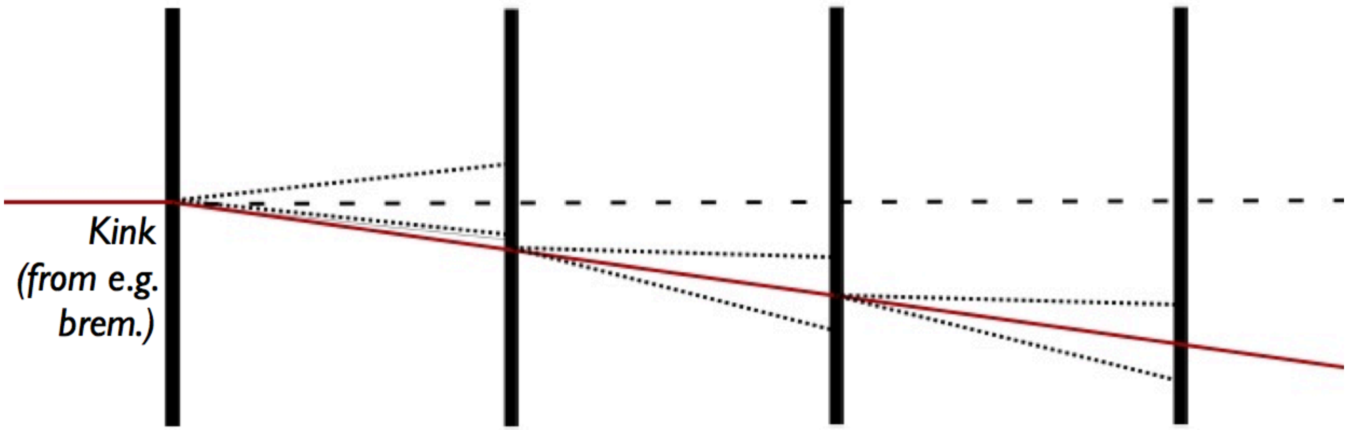
Parameterization and conventions inherited from lcsim

⇒ B-field in z-direction, beam in x

⇒ Rotation from natural coord. system



Multiple Scattering Model



Hit uncertainty at each layer

- Multiple scattering (MS) uncertainty and spatial resolution added in quadrature
- MS uncertainty from each previous layer are added in quadrature
- No account for correlations across scattering planes or energy loss

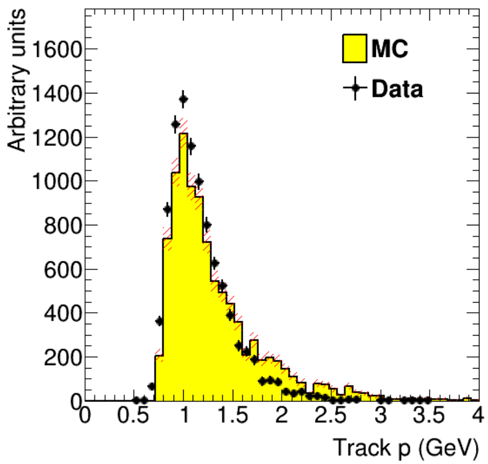
MS uncertainty is on average correct but not an optimal fit

- Good enough for an initial fit

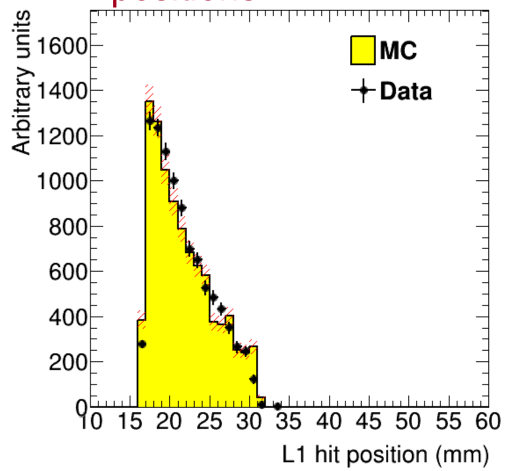
⇒ Different (standard) ways to deal with this problem

