SVT Alignment Overview

PF

05/02/2022

U.S. DEPARTMENT OF ENERGY | Stanford University
SLAC NATIONAL ACCELERATOR LABORATORY

Track Based alignment of the SVT Detector

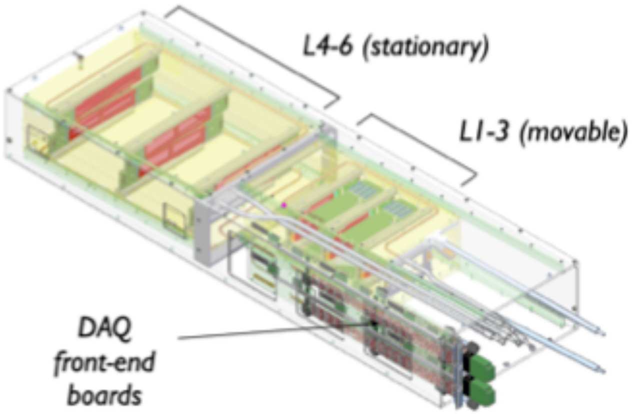
- The HPS SVT Alignment framework
  - GBL-MPII implementation
  - External measurements
  - Geometrical constraints, soft-mode cut, external points on track, momentum constraint
  - Monitoring and unbiased residuals

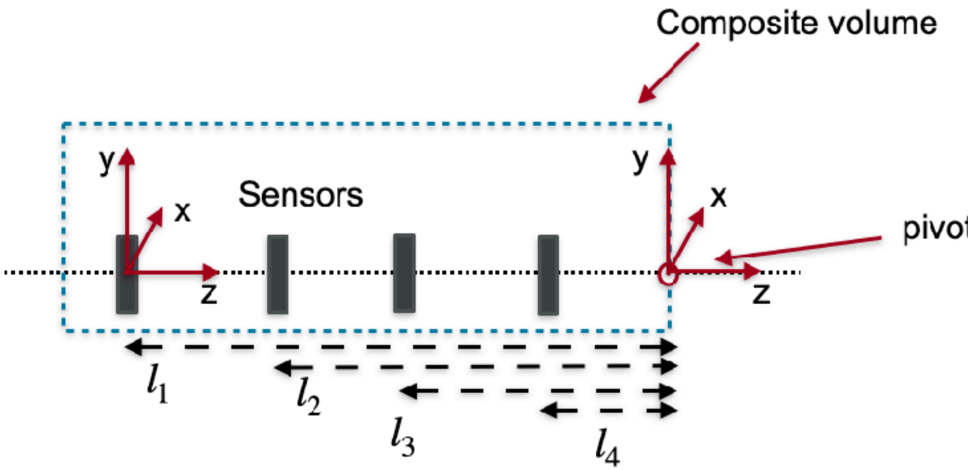
- Applications to 2019 data:
  - 2019 alignment - 2021 pass
  - 2019 alignment - 2022 pass (on top volume)

Introduction

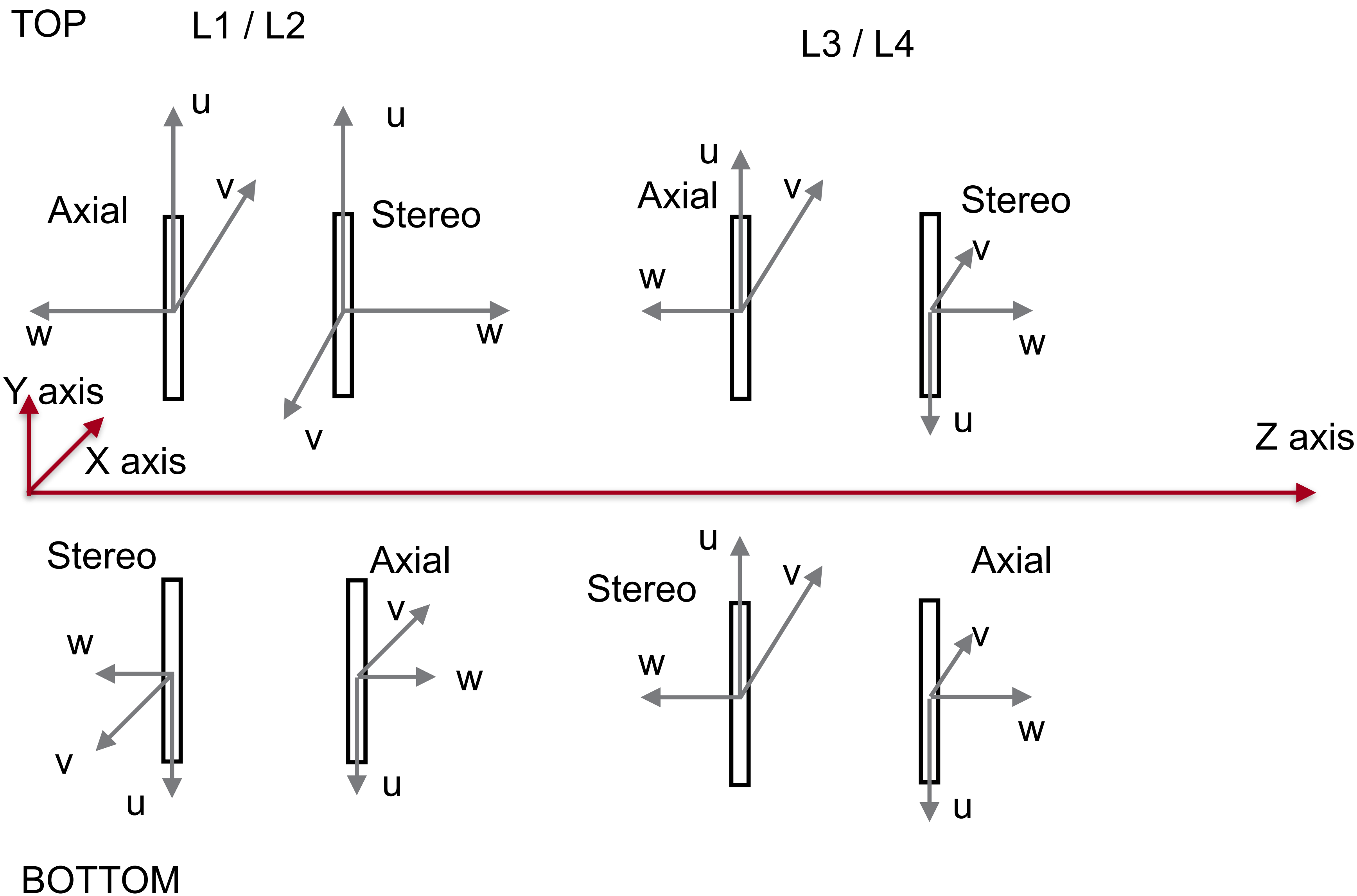
GBL + MPII

- The HPS track-based alignment framework is based on the General Broken Lines (GBL) and Millepede II (MPII)
- HPS Tracker Geometry can be split into split in:
  - 4 U-Channels structures
  - 7 Modules structures
  - 20 Single sensors structures
- Each structure location and orientation is defined by 6 DoF:
  - 3 Translations : T_x, T_y, T_z
  - 3 Rotations : R_x, R_y, R_z
- Global χ^2 minimisation technique
- Weak mode constraints employed:
  - Momentum constraint
  - Beamspot location constraint
  - Soft mode cut

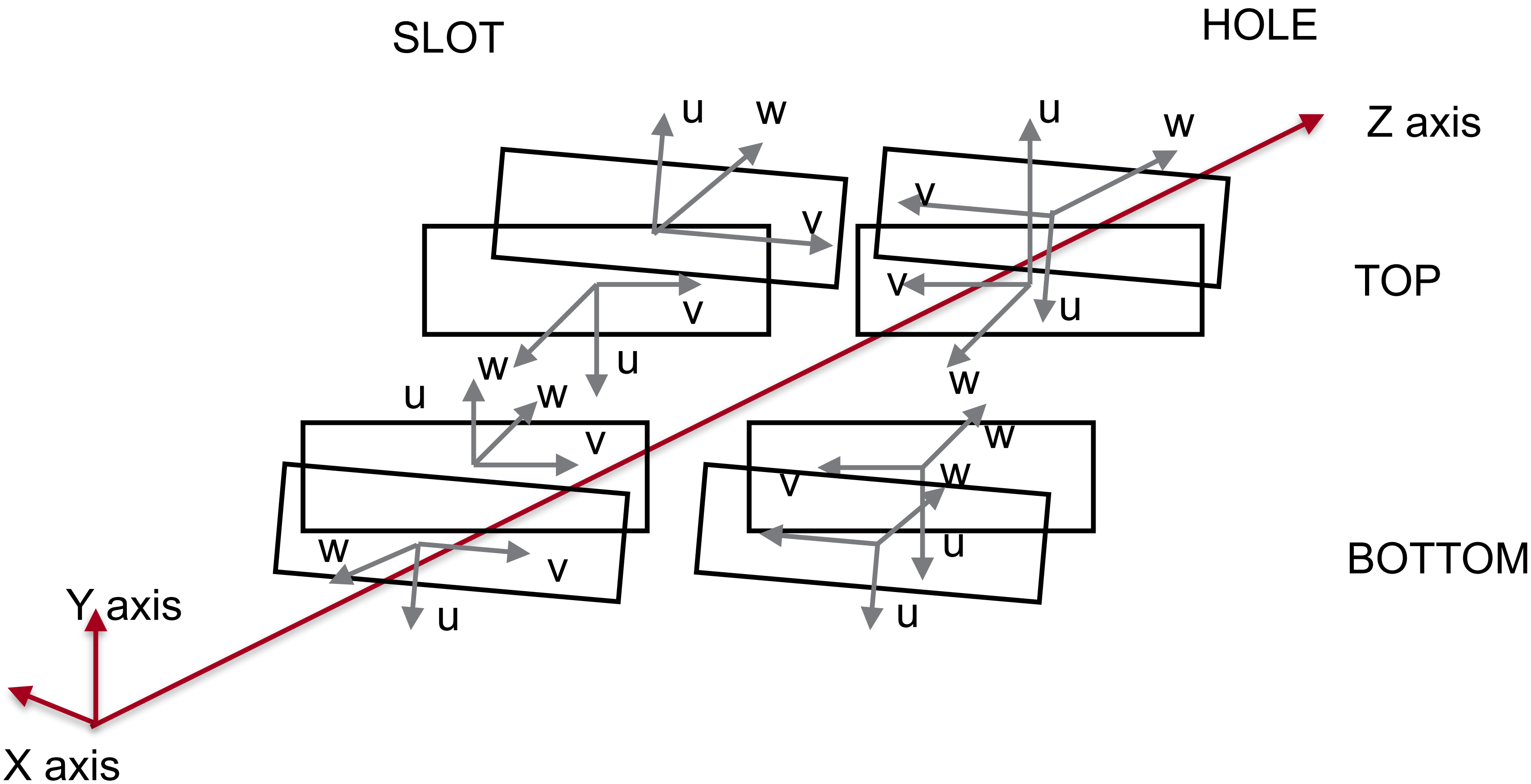




Local frames on sensors - a reference



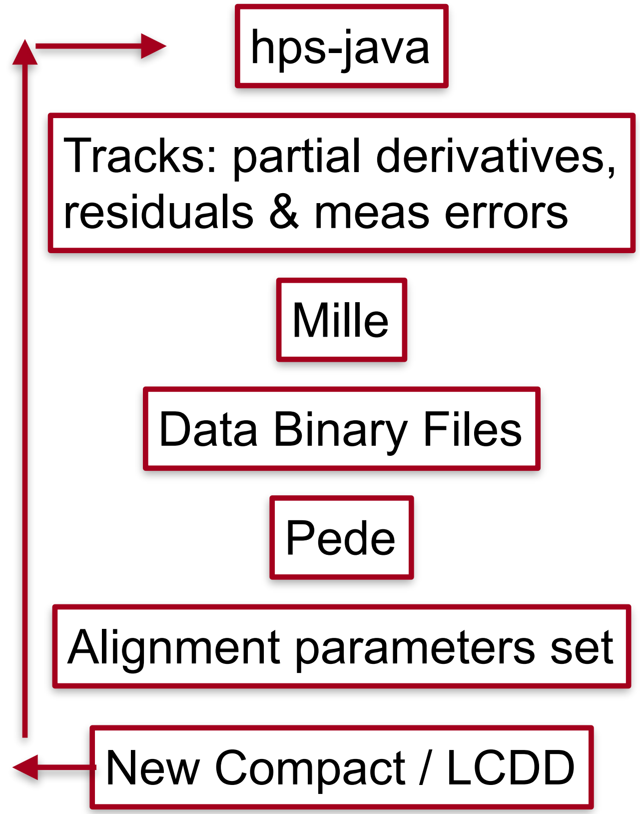
Local frames on sensors - Back U Channel - a reference



MPII + GBL Aligment implementation

Aligment framework validation for 2019/2021 geometry

- MPII is the software used for aligning the single sensors in HPS (internal/local alignment)
- Assessed 2019 MPII framework using ideal MC and introducing misalignments
  - Tested thin layers implementation in the framework
  - Fixed rotations of thin sensors in the alignment framework
- These tests were done on MC samples with perfect geometry
- Used TriTrig 2019 with new thin layers L1/L2 geometry modelling and **no beam overlay**



Local Alignment Tests - MPII steering settings and results

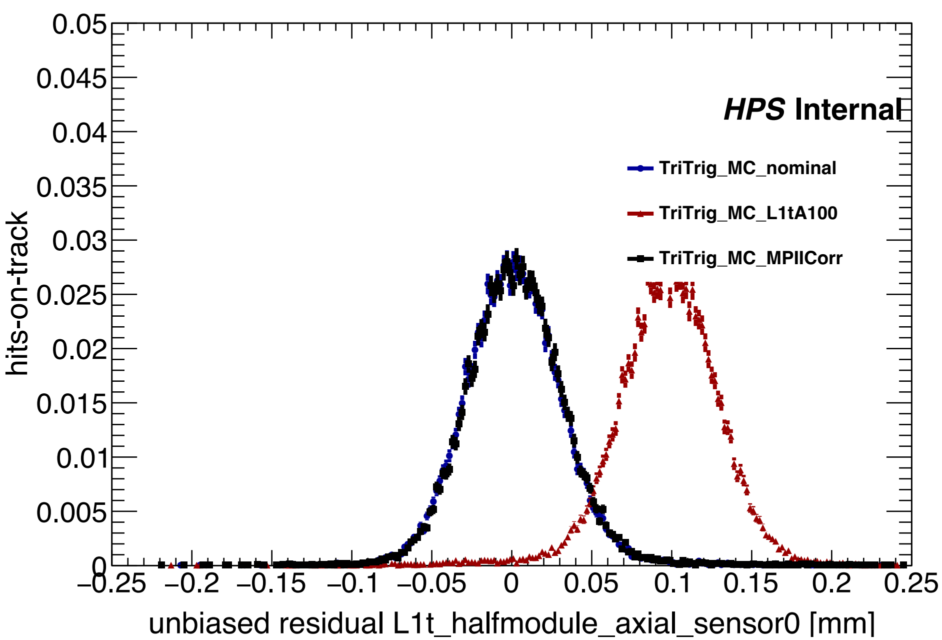
- **Re-alignment strategy:**
  - Fix all outer layers
  - Re-align both L1 top Axial and Stereo
- This is to check if MPII:
  - can recognize displacements on single side when two sides are realigned
  - can recover the same degree of misalignment
- Results are **OK**:
  => MPII finds that L1At is moved of 100um with sub-micron precision
  => MPII finds that L1St is moved of 0.8um with sub-micron precision
- **Simple translations on the new thin sensors can be recovered**

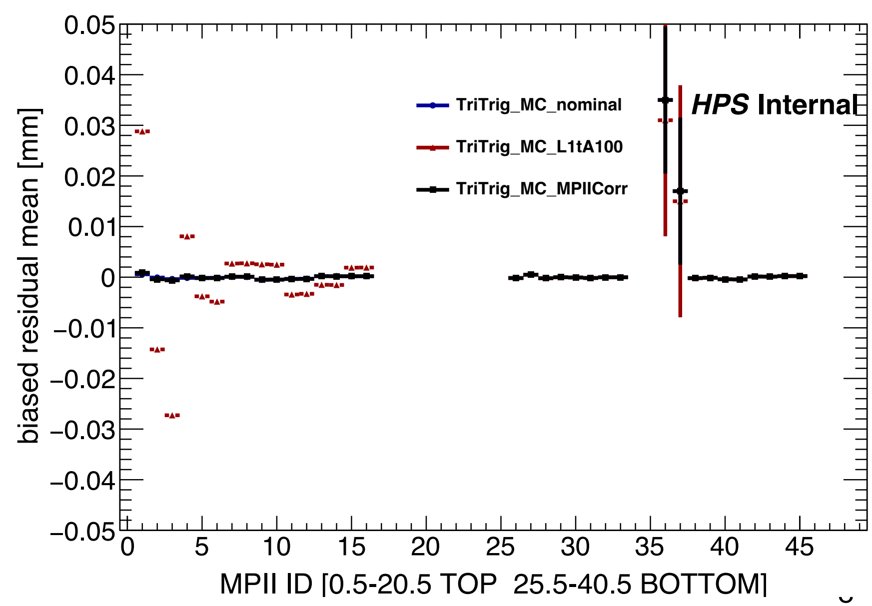
Single L1At movement

```
Solution algorithm:
========================
                solution method:   diagonalization
     convergence limit at Delta F=   0.10E+01
      maximum number of iterations=   6
                using pre-sigmas:   no
                 regularization:   no
   Chi square cut equiv 3 st.dev applied
     ... in first iteration with factor   0.30E+02
    ... in second iteration with factor   0.60E+01
     (reduced by sqrt in next iterations)
Checking feasibility of parameters:
   parameters are feasible  (i.e. satisfy constraints)
```

```
PTLINE: INFO=1  convergence reached
```

```
Parameter   ! first 3 elements per line are significant (if used as input)
    11101    0.99313E-01   0.0000      -0.68652E-03   0.19138E-03
    11102    0.87519E-03   0.0000       0.50875E-01   0.18948E-03
    11103     0.0000      -1.0000
    11104     0.0000      -1.0000
    11105     0.0000      -1.0000
```





Local Alignment Tests - MPII steering settings and results - “multiple correlated misalignment”

SLAC

- **I also tested global movements**
  - L1tL2t A+S = +100um
  - L1bL2b A+S = +500um and was able to re-align correctly [however this assumes the rest is correctly placed]

```
<millepede_constant name="11101" value="0.100"/>
<millepede_constant name="11102" value="0.100"/>
<millepede_constant name="11103" value="0.100"/>
<millepede_constant name="11104" value="0.100"/>
<millepede_constant name="11105" value="0.0"/>
<millepede_constant name="11106" value="0.0"/>
<millepede_constant name="11107" value="0.0"/>
<millepede_constant name="11108" value="0.0"/>
<millepede_constant name="11109" value="0.0"/>
<millepede_constant name="11110" value="0.0"/>
<millepede_constant name="11111" value="0.0"/>
-UUU:----F1  compact.xml    30% L199    (nXML Valid) ------
<millepede_constant name="21101" value="0.50"/>
<millepede_constant name="21102" value="0.50"/>
<millepede_constant name="21103" value="0.50"/>
<millepede_constant name="21104" value="0.50"/>
<millepede_constant name="21105" value="0.0"/>
<millepede_constant name="21106" value="0.0"/>
<millepede_constant name="21107" value="0.0"/>
```

Compact entries for global movement

```
Parameter   ! first 3 elements per line are significant (if used as input)
 11101    0.10191     0.0000     0.10191     0.48554E-03
 11102    0.10095     0.0000     0.10095     0.49947E-03
 11103    0.10052     0.0000     0.10052     0.32385E-03
 11104    0.10053     0.0000     0.10053     0.33074E-03
 11105     0.0000    -1.0000
 11106     0.0000    -1.0000
 11107     0.0000    -1.0000
 11108     0.0000    -1.0000
 11109     0.0000    -1.0000
 11110     0.0000    -1.0000
-UU-:----F1  millepede--L1_L2_MC_GlobMov.res   Top L1    (Fundamental) ----
 12316     0.0000    -1.0000
 12317     0.0000    -1.0000
 12318     0.0000    -1.0000
 12319     0.0000    -1.0000
 12320     0.0000    -1.0000
 21101    0.50224     0.0000     0.50224     0.42774E-03
 21102    0.50398     0.0000     0.50398     0.41657E-03
 21103    0.50071     0.0000     0.50071     0.28678E-03
 21104    0.50135     0.0000     0.50135     0.27921E-03
```

