SVT Alignment: status and update

PF

09/15/2020

U.S. DEPARTMENT OF ENERGY | Stanford University

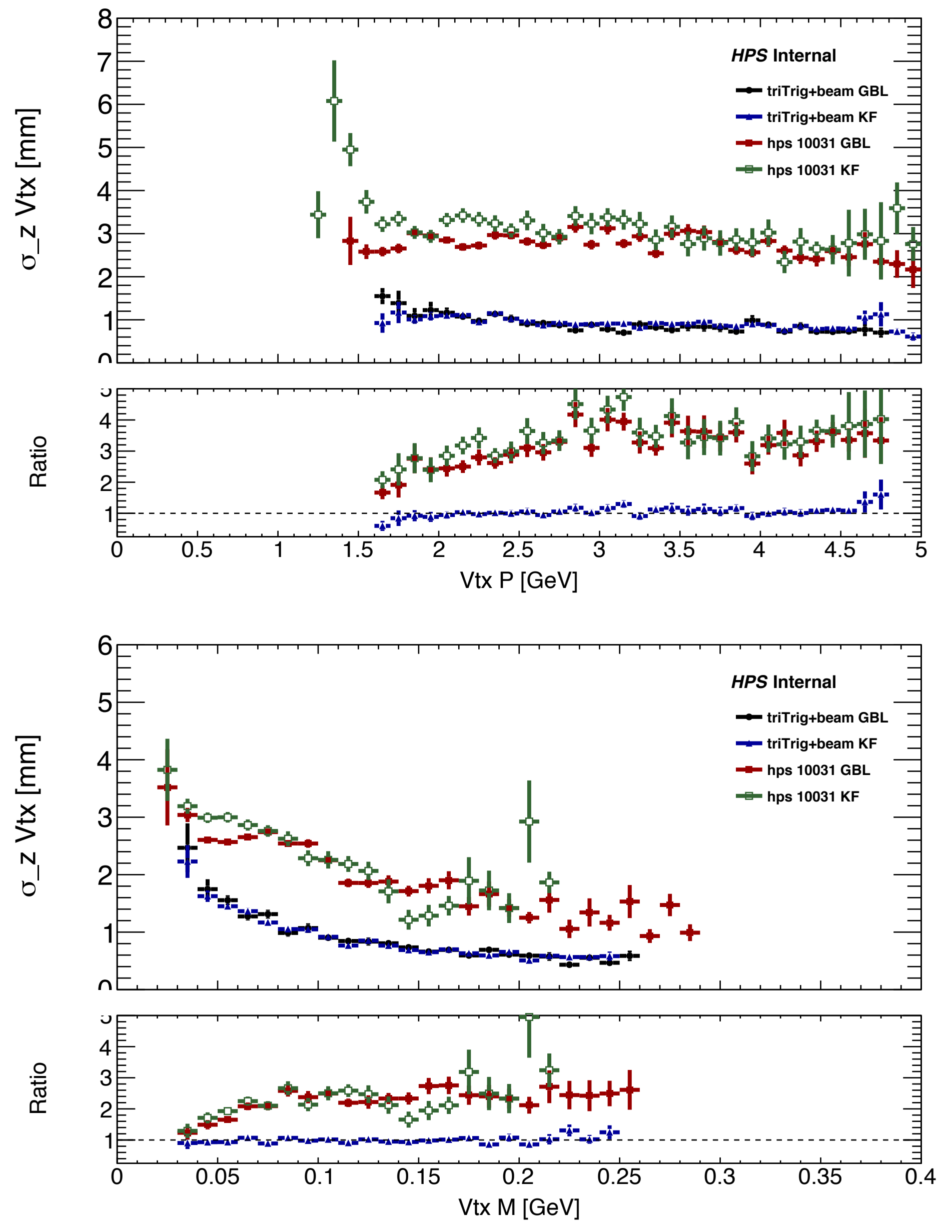
SLAC NATIONAL ACCELERATOR LABORATORY

Outline

- **Recap from previous results at workshops/collaboration meeting - May 2020**
  - Results from the April analysis workshop
  - Detector alignment from last collaboration meeting
- **Introduction**
  - Issues when combining global alignment with local alignment with MPII
  - Datasets and external constraints available
- **Updates to alignment framework in hps-java**
  - Updates to alignment monitoring
  - Mathematical formalism and implementation in hps-java of:
    - Global structures alignment
    - Impact parameters constraints
    - External point constraints (i.e. beamspot location)
  - Test on single electron MCs
  - Application on 2019 FEE data
- **Current SVT calibration statusTim**
  - Vertex resolution
  - FEE momentum scale and resolution
- **Conclusion, next steps and timescale**

Introduction - Highlights from April's workshop

- First results on 2019 data and MC readiness were shown at the April's 2019 Analysis Workshop
- Details on the selection in the backup
- These plots were made on a large fraction 10031 events and triTrig+beam generated by TT back in end of March. No skims back then
- MC reproduces the expected resolution plot produced before the upgrade (see slide 25 of this talk)
- Alignment is top priority for 2019 data processing
- First results have shown a x2-3 worse resolution wrt trident MC + beam



Introduction - Calibration Data and MC samples

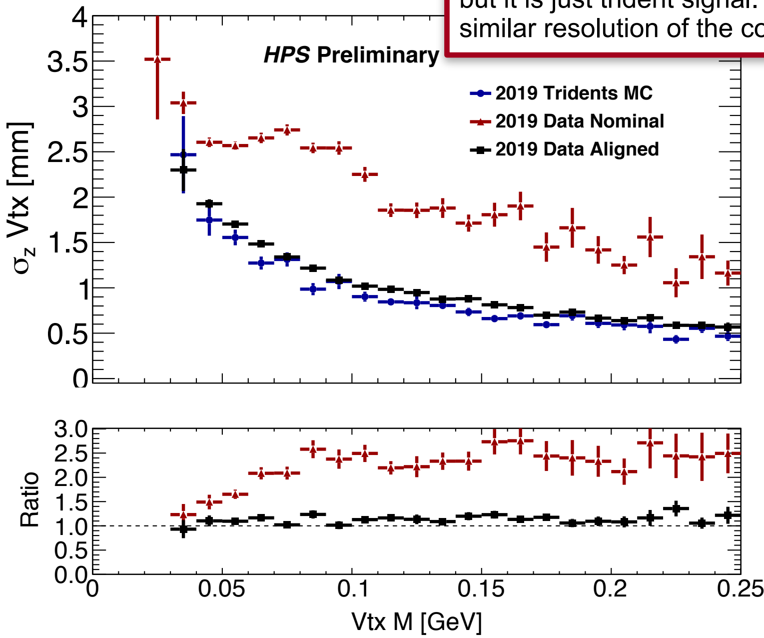
SLAC

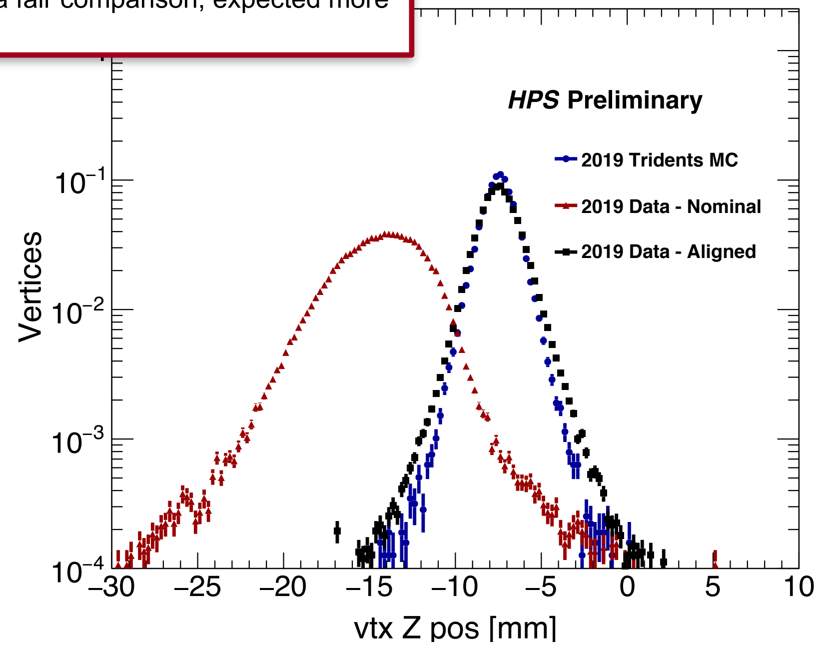
- A set of samples have been selected for the SVT calibration:
  - Full Energy Electron (FEE) trigger: 10103 and 10104 B-Field ON
  - FEE trigger: 10101 B-Field OFF
    - FEE have high momenta tracks to minimise MCS
  - V0 skims: 10031 both with Ecal Cluster on Track (V0Skims) or without (V0SkimsLoose)
    - Illuminate both electron (hole) and positron (slot) sensors
- The data sets information is summarised SVT Alignment Skims
- In addition MC samples used for checking perfect geometry are (for the moment):
  - Tridents (TriTrig): signal only and signal + beam overlay

- All through future talks on alignment I'll use L1-L7 nomenclature.

Results provided to Jeopardy

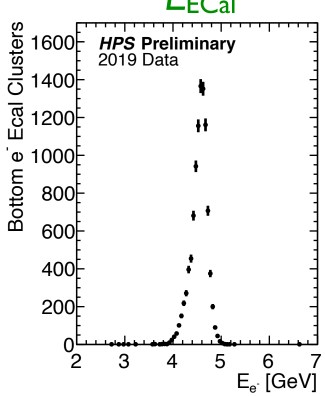
"Optimistic" MC Simulation, as does not contain beam (pile-up) but it is just trident signal. With a fair comparison, expected more similar resolution of the core



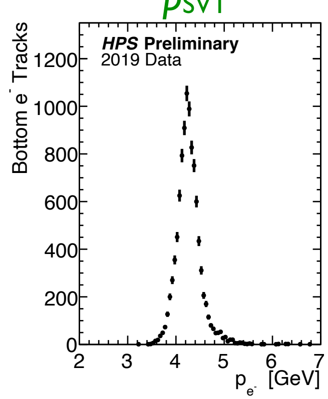


full energy electrons

E_{ECal}

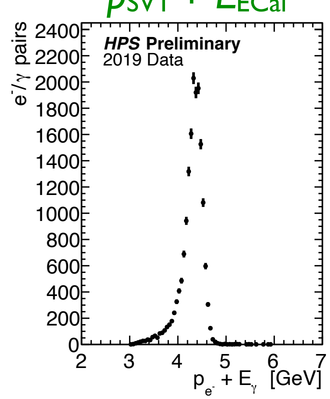


p_{SVT}



e^-+ brem γ

p_{SVT} + E_{ECal}

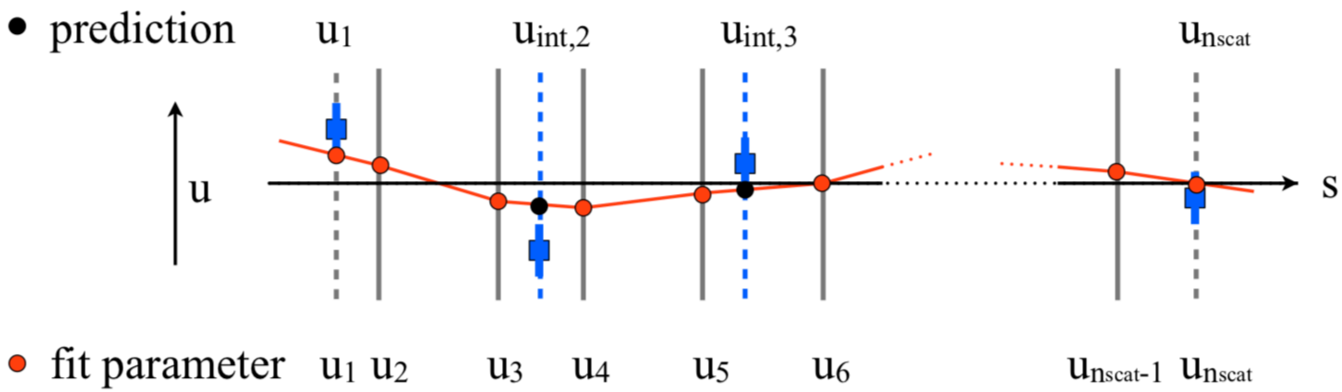


- Achieved:
  - Factor ~3 improvement in vertex resolution (almost at the level of no-pileup MC simulation). Large tail reduction.
  - Work is ongoing for momentum scale: alignment difficulties arise due to charge-asymmetry statistics when using elastically scattered electrons
- Results were **delivered in time for the HPS experiment review** for 2021 at [PAC 48 Jeopardy Meeting]

SVT Alignment procedure

GBL Tracking - Recap

- General Broken Lines (GBL) is a track refit algorithm that add the description of multiple scattering to an initial trajectory
  - Based on propagation in magnetic field
  - Constructed from a sequence of thin scatterers
  - In the case of silicon detector a scatter also has a measurement (in the form of local residual in the sensitive u direction)



- The initial trajectory should be ‘close enough’ to the solution and provide a reasonable estimate of the particle trajectory
- GBL is used in hps-java to refit helical track fits
- It is iterated (5 iterations) in our code to ensure convergence of the track parameters corrections

Introduction - SW status and readiness - just for reference

- Majority of alignment software is in place since 2016 alignment campaign.
- We use hps-java with custom steering files for producing
  - Output monitoring files - ROOT format - hps-DQ-macros
  - Millepede input files for local alignment for hps-mille
  - SLCIO files for dedicated analysis of the results using hpstr
- Work in the past month has been made on the alignment chain:
  - GBL Code review for global derivatives for local alignment
  - Fix our MPII wrappers for 2019 geometry. MPII can now run on 2019 data/MC
  - Tests on MC misalignments for validation
  - Use of pre-fitted hits for faster processing of iterations
  - Improved monitoring plots/tools and collect all available monitoring drivers useful for alignment purposes
- More informations available 2019 HPS Alignment Notes