Tracking works

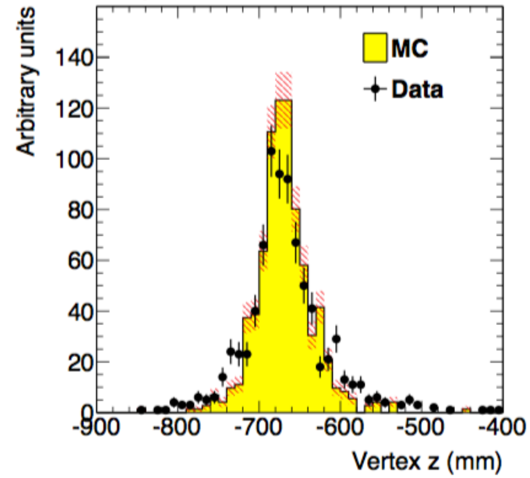
Track momentum



Vertical stereo hit positions

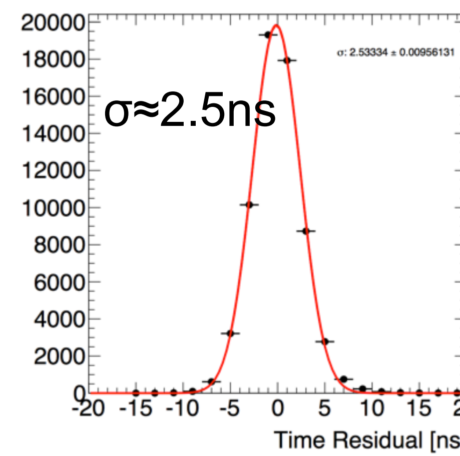


Converter (vertex) position



Tracking software already exercised in Test run

- Used in both online monitoring and offline analysis
- Good performance
- Speed exercised fully in mock data challenge



⇒ The basic software for HPS operation is already there

Tracking Software for 2014

Test run proved that tracking software works

Topics we'd like to improve

- Better handling of multiple scattering
- Track-based alignment
- Inhomogeneous magnetic field

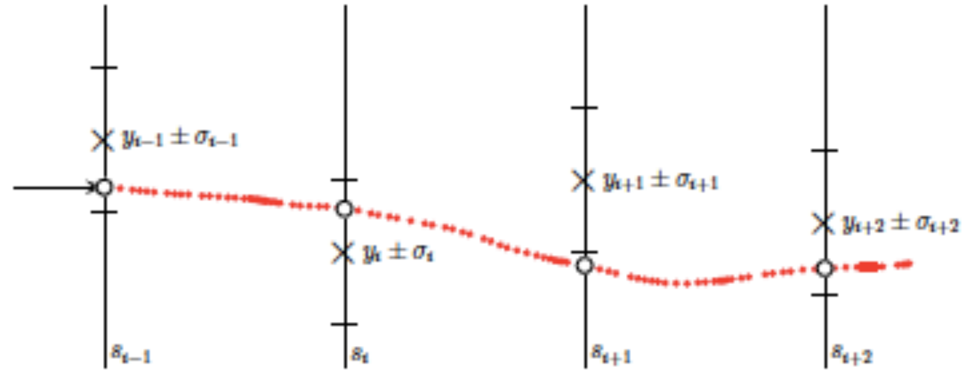
Performance analysis

- Momentum scale
- Momentum resolution

Generalized Broken Lines (GBL)