MPII residuals after accumulation and solving

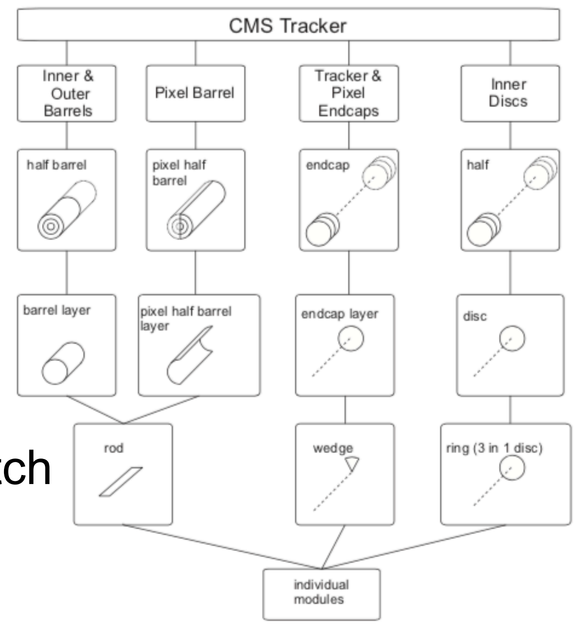
Constraints

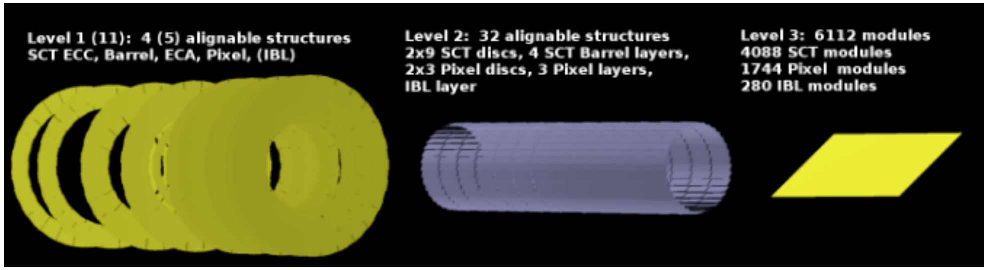
Hierarchical alignment and external constraints

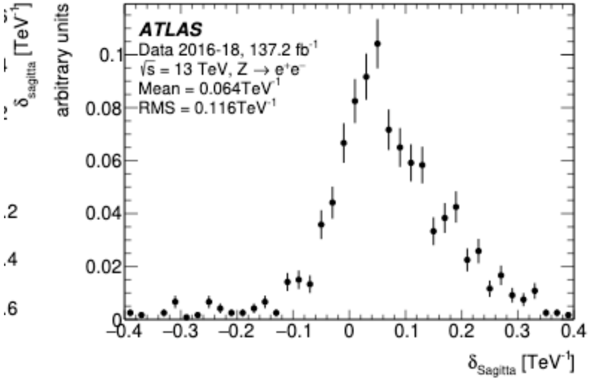
- Implement an hierarchical alignment procedure:
  - **Same way to solve global and local misalignments**: just accumulate all information and decide which structure we want to align.
  - **Sensor positions and orientations will be relative to composite structures** and there is a **natural way to include 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



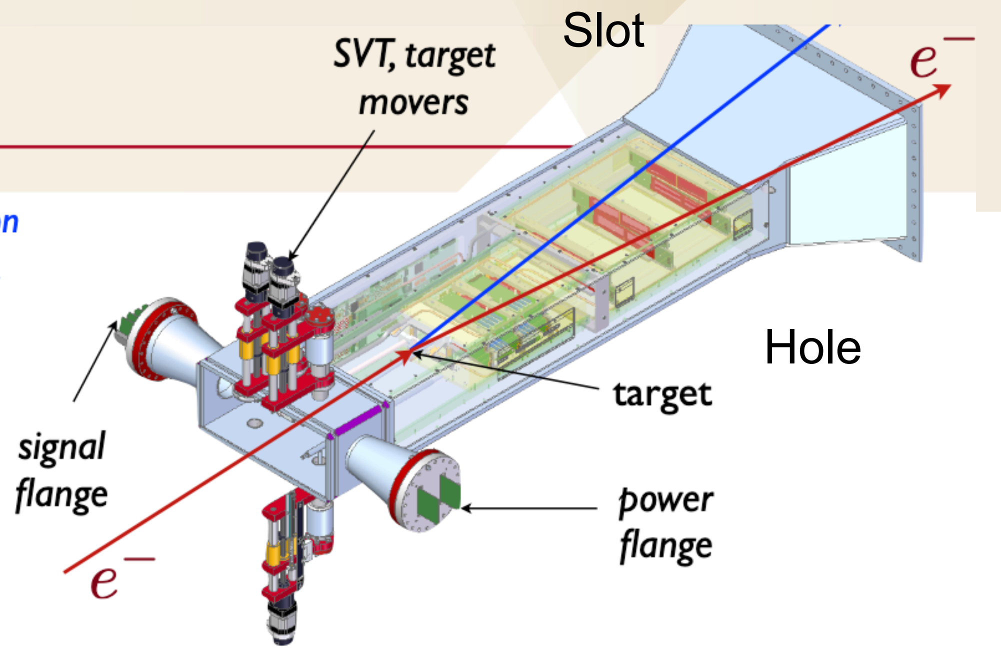




ATLAS sketch

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

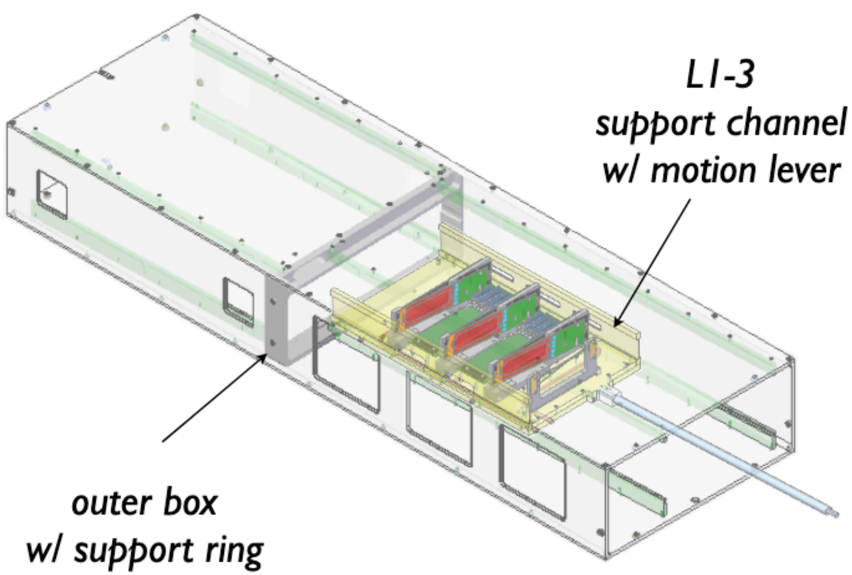
The HPS SVT

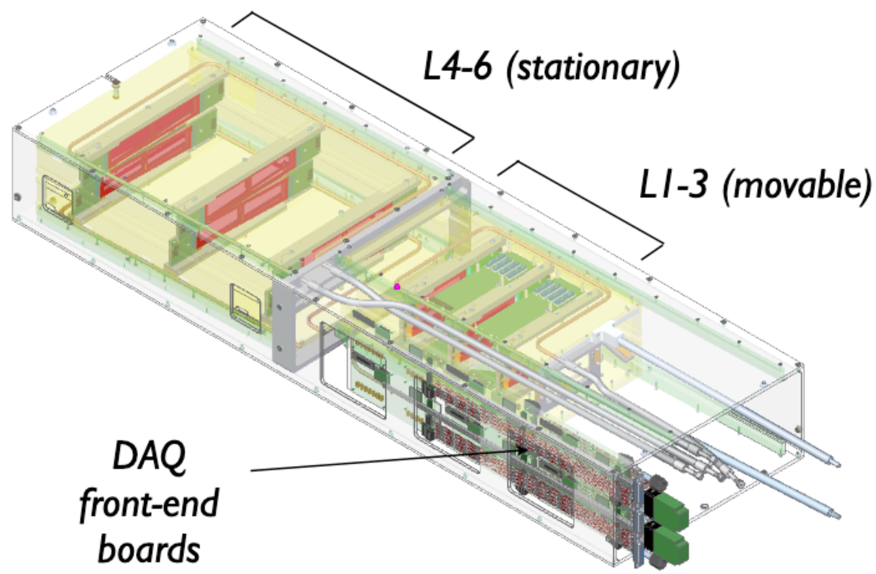


7 double-layers of silicon strips, each plane measures position (~6-10 μm) and time (~2 ns) *with ~0.2% – 0.35% X_0/hit.*

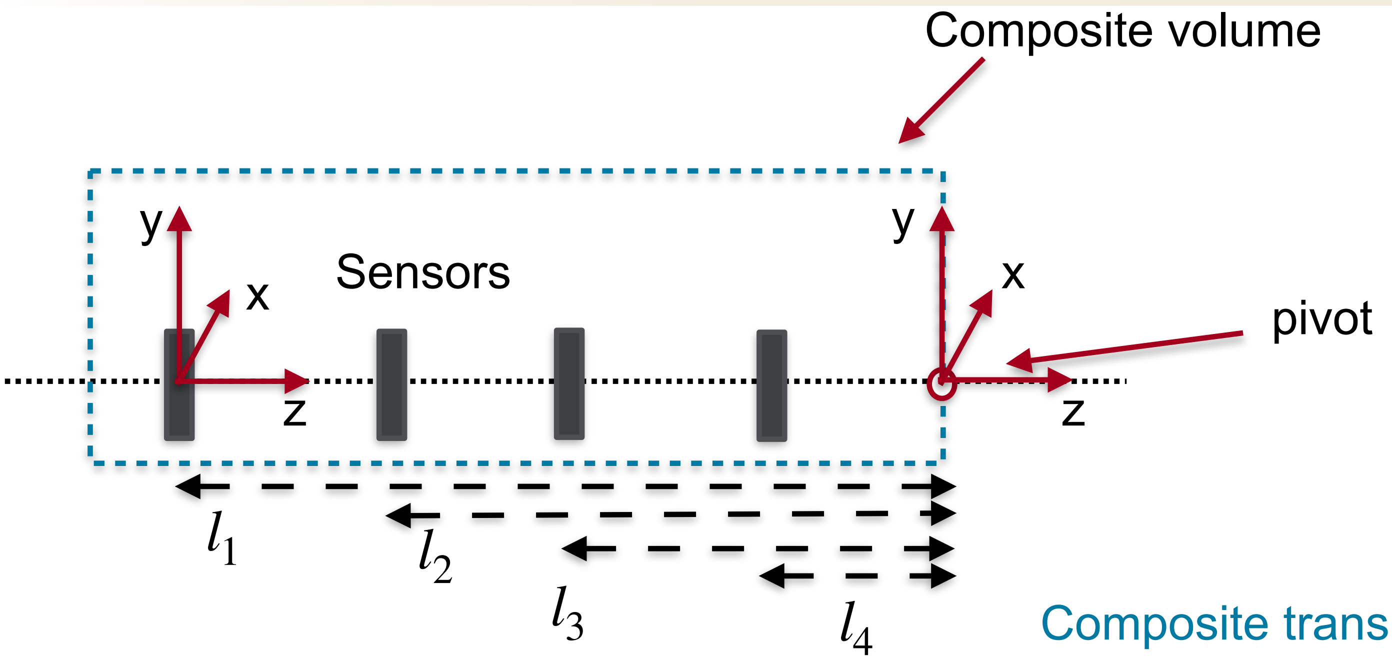
Operates in an extreme environment:

- 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)



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

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

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

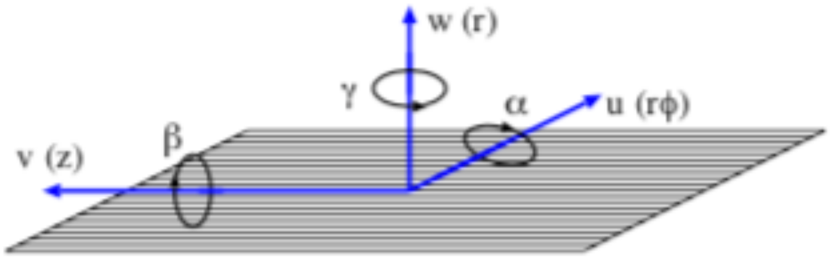
$$\mathbf{r} = \mathbf{R_s}^T\mathbf{q} + \mathbf{T_s}$$

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

$$\mathbf{r} = \mathbf{R_s^T \Delta R_s (q + \Delta q_s) + T_s}$$

- 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

SLAC

- 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

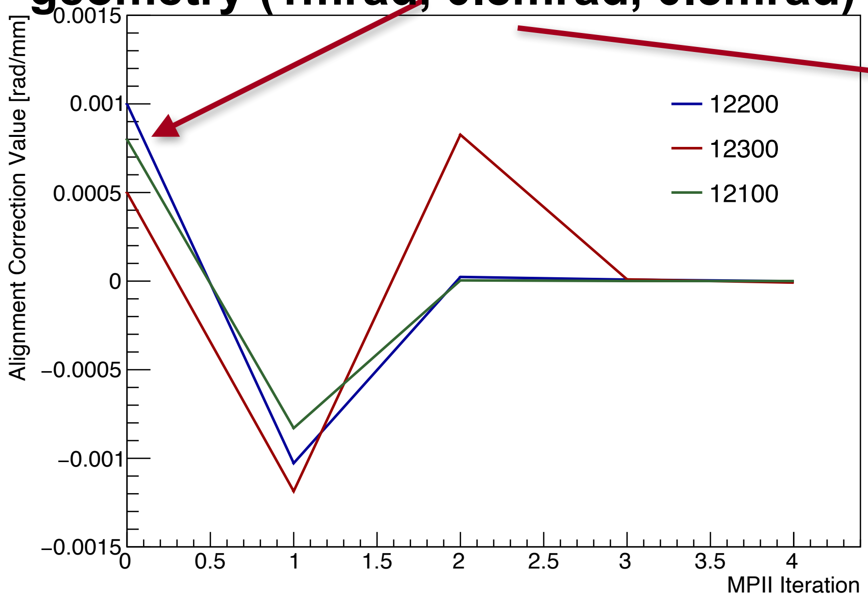
$$\mathbf{a}_c = \sum^{i=n} \mathbf{C}_i^{-1}\mathbf{a}_i$$ ← Inverse relation between sub-components to composite corrections

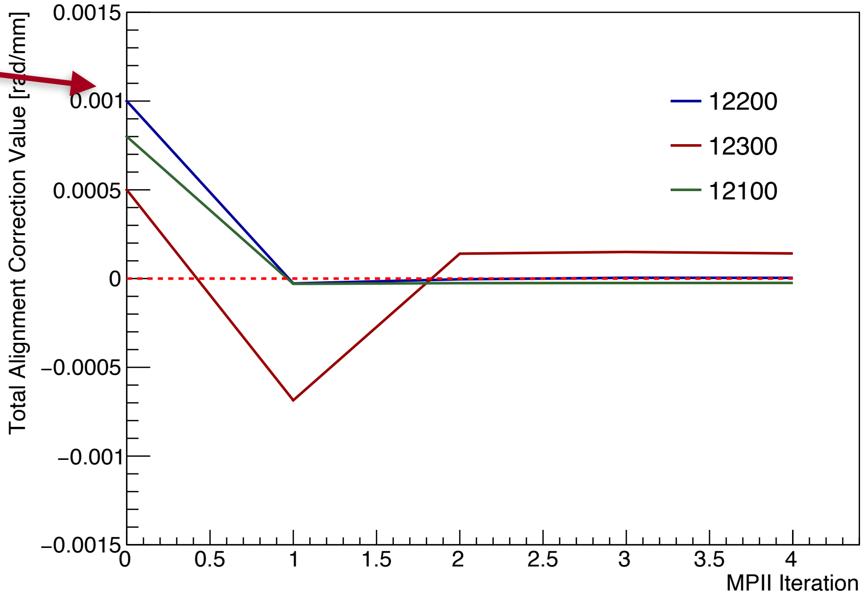
$$0 = \sum_{i=0}^{i=n} \mathbf{C}_i^{-1}\mathbf{a}_i$$ ← Natural hierarchical constraint: sub-components movements keep the pre-aligned global structure fixed. Constraint format supported by MPII

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)

- **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)

Momentum Constraint

Formalism of seed constrained alignment

SLAC

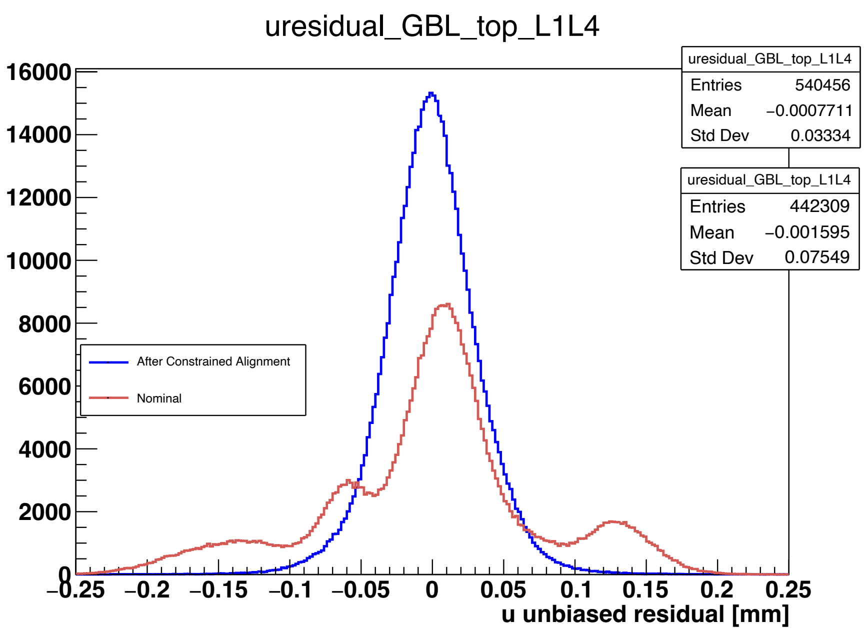
- In order to add an extra handle to improve the alignment solution, external constraints should be used.
- They can take the form of survey measurements, beamspot determination, but also constraints on the track parameters, i.e. momentum constraint from beam energy or calorimeter measurement.
- **If 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**

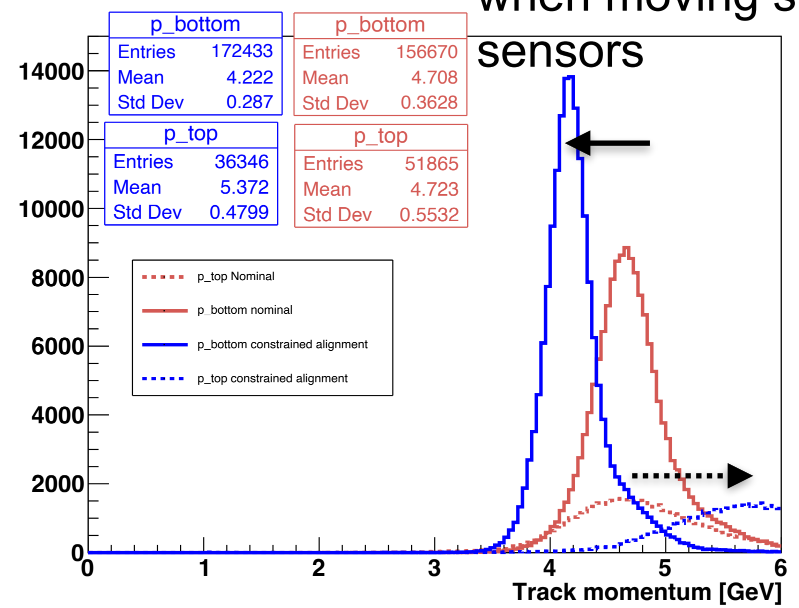
- **Tested a constrained alignment on data fixing the u-channel to nominal positions**

  - 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

Large biases in momentum appear when moving stereo sensors





Implement track parameters constraints in MPII

SLAC

- MPII refits tracks solving for $\partial f / \partial \mathbf{q}_\mathrm{j}$ at each $\mathbf{p} \to \mathbf{p} + \mathbf{dp}$ iteration ($\mathbf{p}$ are the alignment parameters, $\mathbf{q}$ are the track parameters)
- If the local derivatives are "small" then $\mathbf{dq}_\mathrm{j}$ 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

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

The dimension of the label set is arbitrary

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

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

z = residual $(\equiv y_i - f(x_i, \boldsymbol{q}, \boldsymbol{p}))$ σ = standard deviation of the meas

