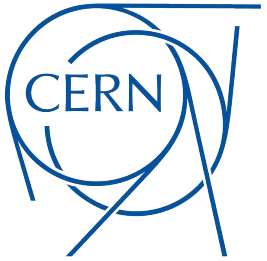
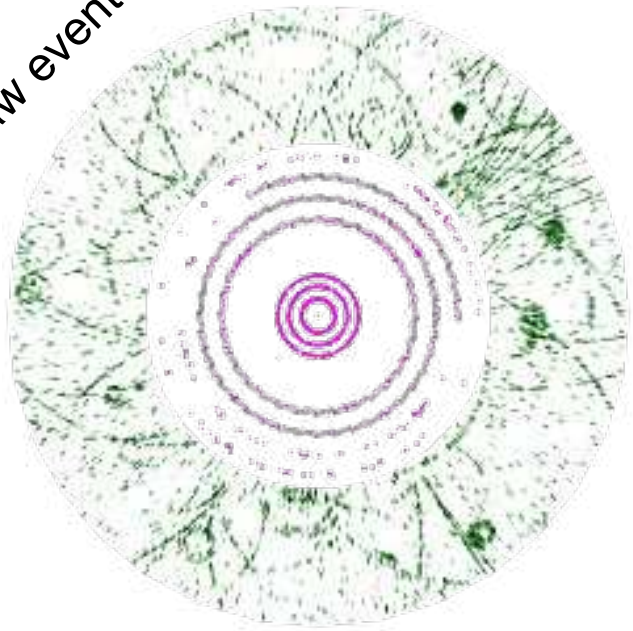
Kalman filter for gas detectors

Fabian Klimpel
CERN, TU Munich
fklimpel@cern.ch
25.02.2021

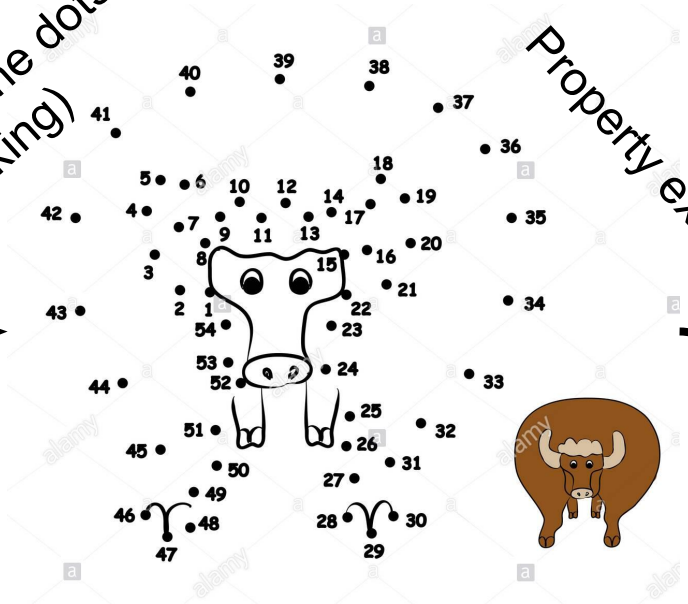


Tracking in a nutshell

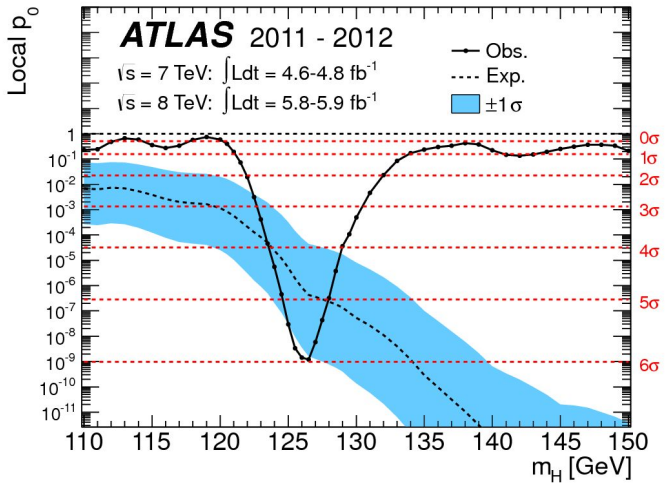
Raw event



Connecting the dots (Tracking)



Property extraction

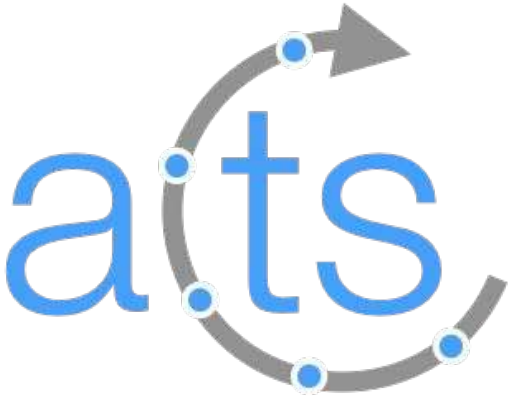


Tracking software: Acts

A project providing all required tracking components is Acts
(**A** **C**ommon **T**racking **S**oftware)

- Detector independent tracking software
- Development ongoing since ~6 years

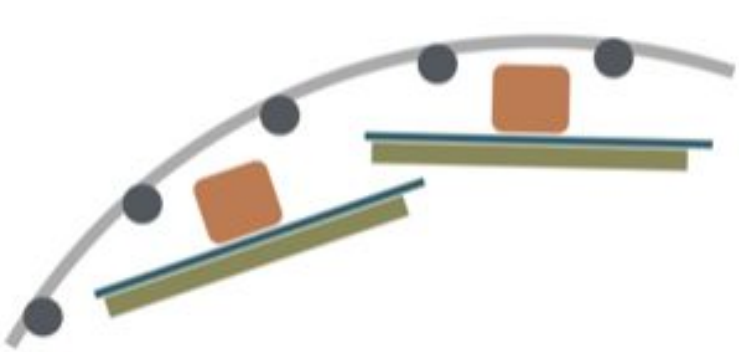
- Guidelines:
  - Minimal external dependencies
  - Optimised hardware usage
  - Workload at compile-time
  - Provide long time maintainability



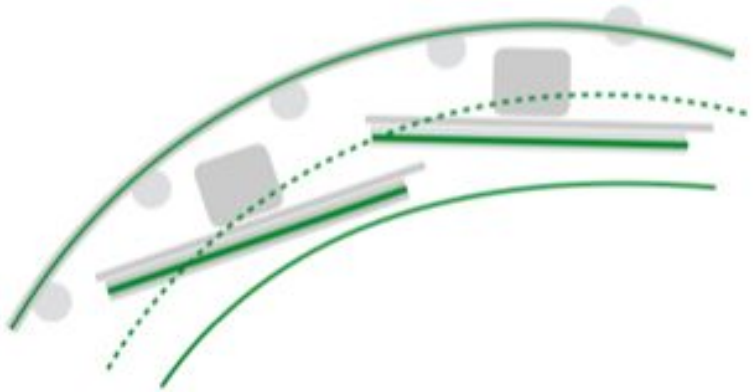
- Based on rewritten tracking algorithms
  - Modular design
  - Allows comparing, testing and improving the code
- Reasonable R&D platform

Detector geometry

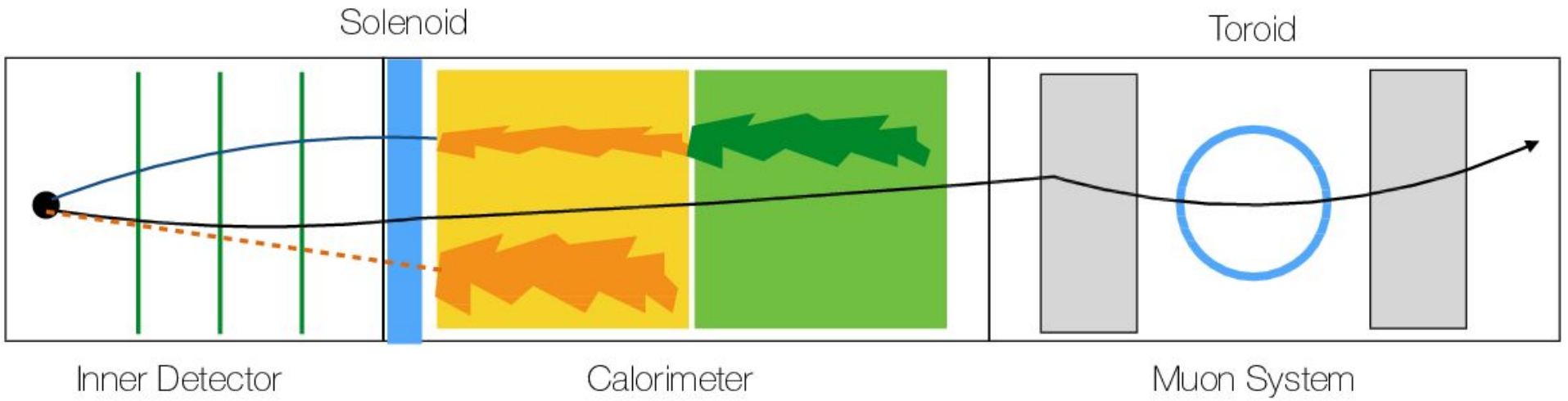
- Detector approximated as set of **volumes & surfaces**
- **Material is mapped** from Geant4 onto the surfaces/volumes
- Surface either active (= detector module) or passive (= pure material)



