Using Time Information in ACTS Vertexing Algorithms

Ideas & Concepts

Felix Russo

09/11/2023

Task

Set of all tracks T → Vertexing Algorithm → Set of all vertices V

$$t \in T \sim \hat{\mathbf{q}} = \left(d_0, z_0, \phi, \theta, \frac{q}{p}, t_0\right)^1$$

$$v \in V \sim (x, y, z, t)$$

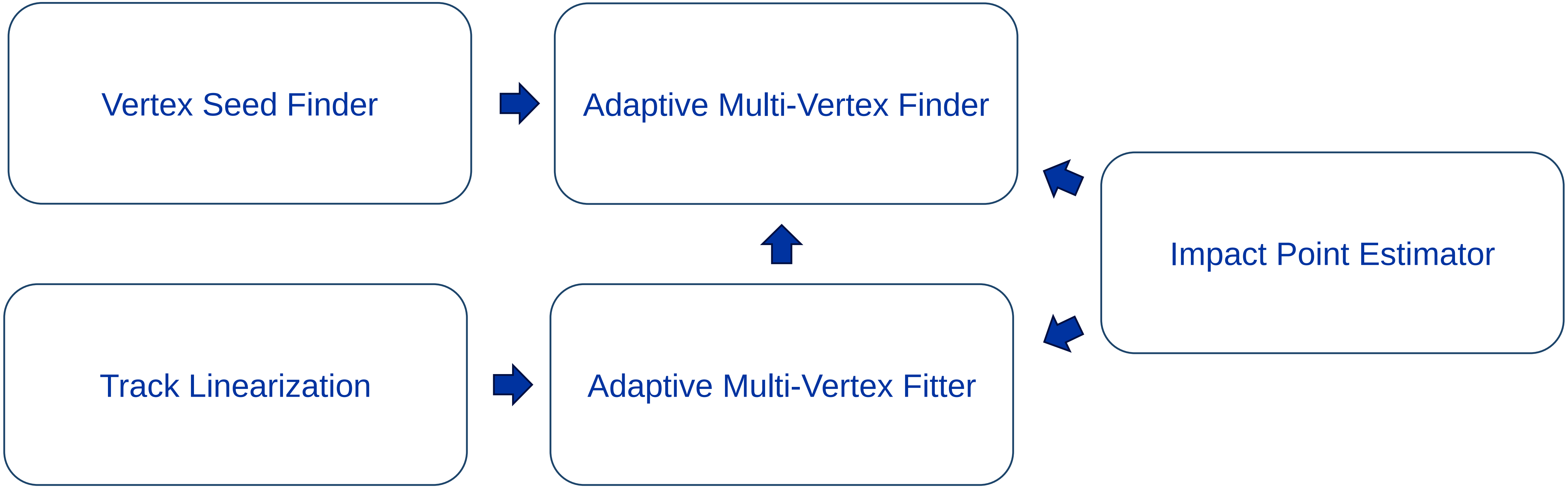
and

$$\mathbf{p} \; \forall \; t \in T_v \subset T$$

All quantities have associated covariance matrixes, e.g., $\mathbf{K}_{\hat{\mathbf{q}}}$ for the track parameters.

[1] with respect to some reference line (e.g., the z-axis).

Simplified Structure



Vertex Seed Finding

AdaptiveGridTrackDensity.*pp & AdaptiveGridDensityVertexFinder.*pp

- Goal: Find a first estimate of the vertex position from a set of tracks
- Tracks are modeled as a 2D Gaussian distribution in the d-z-plane[1]
- Density at $d = 0$ is calculated at discrete z-values for each track and added to a map

$$M\colon \text{bin} \rightarrow \text{densityValue}$$

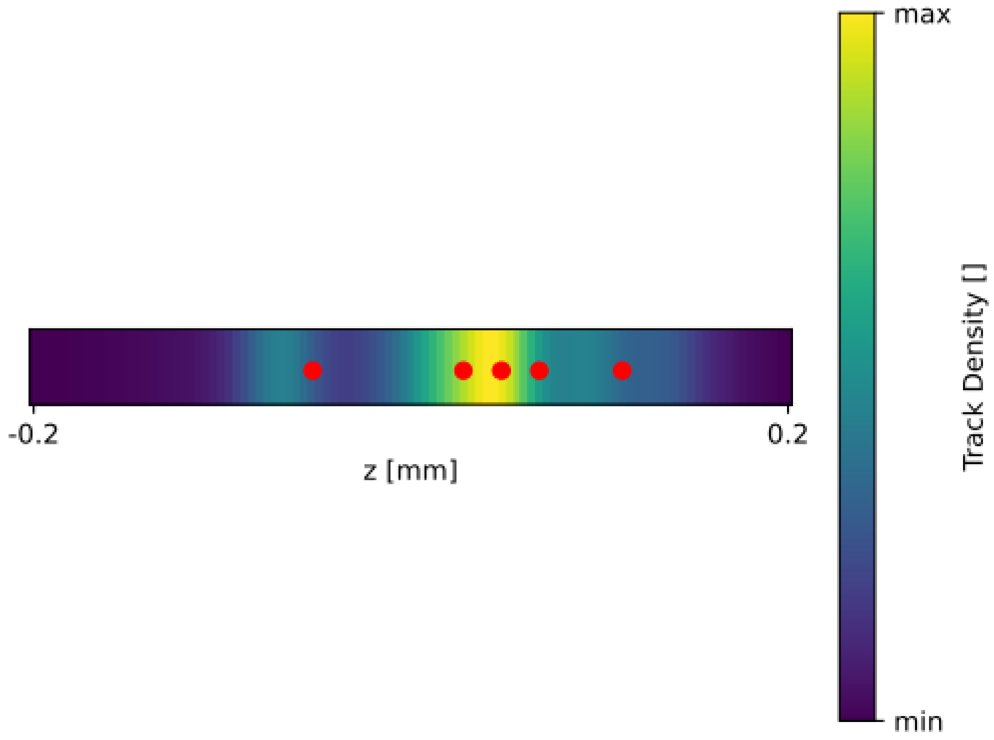
- Cache track densities → can be removed without recalculating their contribution
- We effectively have a 1D density grid; its maxima are the vertex seeds $(0,\ 0,\ z_{\text{max}},\ 0)$