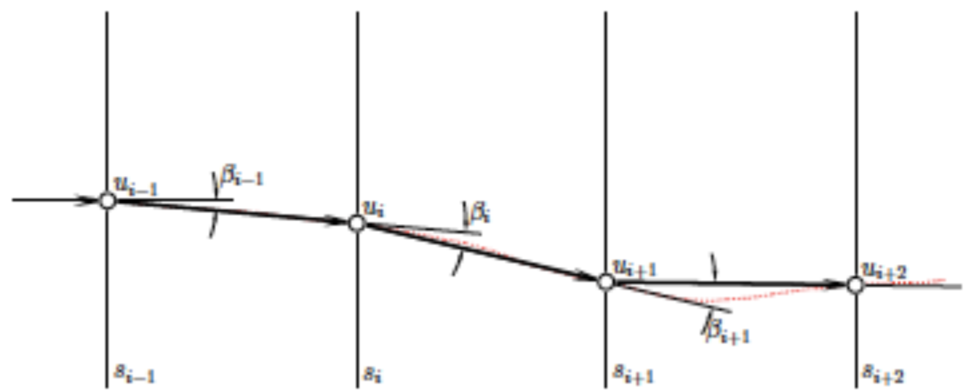
Generalized Broken Lines (GBL)

- A track fit with multiple scattering
- Widely used, e.g. CMS detector alignment



GBL is a track refit

- Initial fit to estimate residuals and momentum (using SeedTracker)
- Use residuals and estimated momentum, in a second fit that includes multiple scattering
- Covariance matrix of all track parameters are available (at each point)



Iteration needed for energy loss

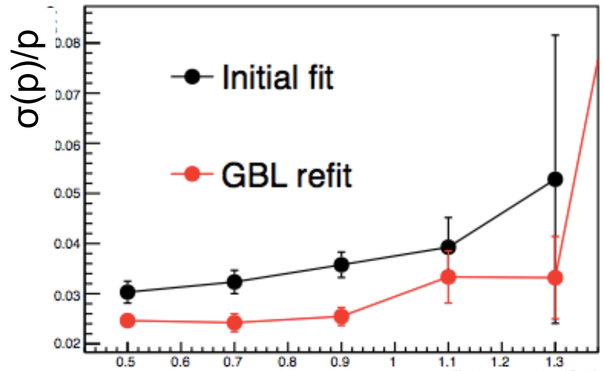
$$S(u) = \sum_{i=1}^{n} \frac{(y_i - u_i)^2}{\sigma_i^2} + \sum_{i=2}^{n-1} \frac{\beta_i^2}{\sigma_{\beta,i}^2}$$

⇒ Alignment software (Millepede-II) "supported"