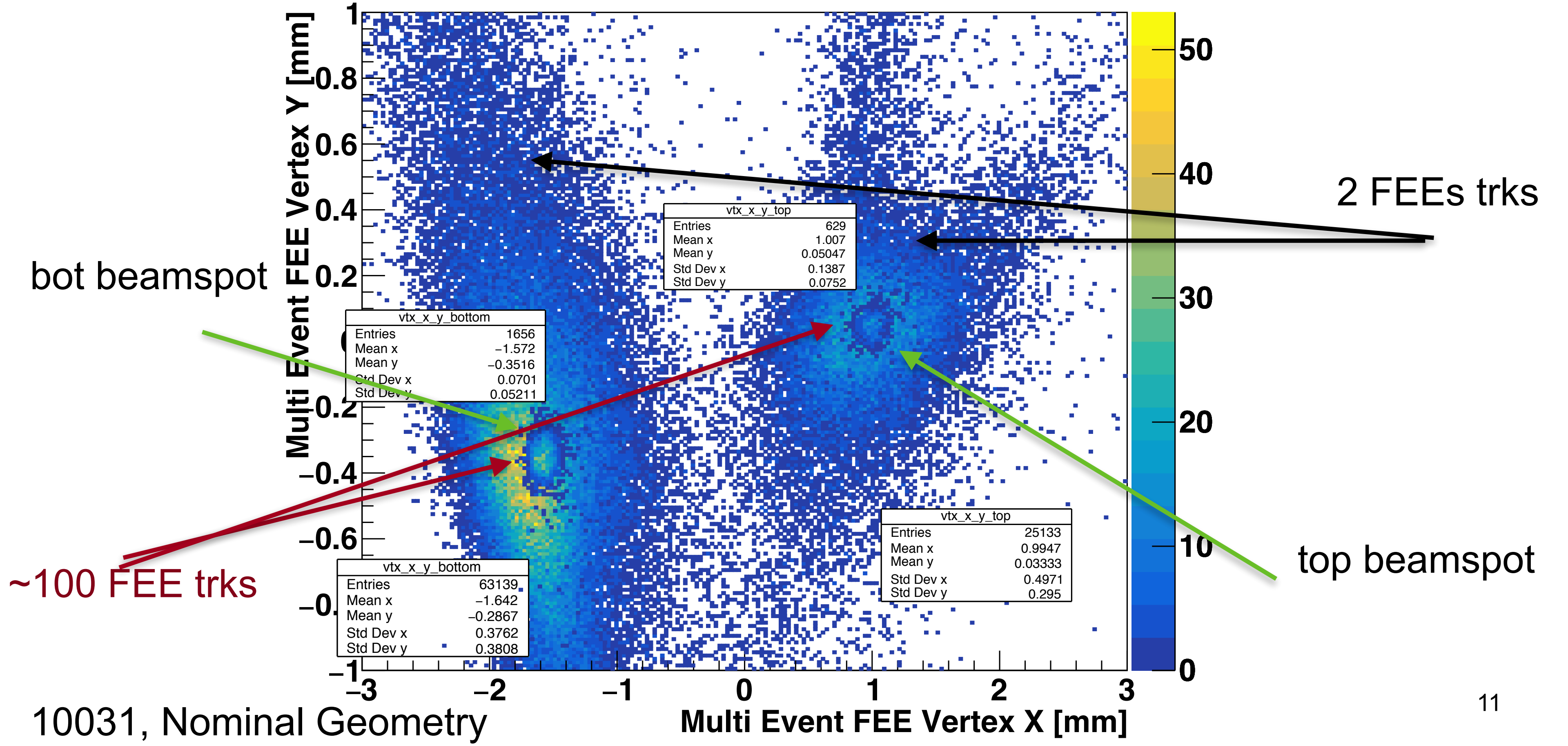
HPS Alignment strategy

- HPS geometry is implemented in the software without a direct support for MPII global structures alignment
- Since 2016, the strategy to align the detector was divided in aligning first global structures, i.e. front vs back of the detector, top/bottom angles and relative positions … and then MPII was invoked for aligning the single sensors
- I will go through
  - Updates to the alignment monitoring
  - Updates to the alignment framework
  - Current performance on V0 skims
  - Current Momentum scale and resolution
  - Next steps

Alignment monitoring updates

Global Alignment - Multi events FEE vertexing

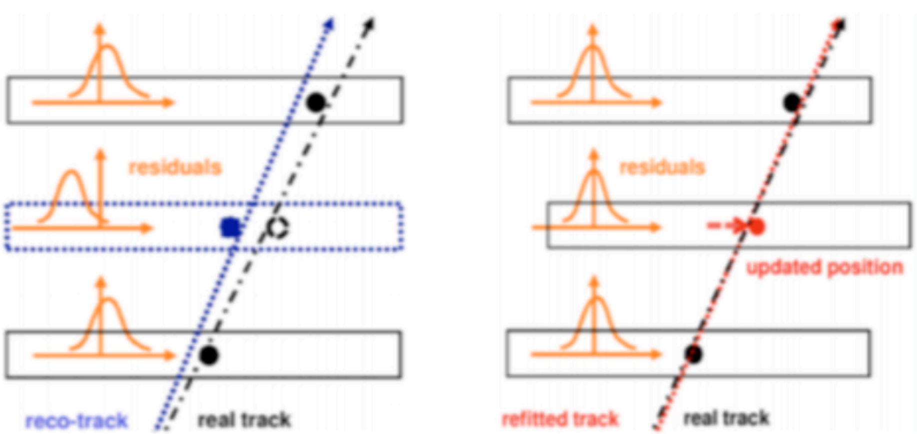
- First, we updated the Multi Event FEE Vertexer to accept more than 2 tracks per event
- Clear effect on the x-y position resolution wrt 2-tracks vertices
- Events are collected, vertices are fitted in 100 tracks chunks, or less if not available: i.e. if 150 tracks are found 2 vertices are formed with 100 and 50 tracks, respectively.
- This can be extended to e+/e- pairs from multiple events in order to exploit tracks with opposite curvature.
- **In case of top-bottom consistent alignment, the locations of the separate beamspots should coincide**

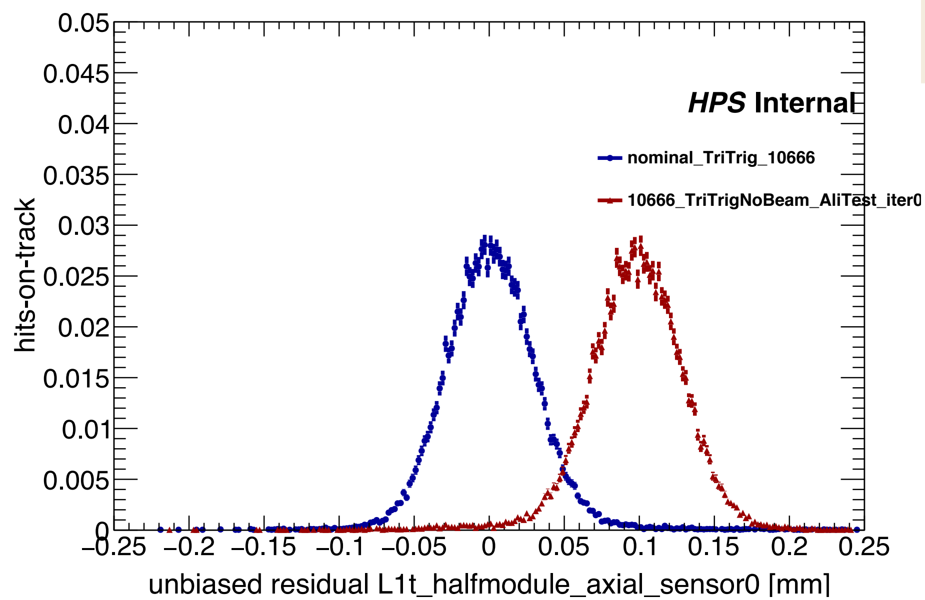


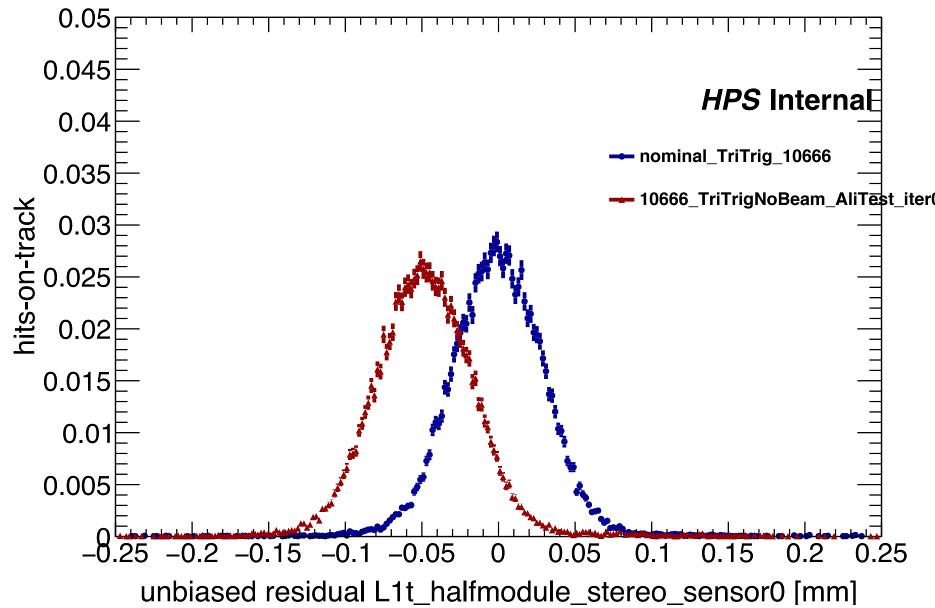
10031, Nominal Geometry

Unbiased residuals

- The unbiased residuals show the degree of misalignment for each sensor / module / structure
- The unbiased residual is obtained by refitting the tracks removing the hit we want to check the hit-on-track residual: now done automatically via GBL C++ library via JNA
- In this example: MC tridents, moved **L1 top axial sensor** of $du = +100\ \mu$m to validate alignment chain
- The size of misalignment can be identified by GBL unbiased residuals
- The bias in the stereo residual is due to the reflection of the axial side misalignment -> **need to be recognised as correctly placed in the solution**

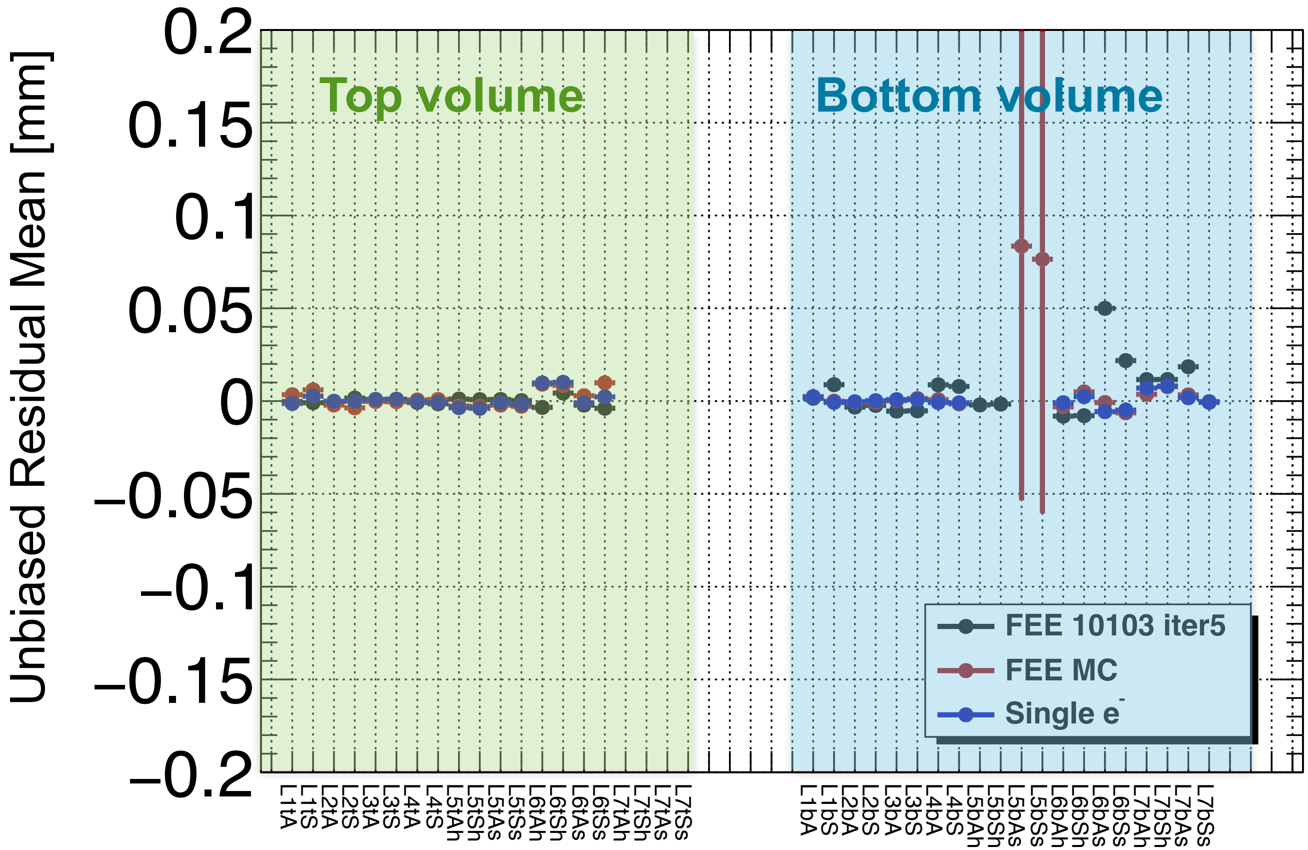






Unbiased residuals maps and kinks maps

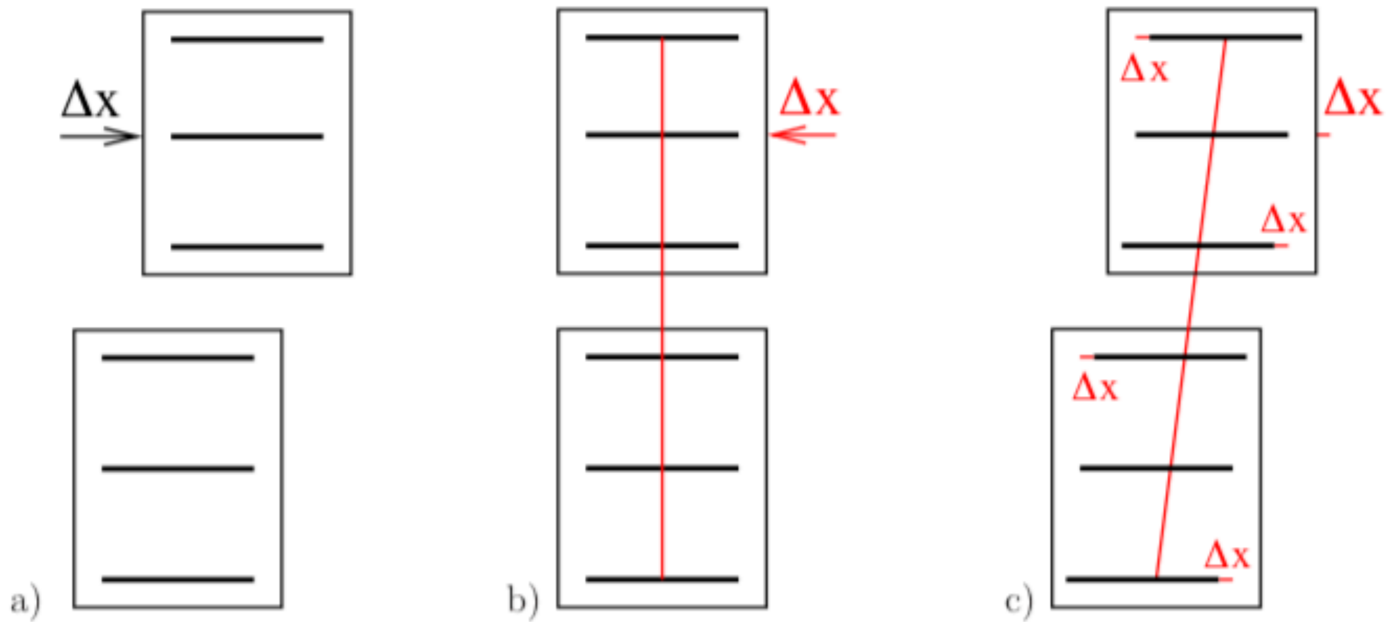
- These plots show the mean value of the unbiased residuals and of the kink distributions
- In perfect alignment case we expect to see residuals and kinks centred at 0



One point for each sensor

Alignment framework updates - global structures alignment

Why global structures first?