[1] Schlag, Bastian: Advanced Algorithms and Software for Primary Vertex Reconstruction and Search for Flavor-Violating Supersymmetry with the ATLAS Experiment

Vertex Seed Finding

AdaptiveGridTrackDensity.*pp & AdaptiveGridDensityVertexFinder.*pp

- Data:
  - Five vertices (red dots) at random positions
  - 4 Muons per vertex
  - Tracks reconstructed using default seeding and Combinatorial Kalman Filter



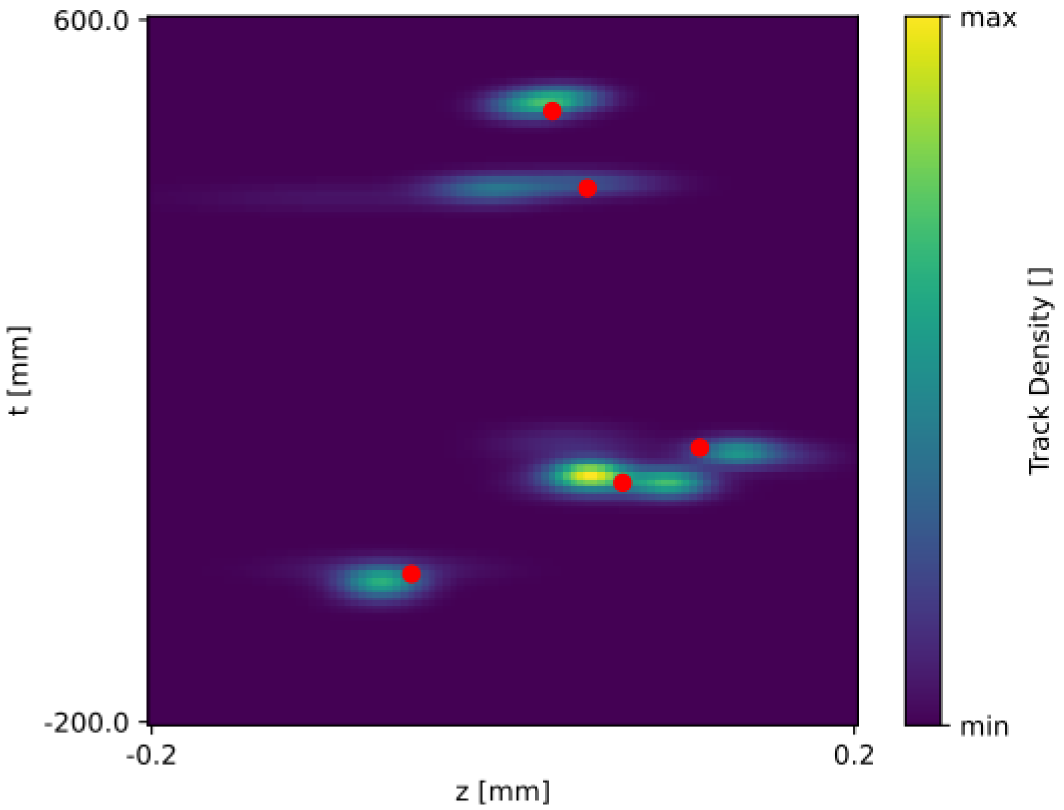
- In high-luminosity environments, the 1D description is not sufficient to resolve all vertices[1]

[1] See also: https://cds.cern.ch/record/2870326/files/ATL-PHYS-PUB-2023-023.pdf

Vertex Seed Finding

AdaptiveGridTrackDensity.*pp & AdaptiveGridDensityVertexFinder.*pp

- Time can be included using a 3D distribution in the d-z-t-plane
- As before, we evaluate at $d = 0$ to obtain a 2D grid
- Its maxima are the vertex seeds $(0,\ 0,\ z_{\mathrm{max}},\ t_{\mathrm{max}})$
- Vertices can be resolved now!