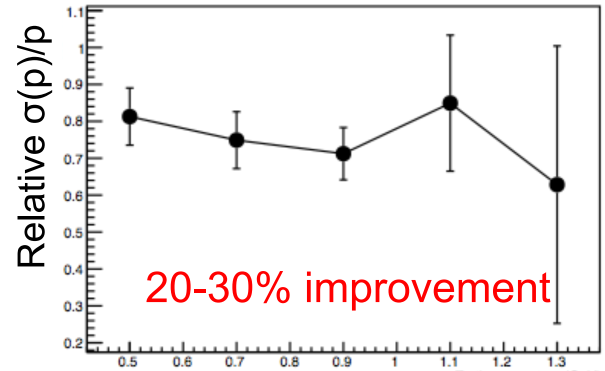
GBL Already Implemented

Momentum resolution

A' (40MeV) events

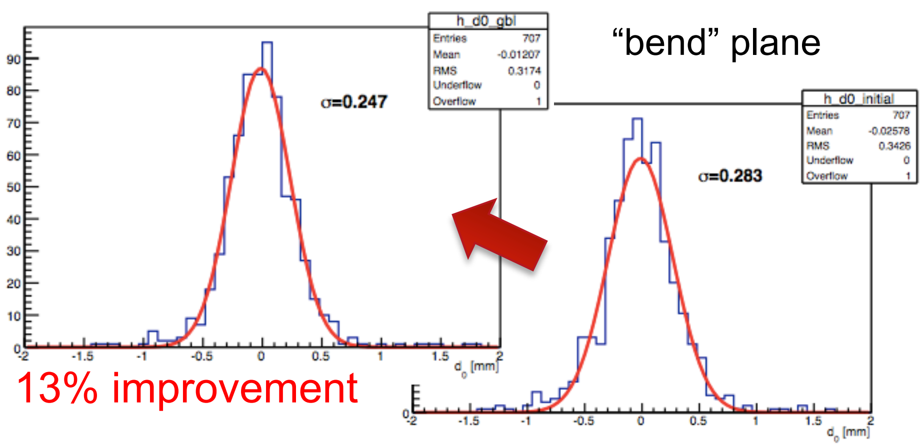


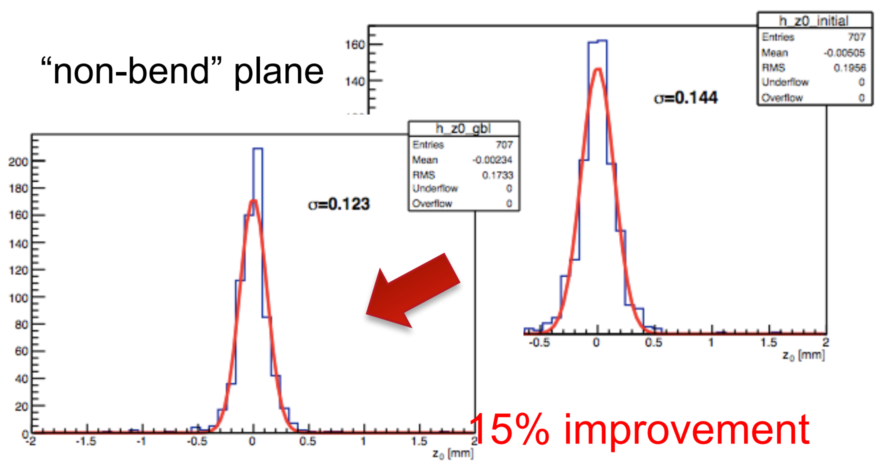
Track momentum (GeV)



Track momentum (GeV)

Impact parameter resolution





Currently implemented in python (used here) and C++

⇒ would like to port to Java, but not critical

Alignment

Roughly, needs to be significantly better than single hit resolution (~<5um)

- Use survey + track-based alignment
- Test run alignment experience

Silicon module survey (optical)

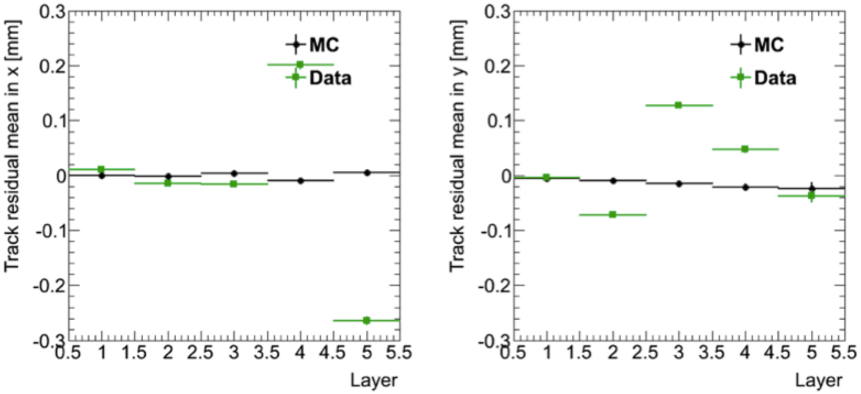
⇒ Relate sensors to support plate

Support structure survey (touch probe, post-assembly)

⇒ Relates support plates to support structure

Beamline survey

⇒ Relate support structure to beam



Mean of biased track residuals vs tracker layer

⇒ Scheme worked (residuals < 300um) in Test run (w/ some pain)