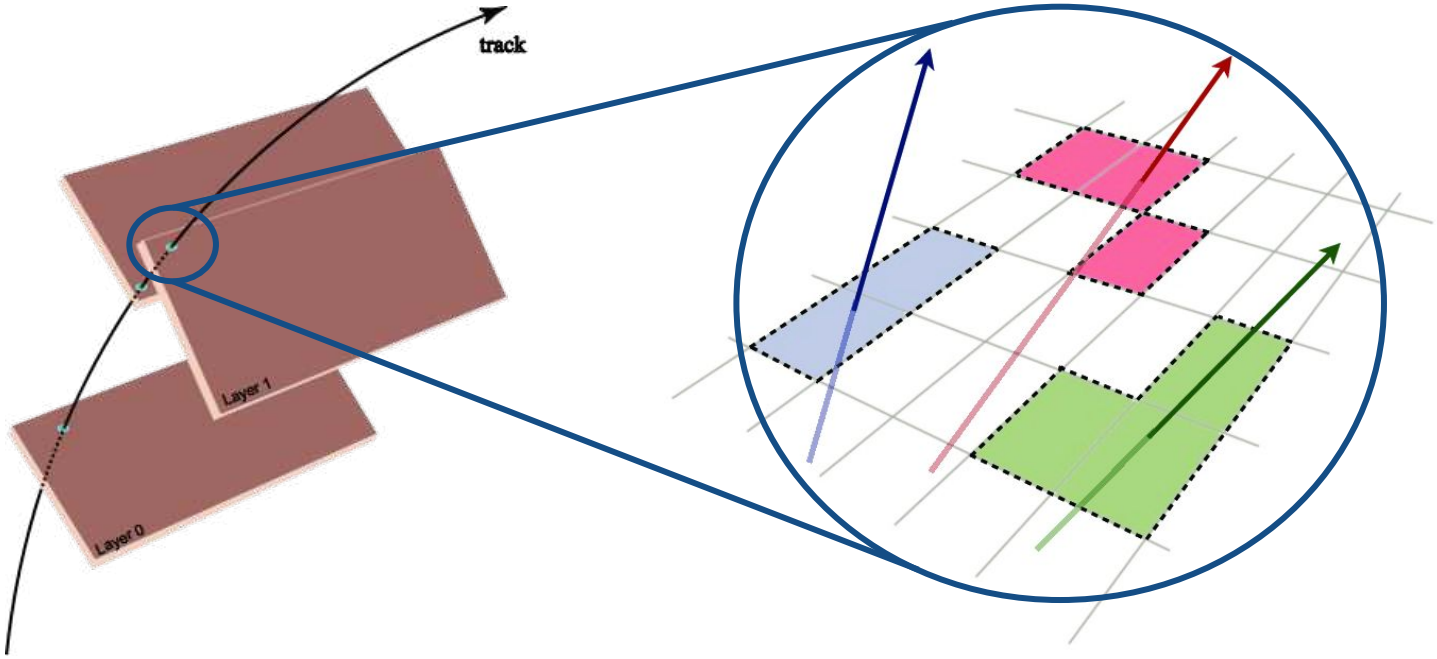
Detailed description



Approximated by surfaces



Measurements



- Particles traverse active surfaces
- Active surfaces measure particle properties
- (Subset of) Parameters $(x, y, \phi, \theta, q/p)$ measured
- Parametrized as (multivariate) normal distribution
- Associated with the surface

Goal:
Utilise data to learn about the particle

Kalman filter

Kalman filter = progressive parameter update

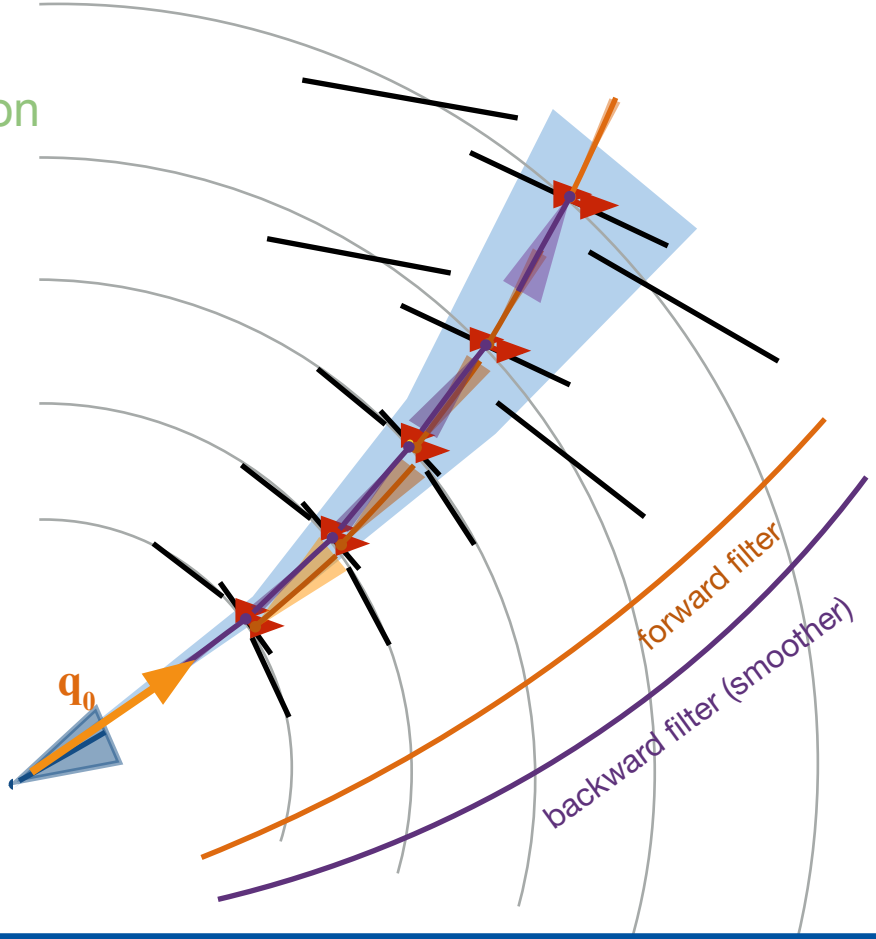
- Initially developed by R. Kalman to track missiles
- **Bayes filter** for normal distributed data and conjugate prior
- For HEP pioneered by P. Billoir and R. Frühwirth

see here:
Kalman
Billoir
Früwirth

Transport of track parameters allow prediction
Measurements used to update prediction
→ Filtered state

Filtered state allows next prediction etc.

At outermost surface:
Filtered state carries all data
→ Also applied onto inner surfaces
→ Predicting parameters at vertex



Track extrapolation

- Step-wise extrapolation of global mean

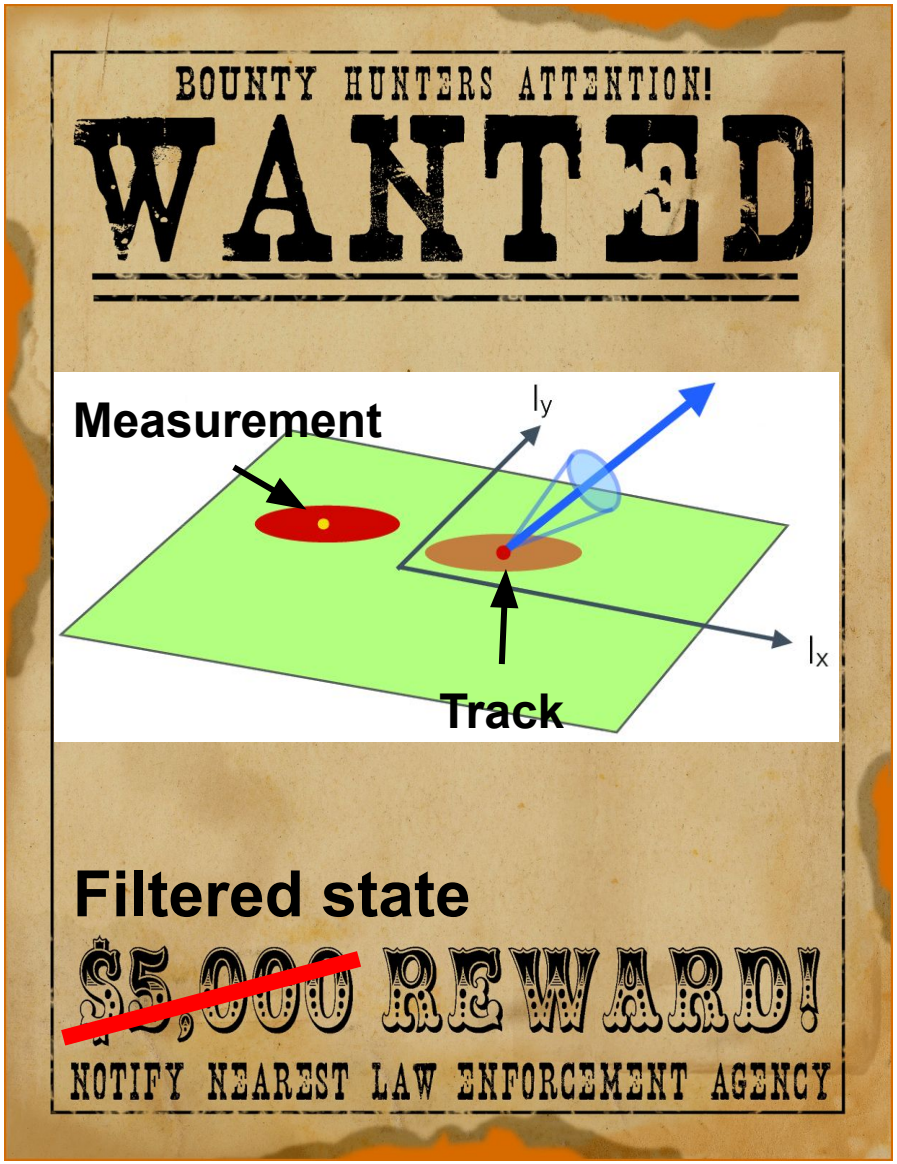
$$\frac{d^2\vec{r}}{ds^2} = \frac{q}{p}\left(\frac{d\vec{r}}{ds} \times \vec{B}\left(\vec{r}\right)\right)$$