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

$$p_T = \sqrt{p_x^2 + p_y^2}$$

$$\cos\lambda = \frac{1}{\sqrt{1 + (\tan\lambda)^2}}$$

$$p_T^{corr} = E_{beam} * \cos\lambda$$

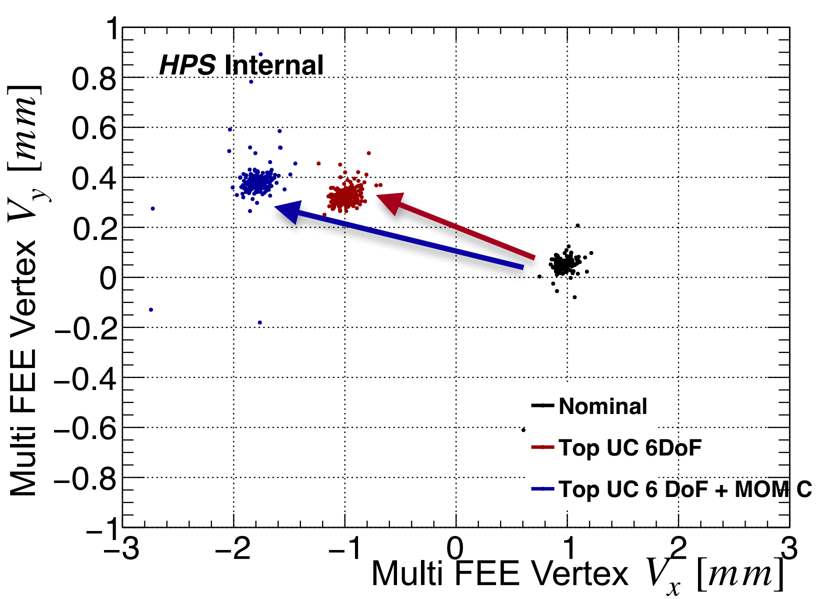
$$\omega = \pm(2.9979e^{-4} \times B)/p_T^{corr}$$

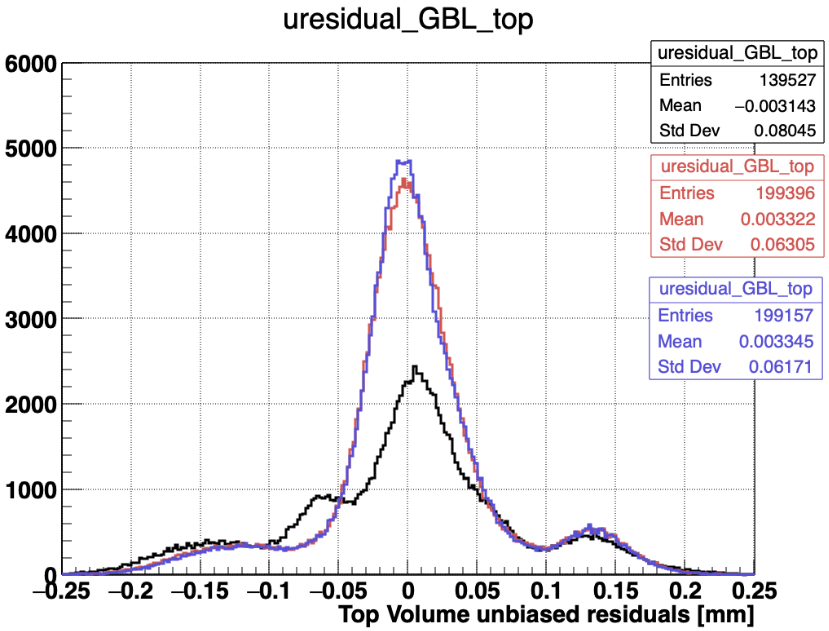
$$\chi^2(\mathbf{x}) = \sum_{i=1}^{n_\mathrm{meas}} (\mathbf{H}_{m,i}\mathbf{x} - \mathbf{m}_i)^T \mathbf{V}_{m,i}^{-1} (\mathbf{H}_{m,i}\mathbf{x} - \mathbf{m}_i) \quad \text{(from measurements)}$$
$$+ \sum_{i=2}^{n_\mathrm{scat}-1} (\mathbf{H}_{k,i}\mathbf{x} + \mathbf{k}_{0,i})^T \mathbf{V}_{k,i}^{-1} (\mathbf{H}_{k,i}\mathbf{x} + \mathbf{k}_{0,i}) \quad \text{(from kinks)}$$
$$+ (\mathbf{H}_s\mathbf{x})^T \mathbf{V}_s^{-1} (\mathbf{H}_s\mathbf{x}) \quad \text{(from external seed)} \quad (9)$$

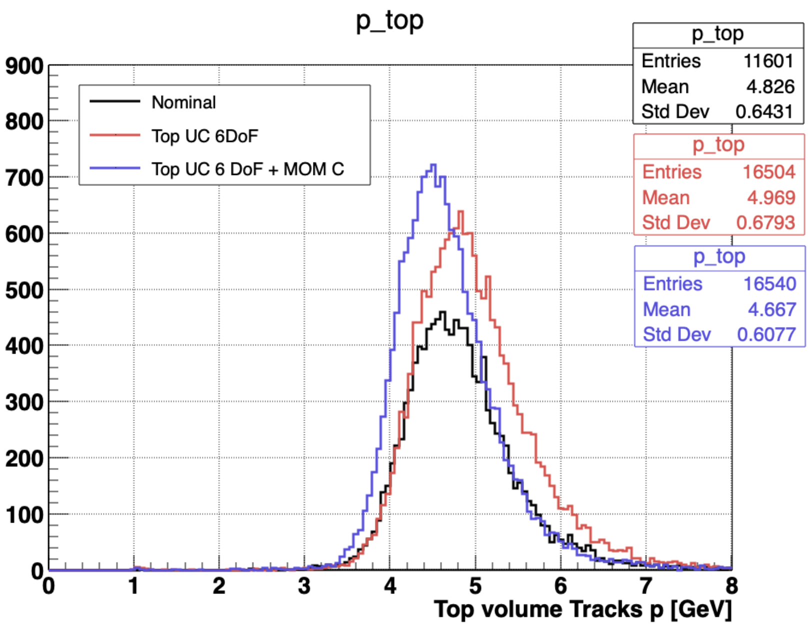
GBL Manual

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.

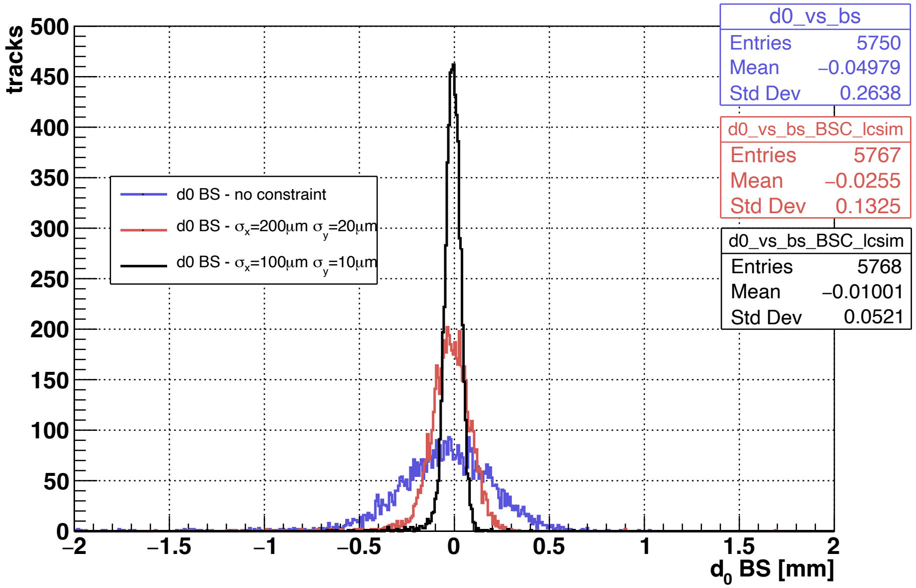
$$\mathbf{U} = \frac{\mathbf{Z} \times \mathbf{T}}{|\mathbf{Z} \times \mathbf{T}|} = \frac{\mathbf{Z} \times \mathbf{T}}{\cos\lambda}$$

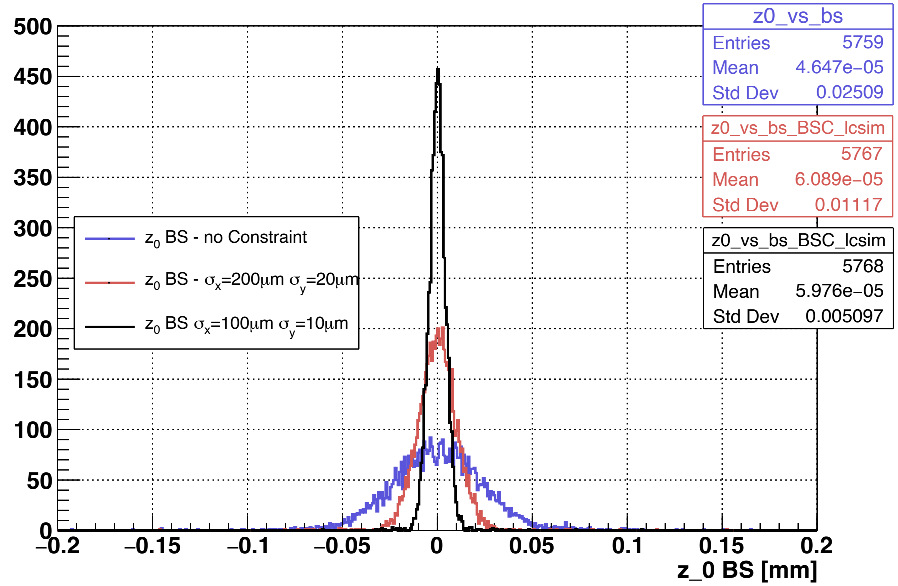
$$\mathbf{V} = \mathbf{T} \times \mathbf{U}$$

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:

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

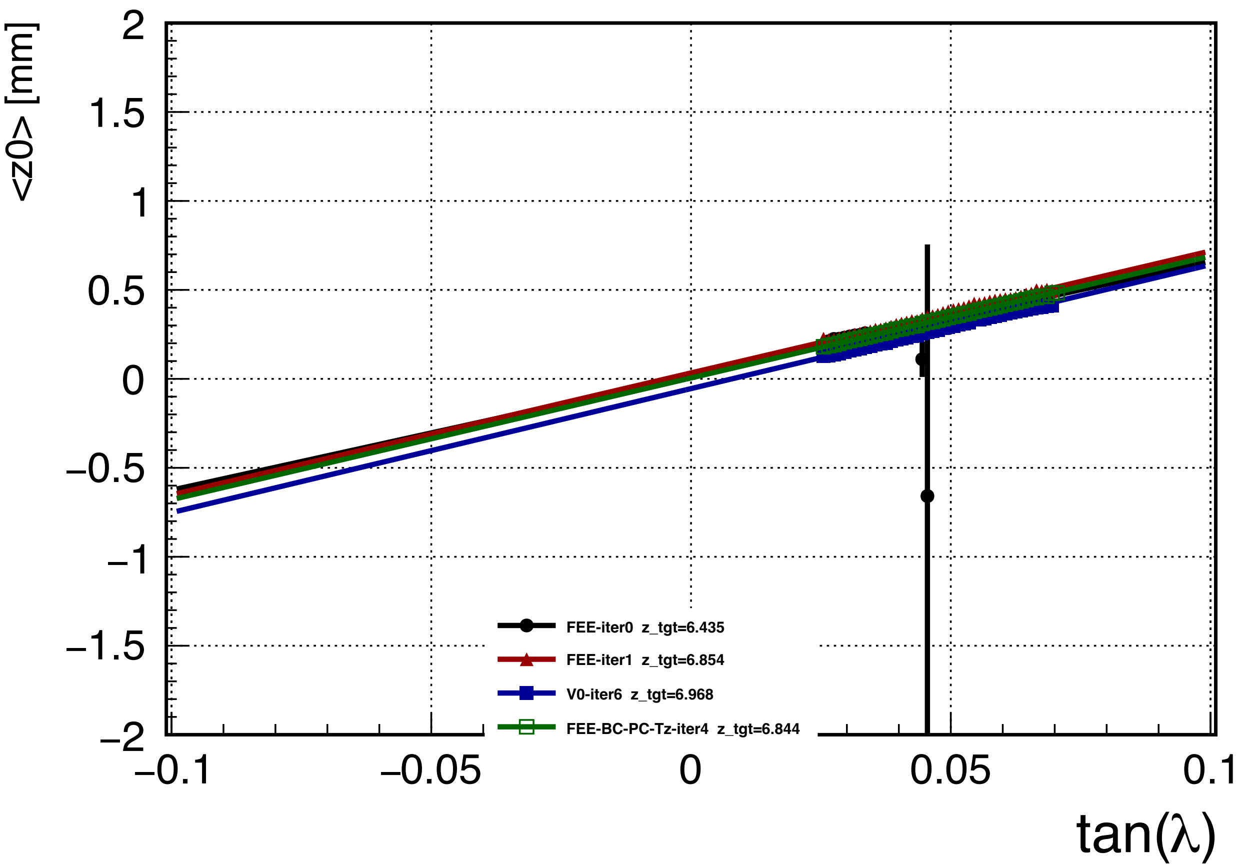
$$P_{p2cl} = \begin{pmatrix} 0 & -1 & 0 \\ \sin\lambda & 0 & \cos\lambda \\ -\cos\lambda & 0 & \sin\lambda \end{pmatrix}$$

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

SVT driven beamspot determination

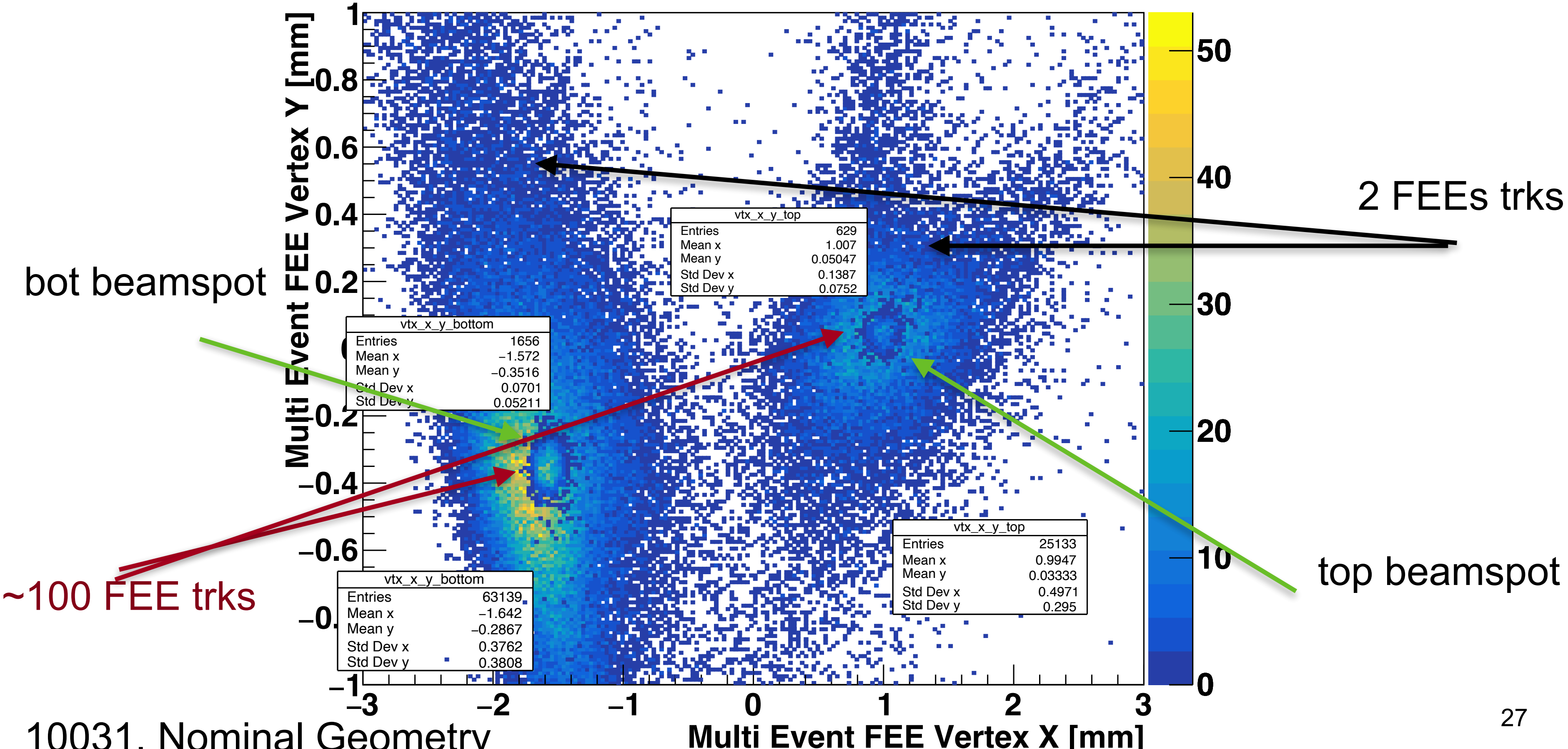
Z0 vs tan Lambda fit

- Use the z0 vs tanLambda fit to determine the Z of the target
- The assumption is that the z_tgt has small dep wrt internal alignment of the sensors. The assumption is valid within O(500um)
- Use FEEs to determine the z_tgt (but e+/e- work similarly for p>1GeV and tanL > 0.03)
- We can the internal z position of the target at each iteration to reach convergence



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

Soft mode cut

Additionally it is possible to add external measurements
On the corrections together with an initial sigma:
free more parameters avoiding largely unconstrained movements

Global parameter measurements

Measurements with measurement *value* and measurement *error* for linear combinations of global parameters in the form

$$y = \sum_{\ell \in \Omega} f_\ell \cdot p_\ell + \epsilon$$

can be given, where ϵ is the measurement error, assumed to follow a distribution $N(0, \sigma^2)$, i.e. zero bias and standard deviation σ. The format is:

```
Measurement     value     sigma
label     factor
 ...
label     factor
```

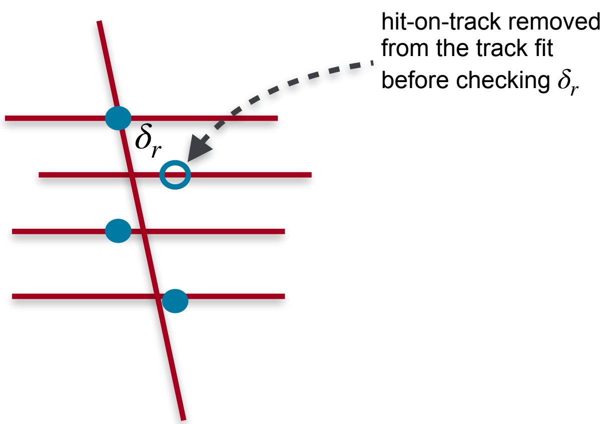
```
Parameter
label    initial_value    presigma
 ...
label    initial_value    presigma
```

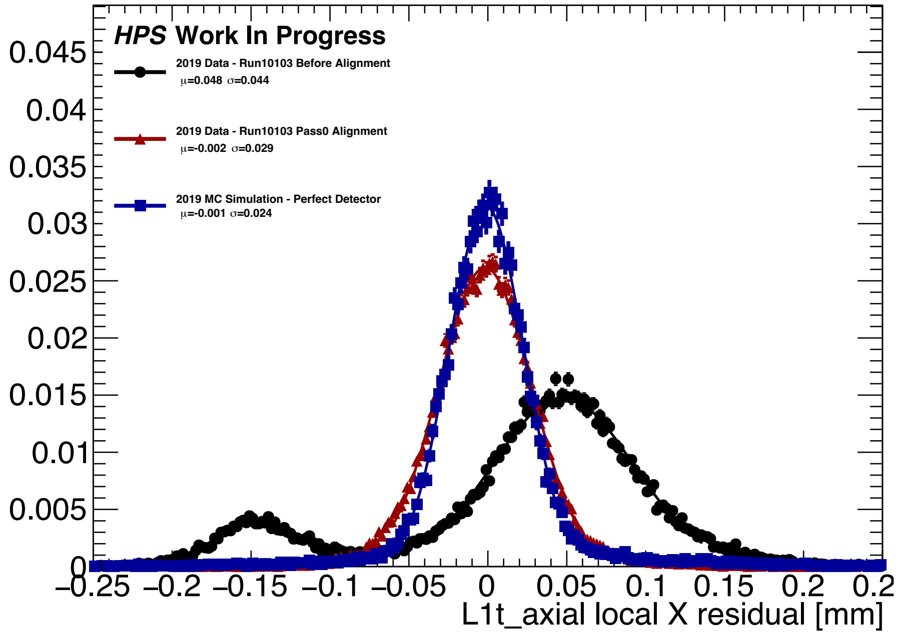
$s_\ell > 0$: The parameter is *variable* with the given initial value. A term $1/s_\ell^2$ is added to the diagonal matrix element of the global parameter to stabilize a perhaps poorly defined parameter. This addition should *not* bias the fitted parameter value, if a sufficient number of iterations is performed; it may however bias the calculated error of the parameter (this is available only for the matrix inversion method).