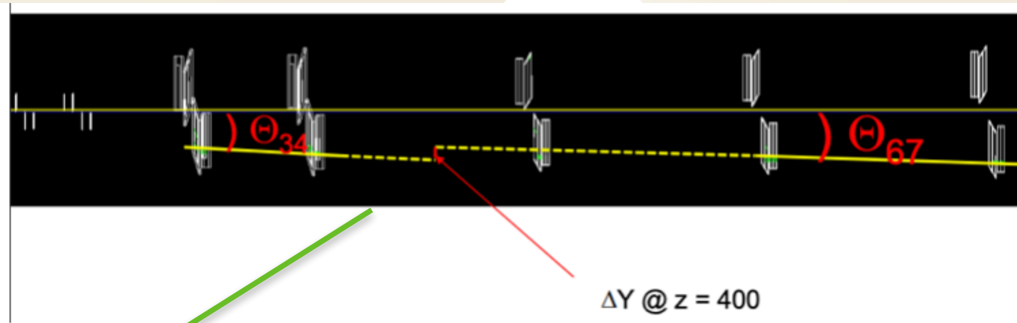
- Illustration of possible misalignment in a telescope.
- b is (a possible) solution if sub-telescopes are preferred
- c is (a possible) solution if single sensors are preferred
- In reality it depends of various factors including:

  - **Constraints** (what moves what not)

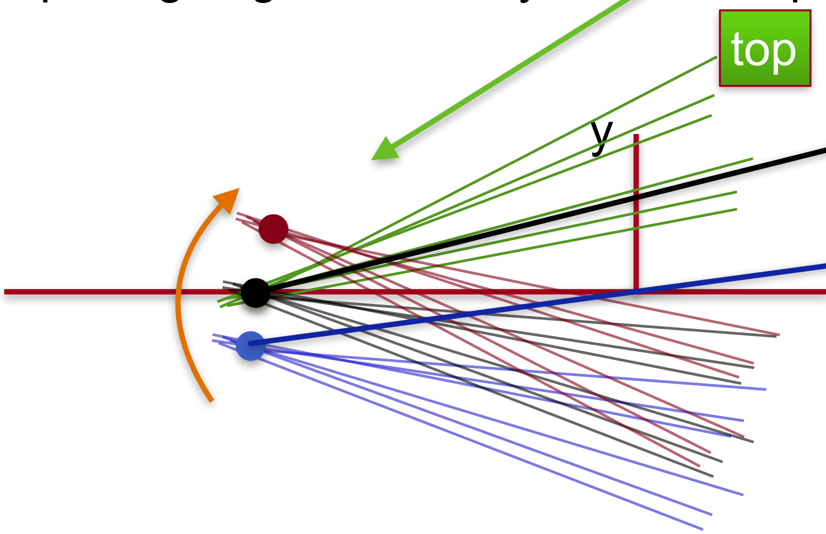
  - **Initial sensor position uncertainty** (we don't use any information on initial uncertainty in MPII solution)

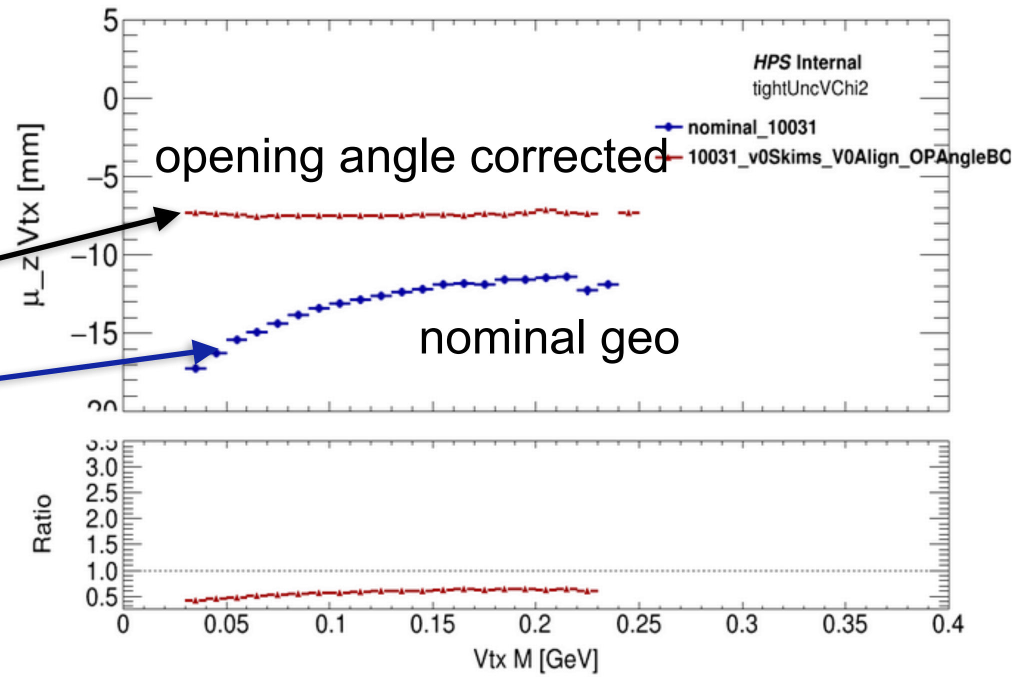
The 2016 alignment strategy didn’t work in 2019 data

- **In my opinion** 2016 strategy is not optimal and I had difficulties to use it in 2019.
- The example shows how moving the front of the detector can fix the vertex location as function of the bottom opening angle.
  - Top is fixed (green tracks)
  - Bottom is moved: blue to black to red tracks
  - When the tracks point in the same location in global Y, then the dependence on mass (opening angle between tracks) disappears



Opening angle can easily fix the z-dep



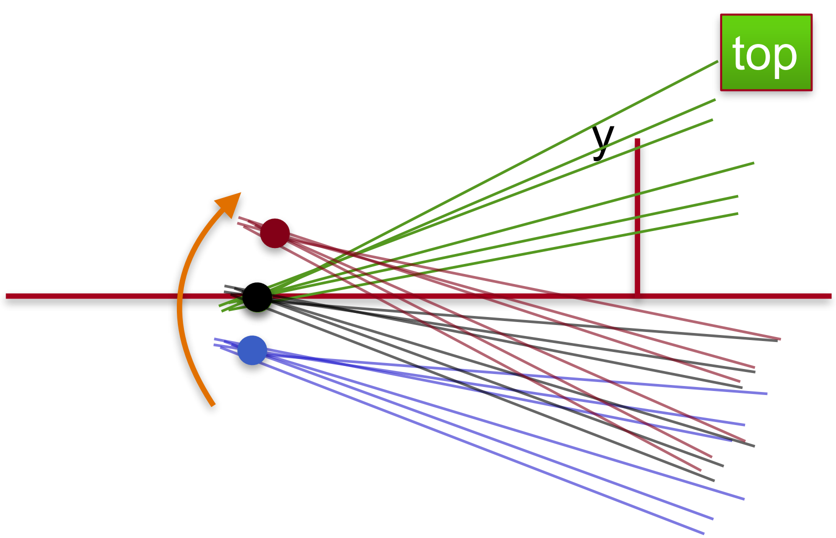


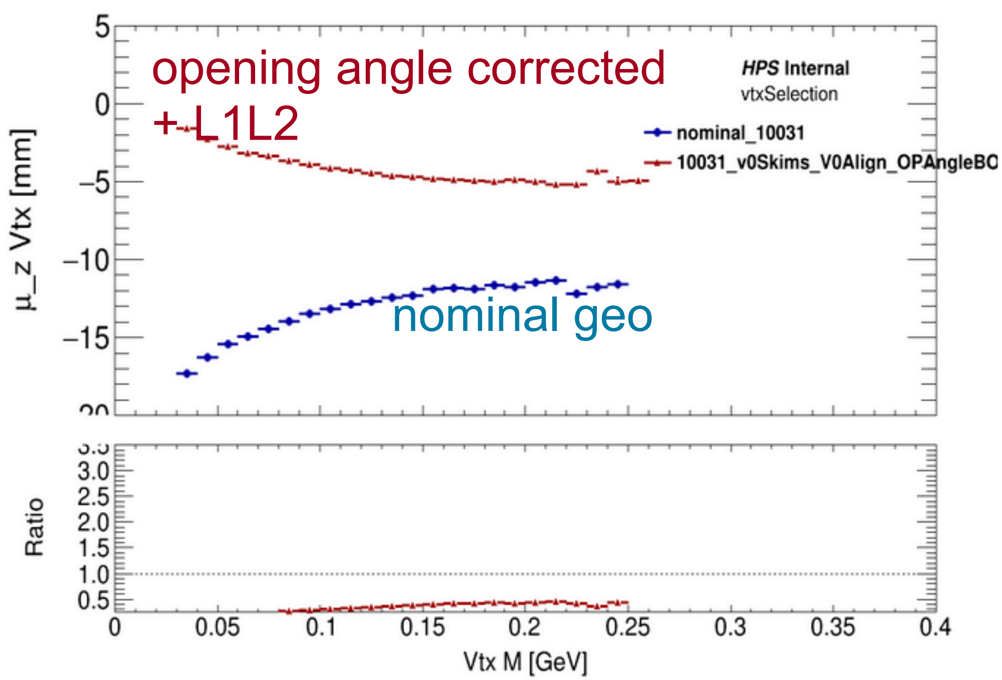
Why 2016 strategy is not working in 2019 detector

- The original implementation of local alignment via Millepede-II doesn't know that global structures are aligned first:
  - Will still try to minimise the unconstrained χ^2 no guarantee that this is the correct solution and local alignments can move globally, i.e. **all stereo move in positive u => global X movement of a Uchannel.**
- **In particular single sensors can move freely, without keeping the pre-aligned global structures.**
- **Bottom line: there isn't a direct way to align global and local structures at the same time and biases can be re-introduced in the procedure**

… and can be reintroduced just aligning L1L2

- Here is shown the following procedure:
  - After opening angle alignment tried to align the Tu of the first two layers
- The innermost layers try to move minimising the Chi2 and that leads to reintroducing a bias in the vertex position as function of invariant mass





Implementation of global structures alignment

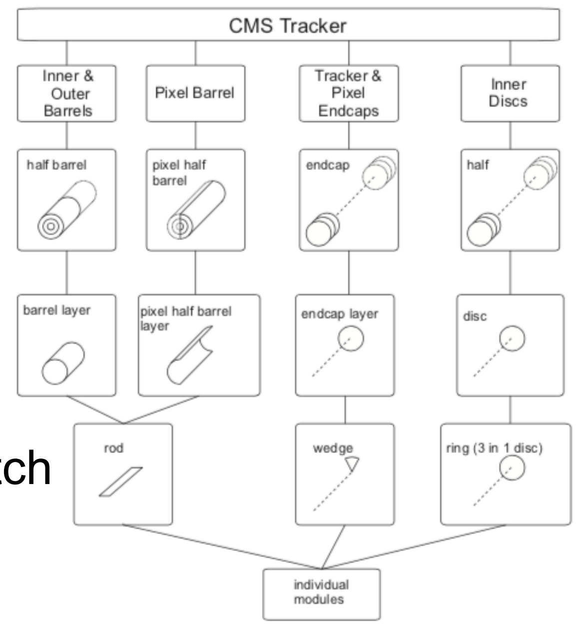
- In order to perform the single modules alignment keeping the global structures in place a formalism on how to constrain the MPII solution should be implemented
- This is a common framework to help solving the track based alignment problem and necessary to reach a solution which maintains the constraints through the various steps
- The necessary ingredients are:
  - **The whole procedure should be done by MPII itself, without external steps**
  - **Computation of the hit-on-track residuals derivatives with respect to the global structures translations and rotations**
  - **Update of the hps-java alignment framework to support for new alignable structures and of hps-mille for the MPII interface**

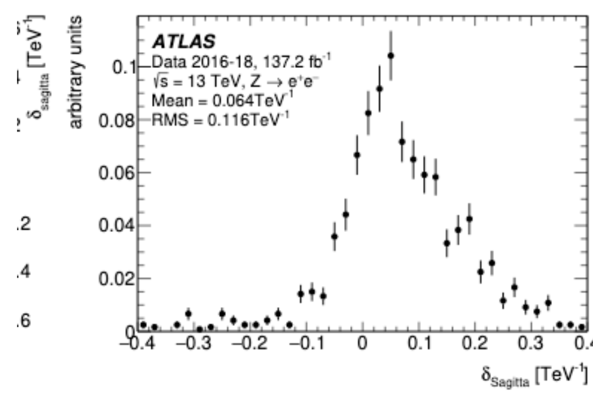
How to improve the current alignment procedure

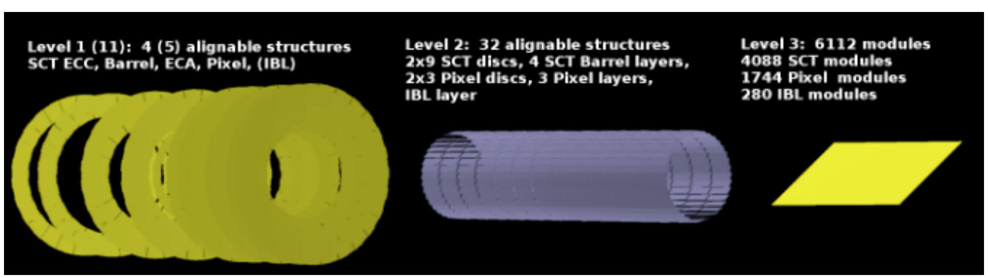
- Implement an hierarchical alignment procedure:
  - **Same way to solve global and local misalignments**: just accumulate all information and decide which structure we want to align.
  - **Sensor positions and orientations will be relative to composite structures** and there is a **natural way to include constraints** to the solution.
  - **Composite structures will be aligned minimising the global** χ^2 and correlations between DoF should be taken care of.
- Introduce external constraints to reduce weak modes:
  - **Beamspot, calorimeter E or beam energy, survey measurements and impact parameters**
- Use a combination of BFieldON/OFF tracks to align single sensor to remove curvature weak mode.

- This procedure is a standard in solving the alignment problem and has been used successfully by other experiments

CMS sketch







ATLAS sketch

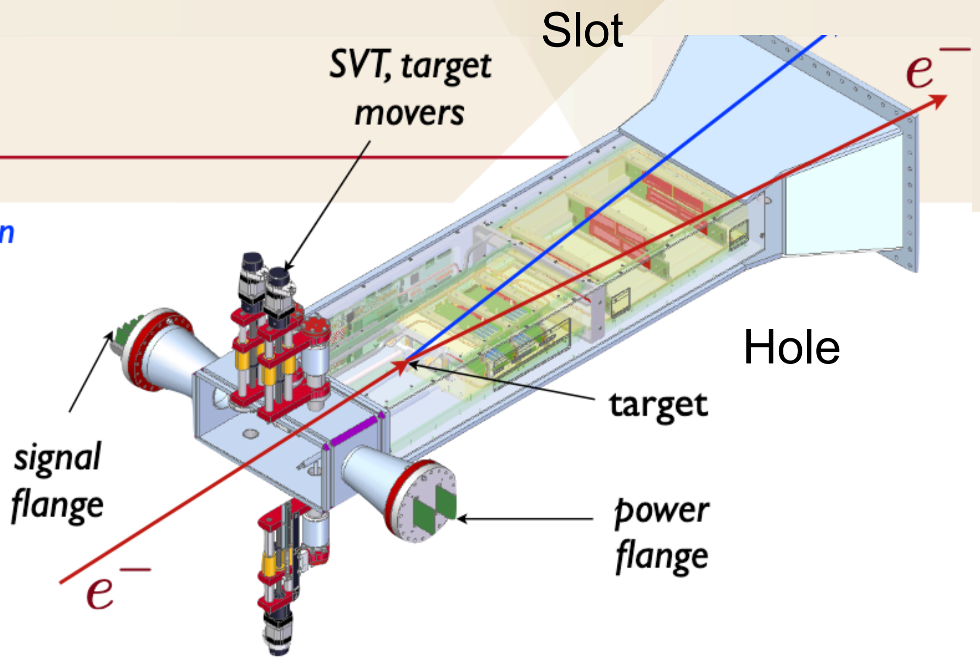
$$\delta_{\text{sagitta}} = \frac{\langle E/p' \rangle^{+} - \langle E/p' \rangle^{-}}{2\,\langle E_{\text{T}} \rangle}.$$