Impact Point Estimation

ImpactPointEstimator.*pp

- Goal: Estimate compatibility between vertices and tracks

- Find the 3D point of closest approach (PCA) between the track and the vertex
  - Use the Newton method to find a minimum of the distance

- Propagate to the PCA, i.e., a plane reference surface with its origin at the vertex
  - x-axis in direction of the PCA
  - z-axis in direction of the momentum
  - y-axis follows from orthogonality

Impact Point Estimation

ImpactPointEstimator.*pp

- Calculate $\chi^2_{\mathrm{vtx,trk}} = \bar{\mathbf{r}}^{\mathbf{T}}\, \mathbf{K}^{-1}_{\mathbf{r}_{\mathbf{trk}}}\, \bar{\mathbf{r}}$, where:

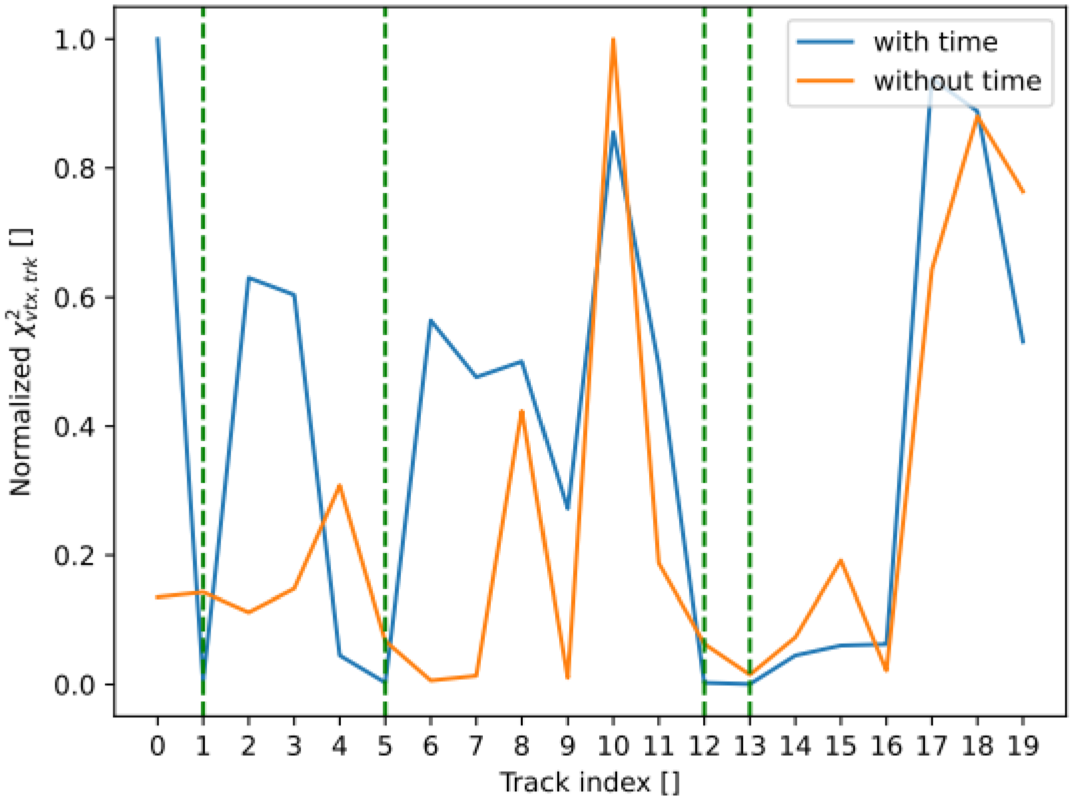
  - $\bar{\mathbf{r}} = \mathbf{r}_{\mathbf{trk}} - \mathbf{r}_{\mathbf{vtx}}$
  - $\mathbf{r}_{\mathbf{trk}} = \begin{pmatrix} x_{\mathrm{trk}} \\ y_{\mathrm{trk}} \\ \mathrm{t}_{\mathrm{trk}} \end{pmatrix}$
  - $\mathbf{r}_{\mathbf{vtx}} = \begin{pmatrix} 0 \\ 0 \\ \mathrm{t}_{\mathrm{vtx}} \end{pmatrix}$

- Note that $z_{\mathrm{trk}} = z_{\mathrm{vtx}} = 0$ because both are on the reference surface

Impact Point Estimation

ImpactPointEstimator.*pp

- Same data as before
- Choose a random vertex
- Calculate $\chi^2_{\mathrm{vtx,trk}}$ for all tracks with and without time
- The green lines indicate the tracks that really originate at the vertex