- Free (global) parametrisation:

$$\left(x, y, z, T^x, T^y, T^z, q/p\right)$$

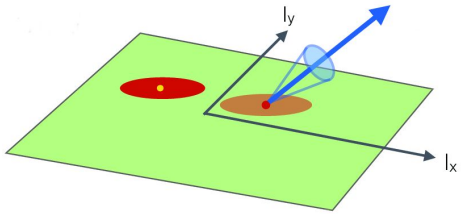
- Coordinate transformation at surface
- Covariance transport:

$$\Sigma_n^{local} = \underbrace{J \cdot \Sigma_0^{local} \cdot J^T}_{\underbrace{J_{g2l} \cdot J_{transport} \cdot J_{l2g}}_{\prod_{i=1}^{n} J_{transport,i}}}$$

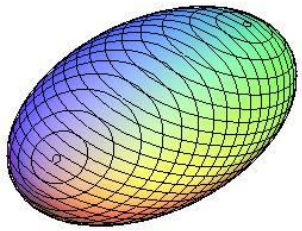


TPC/DC measurements

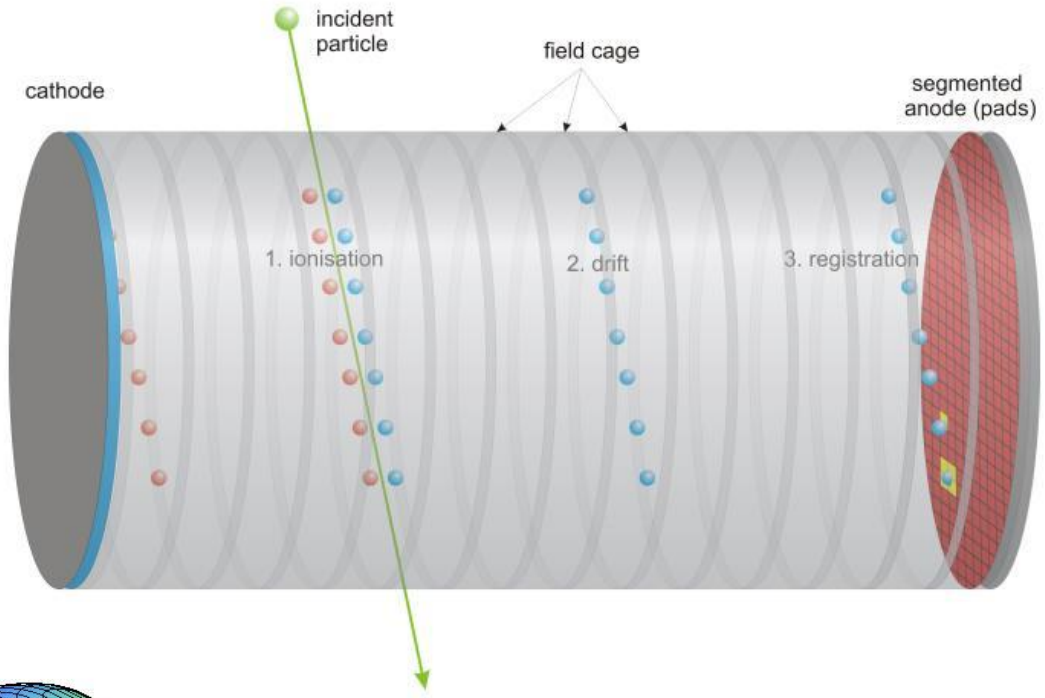
- Gas detectors measure particle not at origin of measurement
  - Displacement
- 3D measurement from pad data
  - Back projection to origin
  - **No surface** constraint
  - Measurement in **volume**
- Underlying problem becomes:



Planar measurement



3D measurement



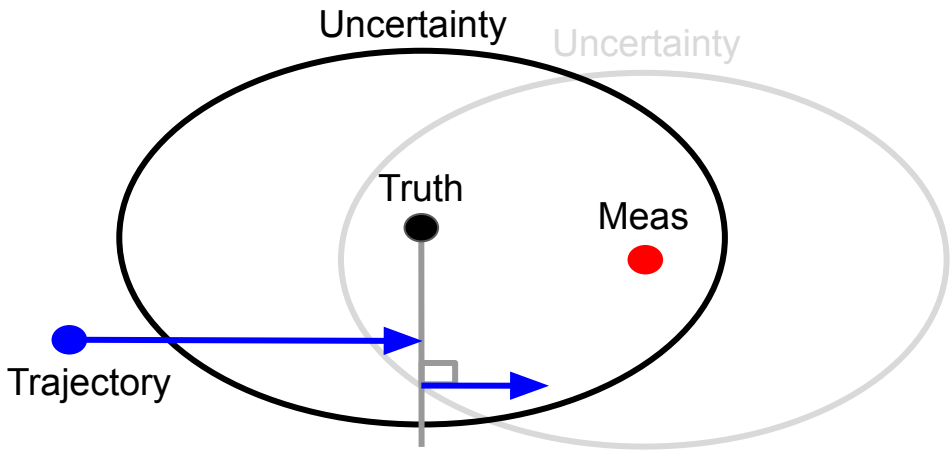
Planar measurement at pad

3D measurement in TPC/DC

- But: Kalman filter updates parameters at one point
  - Planar measurement: At surface
  - What's the point in a volume?

Free Kalman filter

- Unbiased update
  - Update as close to truth as possible
  - Planar: Truth & trajectory at surface
  - 3D: **True position unknown**
    - Unbiased update position unknown
    - Measurement position utilised

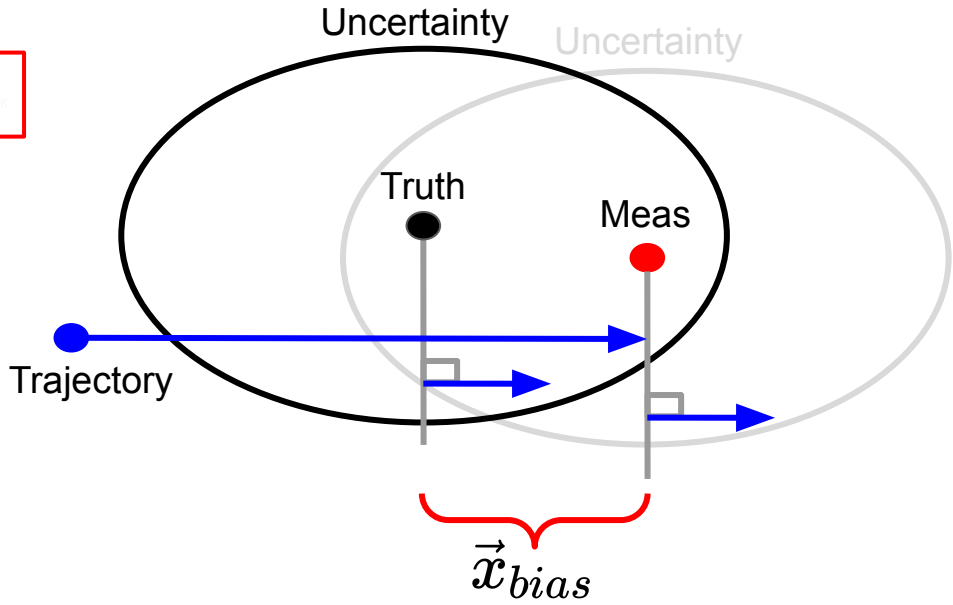


Free Kalman filter

- Unbiased update
  - Update as close to truth as possible
  - Planar: Truth & trajectory at surface
  - 3D: **True position unknown**
    - Unbiased update position unknown
    - Measurement position utilised

$$\vec{x}_{predicted} = \vec{x}_{true} + \vec{x}_{uncertainty} + \vec{x}_{bias}$$

Zero mean contributions

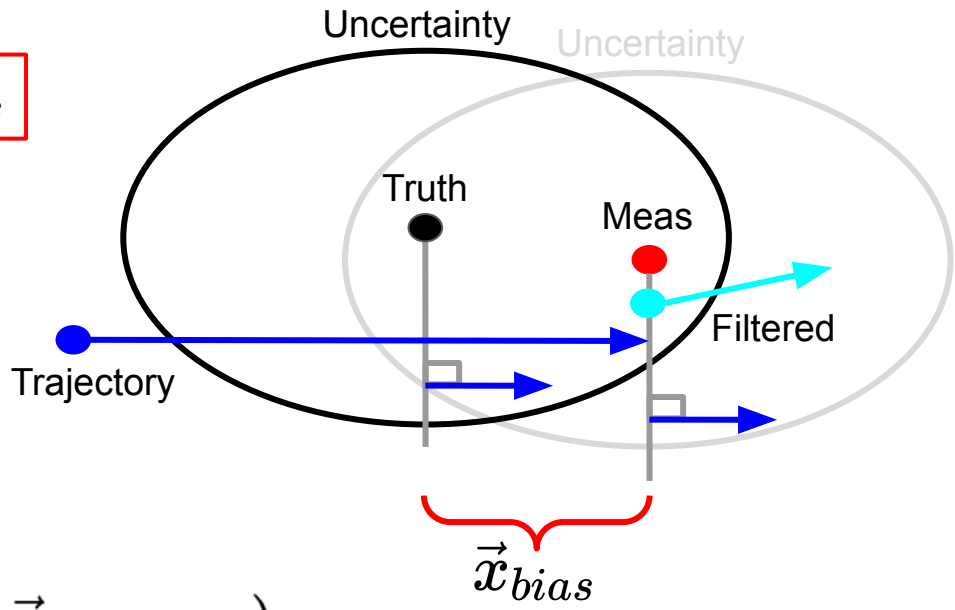


Free Kalman filter

- Unbiased update
  - Update as close to truth as possible
  - Planar: Truth & trajectory at surface
  - 3D: **True position unknown**
    - Unbiased update position unknown
    - Measurement position utilised

$$\vec{x}_{predicted} = \vec{x}_{true} + \vec{x}_{uncertainty} + \vec{x}_{bias}$$