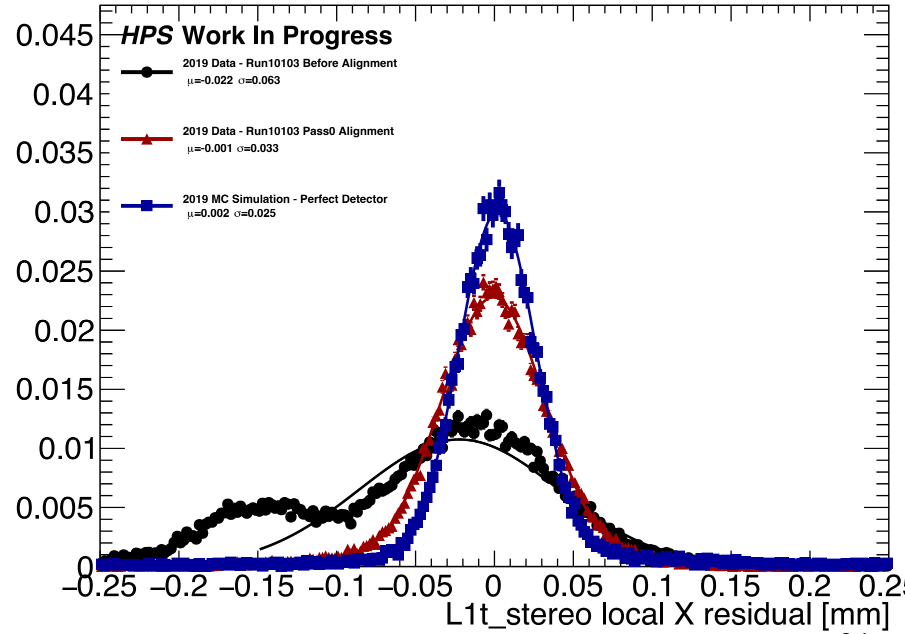
2019 - Pass1-FEE detector, presented for Jeopardy '20

Alignment performance - Unbiased Residuals

- Checked alignment solution quality by evaluating unbiased residuals distributions
- Mean linked to the residual position misalignment
- Large improvement in the newly placed thin-sensors
- Resolution to be improved to get closer to ideal geometry (from perfect MC)

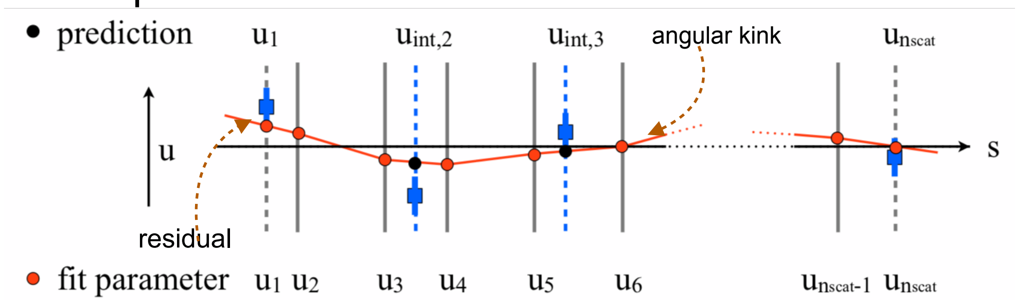


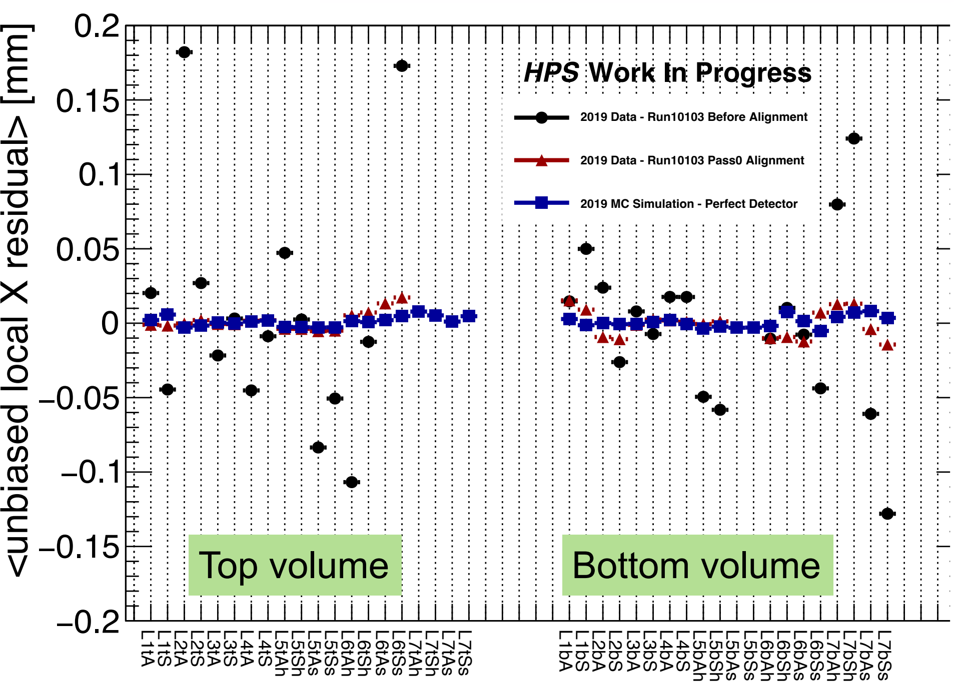


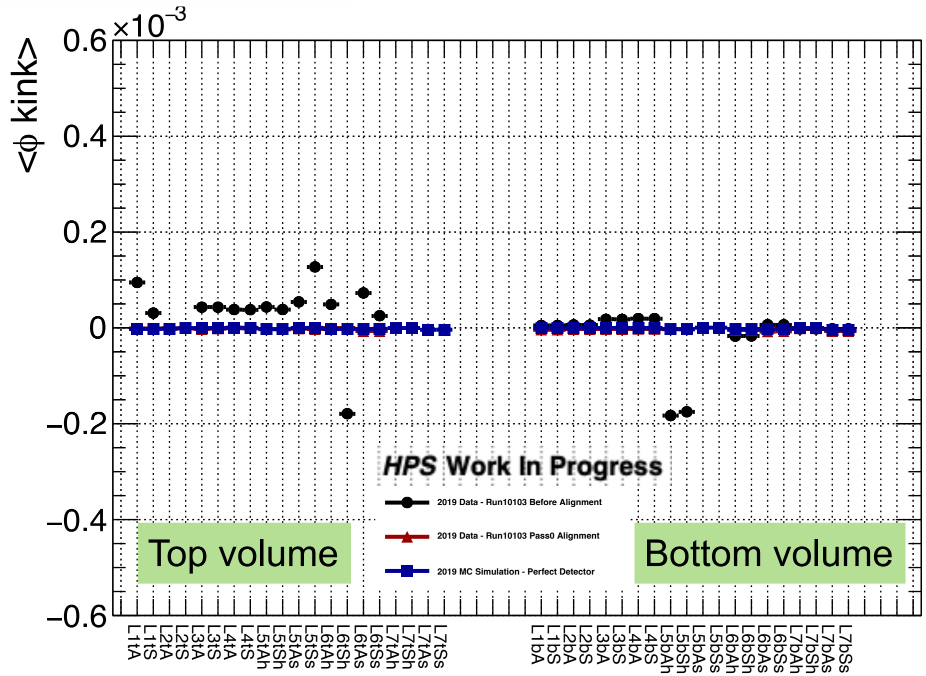


Alignment performance - Unbiased Residuals

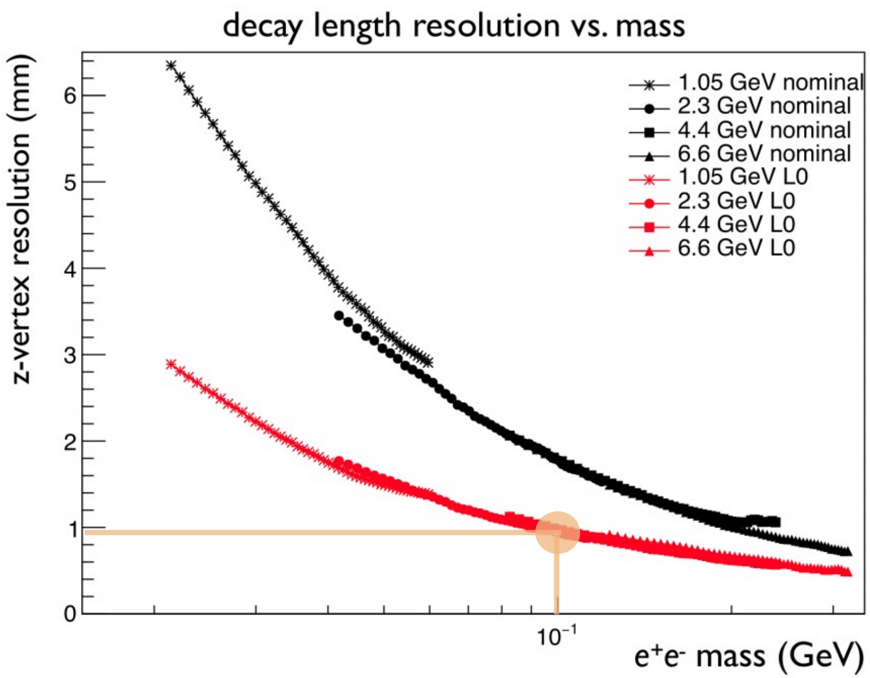
- Initial misalignments up to 200um recovered by current alignment procedure across all detector
- Residual misalignment from first calibration pass ~ 10um, work in progress
- Angular kinks as expected from MC ideal simulation

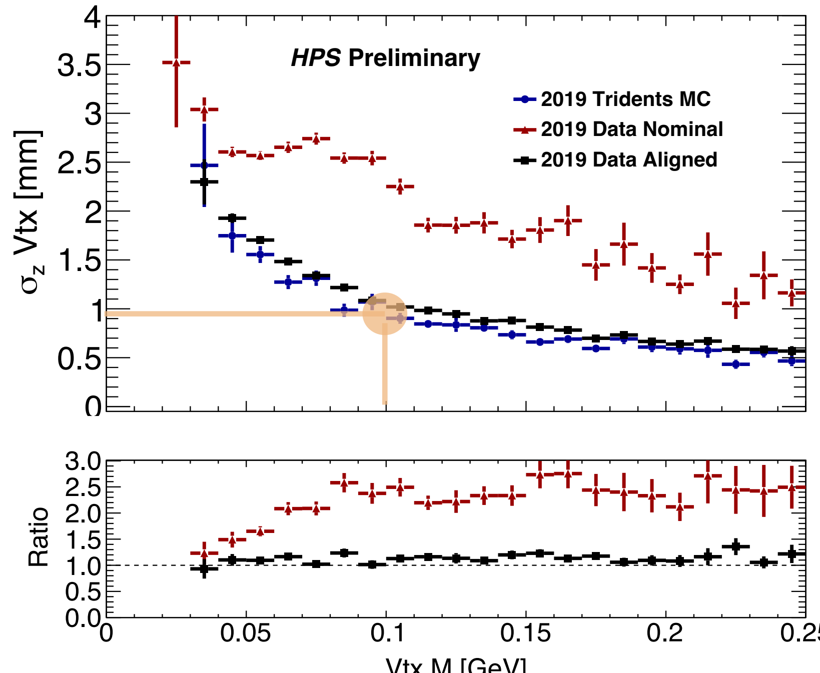






Detector performance - Vertexing

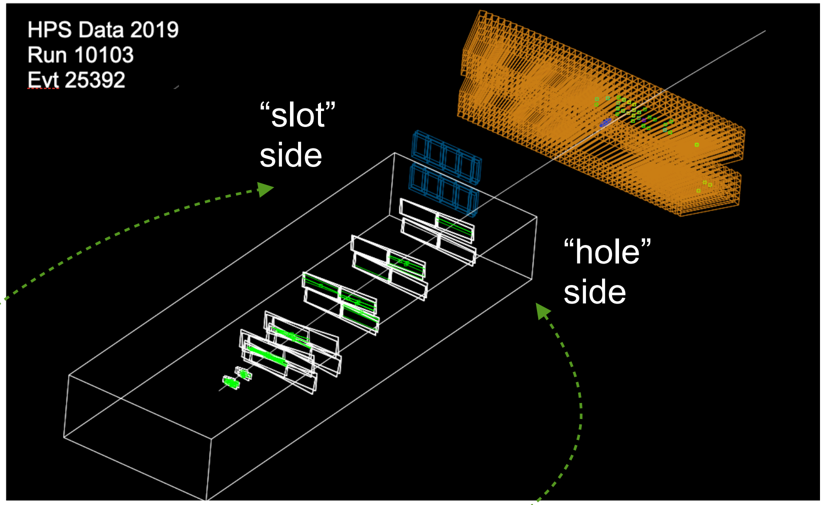


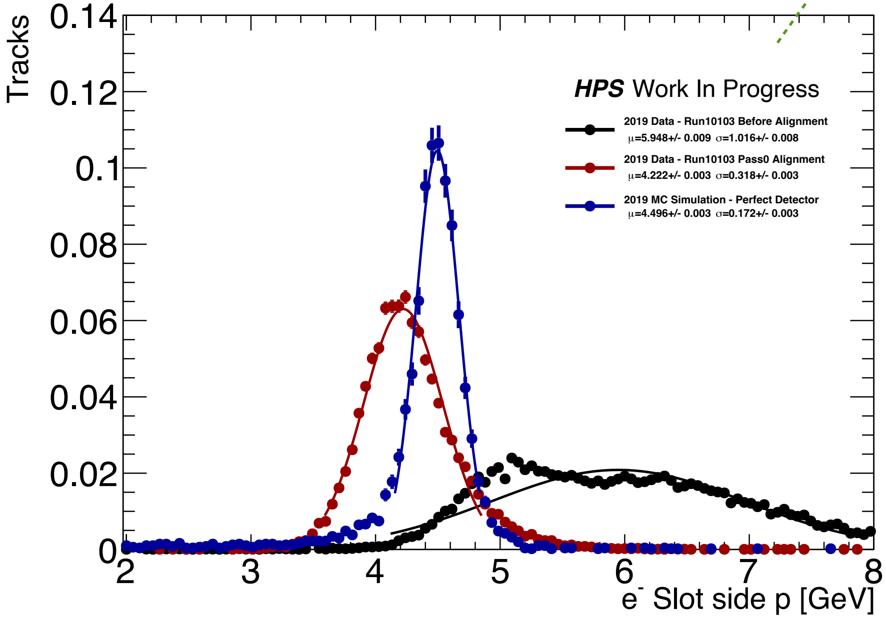


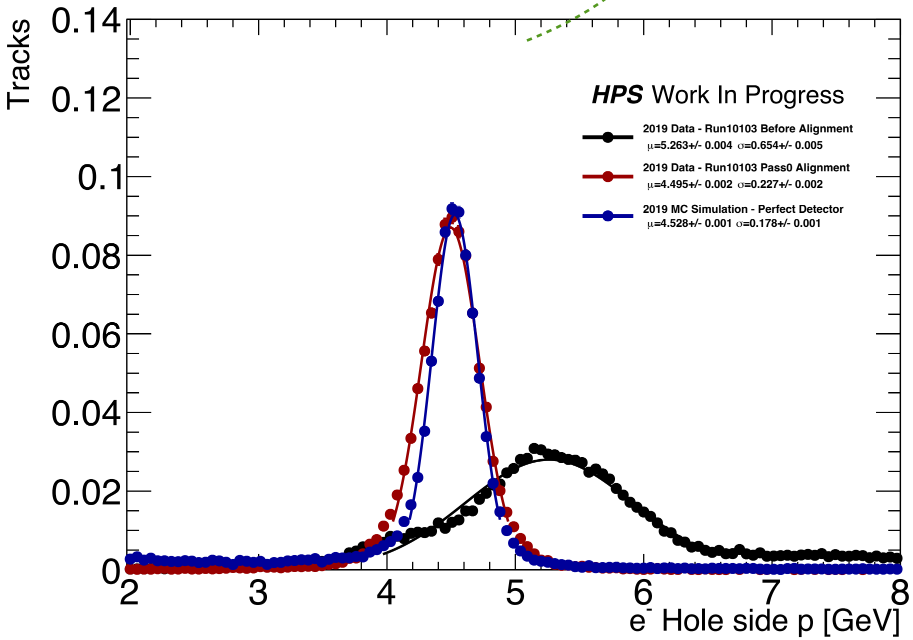
- Preliminary alignment show that HPS reconstruction is able to achieve simulated design performance
- Resolution extracted from gaussian fit on the core of the vertex distribution
- In these results *optimistic* MC simulation has been used (no beam background / pileup included). A simulation that would have similar conditions of data should cover up residual resolution difference

SVT Performance - Momentum Scale and Resolution

- Elastically beam scattered electrons are used to align the SVT with momentum scale constraint
  - Clean event selected by **single high-energy cluster in calorimeter**
  - **Known track momentum for weak-mode suppression**
  - **Only one side of the detector illuminated**:
    - Asymmetry detector halves alignment performance
    - Slot side momentum scale suffers of hole-on-track (one missing working layer for bottom)
- Momentum calibration for positrons/electrons can be checked using E/p method





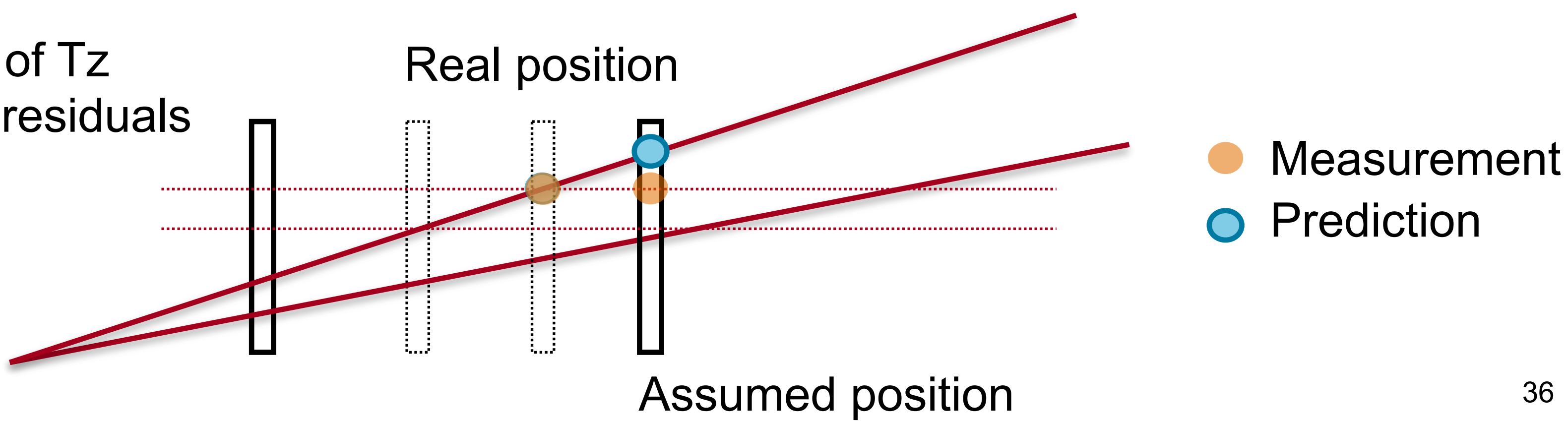


2019 - Top Volume - Pass2 using e+/e- and FEEs

SVT Performance - Momentum Scale and Resolution

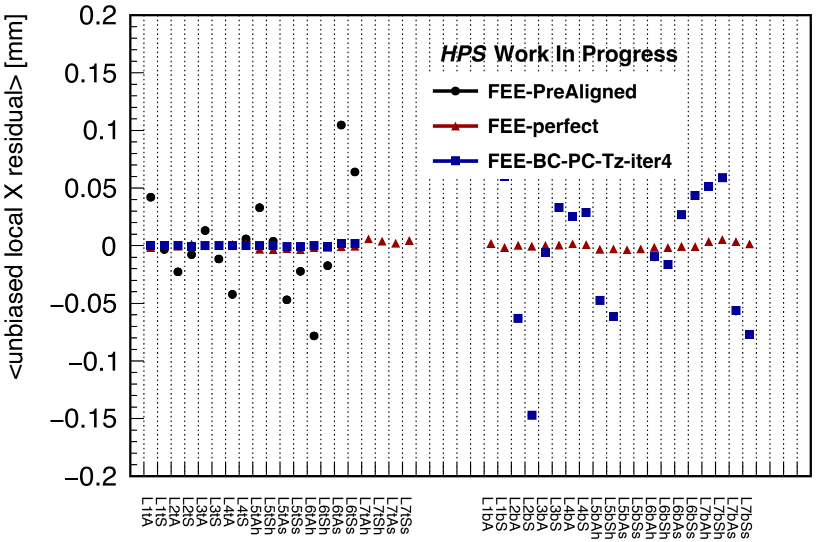
- From 2020 alignment pass several problems appear:
- Top volume has poor momentum performance
  - Also observed strong momentum dependence vs tan Lambda
- Residuals vs v (predicted) and u (measured) show large slopes
  - v-dep can be an indication of Rw to be corrected
  - u-dep can be an indication of Tz to be corrected