Adaptive Multi-Vertex Fitter

AdaptiveMultiVertexFitter.*pp & KalmanVertexUpdater.*pp,

- Based on a Kalman Filtering approach + annealing
- Minimizes $\chi^2_{total} = \sum_{\text{vertices}} \sum_{\text{tracks}} \left(w\left(T, \chi^2_{\text{vtx,trk}}\right) \; \bar{\mathbf{q}}^{\boldsymbol{T}} \, \mathbf{K}_{\hat{\mathbf{q}}}^{-1} \, \bar{\mathbf{q}} \right)$
- $\bar{\mathbf{q}} = \mathbf{q}_{\mathbf{model}} - \hat{\mathbf{q}}$
- $\hat{\mathbf{q}}$ and $\mathbf{K}_{\hat{\mathbf{q}}}^{-1}$ come from the tracking; $\chi^2_{\text{vtx,trk}}$ comes from the impact point estimation
- $\mathbf{q}_{\mathbf{model}} \left(\mathbf{r}_{\mathbf{V}}, \mathbf{p}_{\mathbf{V}} \right) \approx \mathbf{q}\left(\mathbf{r}_{\mathbf{PCA}}, \mathbf{p}_{\mathbf{PCA}} \right) + \mathbf{A} \left(\mathbf{r}_{\mathbf{V}} - \mathbf{r}_{\mathbf{PCA}} \right) + \mathbf{B} \left(\mathbf{p}_{\mathbf{V}} - \mathbf{p}_{\mathbf{PCA}} \right)$
- $\mathbf{A} = \frac{\partial \mathbf{q}}{\partial \mathbf{r}_{\mathbf{W}}} |_{\mathrm{W=PCA}}$ and $\mathbf{B} = \frac{\partial \mathbf{q}}{\partial \mathbf{p}_{\mathbf{W}}} |_{\mathrm{W=PCA}}$ are the Jacobians

Track Linearization

HelicalTrackLinearizer.*pp

- Analytically:

$$\frac{d\mathbf{q}}{dx_w}\Big|_{\mathrm{W=PCA}} \downarrow \qquad \frac{d\mathbf{q}}{dy_w}\Big|_{\mathrm{W=PCA}} \downarrow \qquad \frac{d\mathbf{q}}{dz_w}\Big|_{\mathrm{W=PCA}} \downarrow \qquad \frac{d\mathbf{q}}{dt_w}\Big|_{\mathrm{W=PCA}} \downarrow$$

$$\mathbf{A} = \begin{pmatrix} -\sin(\phi) & \cos(\phi) & 0 & 0 \\ -\frac{|\rho|}{S}\cot(\theta)\cos(\phi) & -\frac{|\rho|}{S}\cot(\theta)\sin(\phi) & 1 & 0 \\ -\frac{\mathrm{sgn}(\rho)}{S}\cos(\phi) & -\frac{\mathrm{sgn}(\rho)}{S}\cos(\phi) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{|\rho|}{c\beta_T S}\cos(\phi) & -\frac{|\rho|}{c\beta_T S}\sin(\phi) & 0 & 1 \end{pmatrix}$$

- $\rho\ldots$ signed helix radius
- $S\ldots\ x-y$ - distance between vertex and helix center
- $c\beta_T\ldots$ speed in the $x-y$ - plane

Track Linearization

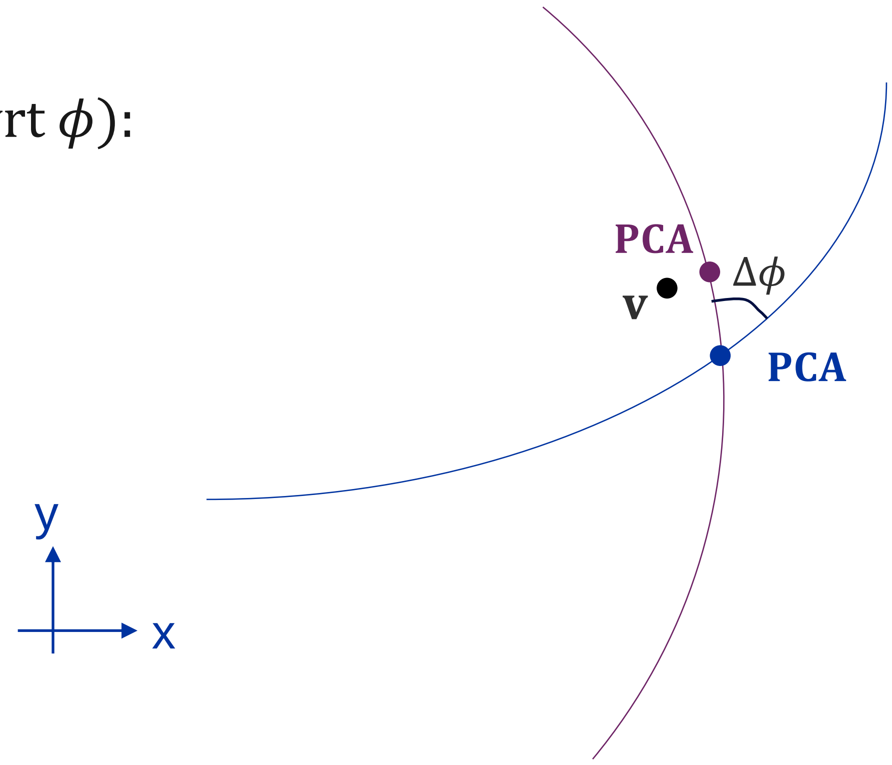
HelicalTrackLinearizer.*pp

- Assumes a constant magnetic field
- For the other Jacobian $\mathbf{B}$, also the last row had to be added (not shown here)
- Derived together with P. Butti – thanks a lot!
- Results checked numerically

Track Linearization

NumericalTrackLinearizer.*pp

- Numerically (here for derivatives wrt ϕ):



- $\frac{\partial \mathbf{q}}{\partial\, \phi}\Big|_{\mathrm{PCA}} \approx \frac{\hat{\mathbf{q}}(\phi+\Delta\phi) - \hat{\mathbf{q}}(\phi)}{\Delta\phi}$
- Computationally expensive, but works for a non-constant magnetic field