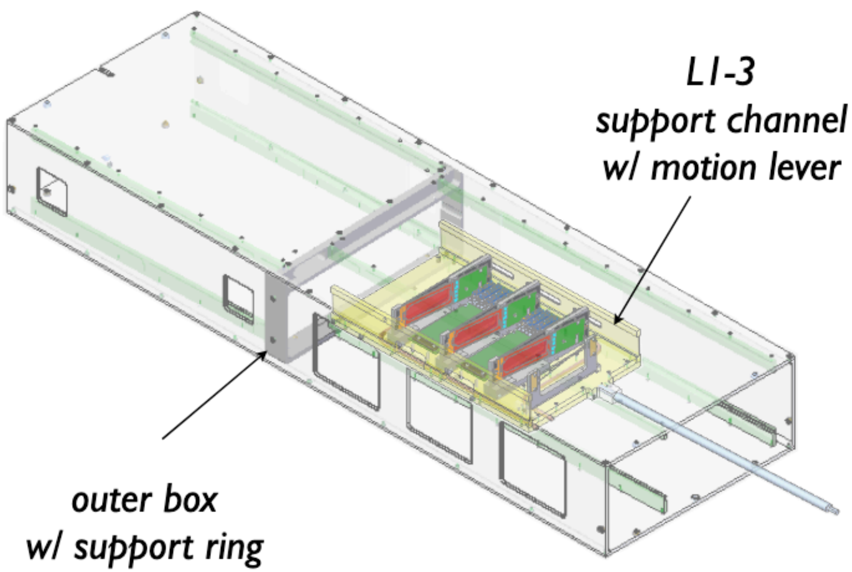
The HPS SVT

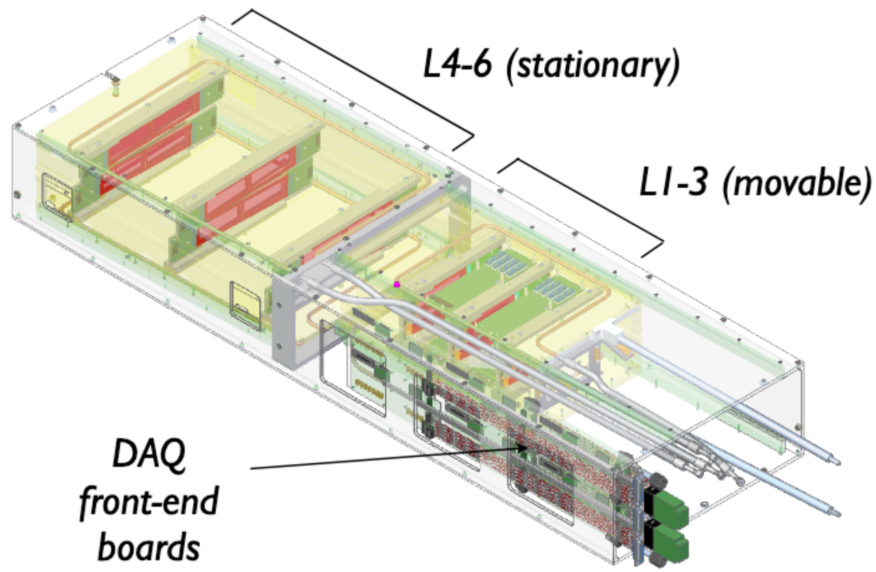
7 double-layers of silicon strips, each plane measures position (~6-10 μm) and time (~2 ns) with ~0.2% – 0.35% X_0/hit.

Operates in an extreme environment:

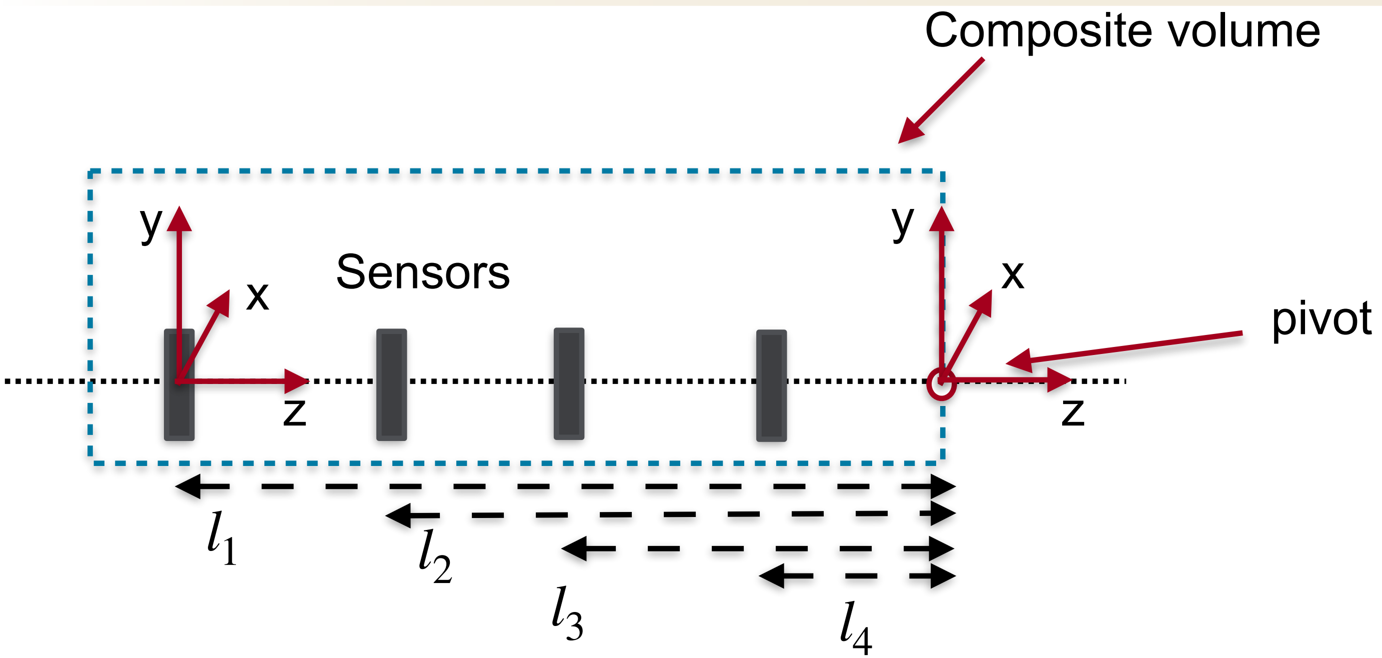
- beam vacuum and 1.5 Tesla magnetic field
  ⇒ constrains materials and techniques
- sensor edges 0.5 mm from electron beam in L1
  ⇒ must be movable, serviceable
- sensors see large dose of scattered electrons
  ⇒ must be actively cooled to -20 ˚C
- 24528 channels can output >100 gb/sec
  ⇒ requires fast electronics to process data





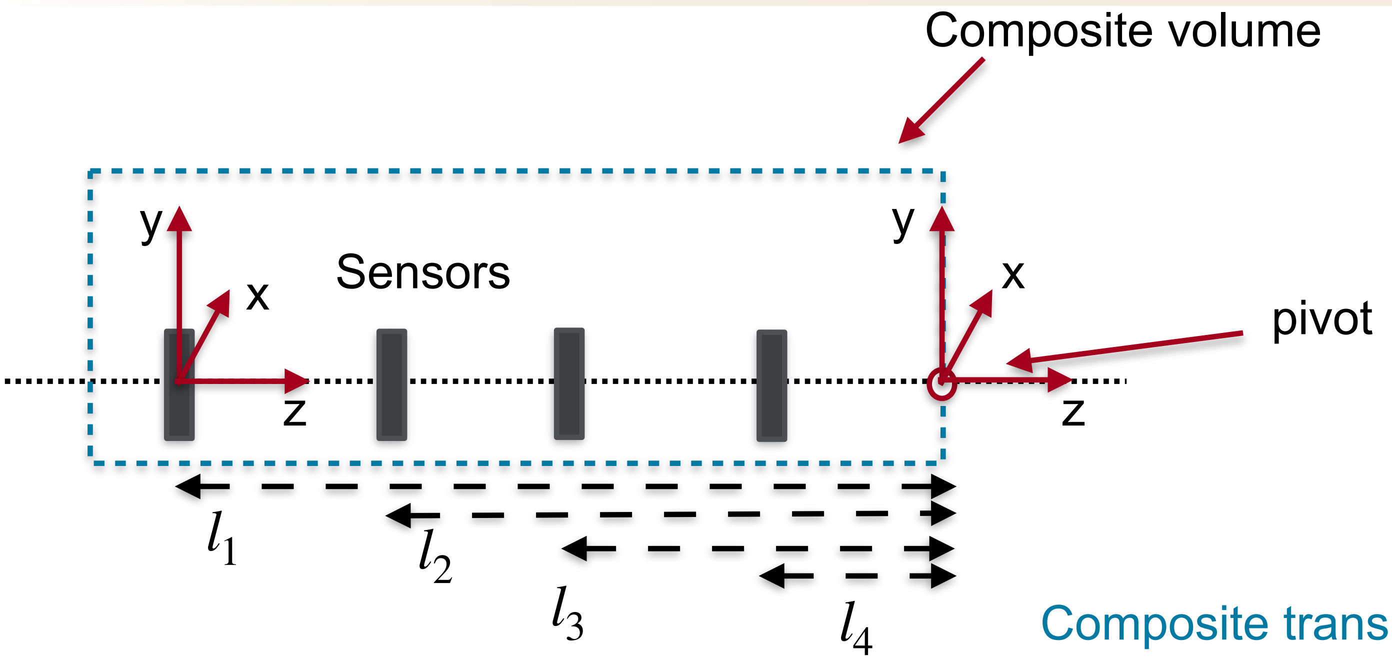


UChannel to sensors relations (simplified example)



$$t_{x,i} = T_x - l_i \cdot R_y$$
$$t_{y,i} = T_y + l_i \cdot R_x$$
$$t_{z,i} = T_z$$
$$r_{x,i} = R_x$$
$$r_{y,i} = R_y$$
$$r_{z,i} = R_z$$

UChannel to sensors relations (simplified example)



$$t_{x,i} = T_x - l_i \cdot R_y$$
$$t_{y,i} = T_y + l_i \cdot R_x$$
$$t_{z,i} = T_z$$
$$r_{x,i} = R_x$$
$$r_{y,i} = R_y$$
$$r_{z,i} = R_z$$

$$\mathrm{a}_i = \begin{pmatrix} t_x \\ t_y \\ t_z \\ r_x \\ r_y \\ r_z \end{pmatrix} \quad C_i = \begin{pmatrix} 1 & 0 & 0 & 0 & -l_i & 0 \\ 0 & 1 & 0 & l_i & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathrm{a}_c = \begin{pmatrix} T_x \\ T_y \\ T_z \\ R_x \\ R_y \\ R_z \end{pmatrix}$$

Composite translation to sub-component translation

Composite rotation to sub-component translation

Composite rotation to sub-component rotation

Composite translations have no effect on sub component rotations

Math behind composite structures alignment

- Hit-on-track residuals are computed in the local coordinates (q) of a sensor and transformed to global frame (r) by

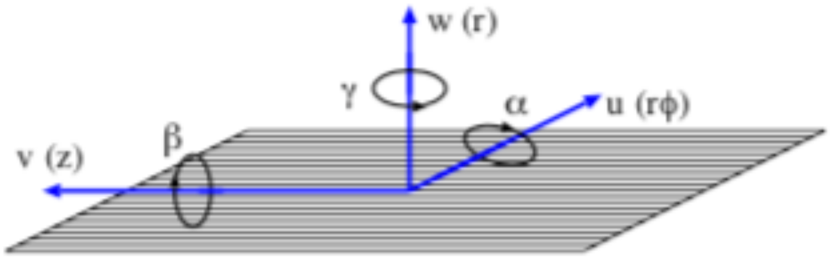
$$\mathbf{r} = \mathbf{R}_\mathbf{s}{}^{T}\mathbf{q} + \mathbf{T}_\mathbf{s}$$

- For individual sensors, alignment corrections are incremental rotations $\mathbf{\Delta R}$ and translations $\mathbf{\Delta q}$ which lead to

$$\mathbf{r} = \mathbf{R}_\mathbf{s}^\mathbf{T}\mathbf{\Delta R}_\mathbf{s}(\mathbf{q} + \mathbf{\Delta q}_\mathbf{s}) + \mathbf{T}_\mathbf{s}$$

- Rotations can be reduced with respect to 3 angles. The alignment parameters become

$$a = (\Delta u \;\; \Delta v \;\; \Delta w \;\; \alpha \;\; \beta \;\; \gamma)$$



u: most sensitive direction
v: least sensitive direction
w: normal to the sensor plane

$$\zeta = \begin{pmatrix} u_r \\ v_r \end{pmatrix} = \begin{pmatrix} u_m \\ v_m \end{pmatrix} - \begin{pmatrix} u_p \\ v_p \end{pmatrix}$$

$$\frac{\partial \zeta}{\partial \mathbf{a}}\Big|_{\mathbf{a}=0} = \mathbf{P}\begin{pmatrix} -1 & 0 & \frac{du_p}{dw} & -v_r\frac{du_p}{dw} & u_r\frac{du_p}{dw} & -v_r \\ 0 & -1 & \frac{dv_p}{dw} & -v_r\frac{dv_p}{dw} & u_r\frac{dv_p}{dw} & u_r \end{pmatrix}$$

Stoye '07

Formalism of composite structures alignment

- Each composite structure has an assigned local coordinate system defined by the orientation matrix $\mathbf{R_c}$ and origin $\mathbf{T_c}$
- The definitions of the composite structure alignment parameters $\mathbf{a_c}$ is the same of the sensor alignment parameters.
- The alignment relations between sub-component to composite structure can be computed by some "simple math" **(see backup)**

- We need to compute the **C-matrices** that translate movements of composite structures to sub-component movements

TransC -> TransS

RotC -> TransS

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{pmatrix}$$

TransC -> RotS => 0

RotC -> RotS => 0

$\mathbf{a}_i = \mathbf{C}_i \mathbf{a}_c$ relation between sub-components to composite corrections

$\frac{\partial \mathbf{r}}{\partial \mathbf{a}_c} = \frac{\partial \mathbf{r}}{\partial \mathbf{a}_i}\frac{\partial \mathbf{a}_i}{\partial \mathbf{a}_c} = \frac{\partial \mathbf{r}}{\partial \mathbf{a}_i}\mathbf{C}_i$ relation between composite to sub-component residual derivatives

$\mathbf{a}_c = \sum^{i=n} \mathbf{C}_i^{-1}\mathbf{a}_i$ Inverse relation between sub-components to composite corrections

$0 = \sum_{i=0}^{i=n} \mathbf{C}_i^{-1}\mathbf{a}_i$ Natural hierarchical constraint: sub-components movements keep the pre-aligned global structure fixed. Constraint format supported by MPII

Current scenario of HPS Alignable structures - Just FYI

SLAC

- Here is reported the set of orientations R and origins T **(*)** for possible alignable structures as it is implemented in the current HPS geometry code
- **Notice:**
  - The 30.5mrad at module level in our geometry structure
  - The modules are located **far** from the sensors and from the support rings (large rot-to-trans cross terms in the C-matrices)
- An alignable structure is just a container of a Rotation and a translation
- C matrices can be computed in a recursive way.
- Tracking volume can be made alienable with identity rotation and null translation

(*) local to global is $R^T q + T$

Alignable Support Ring Top (aka SVT-front)

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad T = [-117.33, 56.857, 417.79]$$

UChannel46 top (aka SVT-back) - check this

$$R = \begin{bmatrix} 0.9995 & 0.0 & -0.0305 \\ 0.0305 & 0 & 0.9995 \\ 0 & -1 & 0.0 \end{bmatrix} \quad T = [14.995, 8.4230, 491.84]$$

Alignable Module Top L1

$$R = \begin{bmatrix} 0 & 1 & 0 \\ 0.9995 & 0 & -0.0304 \\ -0.0304 & 0 & -0.9995 \end{bmatrix} \quad \boxed{T = [-122.61, 59.820, 36.284]}$$

Alignable Sensor Axial L1

$$R = \begin{bmatrix} 0 & 1 & 0 \\ 0.9995 & 0 & -0.0304 \\ -0.0304 & 0 & -0.9995 \end{bmatrix} \quad \boxed{T = [1.1566, 7.8106, 38.366]}$$

Alignable Sensor Stereo L1

$$R = \begin{bmatrix} 0.0998 & 0.995 & -0.0031 \\ -0.995 & 0.0998 & 0.0303 \\ 0.0304 & 0 & 0.9995 \end{bmatrix} \quad \boxed{T = [2.1622, 7.7995, 45.934]}$$