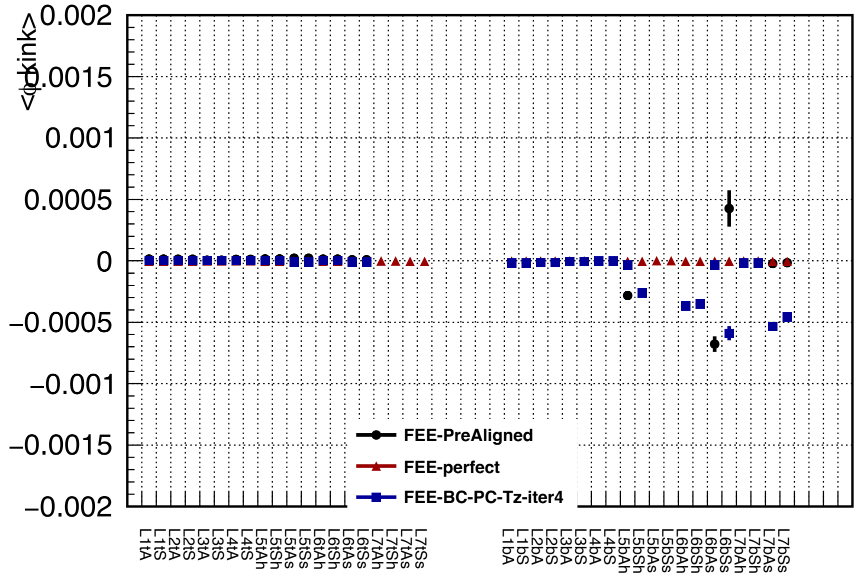
Example of Tz effect on residuals



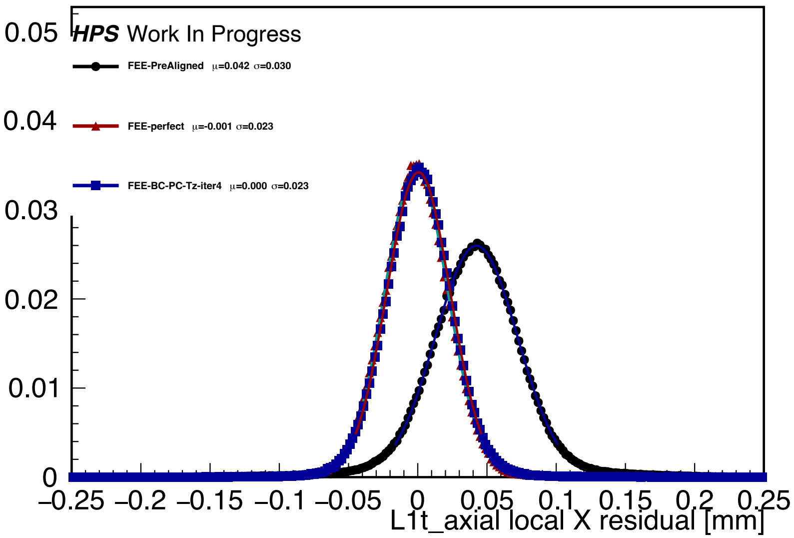
SVT Performance TOP - Realignment

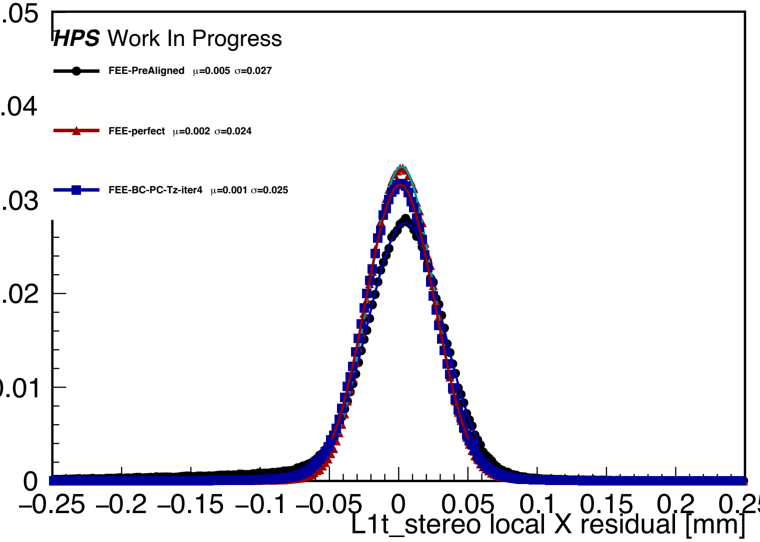
- Performed another pass of Top Volume
  - Used 2016 Survey constants for back UChannel, Modules and Sensors
  - Aligned Tu-Rw of front L1,L2,L3,L4 - Used soft mode cut at 100um, 2mrad.
  - Aligned Tu-Rw of front L1,L2,L3,L4 - Used soft mode cut at 50um, 2mrad.
  - Aligned Tu-Rw-Tz of front L1,L2,L3,L4,L5 - Used soft mode cut at Tu:50um, Rw:2mrad, Tz:200um, Used BS constraint using z from Z0vsTanL slope (-6.9mm), PC from FEEs.
  - Aligned Tu-Rw-Tz of L3,L4,L5,L6 - soft mode cut, BSC, PC using FEEs