Zero mean contributions



→ Re-derivation of formalism

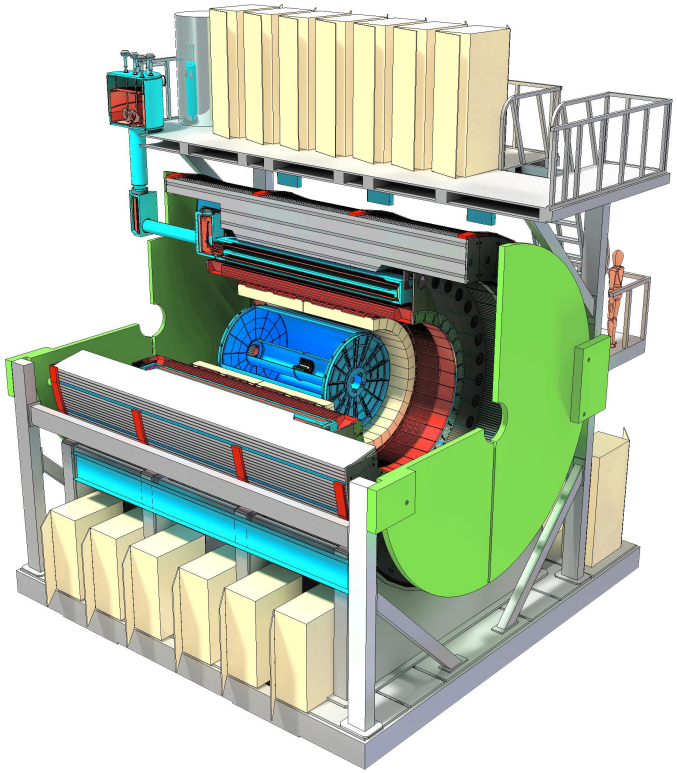
→ Modified and additional terms

$$\vec{x}_{filtered} = \vec{x}_{predicted} + K'(\vec{x}_m - \vec{x}_{predicted})$$

$$\Sigma_{filtered} = (1 - K')\Sigma_{predicted} + \Sigma_{bias} K'^T$$

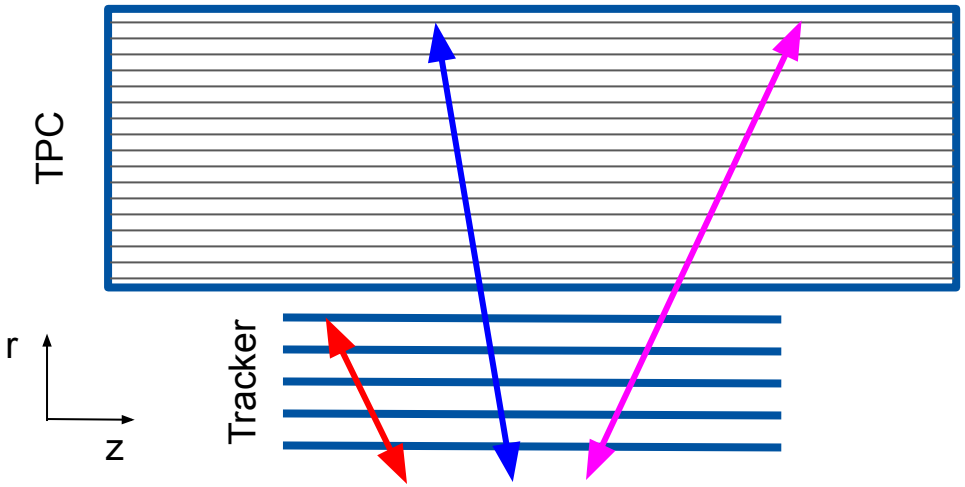
Toy Simulation

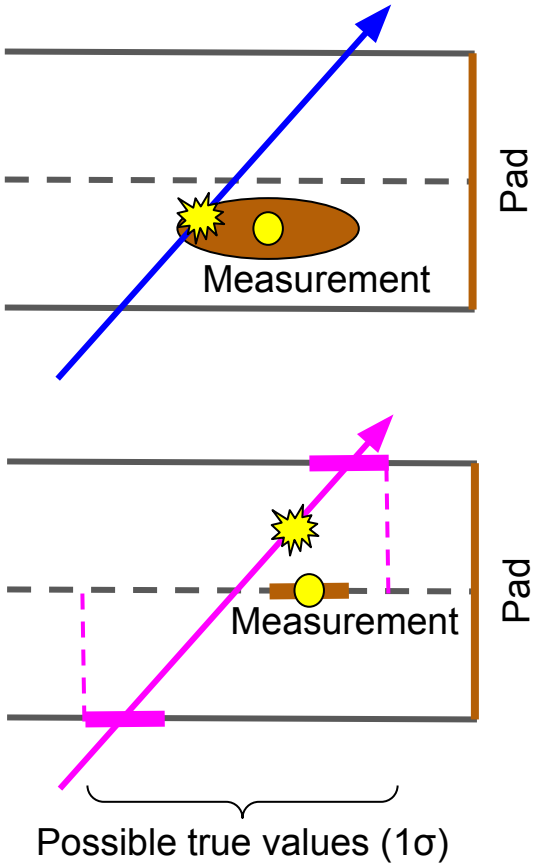
- sPhenix Tracker + TPC
  - 5 layer pixel tracker
  - 48 read-out pads in TPC
  - 1.4 T solenoid magnet
  - **No material**
- Measurement creation
  - Particle gun
    - 10 GeV μ
    - $|\eta| < 1$, full φ coverage
  - Record parameters at sensitive components
  - Gaussian parameter smearing
  - Surface parameters **(x,y)**
  - Volume parameters **(x,y,z)**
- Track fitting only
  - Event contains a single particle
  - Perfect measurement-track association



Track fitting

- 3 configurations for track fitting
  - Tracker only (Baseline)
  - Tracker + TPC with 3D measurements (Free Kalman filter)
  - Tracker + TPC with 2D measurements (Projected Kalman filter)





Learning from measurements

Free parametrisation (Everywhere)

$$(x, y, z, T^x, T^y, T^z, q/p)$$

Bound parametrisation (At surfaces)

$$(x, y, \phi, \theta, q/p)$$

Information from volume measurement

$$(\mu_x, \mu_y, \mu_z) \quad (\sigma_x, \sigma_y, \sigma_z)$$

Information from surface measurement

$$(\mu_x, \mu_y) \quad (\sigma_x, \sigma_y)$$

$$\vec{r} = (x, y, z)$$

$$\vec{r}_{local} = (x, y) = f(\vec{r})$$

$$(T^x, T^y, T^z) = \frac{d\vec{r}/ds}{|d\vec{r}/ds|}$$

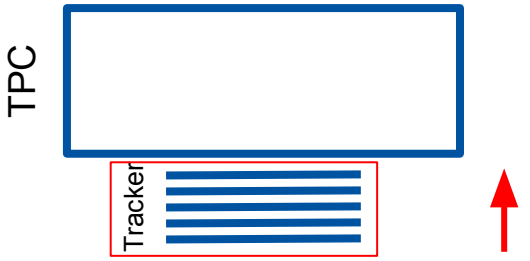
$$\begin{aligned} T^x &= cos(\phi)sin(\theta) \\ T^y &= sin(\phi)\sin(\theta) \\ T^z &= cos(\theta) \end{aligned}$$

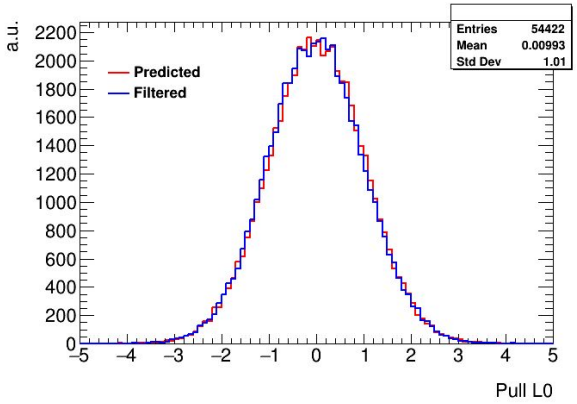
$$\frac{d^2\vec{r}}{ds^2} = \frac{q}{p}\left(\frac{d\vec{r}}{ds} \times \vec{B}(\vec{r})\right)$$

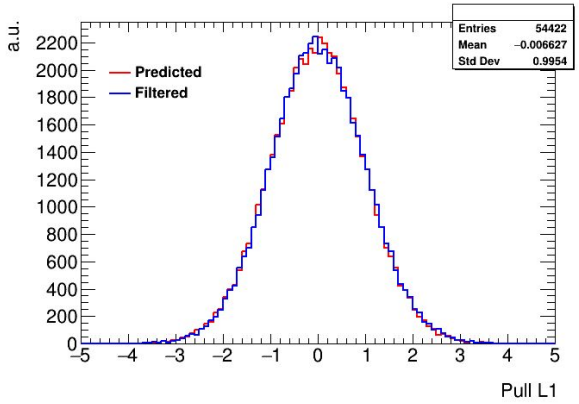
Knowledge gain although parameters not measured