  → Potentially useful for secondary vertexing

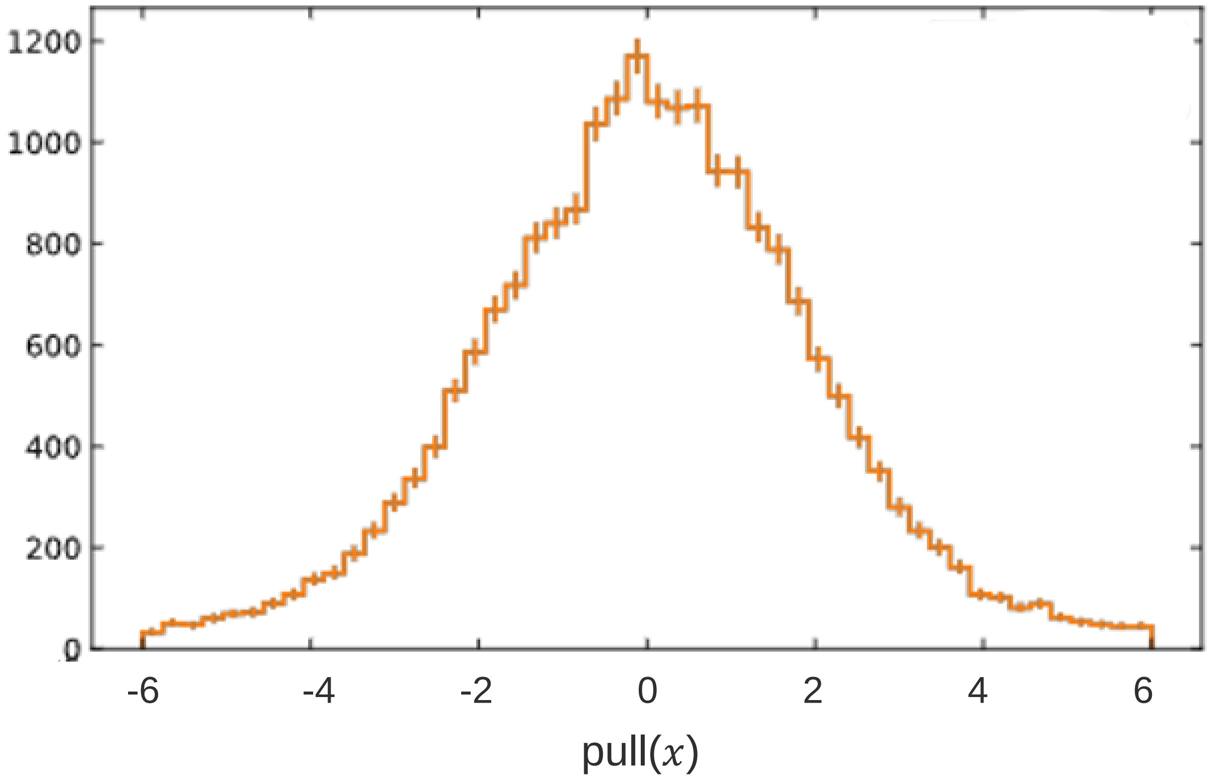
Results from Billoir Vertex Fit

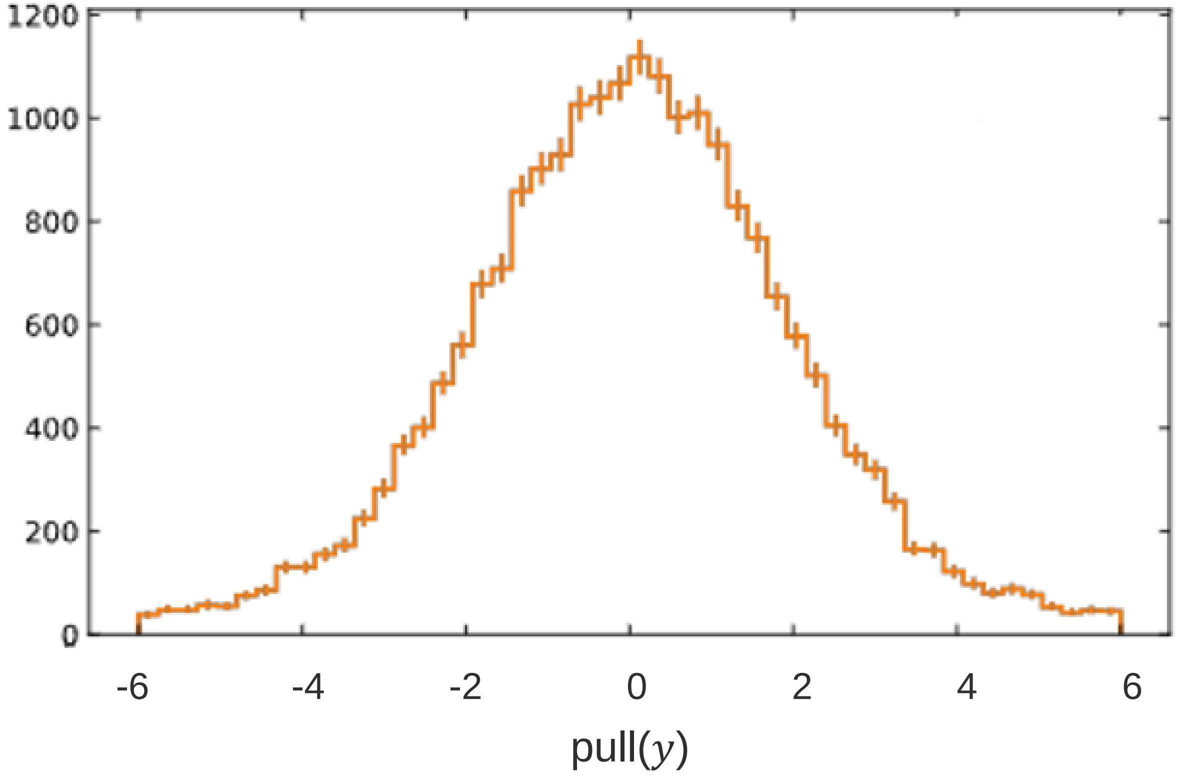
FullBilloirVertexFitter.*pp, physmon_ckf_tracking.py