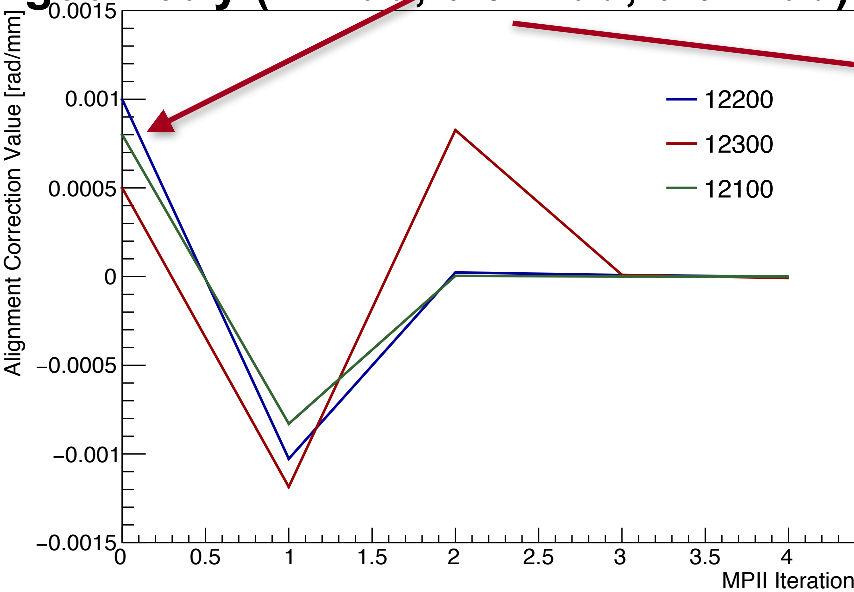
Global Front UChannel alignment test

- **There are 2 Uchannel structures,** front (MillePede ID =80) and back (MPID=90)
- They are characterised in the geometry by 6 MPID parameters, 3rot, 3tr.
  - For example: front U-Channel:
    - **11180** (T_u), **11280** (T_v), **11380** (T_w)
    - **12180** (R_u or opening angle), **12280** (R_v or yaw angle), **12380** (R_w or roll Angle)
- In the following plots MPID=00 is used instead of 80, but is the same structure.
- The derivatives of the UChannels movements are given by the sensor derivatives times the **C-matrices**
- I've started cross-checking the MPII global solution using a misalignment geometry. The starting point is:
  - TOP UChannel misaligned: R_u (R_x) = +0.8 mrad, R_v (R_z) = 1mrad , R_w (R_y) = +0.5 mrad
  - MPII solution obtained keeping the back UChannel and others dof fixed. Use of outlier suppression + Matrix Inversion (small number of Dofs)
  - Set up 4 iterations of accumulation + solving.

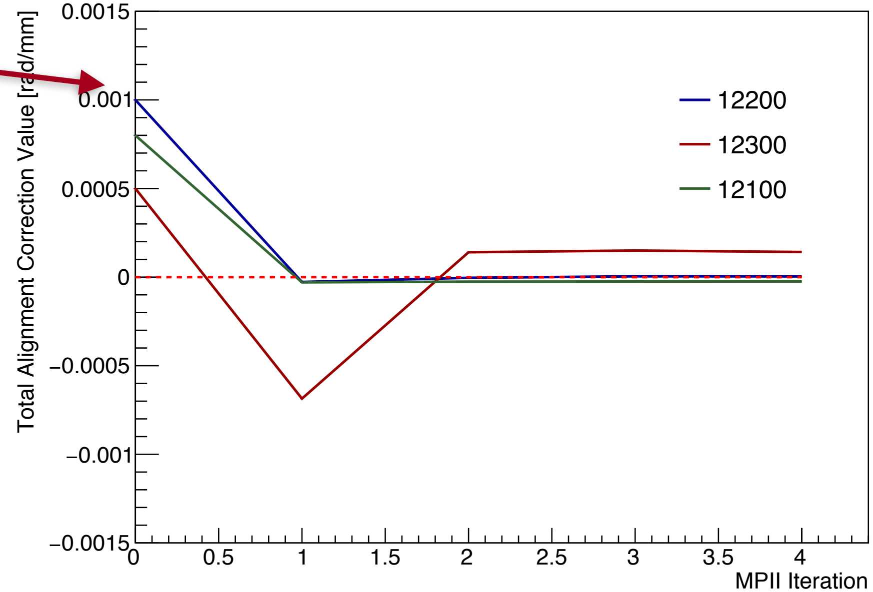
Global rotations MPII corrections convergence check

SLAC

Initial misalignments on perfect geometry (1mrad, 0.8mrad, 0.5mrad)



MPII correction at each iteration (should go to 0)



Sum of the corrections (should be 0)

- The Alignment corrections of the Top Front UChannel rotations ru,rv,rw converge.
  The corrections per iteration rapidly go to 0. Some checks need to be done to MPII solution still.
- Red dotted line is the perfect geometry result.
- **r_w (Ry in LCIO system)** is harder to get right (mostly along sensor v, only stereo information, derivatives might be wrong…). Probably additional constraints such as momentum constraint is needed.
- The cumulative corrections over 1 iterations recover the initial geometry in r_u and r_v. I still see a bias in r_w **(Ry in LCIO system)** for the front UChannel at convergence after 2 iterations (in this case 0.1mrad)

Test on hierarchical structures constraints

- FEEs MC, Perfect geometry.
- **Tried releasing L1-L2-L3-L4 t_u only.**
  - (1) Used $T_y = 0$ and $T_x = 0$ constraints
  - (2) No constraints
- The constraint file is generated automatically
- The $T_x = 0$ only implies stereo sensors constrains

$$c = \sum_{\ell \in \Omega} f_\ell \cdot p_\ell$$

The constraint value c is usually zero. The format is:

```
Constraint    value
label         factor
  ...
label         factor
```

$$0 = \sum_{i=0}^{i=n} \mathbf{C}_i^{-1} \mathbf{a}_i$$

$$T_y = 0 = t_{u,L1A} + 0.995 t_{u,L1S} + t_{u,L2A} + 0.995 t_{u,L2S} + t_{u,L3A} - 0.995 t_{u,L3S} + t_{u,L3A} - 0.995 t_{u,L4S}$$

$$T_x = 0 = 0.0998 t_{u,L1S} + 0.0998 t_{u,L2S} + 0.0998 t_{u,L3S} + 0.0998 t_{u,L4S}$$

MPII residuals solution with L1L2L3L4 t_u floating

11101	-0.14129E-03	0.30922E-03
11102	0.18741E-02	0.38248E-03
11103	0.21053E-03	0.20933E-03
11104	0.66097E-03	0.23083E-03
11105	0.29855E-03	0.19284E-03
11106	0.89110E-03	0.21868E-03
11107	-0.36779E-03	0.25725E-03
11108	0.16440E-02	0.33194E-03

Updated alignment procedure: movements $O(1\mu m)$ compatible with resolution

MPII residuals solution with L1L2L3L4 t_u floating

11101	-0.93741E-02	0.16217E-02
11102	-0.78525	0.44262E-01
11103	-0.64972E-02	0.13765E-02
11104	-0.64422	0.36259E-01
11105	-0.28579E-02	0.94001E-03
11106	0.40327	0.22621E-01
11107	-0.14929E-02	0.60756E-03
11108	0.21573	0.12037E-01

Original alignment procedure: Stereo corrections are $O(100\mu m)$ due to lack of global movements constraints

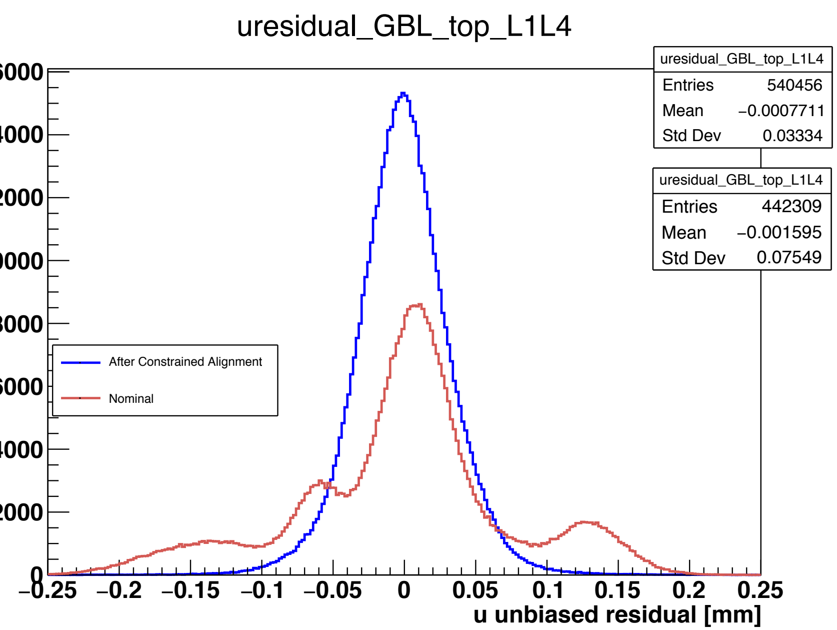
Alignment framework updates - Momentum Constraint

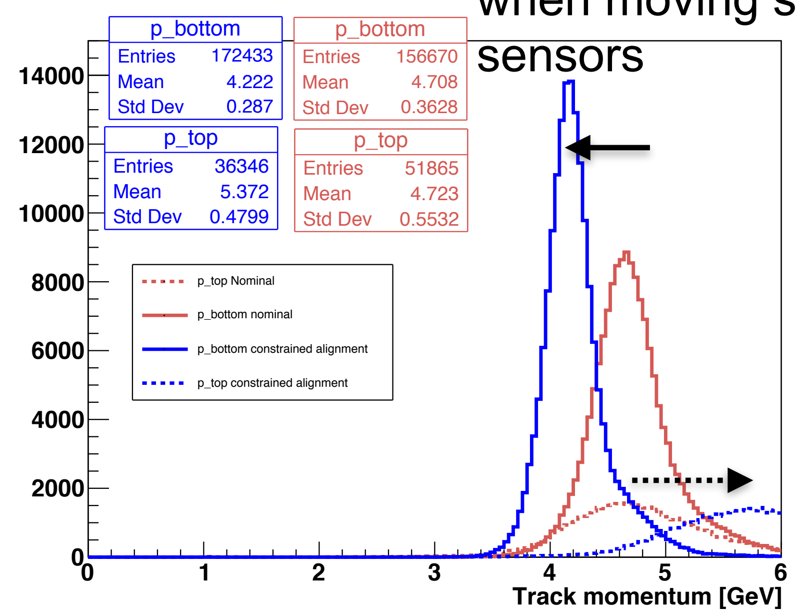
Formalism of seed constrained alignment

- In order to add an extra handle to improve the alignment solution, external constraints should be used.
- They can take the form of survey measurements, beamspot determination, but also constraints on the track parameters, i.e. momentum constraint from beam energy or calorimeter measurement.
- **If we use FEEs for alignment we know the momentum of these electrons with good approximation ($\sim E_{beam}$) and that can be used to constrain the alignment parameters**

- **Tested a constrained alignment on data fixing the u-channel to nominal positions**
  - Results are simple a test so plots not worth too much discussing
  - Noticed that sensor corrections were **O(10-30um)**, large improvement on the unbiased residuals, but introduction track-parameter biases (weak modes).
  - Observed in d_0 and p, mostly
- Alignment Strategy:
  - FEEs, L1-L2-L3-L4 A/S tus, Modules + uChannel constraints. [just a check]

Large biases in momentum appear when moving stereo sensors





Implement track parameters constraints in MPII

SLAC

- MPII refits tracks solving for $\partial f / \partial \mathbf{q_j}$ at each $\mathbf{p} \rightarrow \mathbf{p} + \mathbf{dp}$ iteration ($\mathbf{p}$ are the alignment parameters, $\mathbf{q}$ are the track parameters)
- If the local derivatives are "small" then $\mathbf{dq_j}$ can be large to find the χ^2 minimum
- **A track parameter un-constrained fit likely to result in a geometry which leads to biases in the case of curved tracks.**
- A seed-constrained fit is obtained adding a seed precision matrix to the track χ^2, so when minimising an extra term is added to the linear system [this is implemented in GBL]
- **In the case of the momentum, df/d(q/p) is inflated, which means that D(q/p) is smaller-> Dp is computed accordingly -> Momentum constrained alignment.**
- **I now use a way to load the GBL C++ library into hps-java that supports this feature. :**
  - **Seed Tracks are scaled by q/pT -> q/pT + delta**
  - **Then fed to GBL refitting driver.**
  - **Correlation between curvature and other tracks parameters are neglected in this ansatz**
- **For backward compatibility I also translated the relevant parts from C++ to Java.**

track parameter derivatives

$$z_i = y_i - f(x_i, \boldsymbol{q}, \boldsymbol{p}) = \boxed{\sum_{j=1}^{\nu} \left(\frac{\partial f}{\partial q_j} \right) \Delta q_j} + \sum_{\ell \in \Omega} \left(\frac{\partial f}{\partial p_\ell} \right) \Delta p_\ell \, .$$

The dimension of the label set is arbitrary

$$n_{lc} = \text{number of local parameters} \qquad \text{array} : \left(\frac{\partial f}{\partial q_j} \right)$$

$$n_{gl} = \text{number of global parameters} \qquad \text{array} : \left(\frac{\partial f}{\partial p_\ell} \right) ; \text{ label-array } \ell$$

$$z = \text{residual} \quad (\equiv y_i - f(x_i, \boldsymbol{q}, \boldsymbol{p})) \qquad \sigma = \text{standard deviation of the measu}$$

These need to get recomputed for each point and a new trajectory formed

$$\chi^2(\mathbf{x}) = \sum_{i=1}^{n_{\text{meas}}} (\mathbf{H}_{m,i}\mathbf{x} - \mathbf{m}_i)^T \, \mathbf{V}_{m,i}^{-1} \, (\mathbf{H}_{m,i}\mathbf{x} - \mathbf{m}_i) \quad \text{(from measurements)}$$

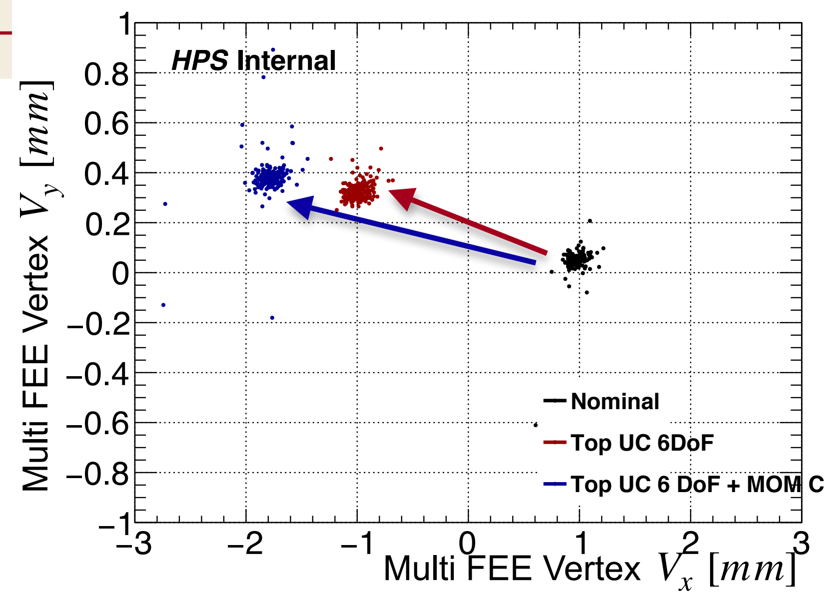
$$+ \sum_{i=2}^{n_{\text{scat}}-1} (\mathbf{H}_{k,i}\mathbf{x} + \mathbf{k}_{0,i})^T \, \mathbf{V}_{k,i}^{-1} \, (\mathbf{H}_{k,i}\mathbf{x} + \mathbf{k}_{0,i}) \quad \text{(from kinks)}$$