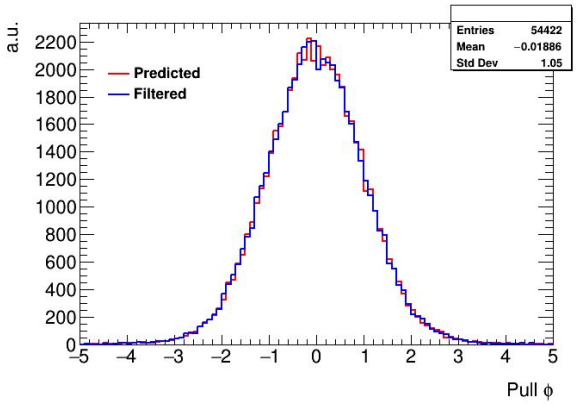
Filtering in Tracker

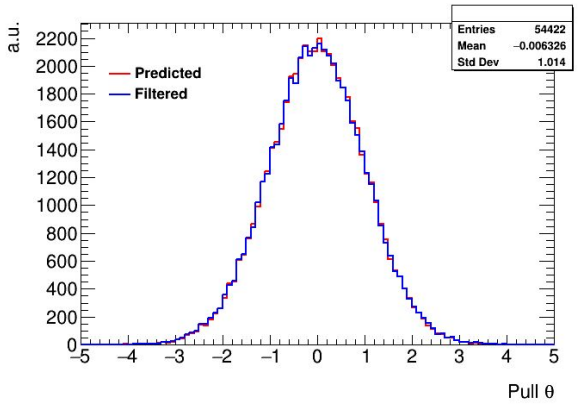
- Filtering identical for all scenarios
- Pull distributions show
  - Unbiased extrapolation (Predicted)
  - Unbiased update (Filtered)

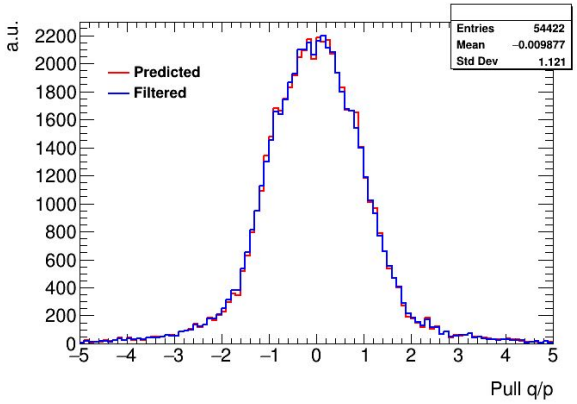






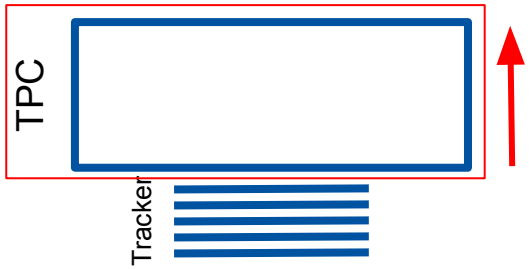


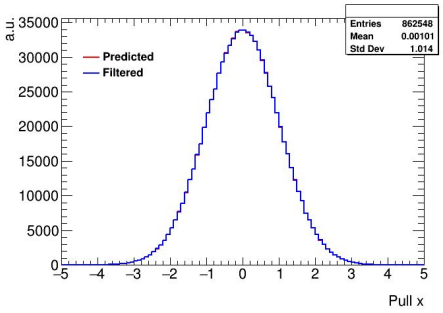


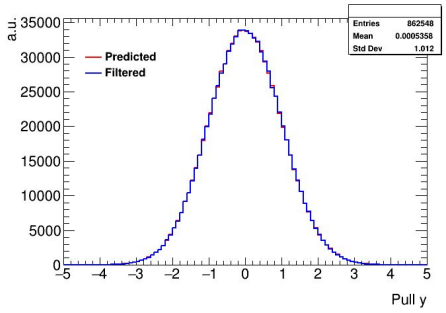


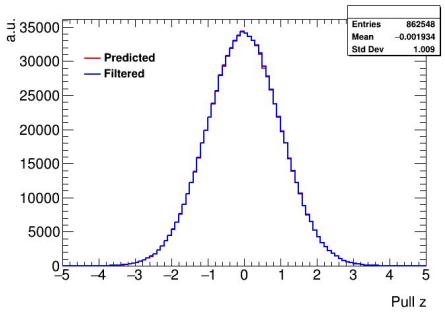
Filtering in TPC

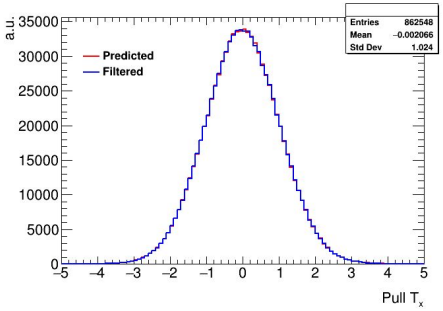
- No issues in free Kalman filter observed
- Too broad for projected Kalman filter pulls
  - Due to error underestimation

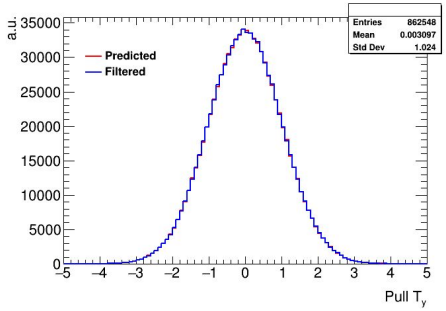


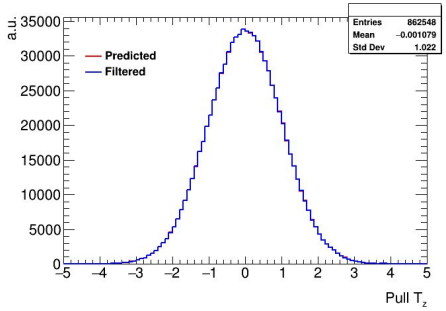


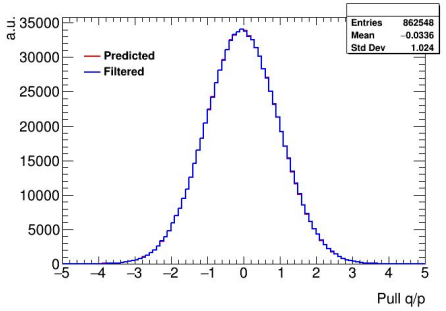






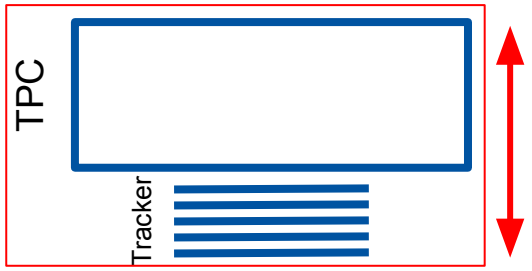


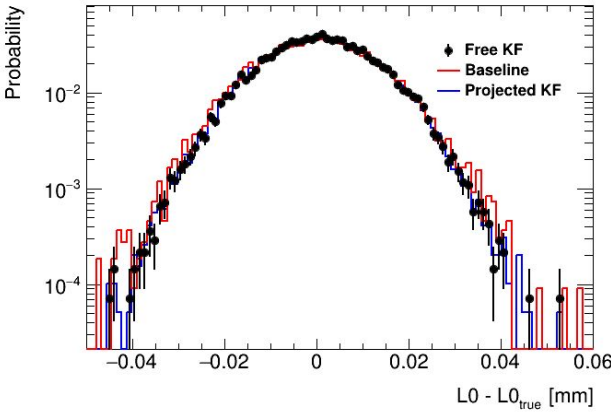


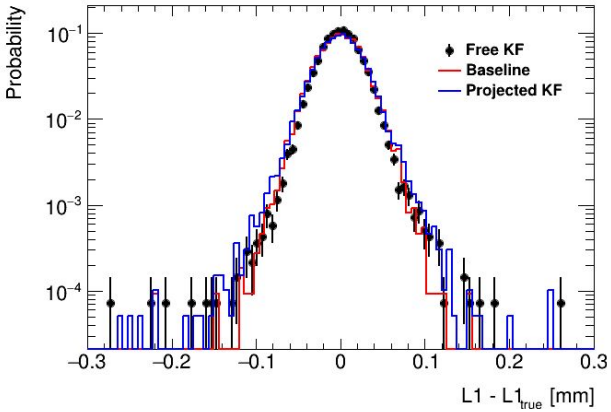


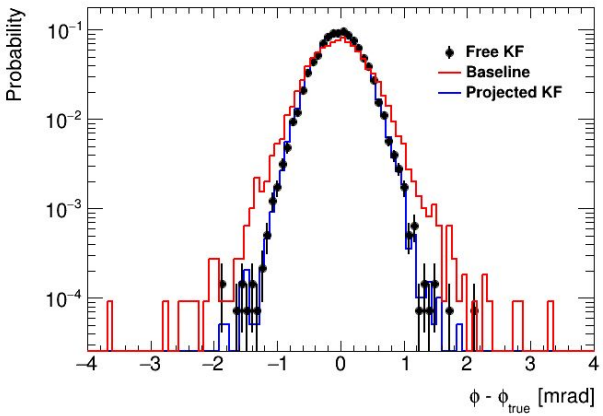
Parameters on innermost layer

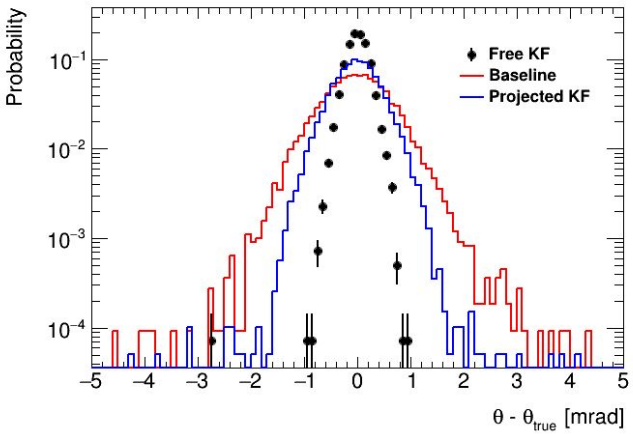
- Spatial resolution dominated by tracker
- Angular and q/p resolution improved
- Projected θ-resolution worse due to
  - Projection approximation
  - Tracker y-resolution