- Fitting the vertex time using the Billoir method
  - Mathematically equivalent to the Kalman vertex fit
  - Kalman vertex fitter is not ready yet
- Using 500 4-Muon events
- Pseudo pile-up of 50
  - Particle gun with vertex smearing
- Default seeding + Combinatorial Kalman Filter

Results from Billoir Vertex Fit

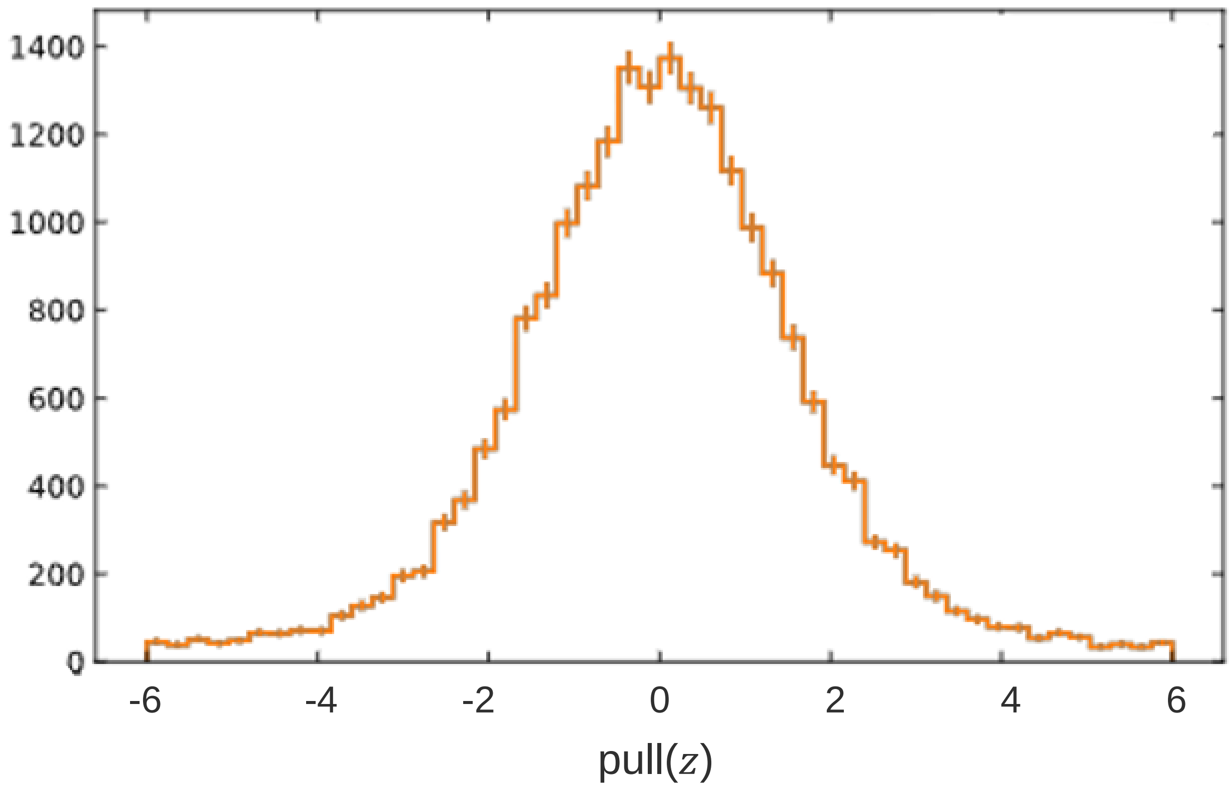
FullBilloirVertexFitter.*pp, physmon_ckf_tracking.py