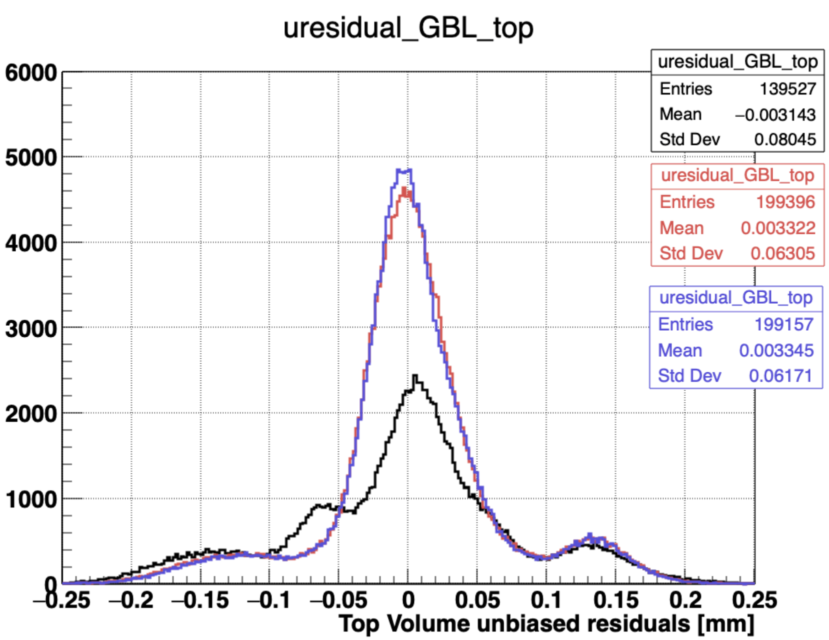
$$+ \boxed{(\mathbf{H}_s\mathbf{x})^T \, \mathbf{V}_s^{-1} \, (\mathbf{H}_s\mathbf{x}) \quad \text{(from external seed)} \quad (9)}$$

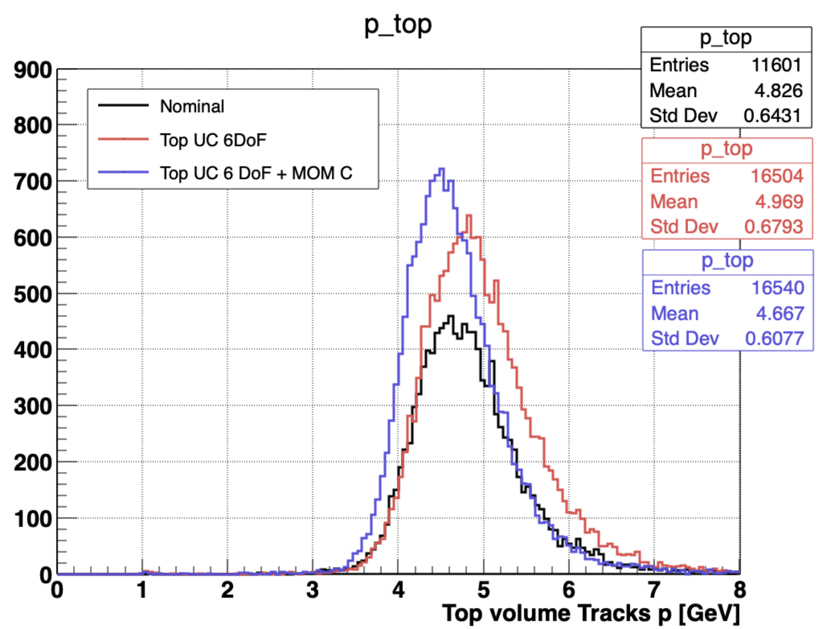
GBL Manual

An example of momentum constrained alignment

- Alignment of the UChannels only, all DoF. No module by module alignment in this tests
- Notice how residuals are compatible between unconstrained and constrained, but momentum is not.
- **However, there are correlations to other track parameters, hence to the common fit position (beamspot position). Constraining the momentum will create tension in beamspot determination.**
- **Additional constraint is needed to avoid such bias.**







Alignment framework updates - Beamspot Constraint

“Beamspot” constrained GBL refit

- I’ve added the beamspot constrained gbl refit to hps-java
- Additional measurements, external to the tracker hits like an external measurement of the beamspot can be added by one GBL point at that location.
- Tracks should be propagated back to the point of closes approach to the point (beamspot) and the distance between this and the beamspot (in XY and Z) are used as measurement.
- For the moment:
  - used slightly simpler approach where I treated the target as a virtual layer with 2D measurement (x-y) and added the point to the GBL track.
  - One has to provide a (x,y,z) location and a (x,y) precision

"Beamspot" constrained GBL refit

- I've used the location: $\mathbf{b} = (-7.5,0.,0.)$, in 'tracking' coordinates.
- The 'sensitive' direction is along global Y so $\mathbf{i} = (0,0,1)$ $\mathbf{j} = (-sin(\alpha), cos(\alpha),0.)$, where α is the SVT angle of 30.5mrad.
- The track-prediction at the beamspot is obtained analytically by helix propagation.
- The RK extrapolation gives very similar result (but our code forces the extrapolation back only starting from (0.0.0), so part of the back-extrapolation is still done by helix assumption).
- The residual at the beamspot is given by: $\mathbf{r} = (b_i - p_i, b_j - p_j) = (-p_i, -p_j)$ where the sub-scripts indicate the projections on the uv-plane.
- To form a GBL point one has to pass the local curvilinear to measurement projection transformation P_{L2M}
- The beamspot precision can be chosen to strengthen this constrain for alignment purposes for example.

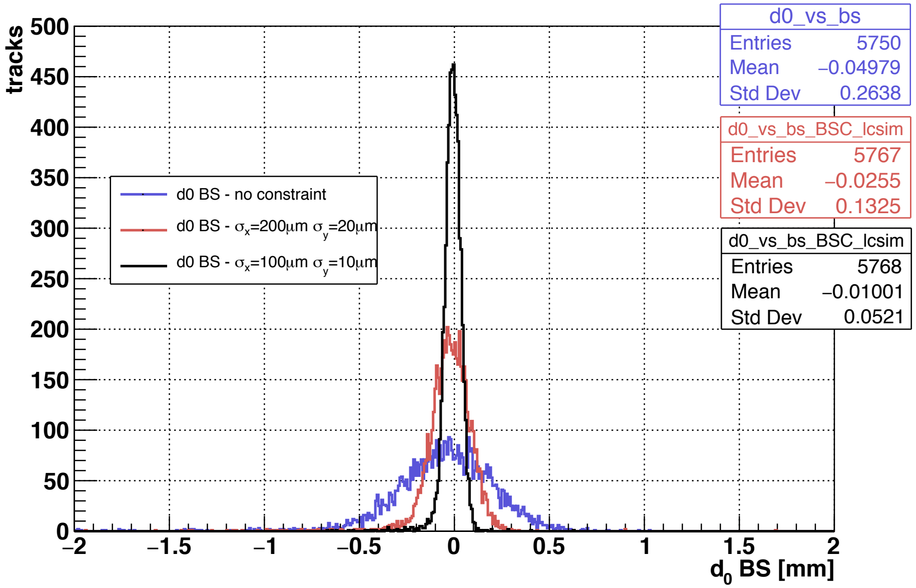
$$\mathbf{U} = \frac{\mathbf{Z} \times \mathbf{T}}{|\mathbf{Z} \times \mathbf{T}|} = \frac{\mathbf{Z} \times \mathbf{T}}{\cos\lambda}$$

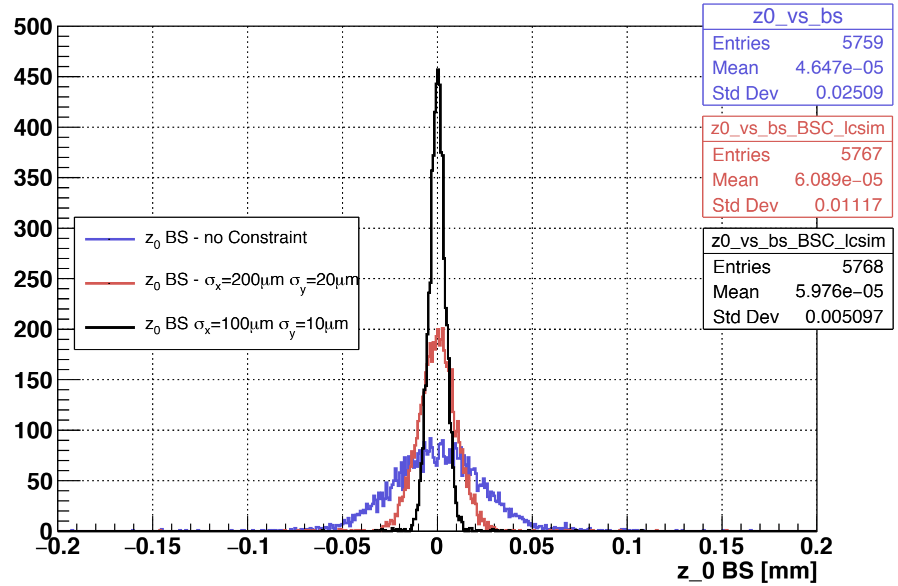
$$\mathbf{V} = \mathbf{T} \times \mathbf{U}$$

Where **T** is the unit-vector tangent to the track direction at a certain s

$$R^{gbl}_{m2c} = \begin{pmatrix} \mathbf{U}\cdot\mathbf{I} & \mathbf{U}\cdot\mathbf{J} \\ \mathbf{V}\cdot\mathbf{I} & \mathbf{V}\cdot\mathbf{J} \end{pmatrix}$$

Results





- The effect of adding the beamspot constrain to the GBL refit is shown.
- The distributions are obtained by changing the helix pivot at the beamspot location
- Red and black distributions are obtained by then adding the GBL local corrections by:

  $d_0^{gbl} = d_0 + \Delta_{d0}$ $z_0^{gbl} = z_0 + \Delta_{z0}$
- The corrections are given by projecting the GBL curvilinear corrections $(\Delta x_T, \Delta y_T, 0)$ to the perigee frame

$$P_{p2cl} = \begin{pmatrix} 0 & -1 & 0 \\ \sin\lambda & 0 & \cos\lambda \\ -\cos\lambda & 0 & \sin\lambda \end{pmatrix}$$

$$\begin{pmatrix} x_\perp \\ y_\perp \\ z_\perp \end{pmatrix} = P_{p2cl} \begin{pmatrix} 0 \\ \epsilon \\ z_p \end{pmatrix}$$

Putting things together: alignment tags produced for testing

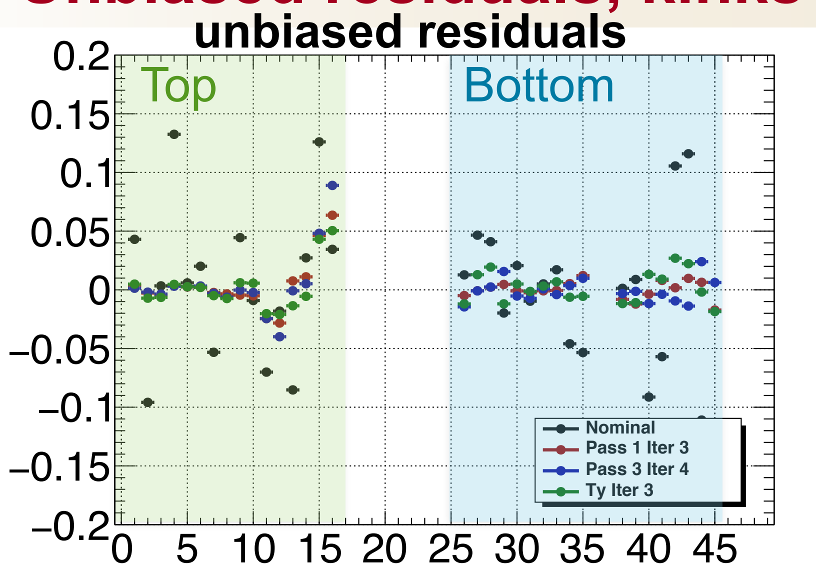
Alignment detectors that have been studied

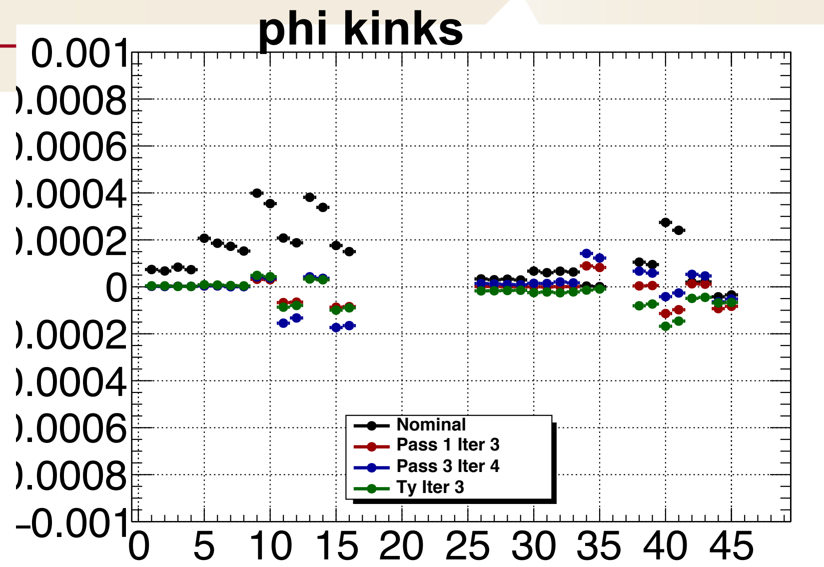
- Back in September I produced 3 aligned detectors with the aim of checking performance of the SVT calibration.
- The tags were made with the purpose of checking the various strategies and included global movements and external constraints:

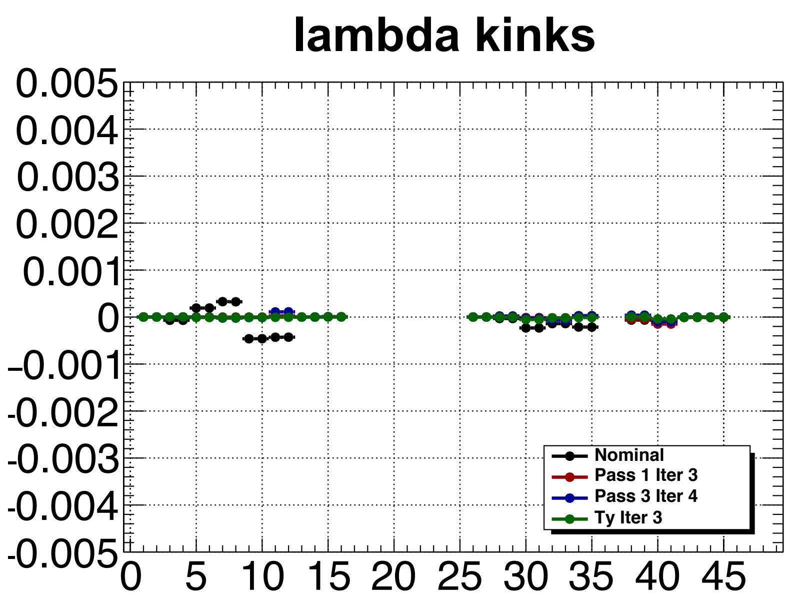
- **HPS_PASS1_iter3**: momentum constrain only, Back UChannel fixed
  - iter1: Tx, Ty, Rx, Ry of front UChannels
  - iter2: Tx, Ty, Rx, Ry of front UChannels
  - iter3: Tx, Ty front UChannels, Tu L1-L4 Modules
  with Tx/Ty constraints
- **HPS_PASS3_iter4**: momenutm constrain and beamspot constrain (0,0,-7.5)
  - iter1-3: Tx,Ty,Rx,Ry,Rz front Uchannels
  - iter4: Tx, Ty + Tu, Tv of L1-L6(7) Modules, with Tx/Ty constrains
- **HPS_TY_iter3**: momentum and beamspot constraints.
  - iter1: Ty and Rx of UChannels
  - iter2: Tu of L1-L6(7) Modules (with Tx Ty constraints)
  - iter3: Tu of L1-L4 Sensors (with module positions and UChannels positions constraints), Rw L1-L4 sensors

- Tracks used for alignment are FEE tracks from run 10103 and 10104.
  - 6 hits in the top volume and 7 hits in the bottom volume
  - Momentum between 3.8 and 5.2 GeV
  - No Chi2 cut
  - Momentum and Beamspot constraints are applied as described before.