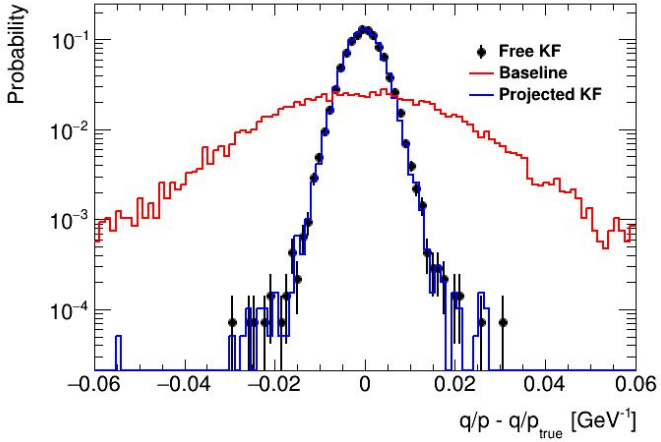






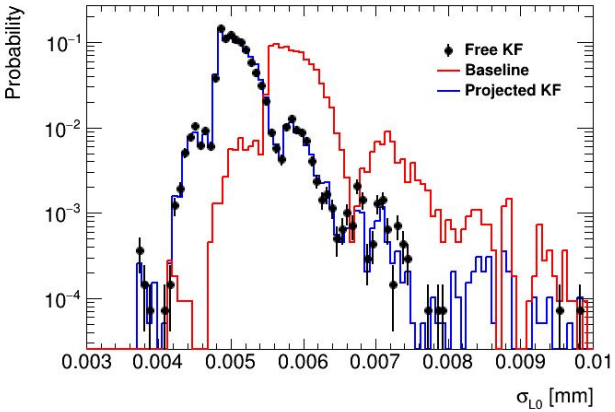


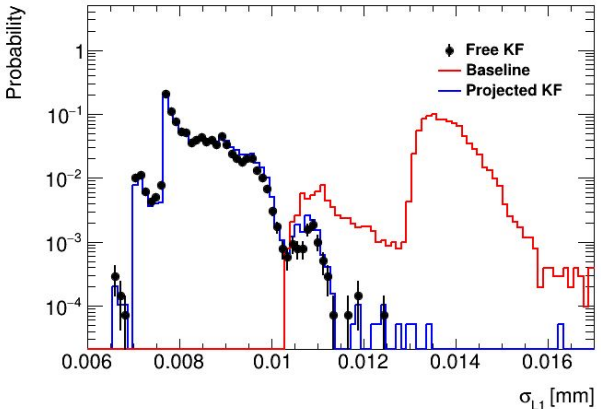


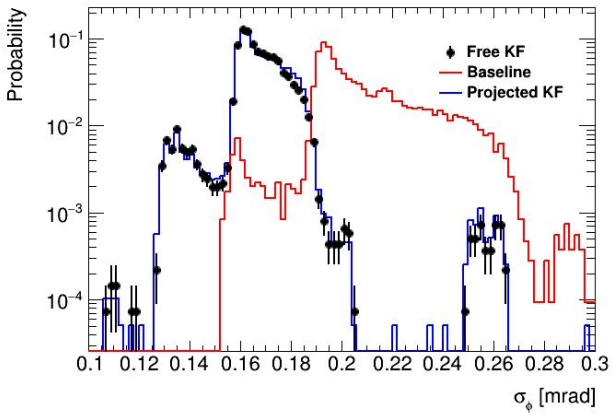


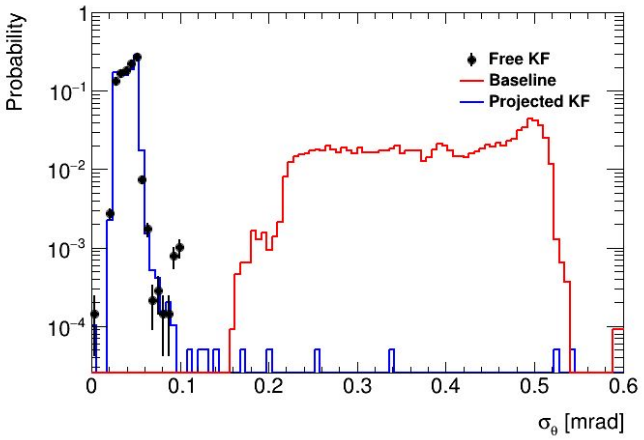
Parameters on innermost layer

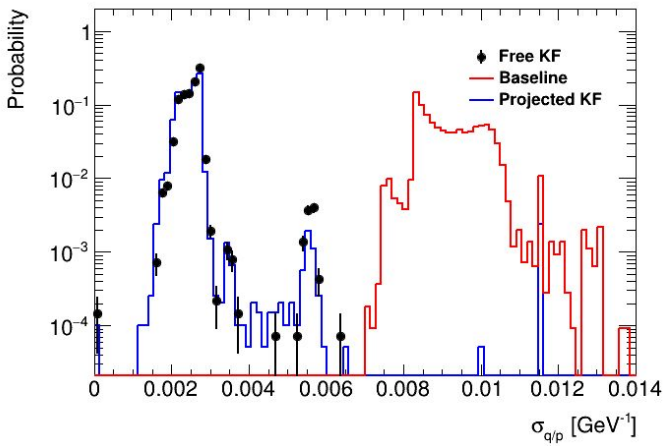
- Uncertainty reduced due to TPC data
- Almost identical result from both TPC scenarios











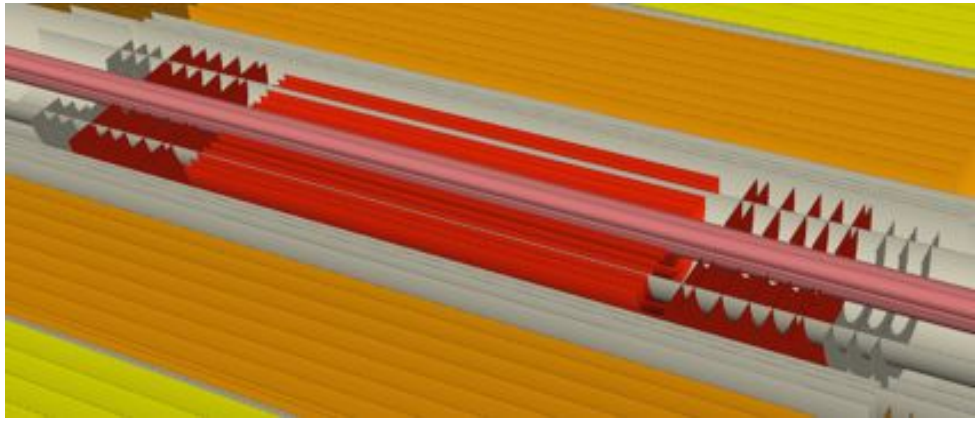
Summary

- Simplified detector and measurements
- Kalman filter
  - Progressive learning
  - Formulated for surface measurements
- TPC/DC
  - 3D measurements in **volume**
  - Kalman filter update position biased
  - **Re-deriving** of expression mandatory
    - If position bias vanishes → “classical” Kalman filter
    - Considerable as generalised Kalman filter
- Toy comparison
  - Filter with 3D measurements **possible**
  - 2D and 3D TPC measurements comparable similar results
    - Uncertainties almost identical
    - Angular resolution improved
- For now: **Conceptual study** rather than real application

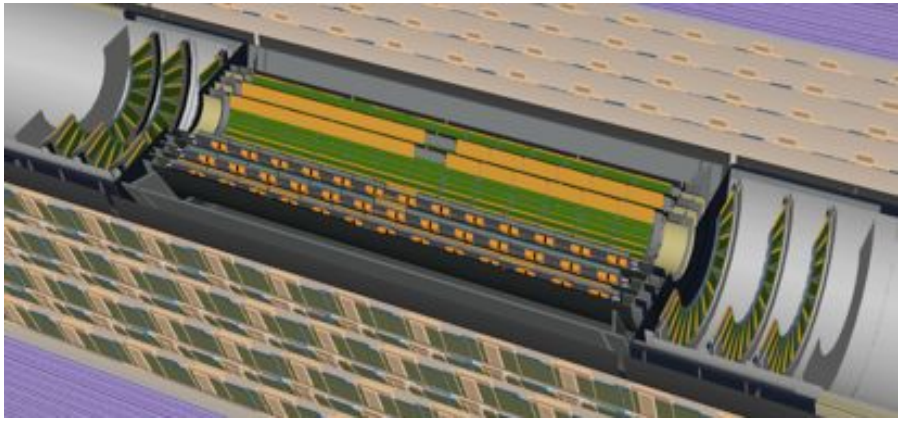
Backup

Detector representations

FastSim: Coarse granularity



Geant4: Fine granularity



ATLAS ttbar event in kSI2k sec:

FastSim: 7.4

Geant4: 1990

Combinatorial Kalman filter

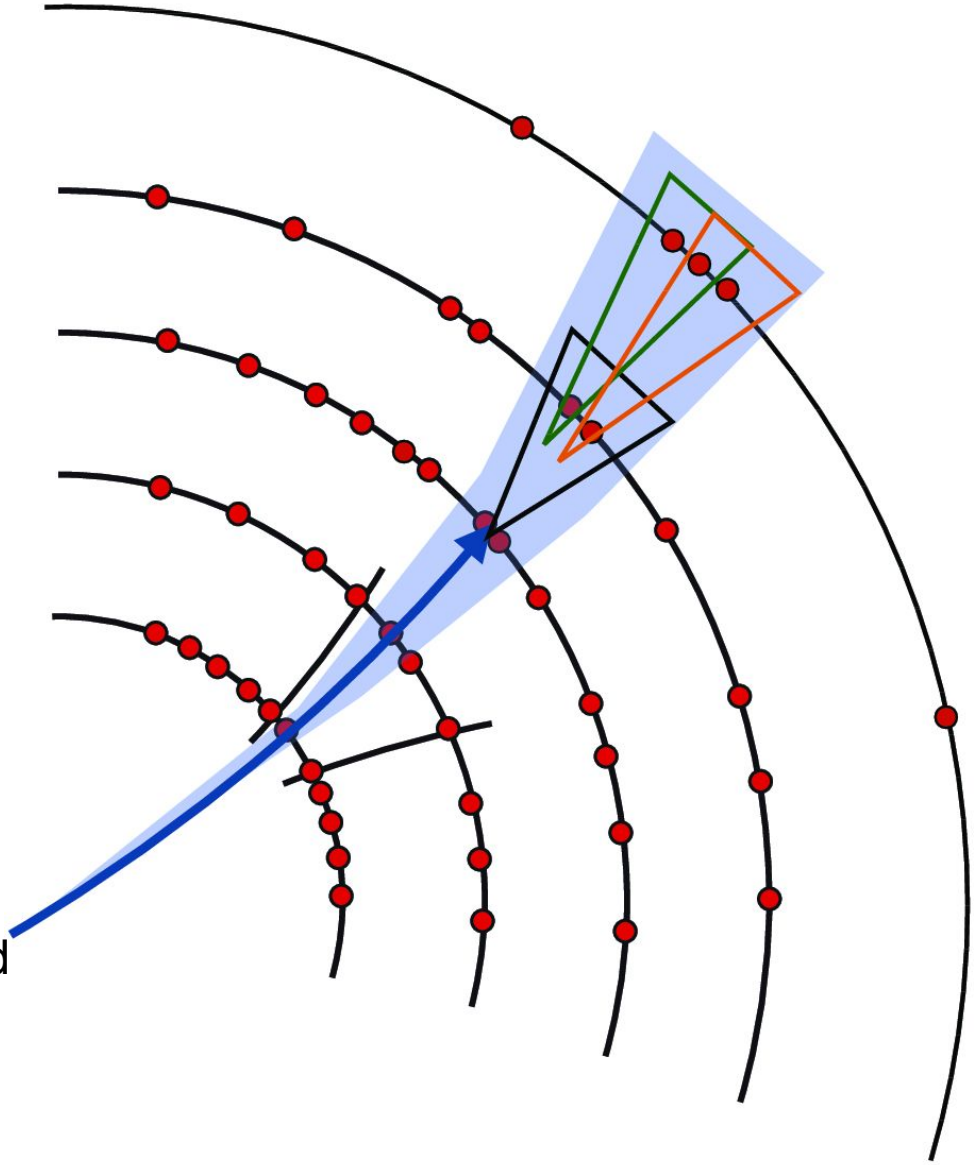
Scenario:
KF encounters multiple hits that may fit

KF can only treat one hit

- Split the state (Forking)
- Each state treats single hit
- Propagate each independently

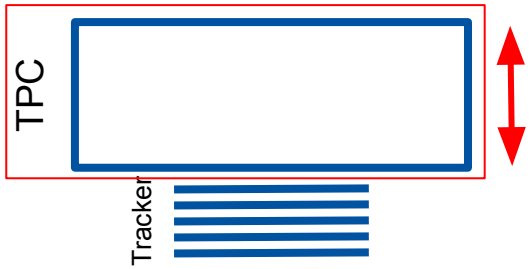
Right track should provide fitting hits

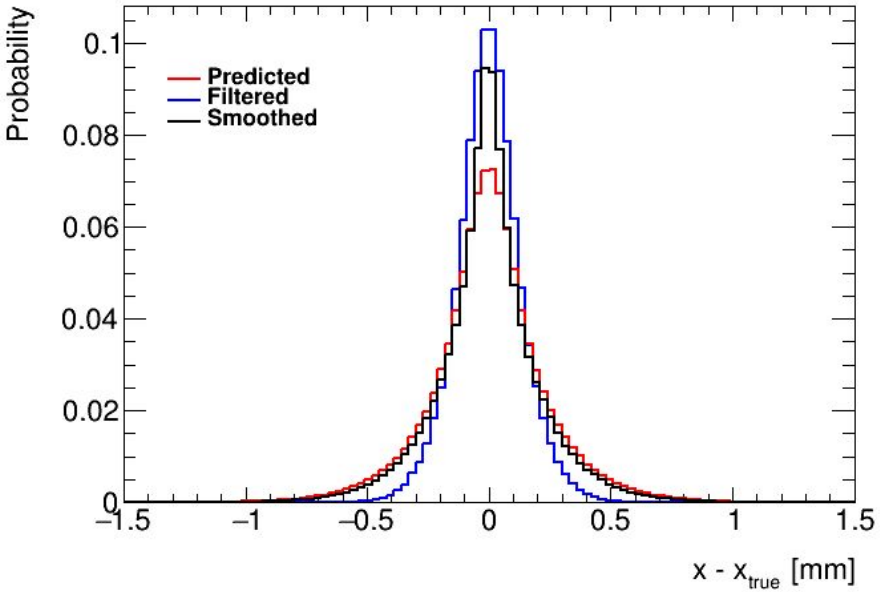
- Further hits will show which one
- Decision postponed to future layers
- Data will show which one is dropped

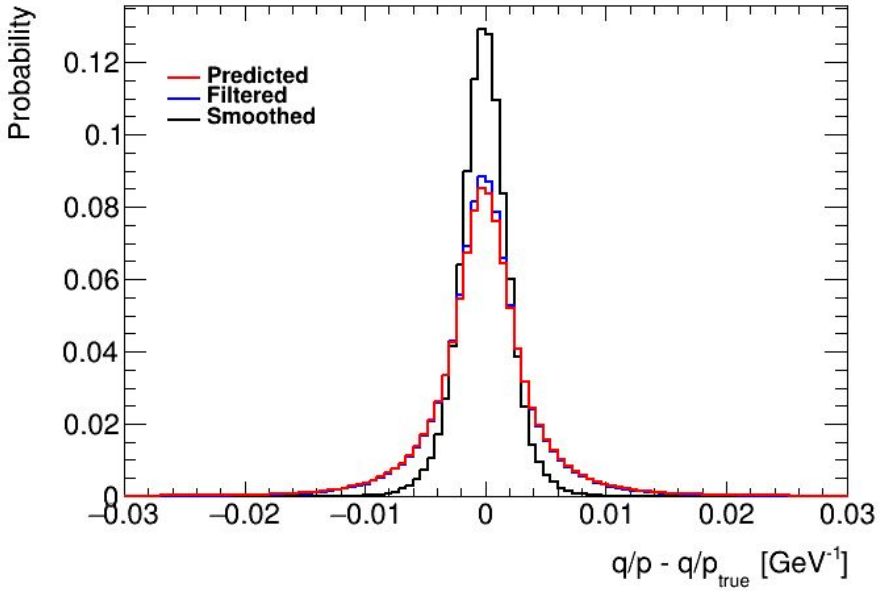


TPC parameters of free Kalman filter

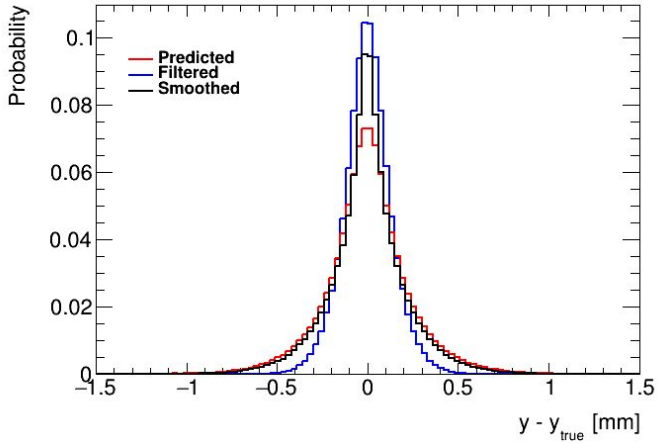
- Overview of all updates
- Means converge to truth trajectory
  - Smoothing wrt predicted position
- Steady q/p resolution improvement