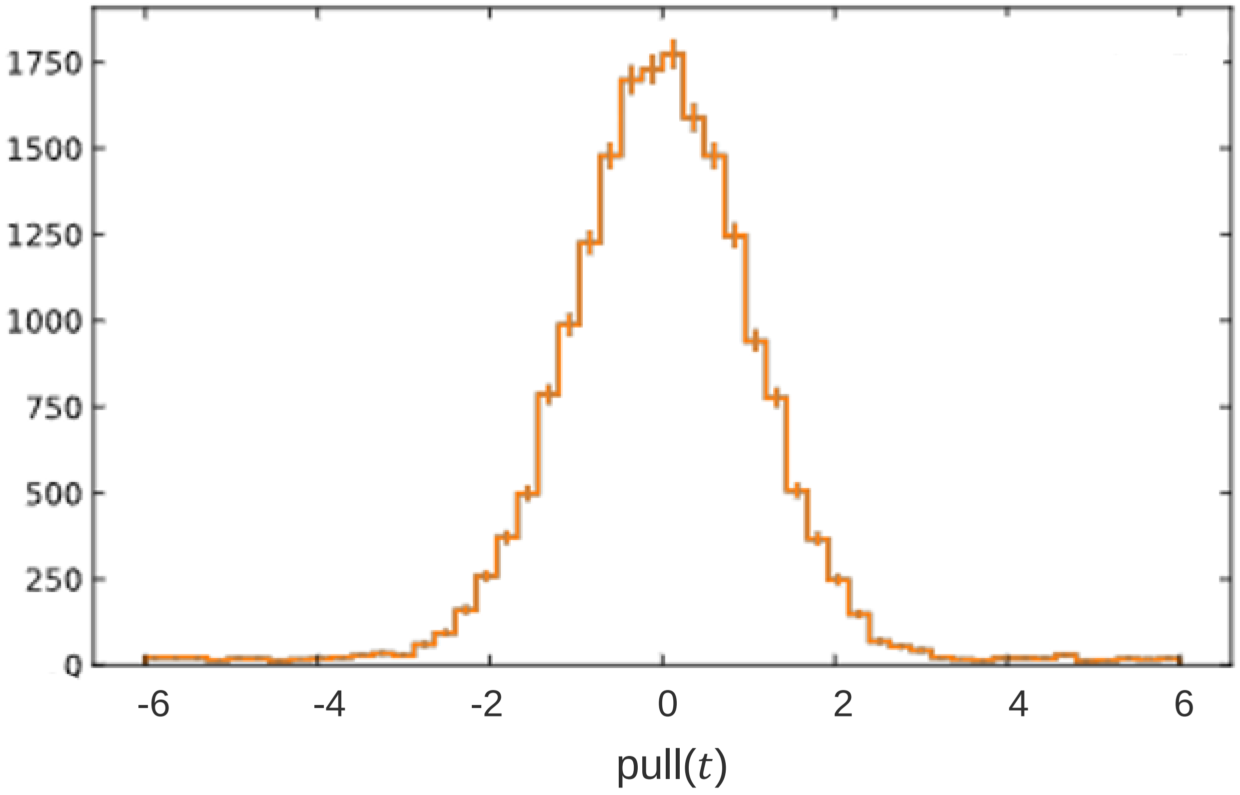




Results from Billoir Vertex Fit

FullBilloirVertexFitter.*pp, physmon_ckf_tracking.py





Conclusion

- Time is gradually introduced into the ACTS vertexing suite
- Can be included elegantly by adding a new dimension
- Partial time measurements should not be a problem
  - If a tracks does not have a time measurement, we inflate the corresponding covariance
- Time vertexing is promising enhanced resolution in high-luminosity environments
- Big thanks to: P. Butti, P. Gessinger-Befurt, A. Salzburger, B. Schlag, A. Stefl

Backup slides

Vertex Seed Finding

AdaptiveGridTrackDensity.*pp & AdaptiveGridDensityVertexFinder.*pp

- Mathematical model

$$P(d, z) \propto \frac{1}{\det\left(\mathbf{K}_{\widehat{\mathbf{q_{IP}}}}\right)} \exp\left(\overline{\mathbf{q_{IP}}}^{T} \, \mathbf{K}^{-1}_{\widehat{\mathbf{q_{IP}}}} \, \overline{\mathbf{q_{IP}}}\right)$$

$$\mathbf{q_{IP}} = \mathbf{q_{IP}} - \widehat{\mathbf{q_{IP}}} \qquad \mathbf{q_{IP}} = \begin{pmatrix} d \\ z \end{pmatrix} \qquad \widehat{\mathbf{q_{IP}}} = \begin{pmatrix} d_0 \\ z_0 \end{pmatrix}$$