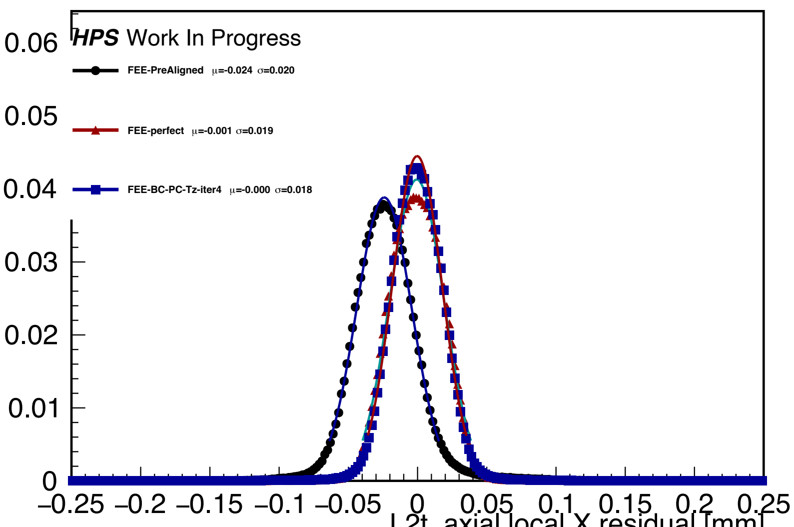


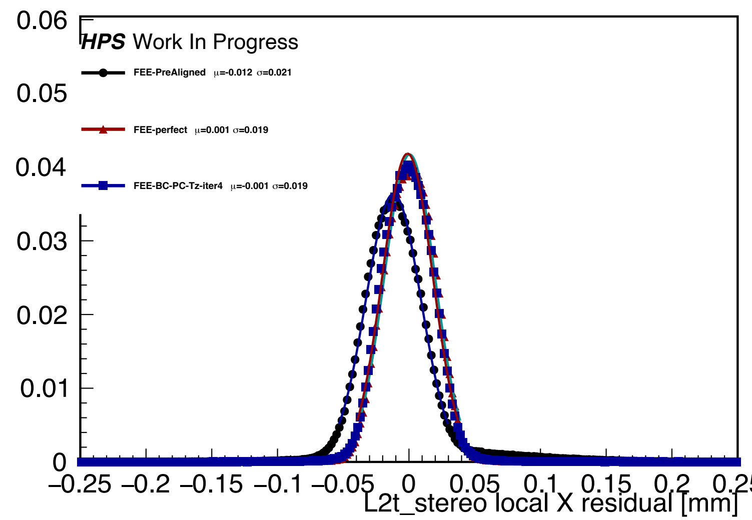


SVT Performance

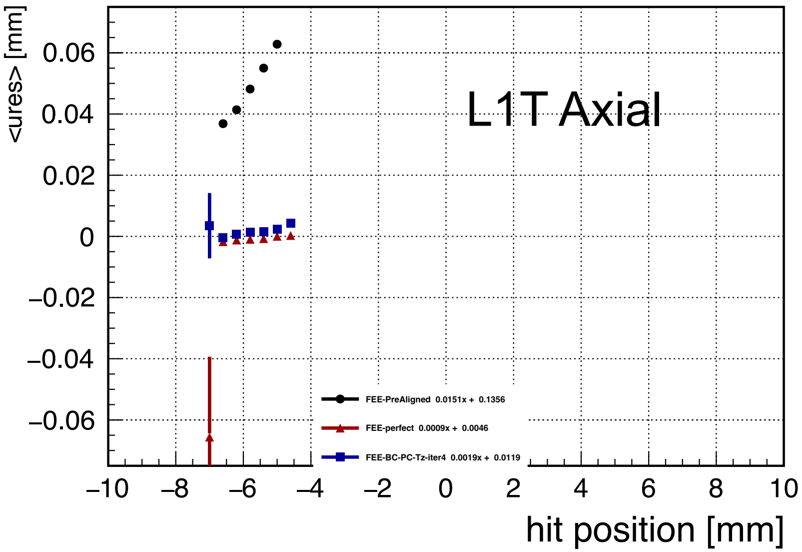


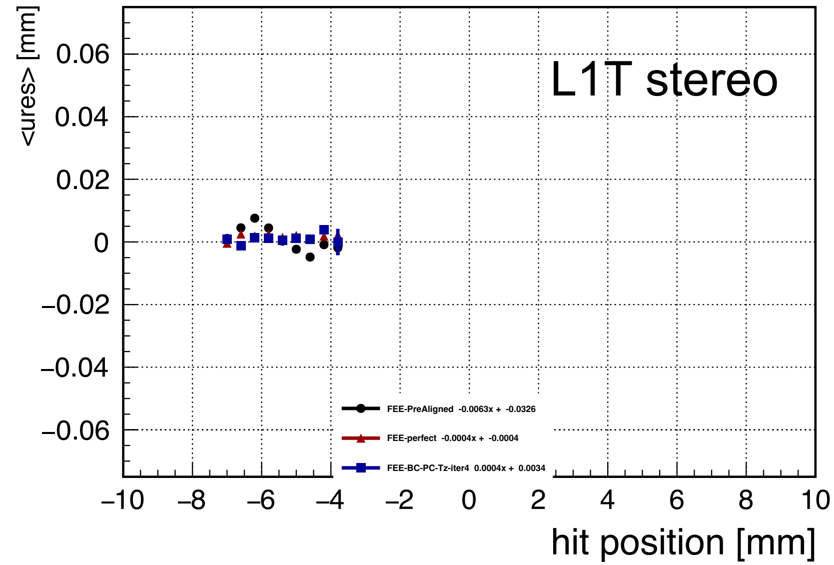


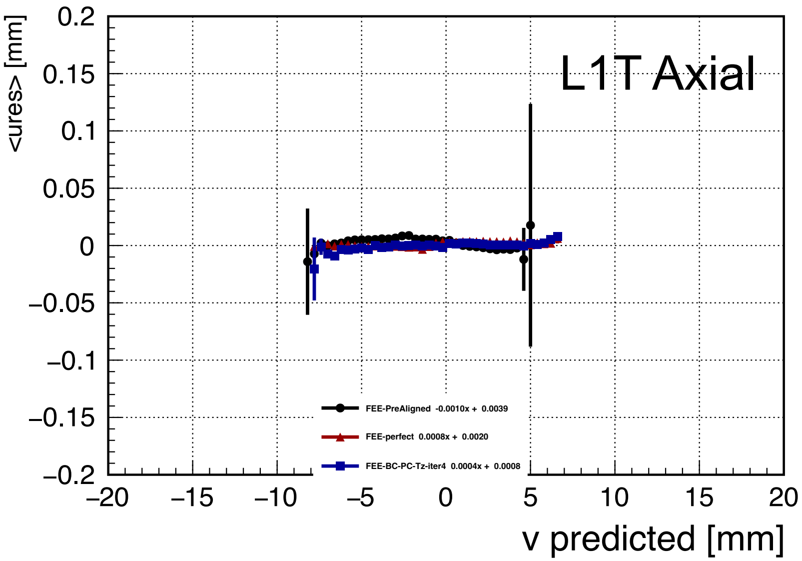


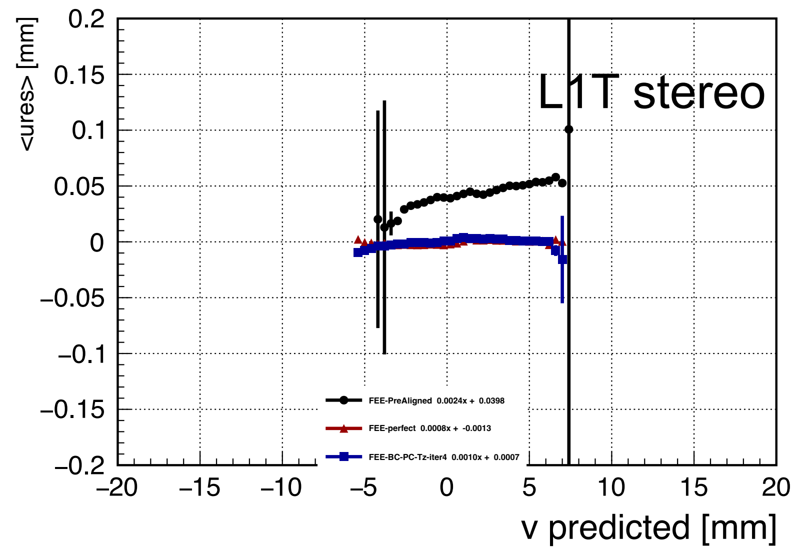


SVT Performance TOP

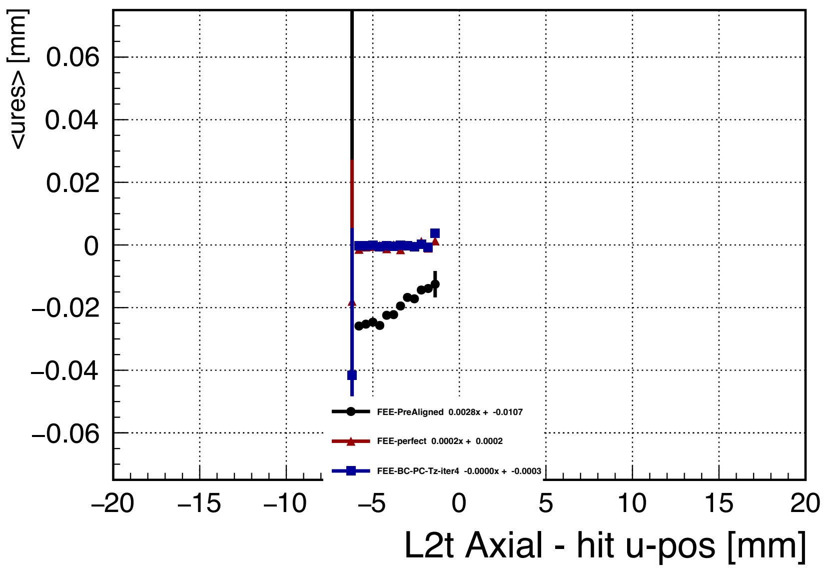


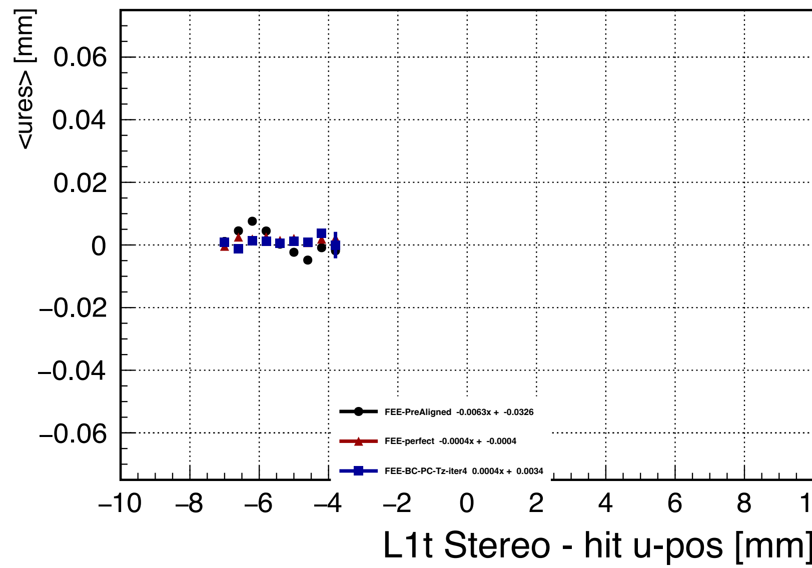


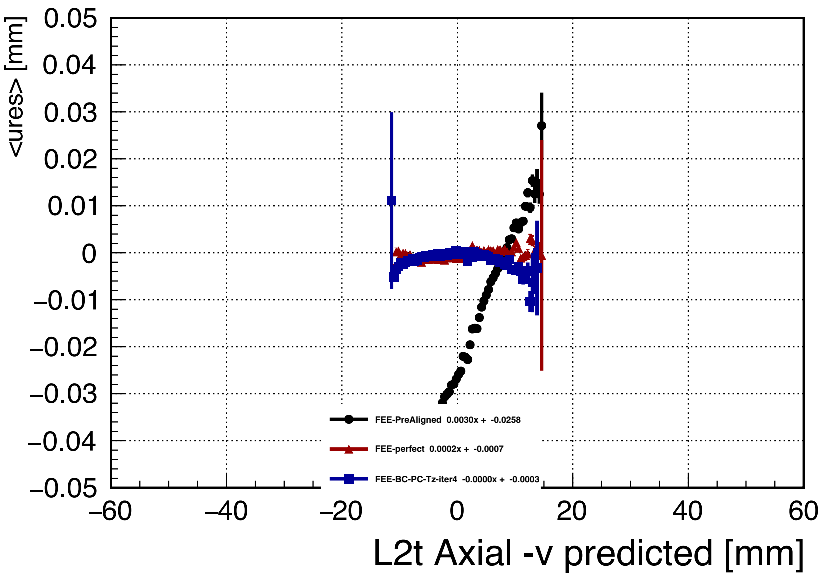


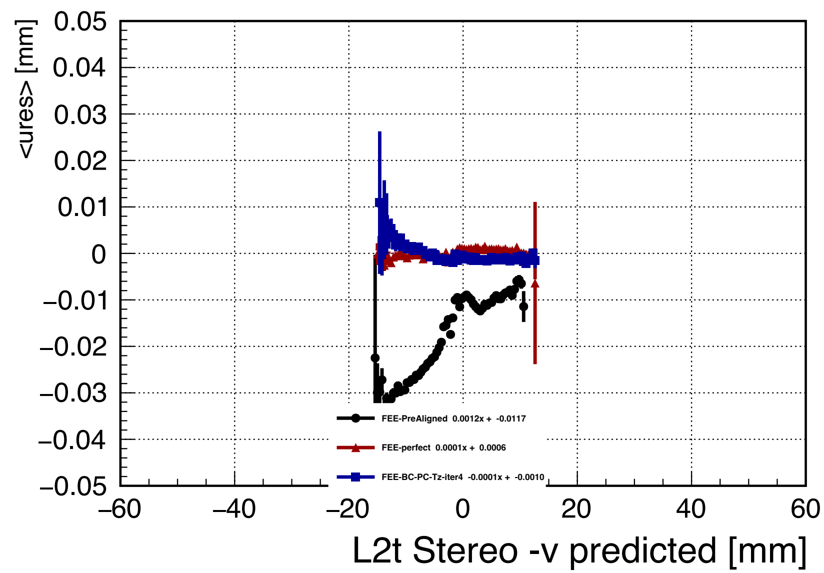


SVT Performance TOP

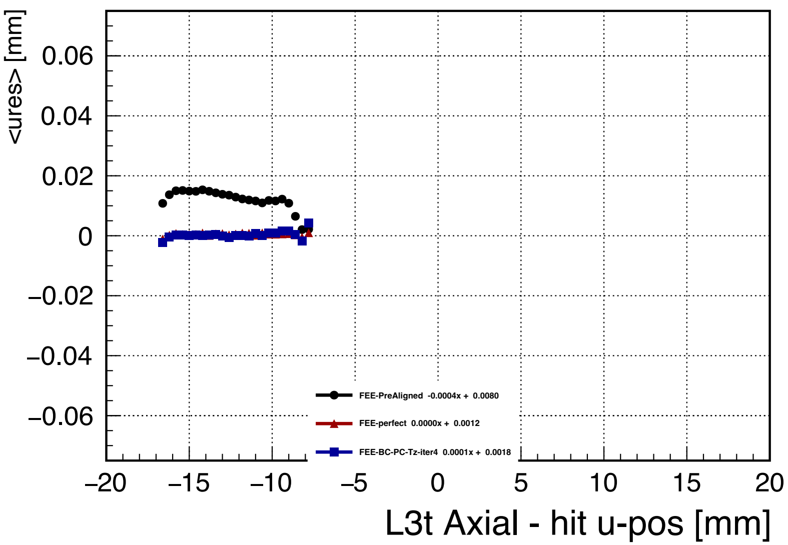


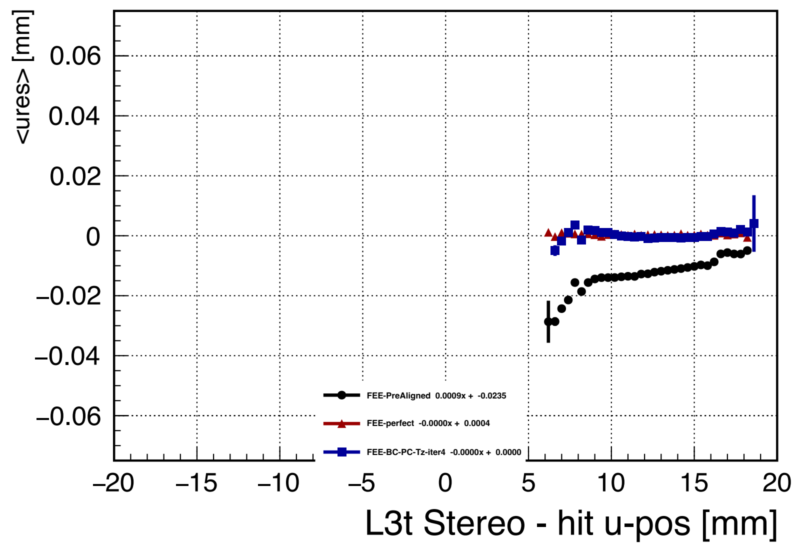


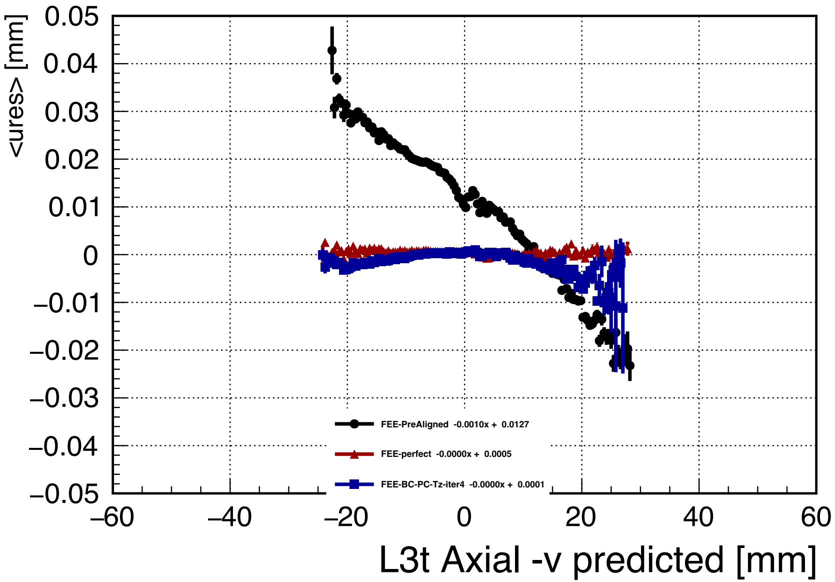


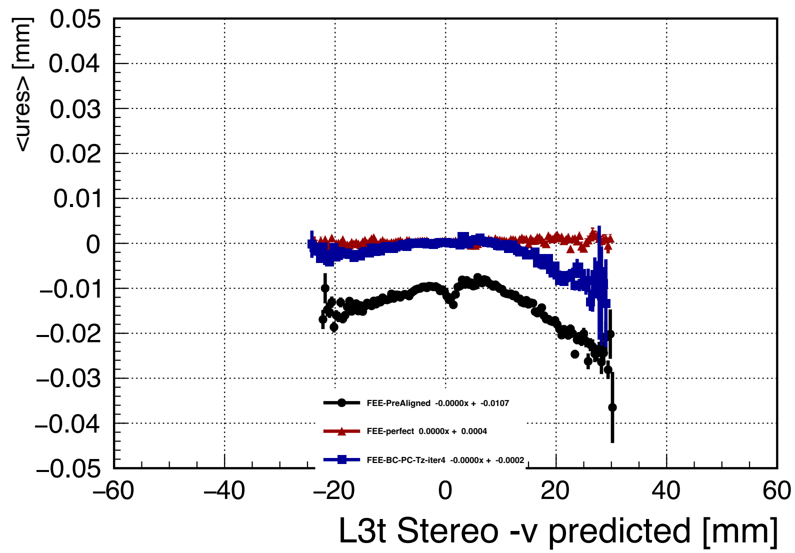


SVT Performance TOP

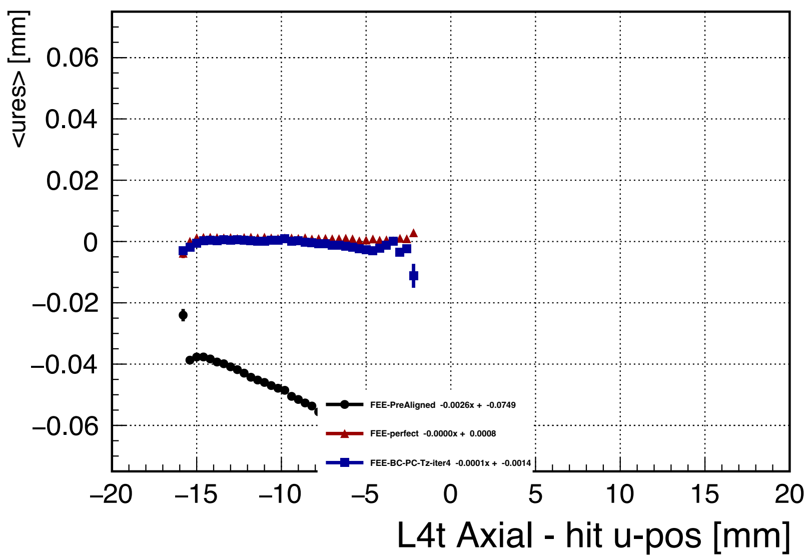


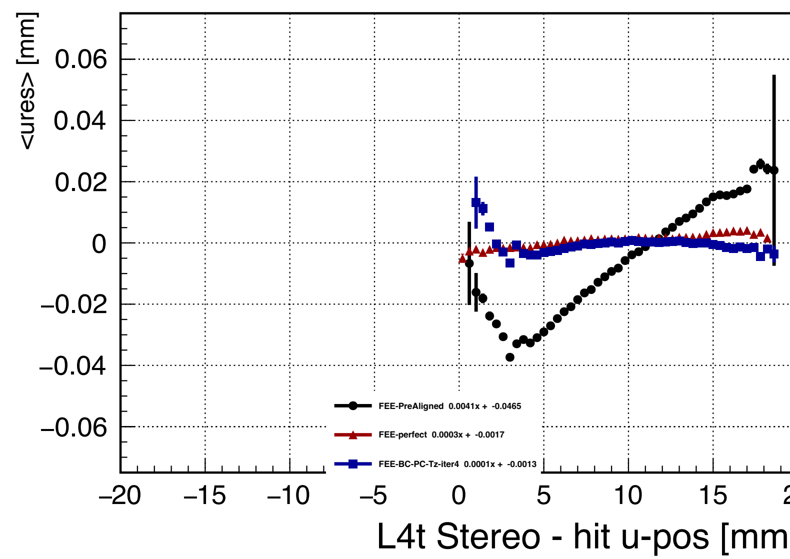


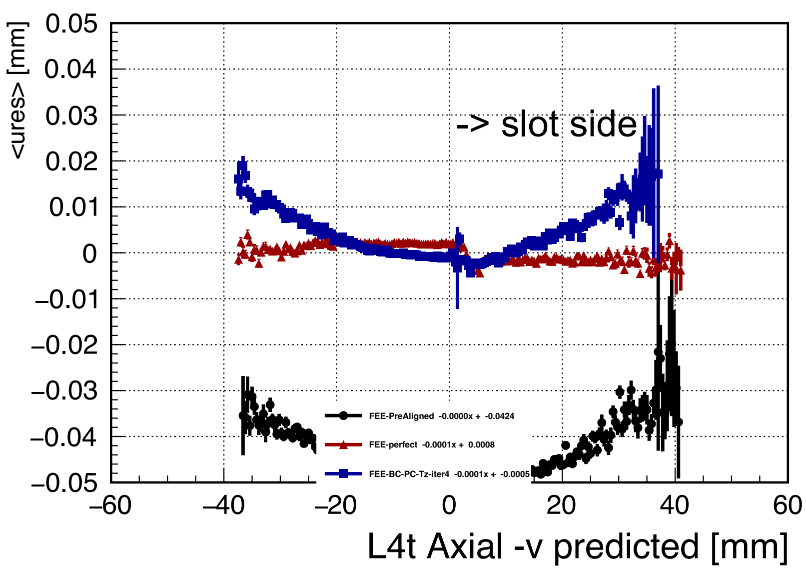


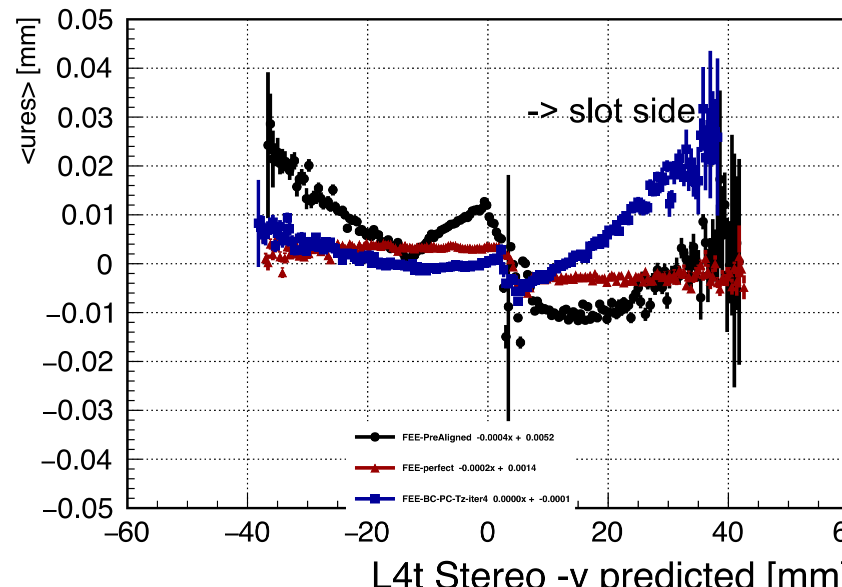


SVT Performance TOP

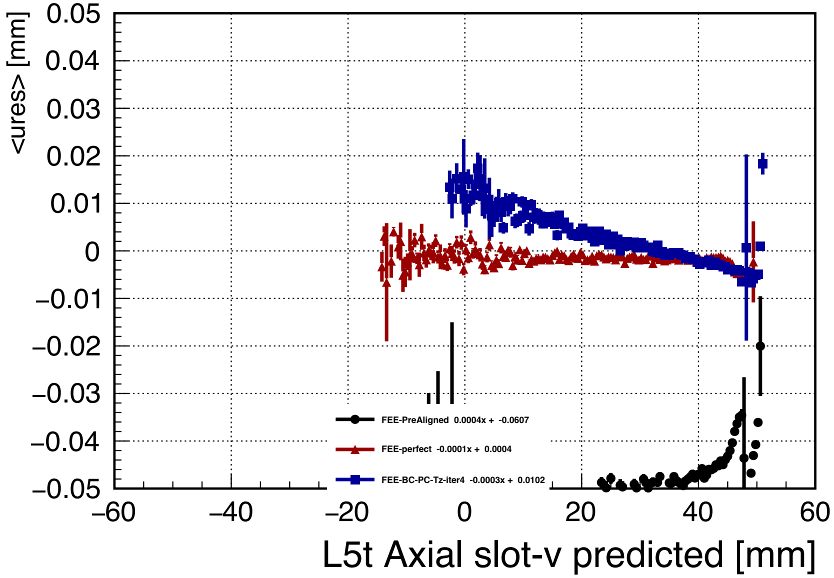


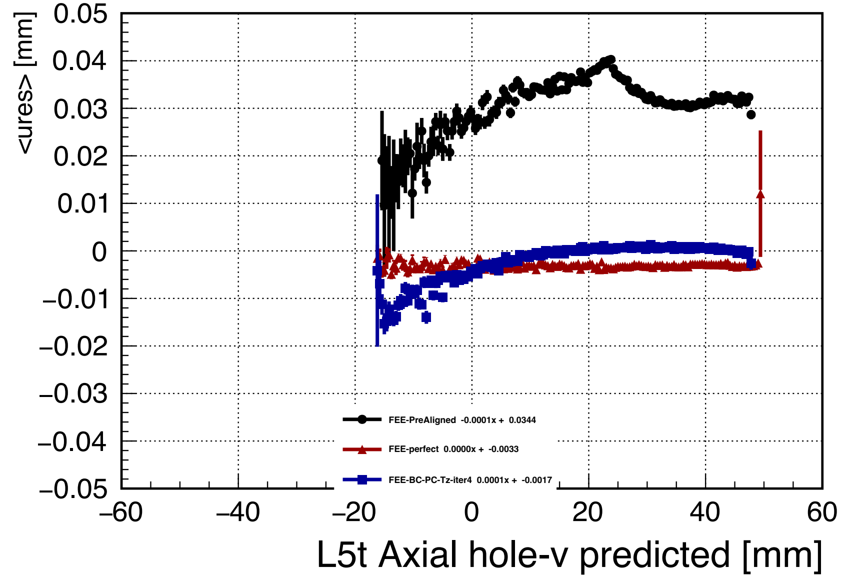


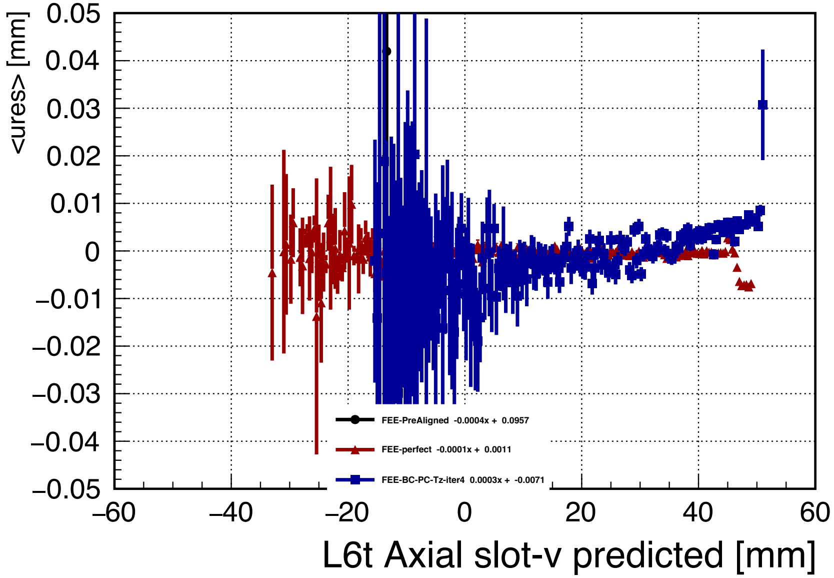


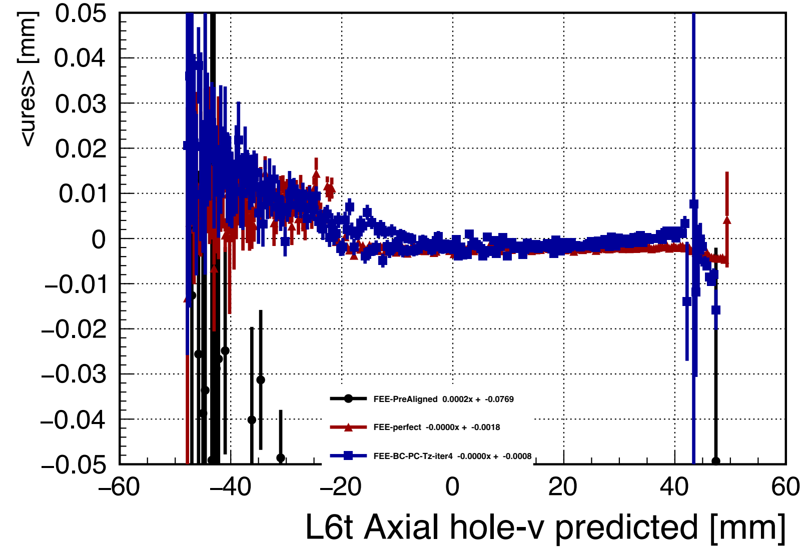


SVT Performance TOP

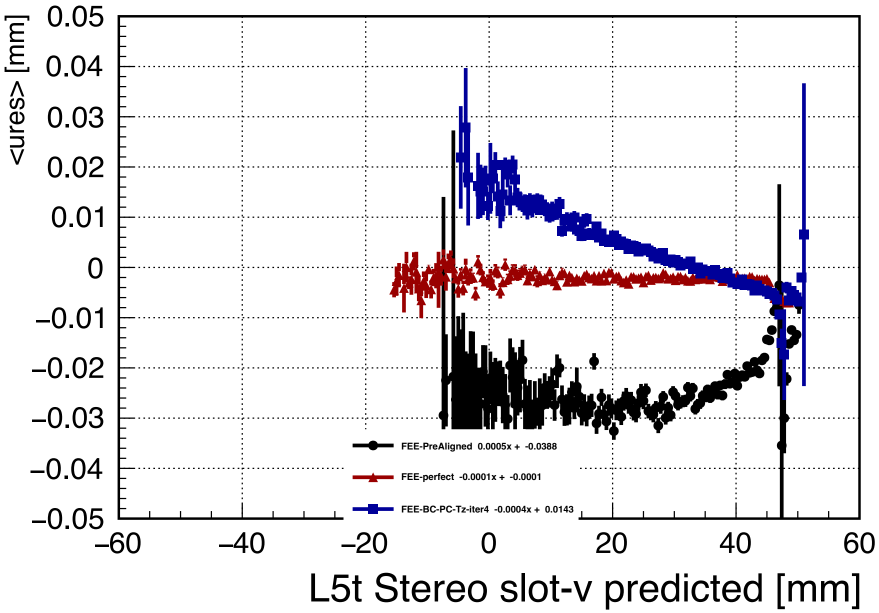


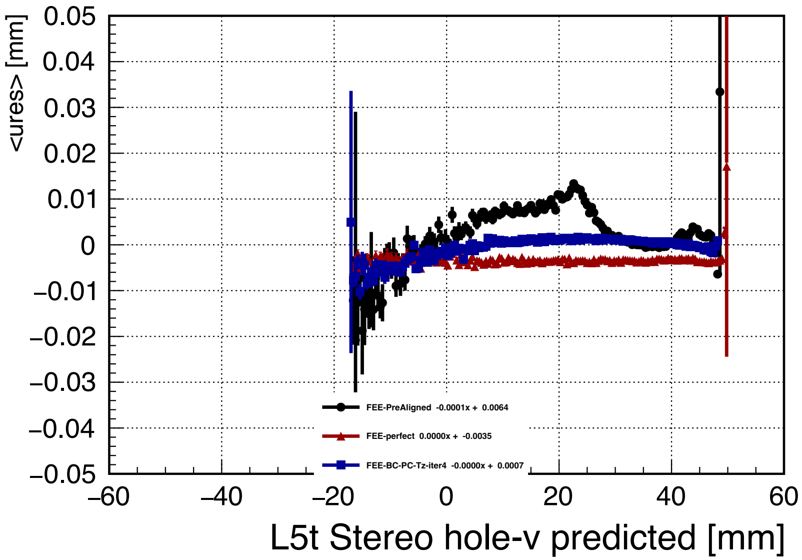


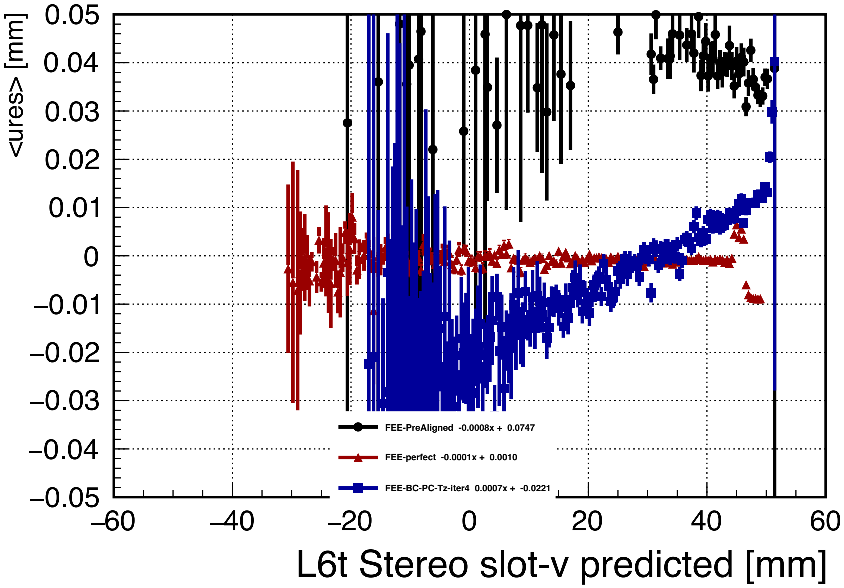


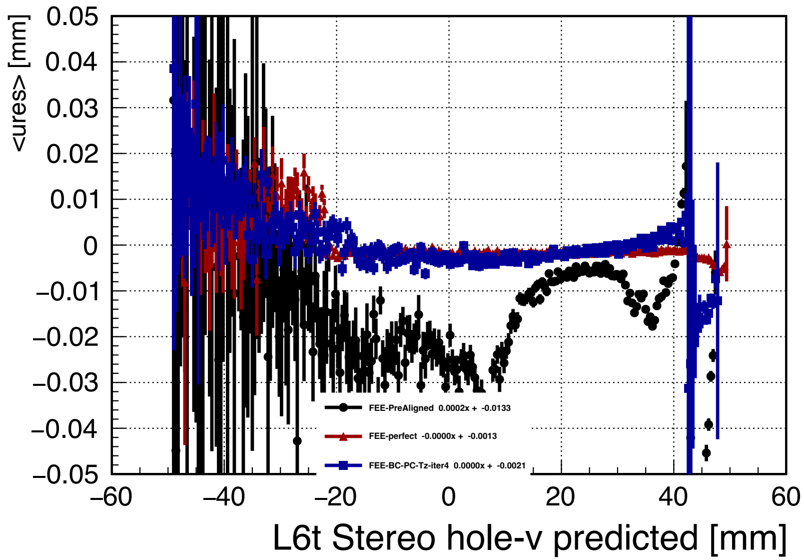


SVT Performance TOP

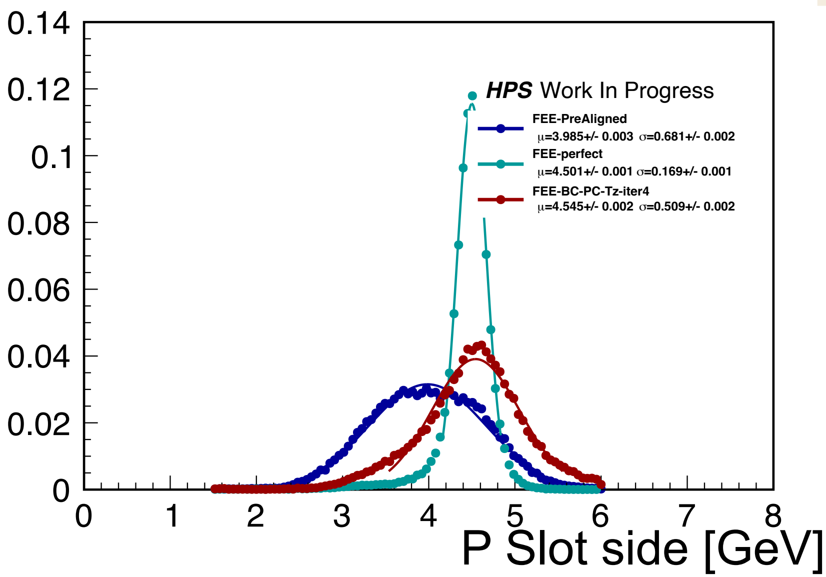


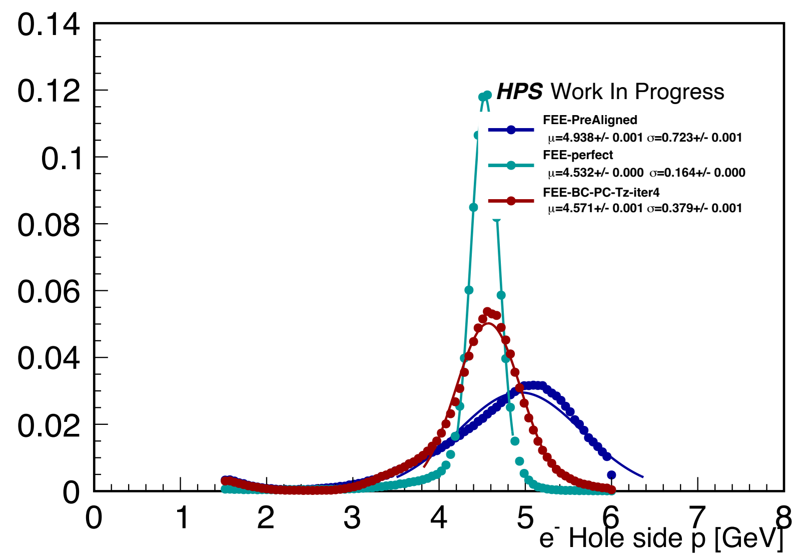


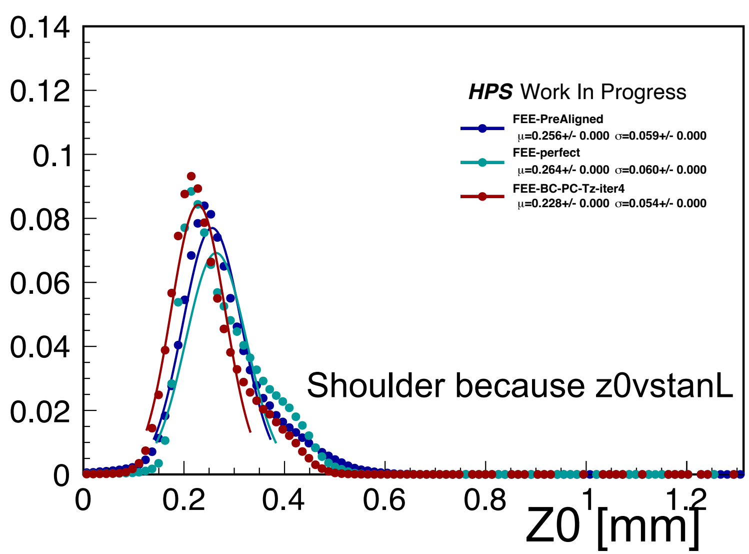


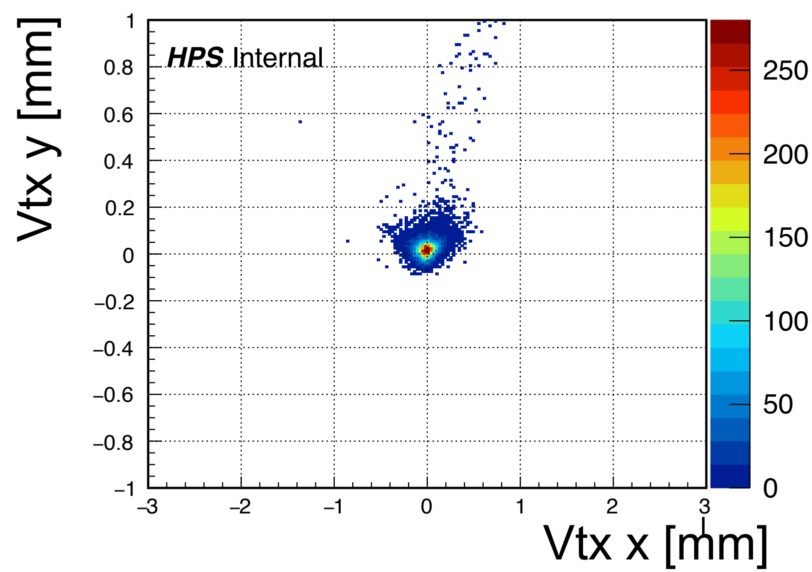


SVT Performance TOP

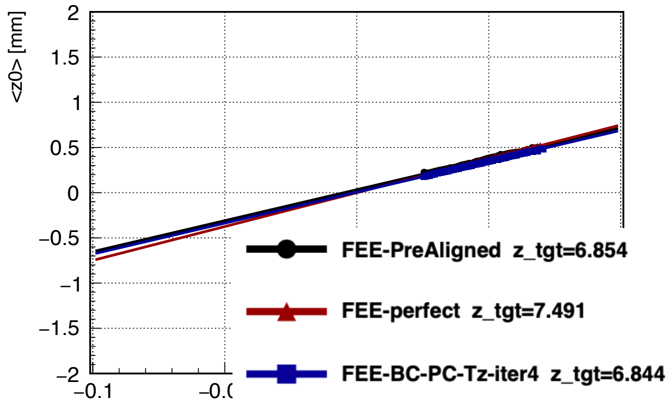




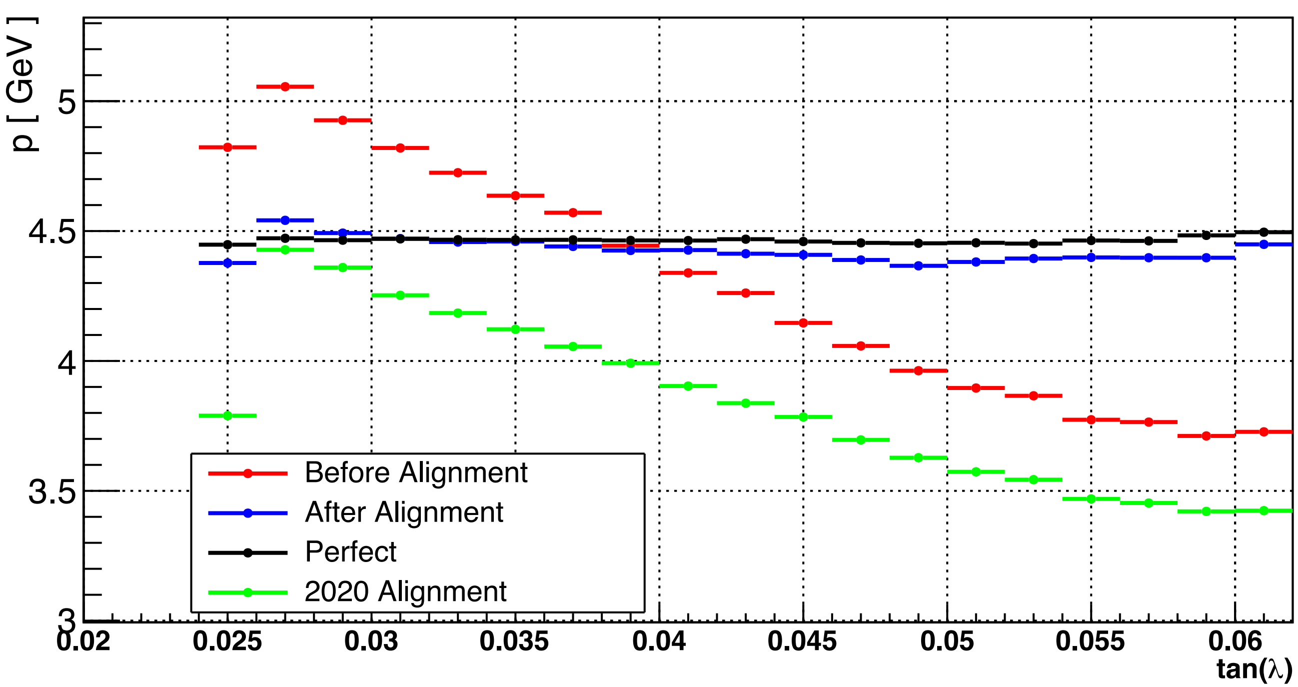




SVT Performance TOP



- This iteration
  - Fixes ures vs u/v dependence in large amount
  - Fixes PvsTanLambda
  - Keeps the BC at 0,0 in x/y with internal constraint at -6.9 mm
  - Fixes hole/slot dependence on momentum
- Necessary other Rw iteration for Back-UChannel Slot side
- Nees to check v0_mu vs InvM and v0_momentum in e+/e- sample



Summary

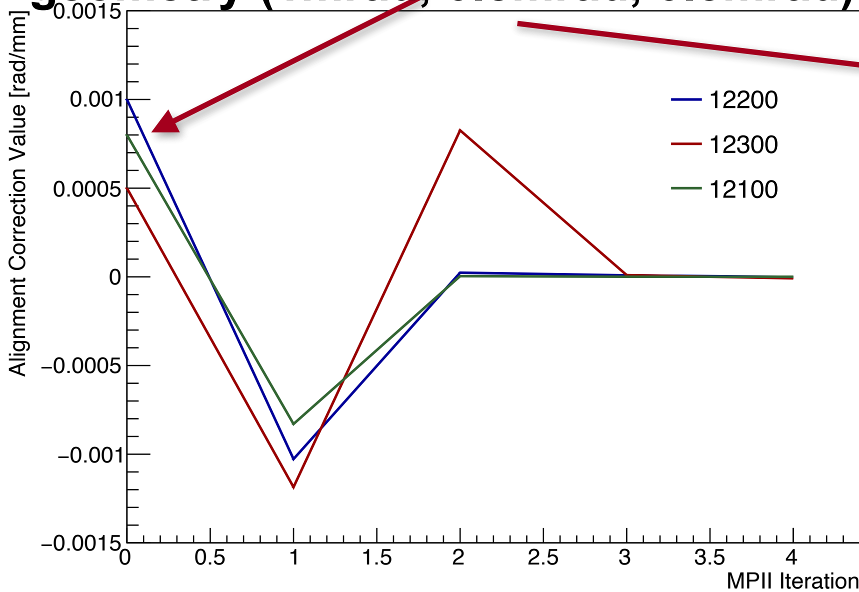
- Presented the available framework for performing SVT alignment in HPS