⇒ Need to improve geometry description

Track-based Alignment

V. Blobel, C. Kleinwort, F. Meier, Computer Phys. Communications (2011)
Kleinwort, NIM A, 673 (2012), 107-110

SLAC

Tracking detector alignment is a standard problem; multiple ways to achieve similar performance

Our approach:

- Do a least square fit of local (track) and global (alignment) parameters
- Millepede-II can do this for us and is “supported” by GBL
- Great support from C. Kleinwort (GBL/Millepede developer)

MPII

Residual z_i for measurement *i*:

5 track parameters

Subset Ω of *n* alignment constants

$$z_i = y_i - f(x_i, \mathbf{q}, \mathbf{p}) = \sum_{j=1}^{5} \left(\frac{\partial f}{\partial q_j} \right) \Delta q_j + \sum_{l=1}^{\Omega} \left(\frac{\partial f}{\partial p_l} \right) \Delta p_l$$

Minimize a “chi2” from entire data sample

$$\chi^2 = F(\mathbf{q}, \mathbf{p}) = \sum_i \left(\frac{(y_i - f(x_i, \mathbf{q}_j, \mathbf{p}))^2}{\sigma_i^2} \right)$$

⇒ Newton minimization problem with large # parameters

⇒ single iteration for **linear** least squares

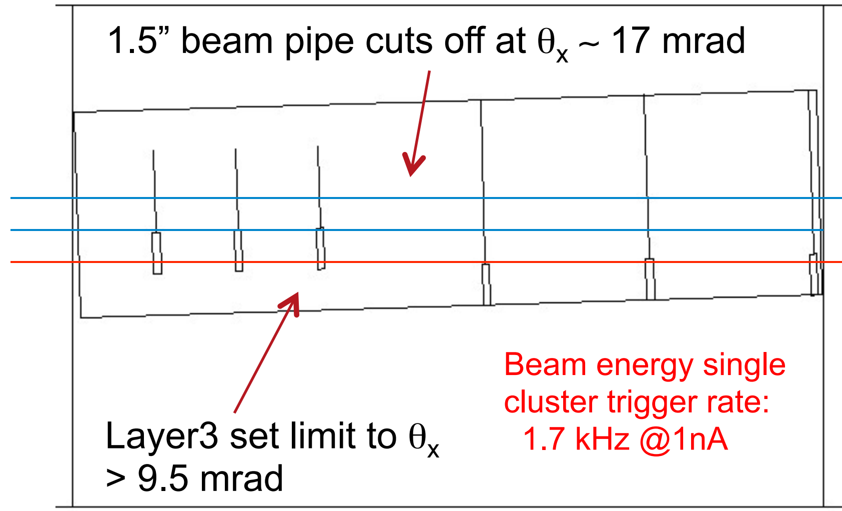
⇒ Millepede’s strength is reducing the dimension of the matrix to be computed to give alignment parameters corrections only

Track-based Alignment – Tools & Operation

Current status

- All derivatives other inputs with GBL track fit is implemented
- It runs and we can update constants in geometry…
- Next big job is to figure out minimal set of global parameters -> bootstrap geometry

⇒ Test run detector is an ideal test bed used for this



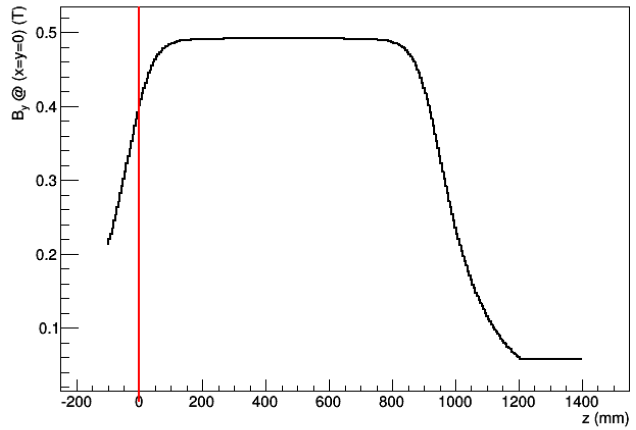
Geometry tools	• Bootstrap geometry from survey constants • Constraints for Millepede-II minimization
Special runs	• Fully simulate straight line track runs • Determine trigger and sample size needed
Operational procedures	• Need to determine (sub-)set of constants after moving the detector • Monitoring – rapid feedback during run (beam spot, chi2, track matching) • Dedicated offline shifter