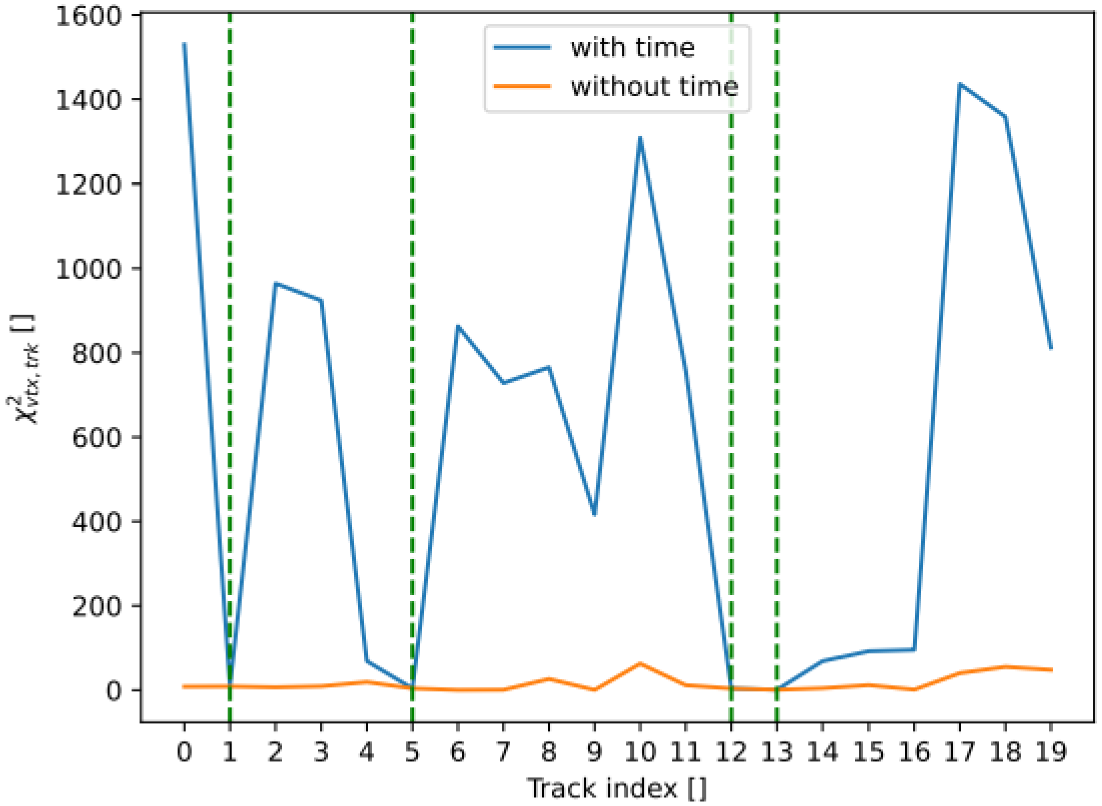
- Using time:

$$\mathbf{q_{IP}} = \begin{pmatrix} d \\ z \\ t \end{pmatrix} \qquad \widehat{\mathbf{q_{IP}}} = \begin{pmatrix} d_0 \\ z_0 \\ t_0 \end{pmatrix}$$

Impact Point Estimation

ImpactPointEstimator.*pp

- Unnormalized $\chi^2_{\mathrm{vtx,trk}}$

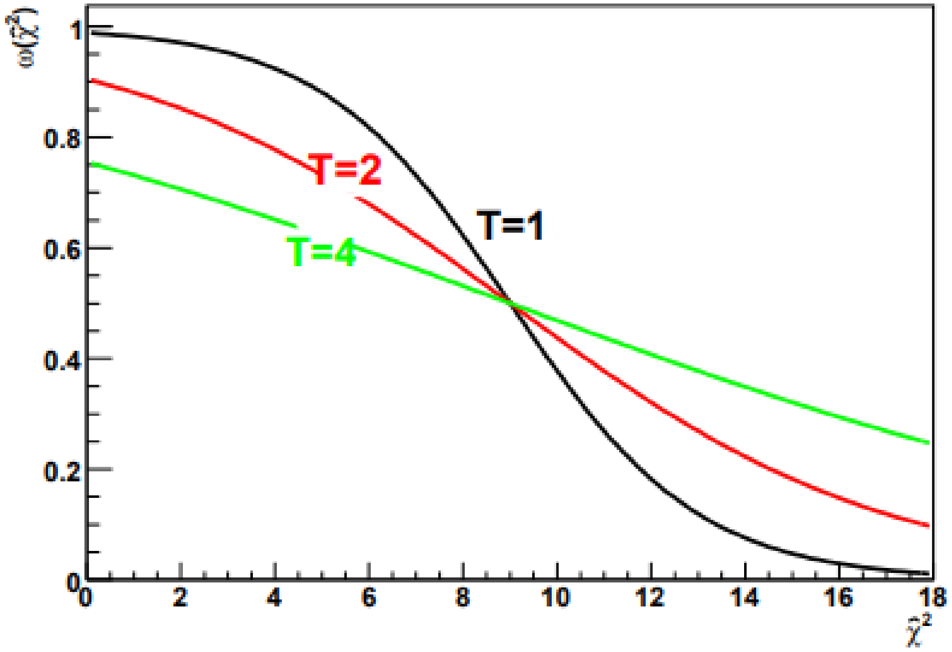


Adaptive Multi-Vertex Fitter

AdaptiveMultiVertexFitter.*pp & KalmanVertexUpdater.*pp,

- Based on a Kalman Filtering approach + annealing
- Adding a new measurement ~ adding a track to the fit
- Temperature is decreased → outliers are weighted down gradually
- $\chi^2 = \chi^2_{\mathrm{vtx,trk}}$ from impact point estimation
- A track can be associated to multiple vertices

  - Vertices are competing for tracks



[1] Piacquadio, Giacinto: Identification of b-jets and investigation of the discovery potential of a Higgs boson in the W H -> l nu b bbar channel with the ATLAS experiment