- The code is available and can be operated successfully (see Cam's talk) on sdf by other users. Instructions and documentation are available
- Extensive checks on :
  - Alignment derivatives, misalignment recovery from known distortions, constraints and monitoring have been done in the past years
- The GBL+MPII implements almost all necessary features:
  - Hierarchical constraints, BS and momentum constraint
  - Soft mode cuts
  - Java Native Access to C++ library together with java wrappers necessary to operate it
- A re-alignment of the top volume has been performed again:
  - Some of the movements are large but the residual, momentum and beamspot location analysis show a good performance wrt expected from data
  - Will run same procedure on Bottom volume and deploy a new detector in 1-2 weeks (compatible with other duties)

BACKUP

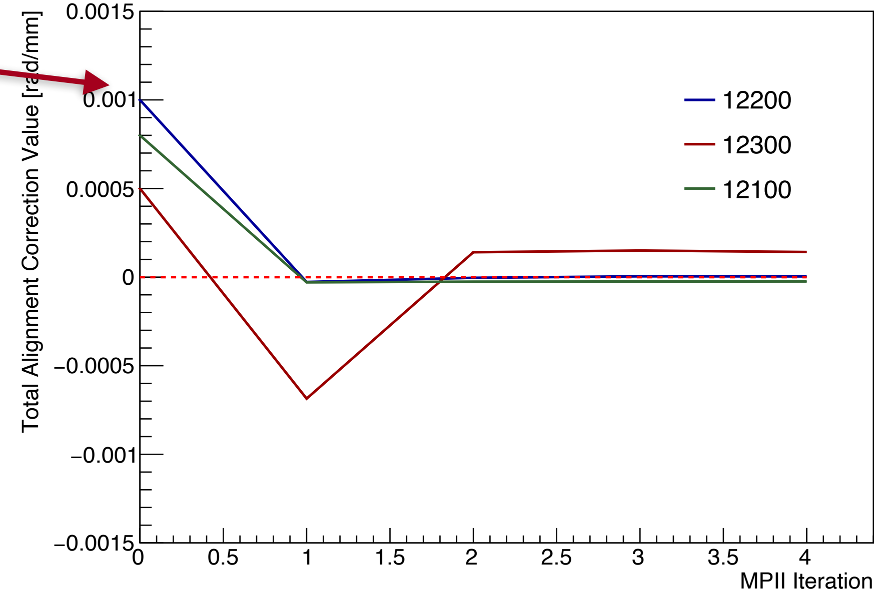
Global rotations MPII corrections convergence check

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)

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.

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

Putting things together: alignment tags produced for testing

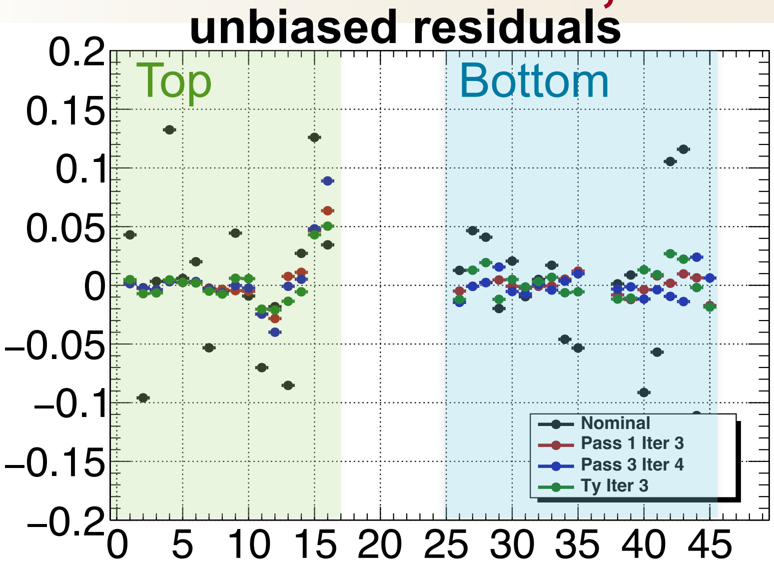
Alignment detectors that have been studied

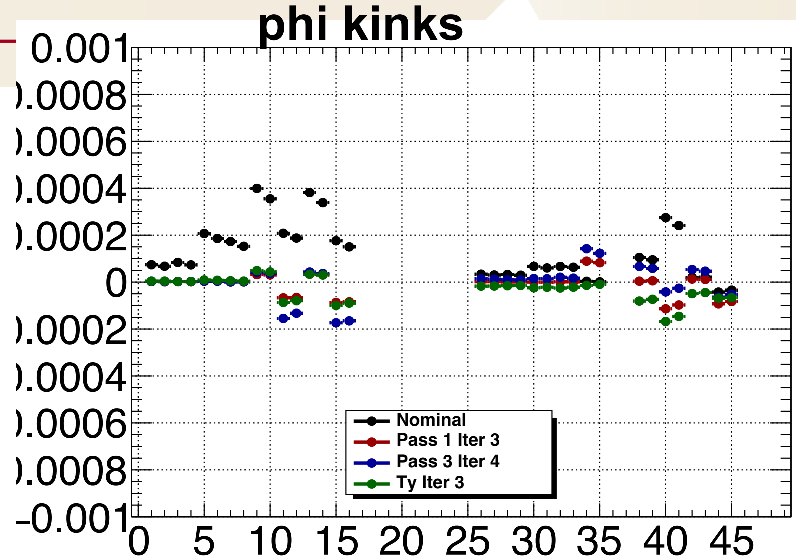
- Back in September I produced 3 aligned detectors with the aim of checking performance of the SVT calibration.
- The tags were made with the purpose of checking the various strategies and included global movements and external constraints:

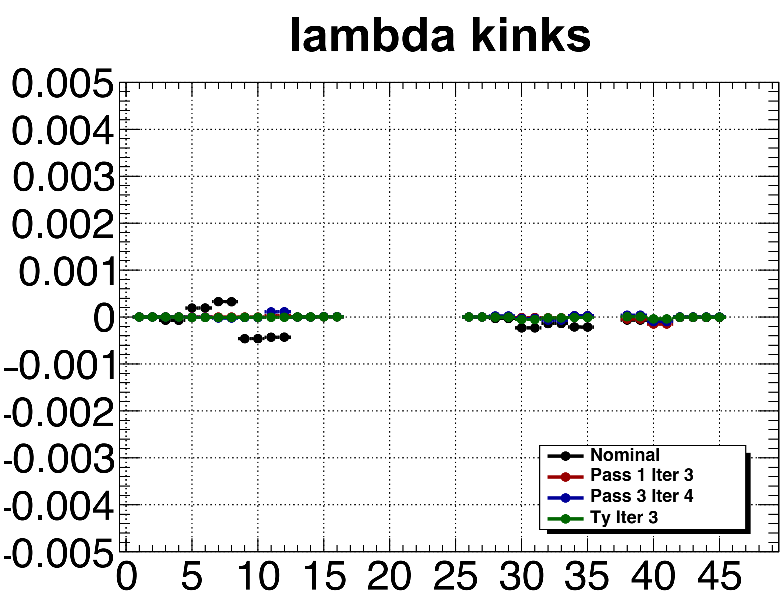
- **HPS_PASS1_iter3**: momentum constrain only, Back UChannel fixed
  - iter1: Tx, Ty, Rx, Ry of front UChannels
  - iter2: Tx, Ty, Rx, Ry of front UChannels
  - iter3: Tx, Ty front UChannels, Tu L1-L4 Modules
  with Tx/Ty constraints
- **HPS_PASS3_iter4**: momenutm constrain and beamspot constrain (0,0,-7.5)
  - 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