Unbiased residuals, kinks

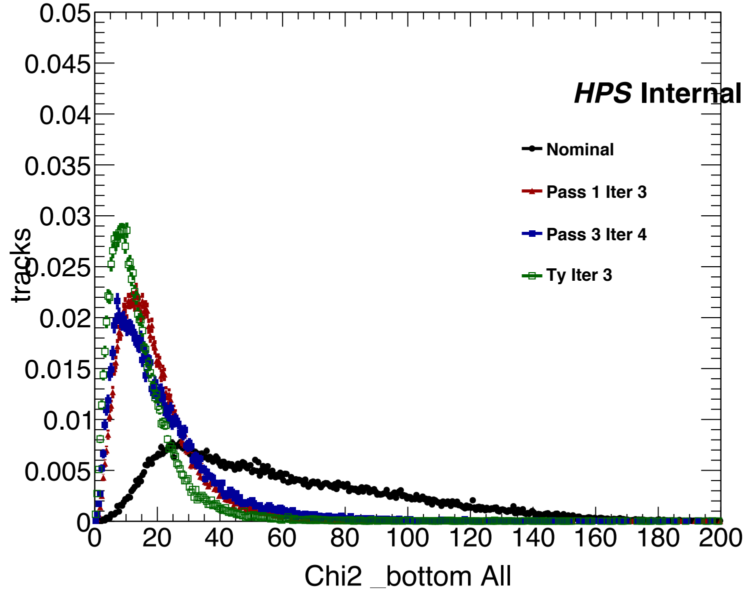


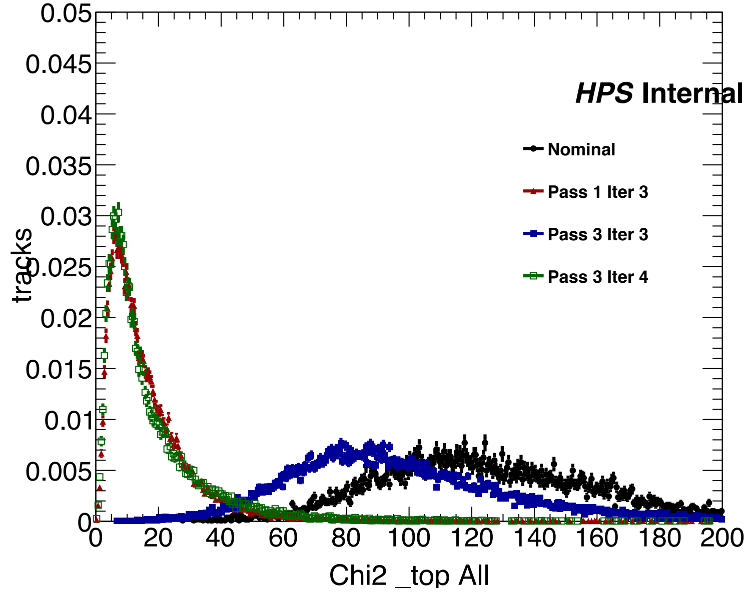


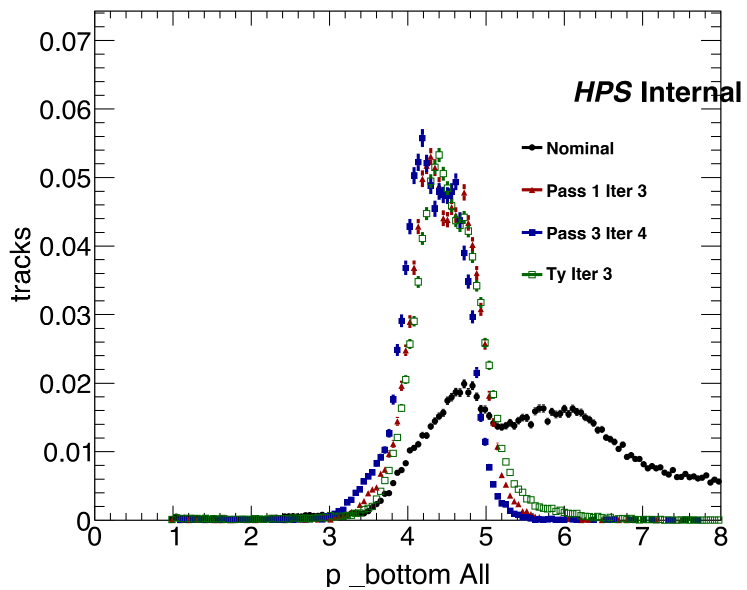


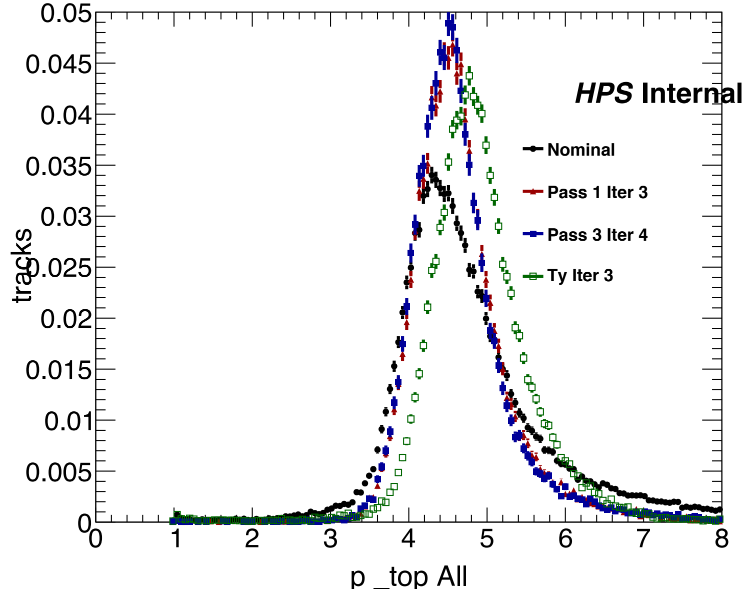
- X-axis is the ID of each sensor: first half represent the top volume, second half represents bottom volume
- Sensors/modules of the back of the detector are not aligned
- Aligning global structures first and then up to module level leads to similar results of aligning Tu up to sensor level.

Chi2, momentum - FEE DATASET (10103)



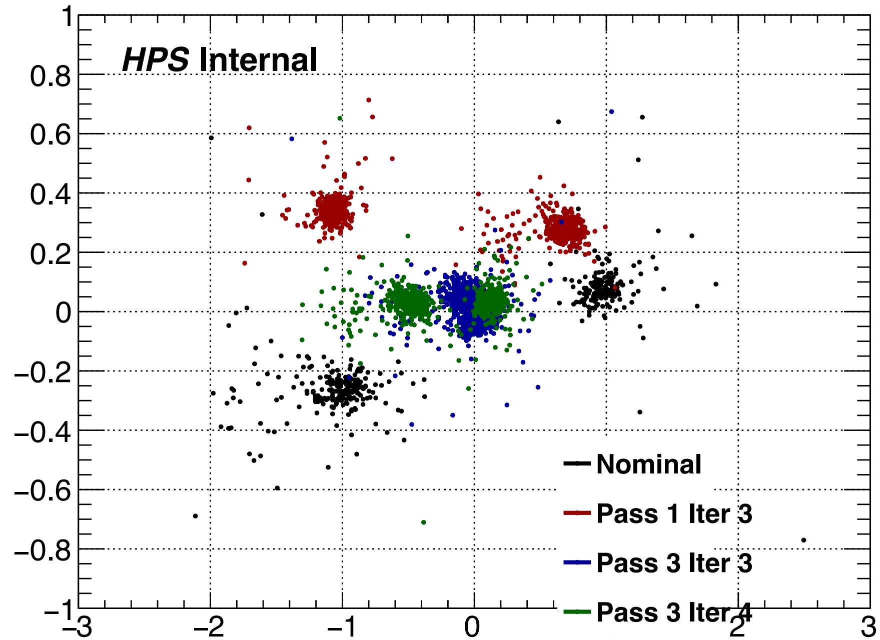




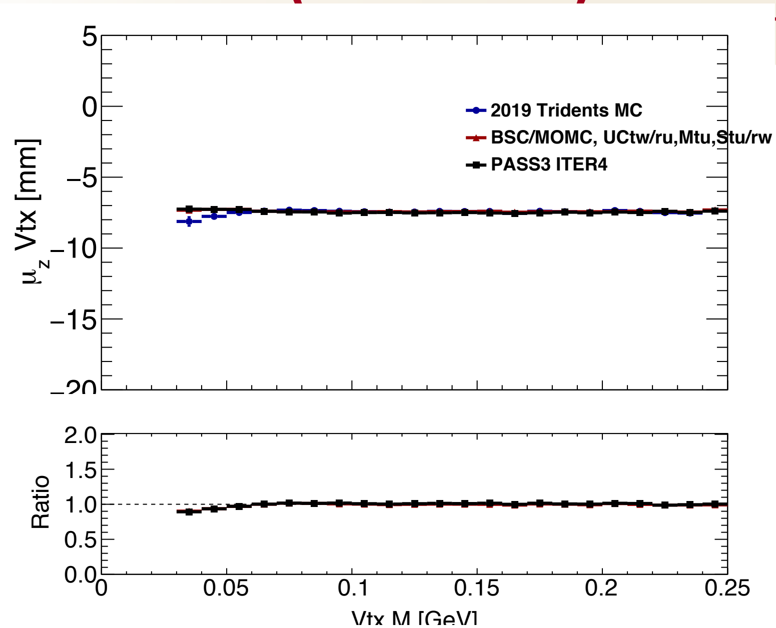


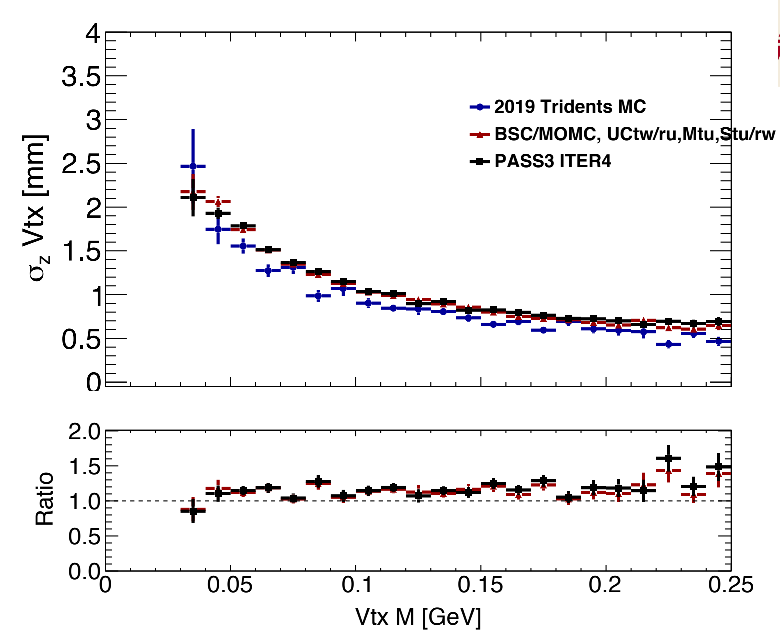
Multi VTX X-Y

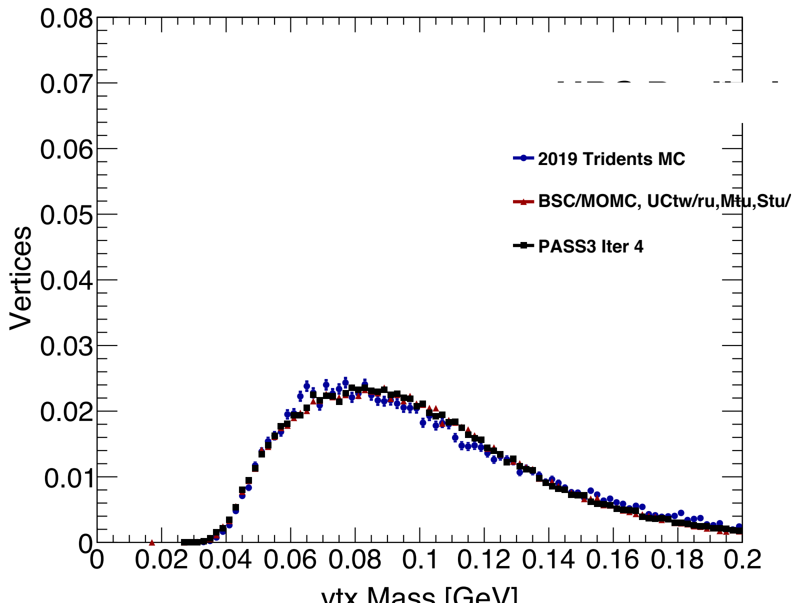
- Effect of beamspot constrain in pass3 shows that top and bottom can be forced to converge to a common point
- When aligning Ty, Tx module by module including back of the detector, a difference in X is noticed.
- When not applying beamspot constraint top and bottom have a large spread in X and around the same Y of ~300um. This solution has been checked that keeps distance between the wires fixed.

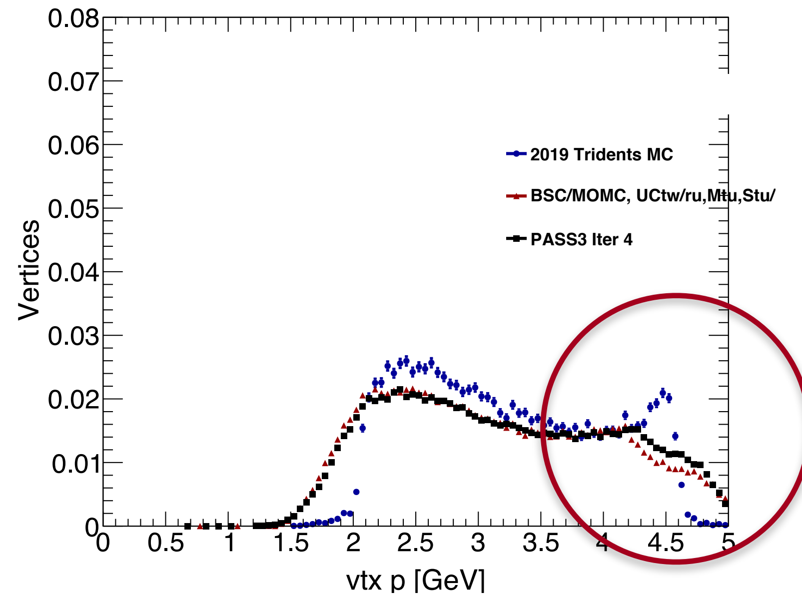


Checks on 10031 - V0 skims comparison with MC Tridents (no beam)