3D Magnetic Field

Primary use case is vertexing
- Target sits in fringe field
- At a minimum we need (By)@(x,z)

Already have this

Existing fringe field desc.



Existing 3D magnetic field support
- Input (Bx,By,Bz)@(x,y,z) on cartesian grid
- Linear interpolation between box of points
- Geometry code to handle field map exists
- Track propagation in inhomogeneous field with Runge-Kutta method

⇒ Need to be integrated and tested with vertexing software

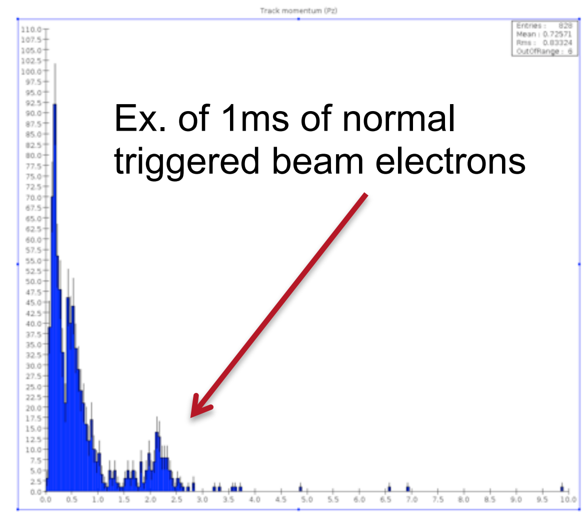
⇒ Not clear we need full 3D map (needs testing)

Momentum Scale and Resolution

SLAC

Couple of ideas exists

- Beam energy electrons “easiest”
- Trident events “for free”; but kinematic fit may be non-trivial
- Elastic peak? (dedicated targets and trigger)



⇒ All of them need more preparatory work/proof of concept…

Type	Purpose	Trigger
Beam energy electrons	Scale, resolution	• Signal trigger (out of time tracks) • Prescaled dedicated single trigger (for coverage)
Trident kinematic fit	Scale (,resolution)	• Signal trigger
Elastic peak	Scale, resolution	• Dedicated CH2/C target • Single electron trigger

Summary

Tracking software was exercised in Test run

- Fast non-iterative fits – speed should not be a problem
- Multiple scattering handled in refit with GBL
- Part of online monitoring for data-taking success
- Track-based alignment is ongoing work

Physics requirements are satisfied

- Test run: S/N (spatial res.), hit time res. and hit eff.
- Assumes track-based alignment is successful

Risks

- Operational success not very dependent on current developments
- Some risk if alignment framework not fully exercised

Manpower & liaisons

- Official overlap with SVT DAQ software
- Large overlap of people in SVT and beamline groups
- We need newcomers – but not critical for operation!

Topic	Manpower
sw infrastructure	McCormick, Moreno, Uemura
General tracking	Graham, Hansson, Graf
GBL	Hansson
Alignment	Hansson, Graham, Nelson
Geometry, B-field	Uemura, Graf, Graham, McCormick
Liaisons	
Production	Uemura
Monitoring	Moreno
SVT DAQ	Hansson, Moreno
SVT	Nelson, Hansson
Beamline	Hansson (overlap)

Backup

Overview of the track reconstruction chain