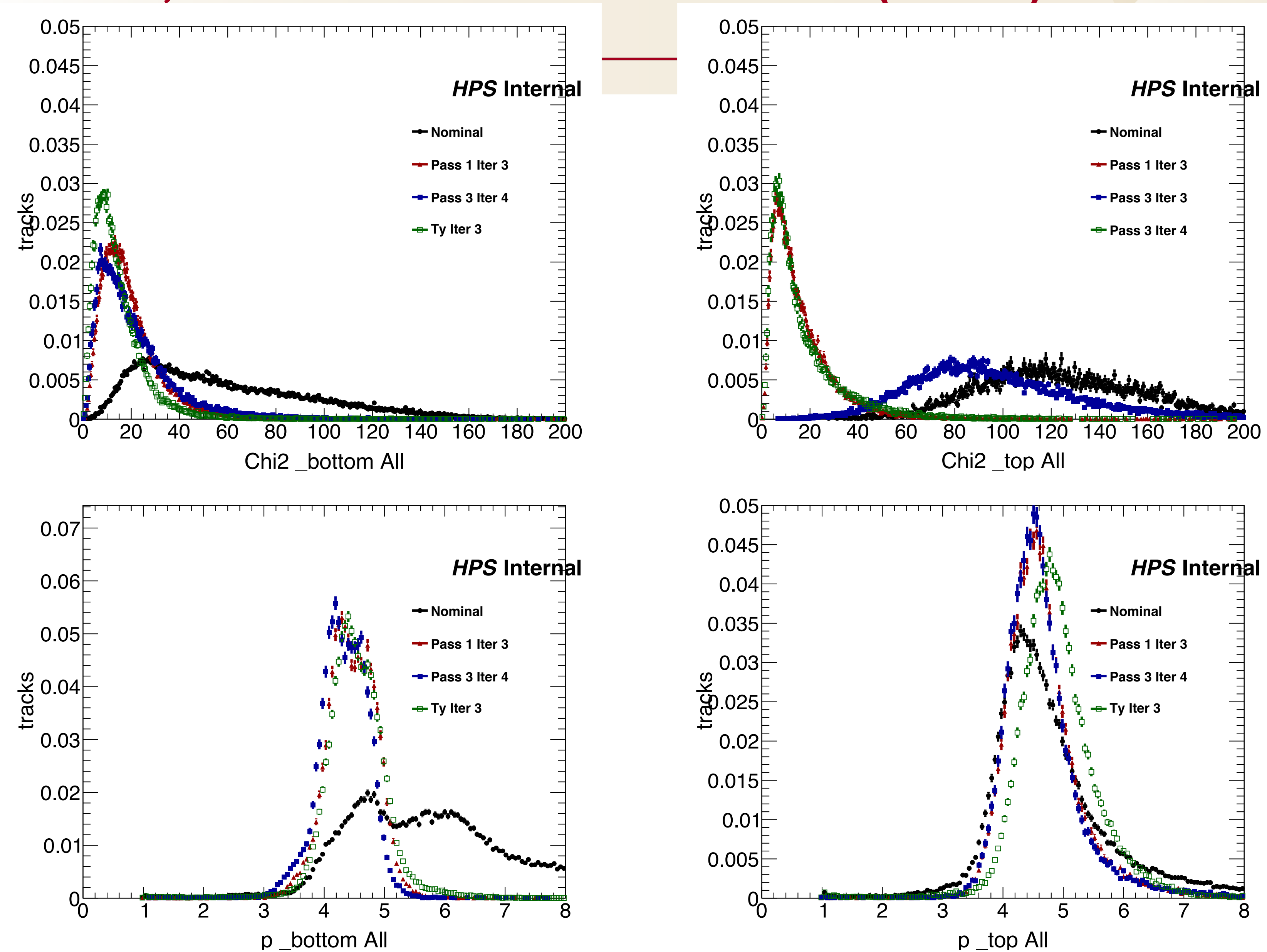






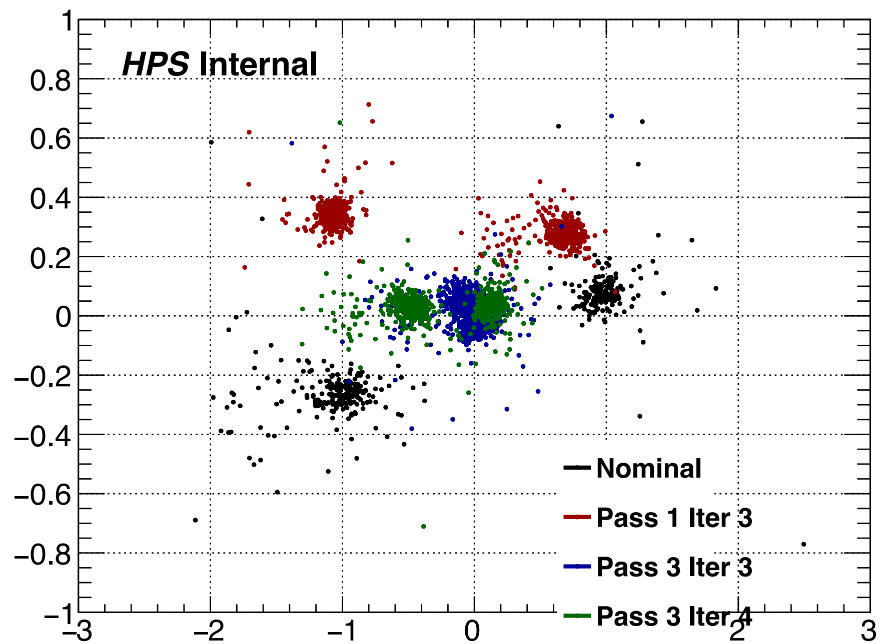
- X-axis is the ID of each sensor: first half represent the top volume, second half represents bottom volume
- Sensors/modules of the back of the detector are not aligned
- Aligning global structures first and then up to module level leads to similar results of aligning Tu up to sensor level.

Chi2, momentum - FEE DATASET (10103)

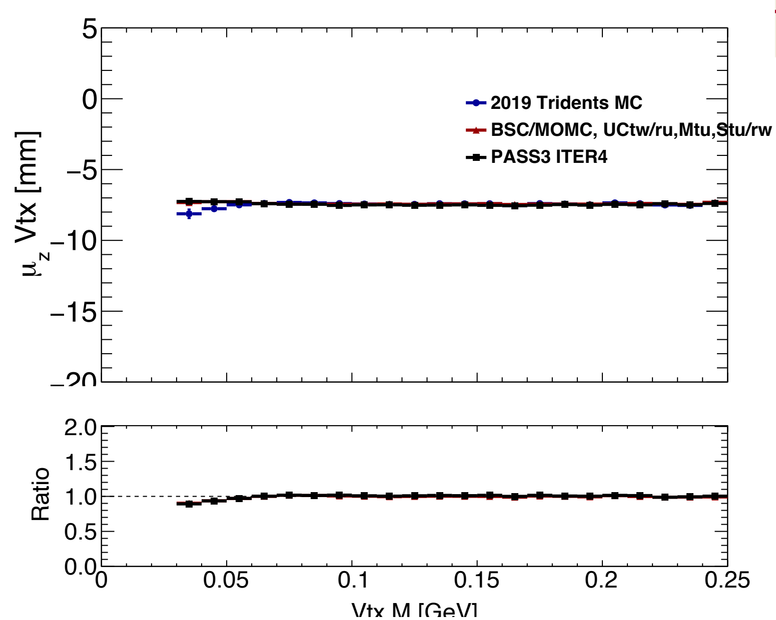


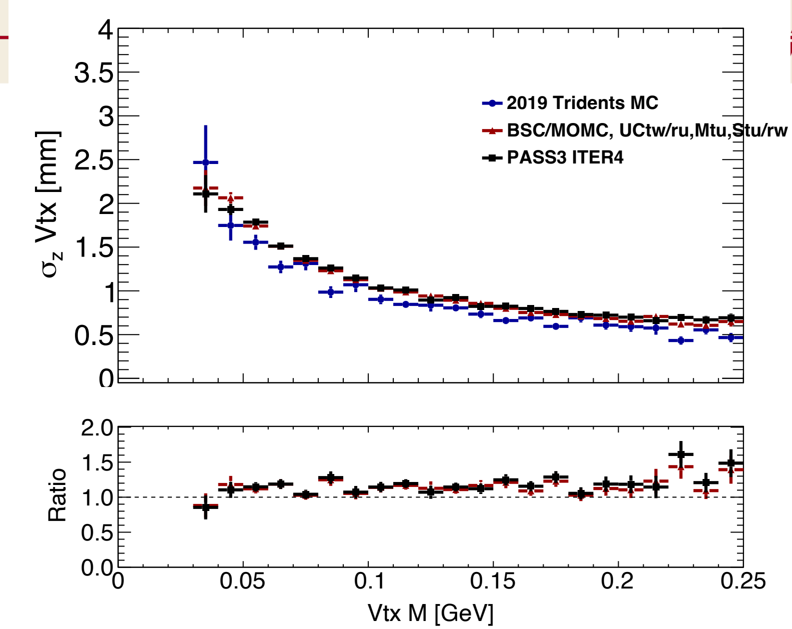
Multi VTX X-Y

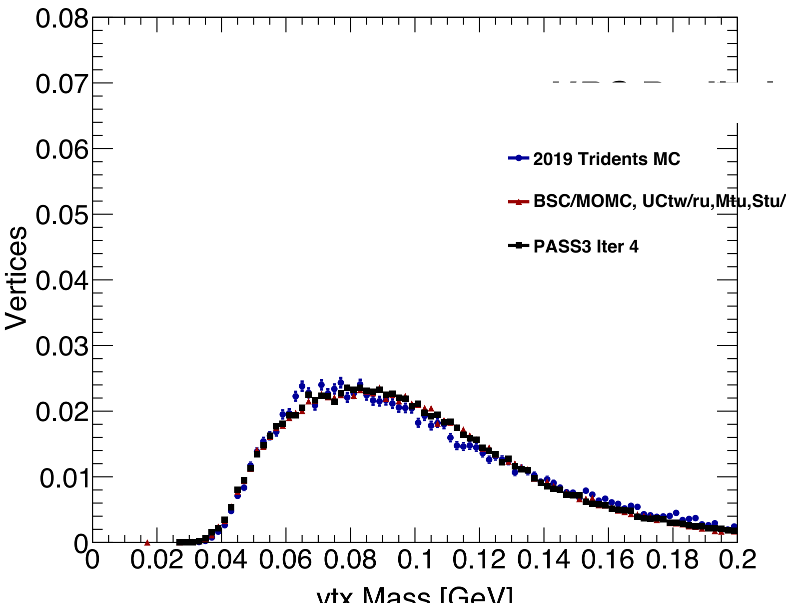
- Effect of beamspot constrain in pass3 shows that top and bottom can be forced to converge to a common point
- When aligning Ty, Tx module by module including back of the detector, a difference in X is noticed.
- When not applying beamspot constraint top and bottom have a large spread in X and around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

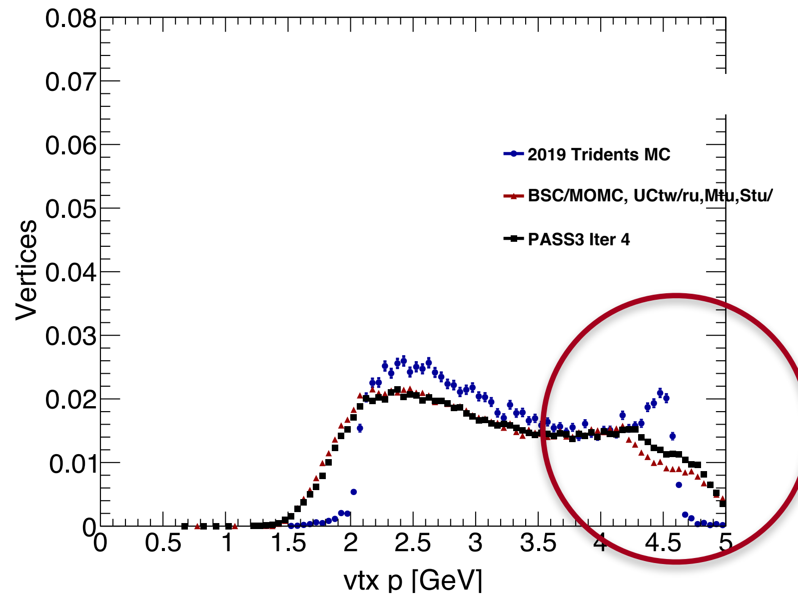


Checks on 10031 - V0 skims comparison with MC Tridents (no beam)



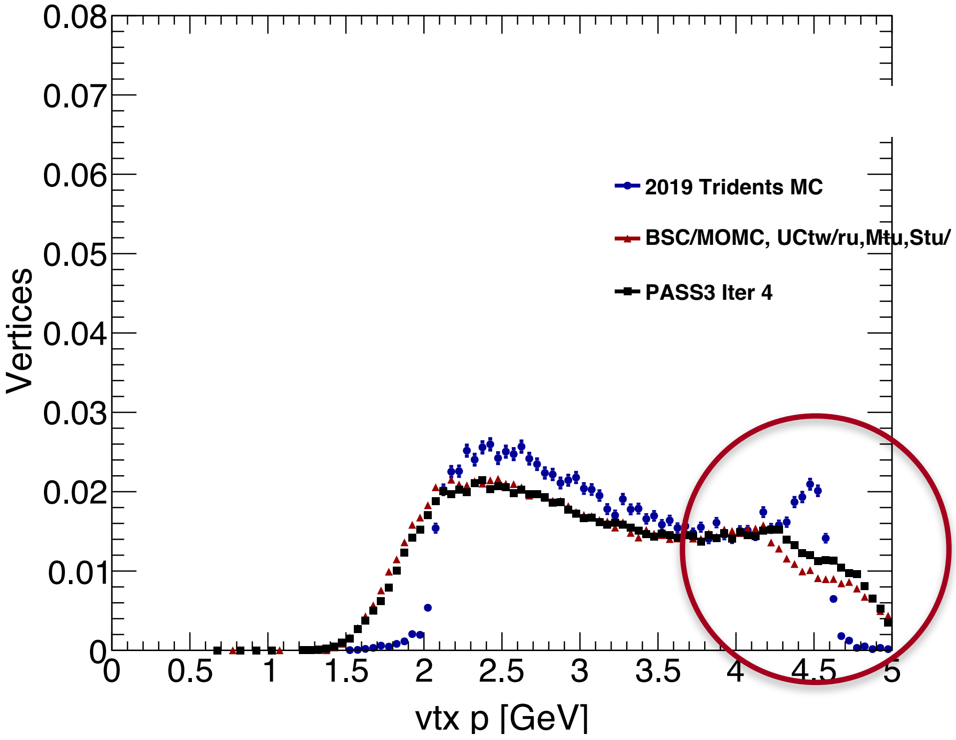




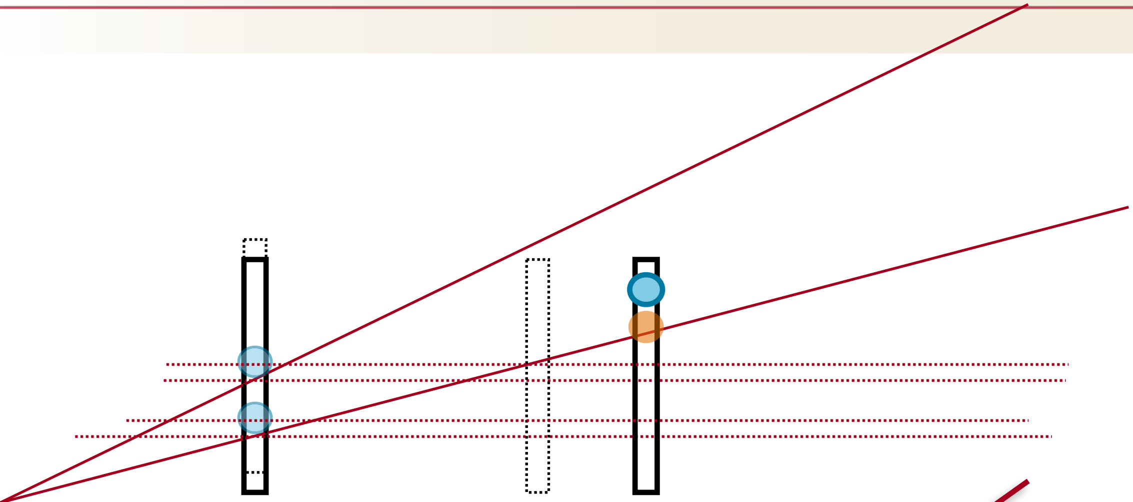


Momentum scale and resolution in FEE samples

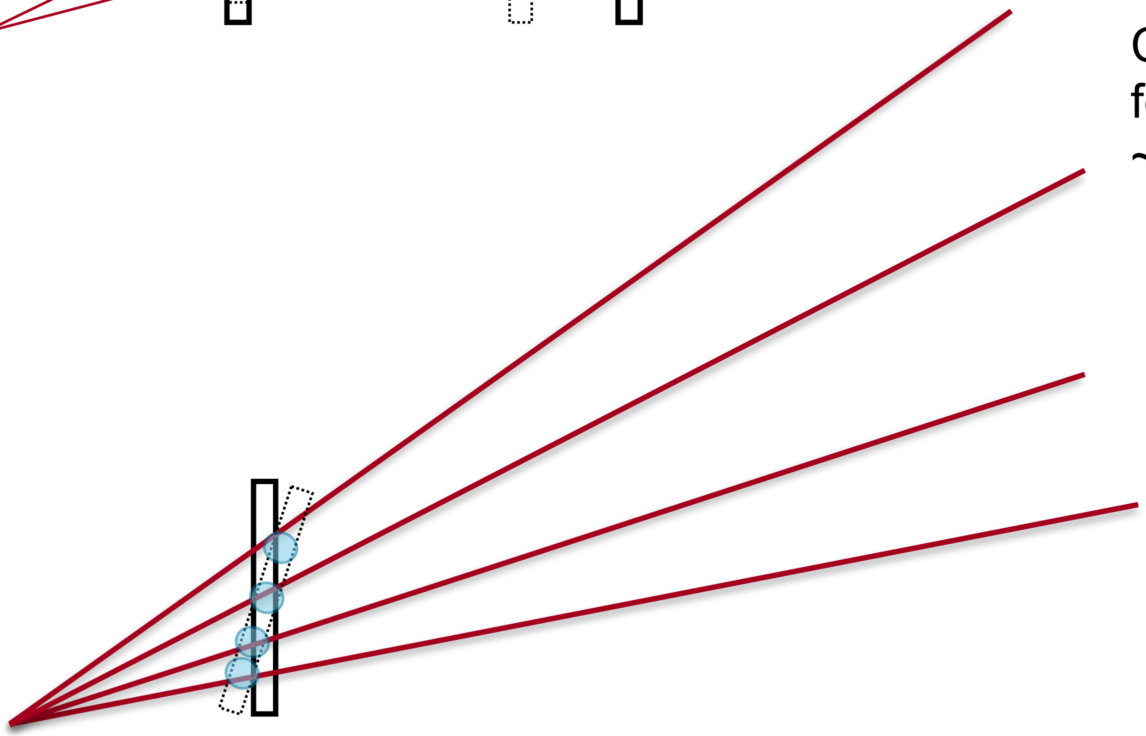
- The three alignment tags perform quite similarly in terms of track chi2, momentum scale and resolution and unbiased residuals.
- Decided to move forward with the tag (HPS_TY_IterX) which only corrects for movements along the most sensitive directions, nominally tu and rw for the sensors.
- Two additional 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



Unbiased residual studies: u-dependence



Cannot be pure Tu of the sensor



Could be pure Rv of the sensor.
for 40um residual, this means
~13mrad Rv, for 3mm lever arm

Current scenario of HPS Alignable structures - Just FYI

- 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

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

Alignable Sensor Axial L1

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

Alignable Sensor Stereo L1

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