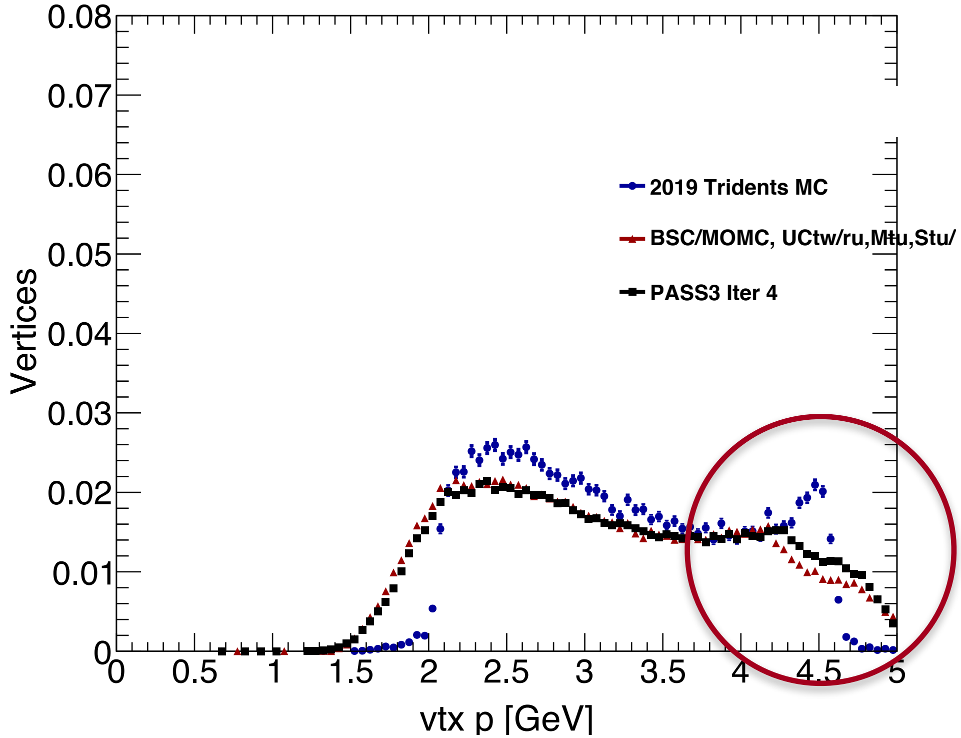






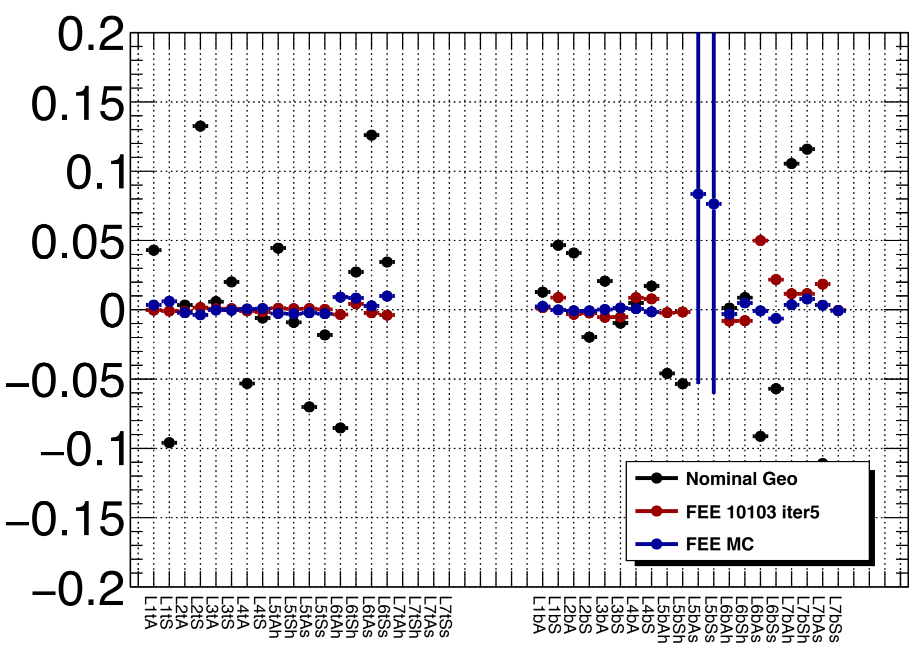
Momentum scale and resolution in FEE samples

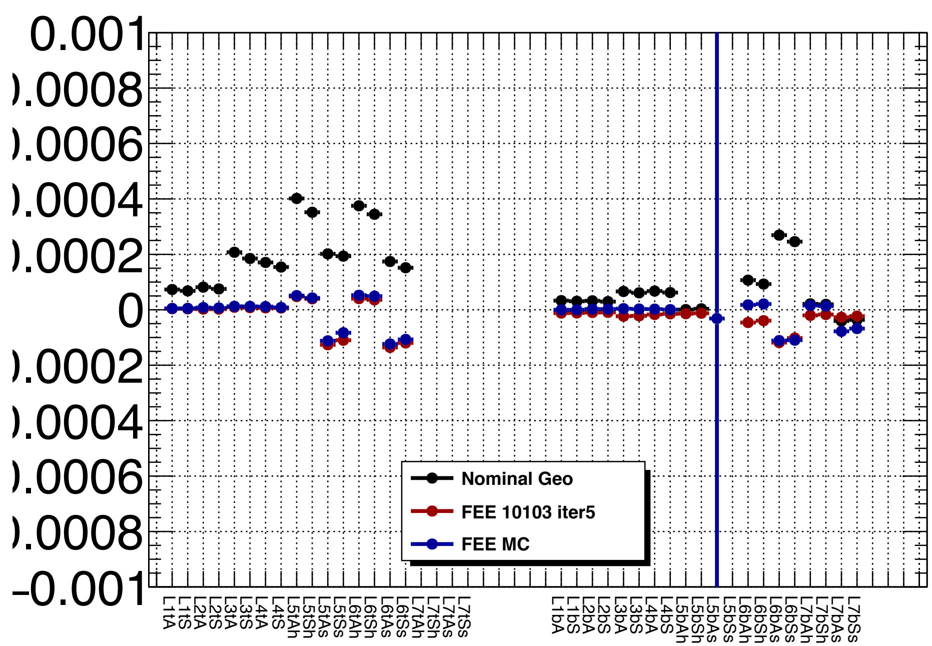
- The three alignment tags perform quite similarly in terms of track chi2, momentum scale and resolution and unbiased residuals.
- Decided to move forward with the tag (HPS_TY_IterX) which only corrects for movements along the most sensitive directions, nominally tu and rw for the sensors.
- Two additional iterations were made for improving top volume alignment and check the momentum resulution and scale with FEE tracks
- In particular checking the Vtx momentum plots one can notice that there is a worse resolution at the beam energy



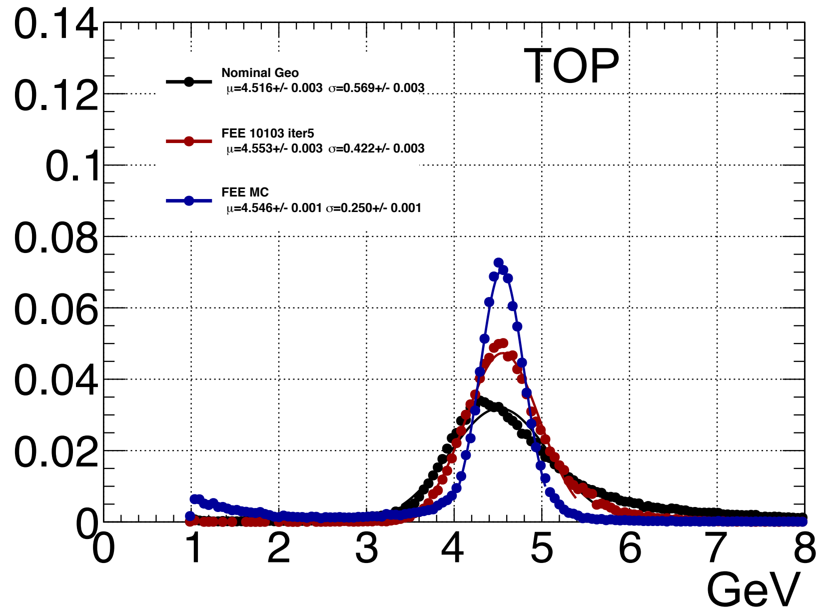
FEE momentum scale and resolution

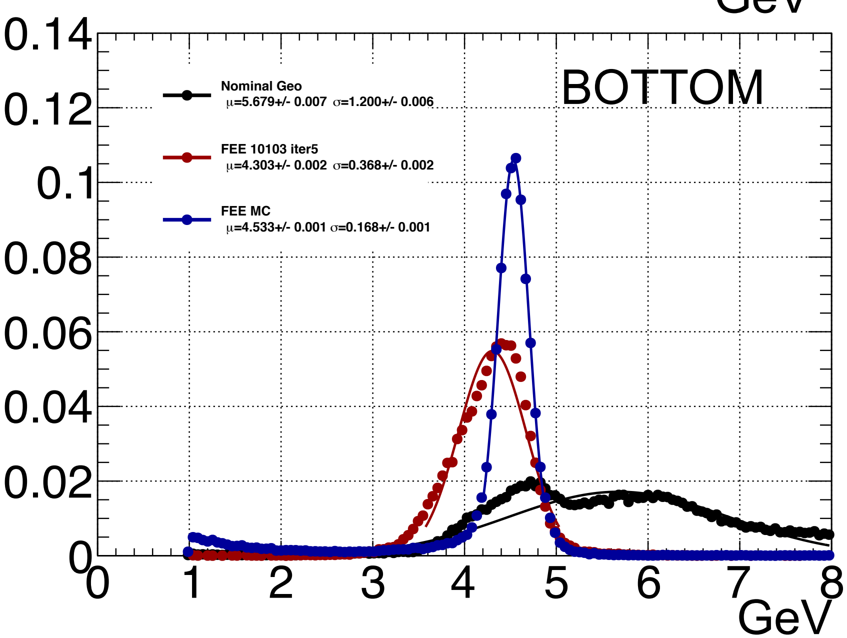
- Momentum scale and resolution is checked by fitting with a gauss distribution the core of the momentum distribution of selected FEE electrons in dedicated runs (10103 and 10104)
- An iterative fit is done to determine mean and sigma
- Additional Alignment iterations were made in the top volume, back u-channel to correct for residual misalignment of tu and rw dofs.
- Unbiased residuals are within ~10um, comparable to MC residuals for the top volume. Bottom volume still need some work.
- Phi kink residuals still show an asymmetry between hole and slot side of the detector: also present in MC (which is not understood)



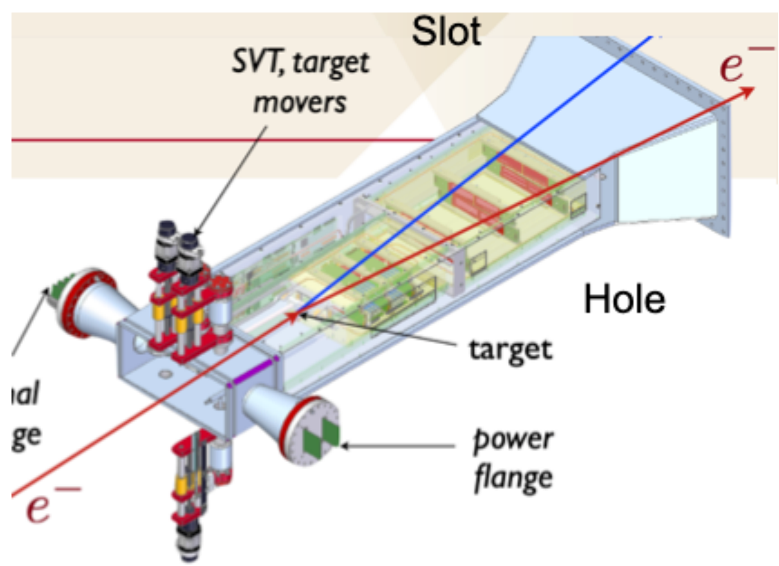


FEE momentum scale and resolution

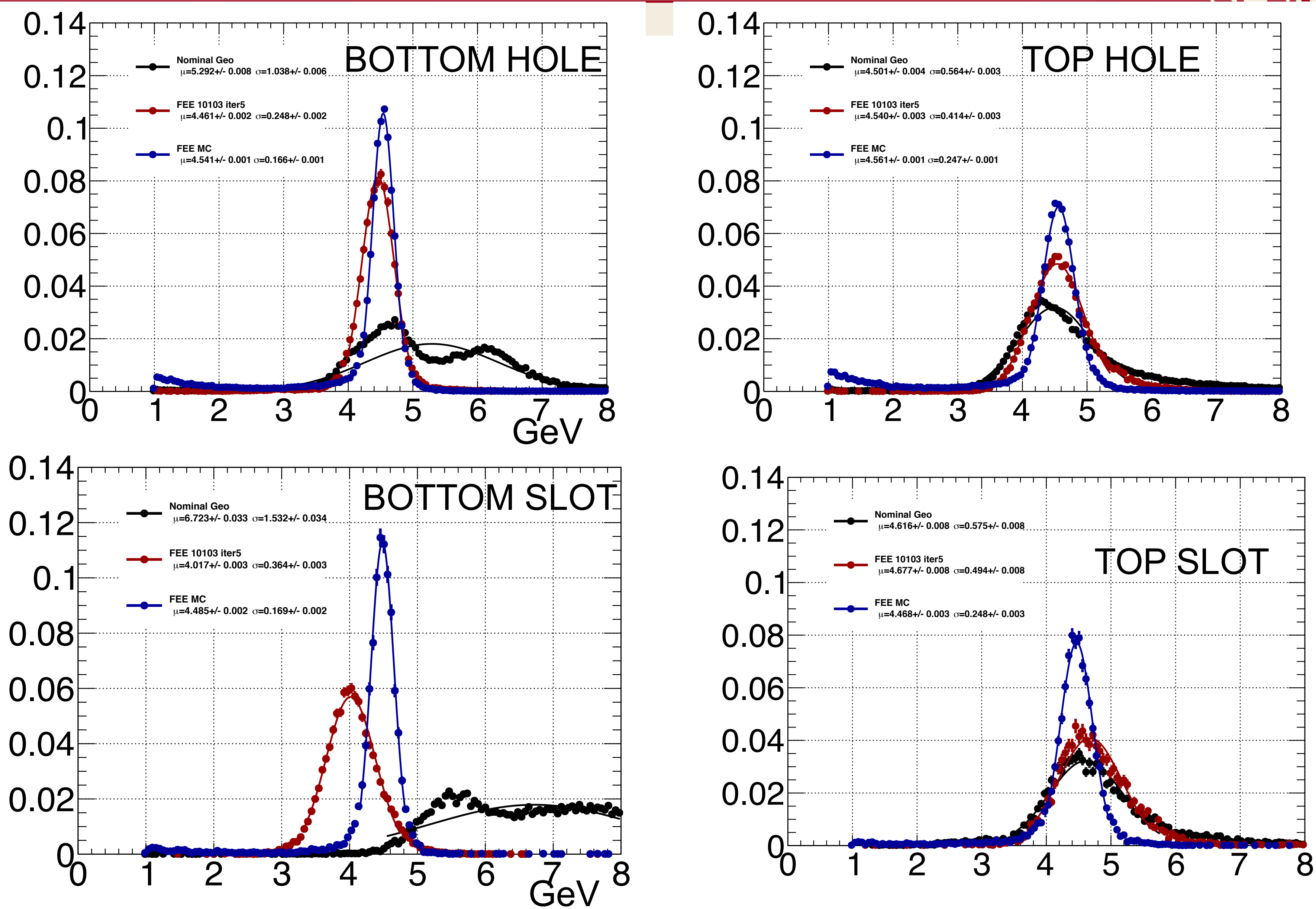




- Large improvement wrt nominal geometry but still about factor 2 worse resolution in data wrt MC simulation
- In particular bottom seem to show a bimodal distribution, so I have split tracks in two halves:
  - Hole tracks are defined as tracks that hit the back of the detector (L5-L6-L7) in the hole side (electron side) only => large statistics for FEEs samples
  - Slot tracks are defined as tracks that hit the back of the detector in the slot side (positron side) only => low stat for FEEs samples



FEE momentum scale and resolution



Summary and next steps

- Largely updated the hps-java alignment framework to include:
  - Additional monitoring tools
  - A basic framework for global structures alignment including hierarchical constraints
  - Momentum and beamspot constraint to the MPII files
  - Fixes for the rw for thin sensors (not shown today)
- Shown jeopardy results which led to 3x better resolution wrt nominal geometry and ~90% of expected MC resolution
- Current focus is on improving momentum resolution:
  - Will investigate the difference between hole and slot side of the SVT
  - With Tim, resumed the effort to crosscheck the geometry code that was ported to hps-java, in particular z and x positioning of the sensors.
- Cameron will show additional results on:
  - Effect of alignment on A' acceptance
  - Additional V0 Data/MC comparisons