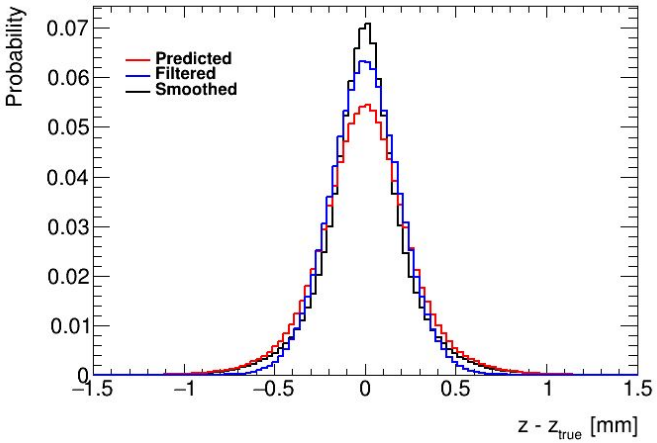


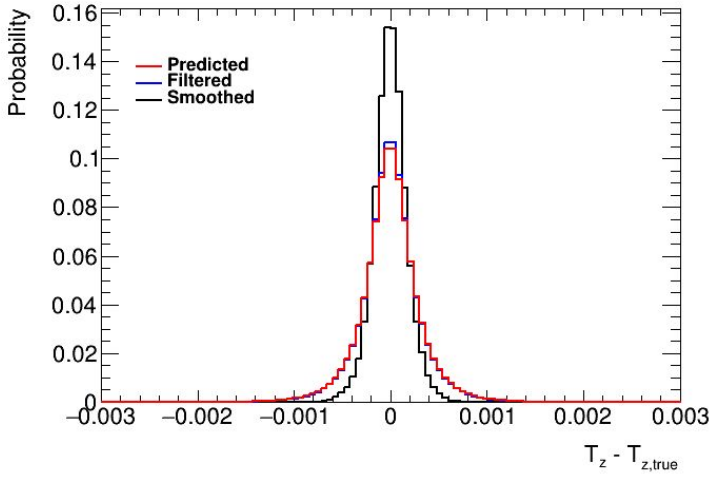


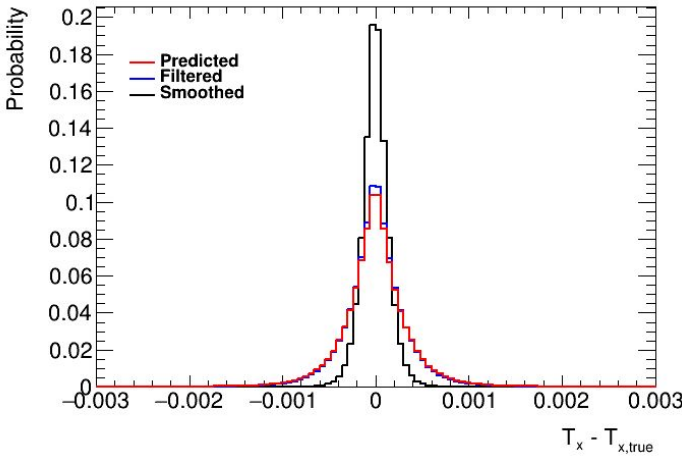


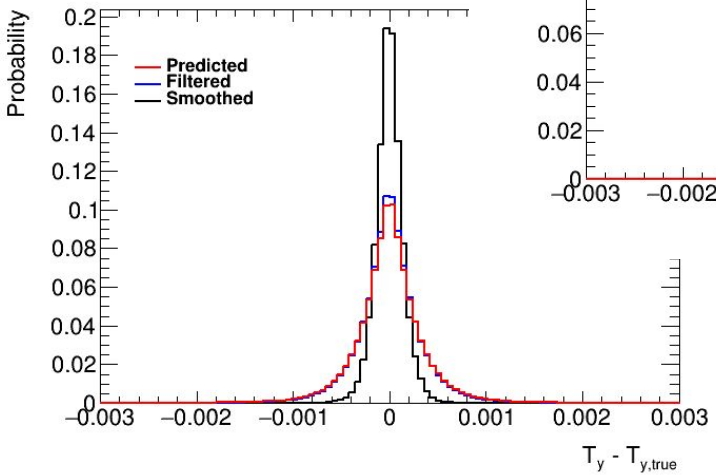
TPC resolutions



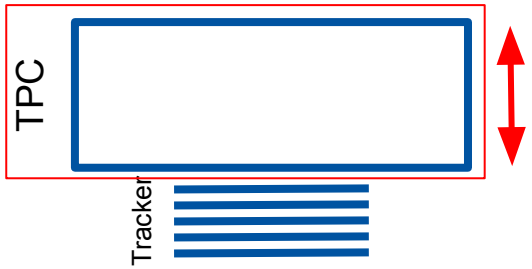




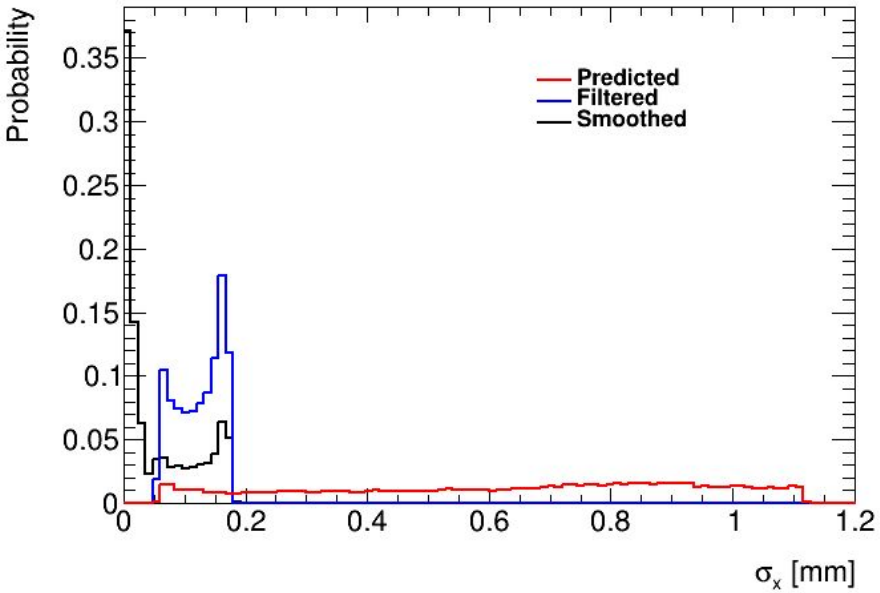


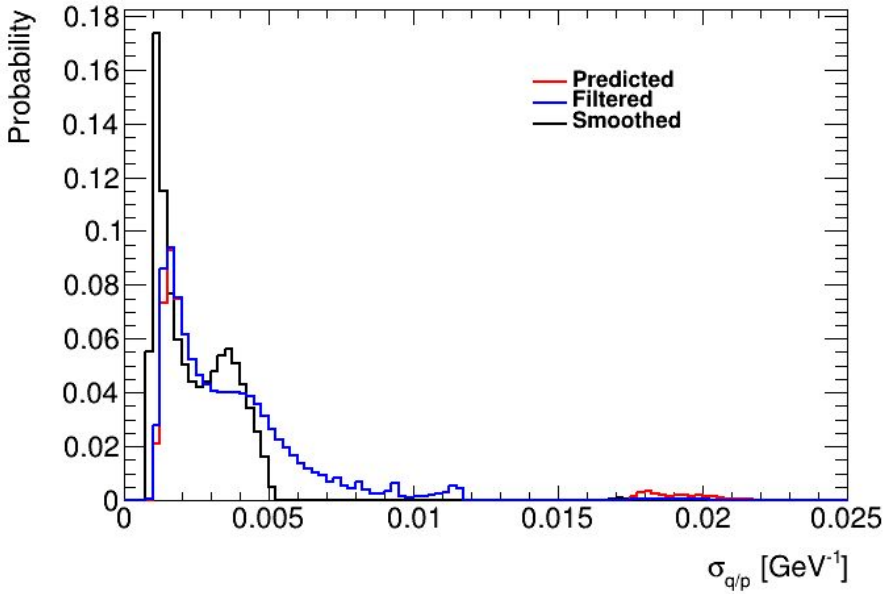


TPC parameters of free Kalman filter

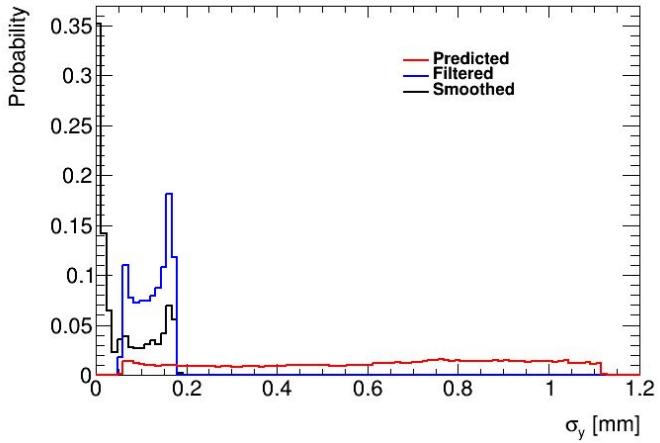


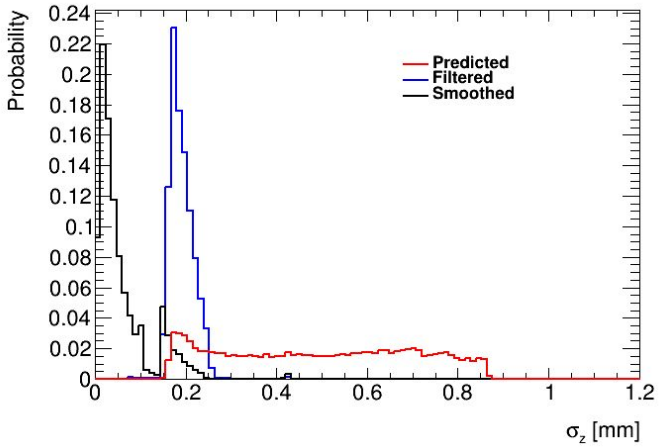
- Overview of all updates
- Means converge to truth trajectory
  - Smoothing wrt predicted position
- Steady q/p resolution improvement
- Data utilisation reduces errors correctly
  - Major absolute impact for spatial parameters from filtering

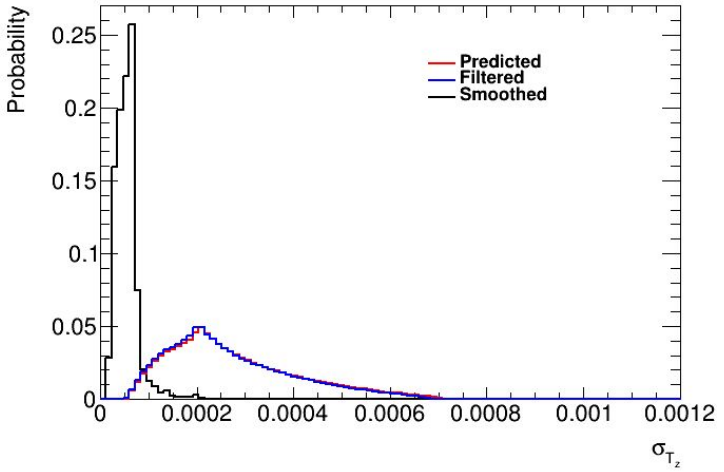


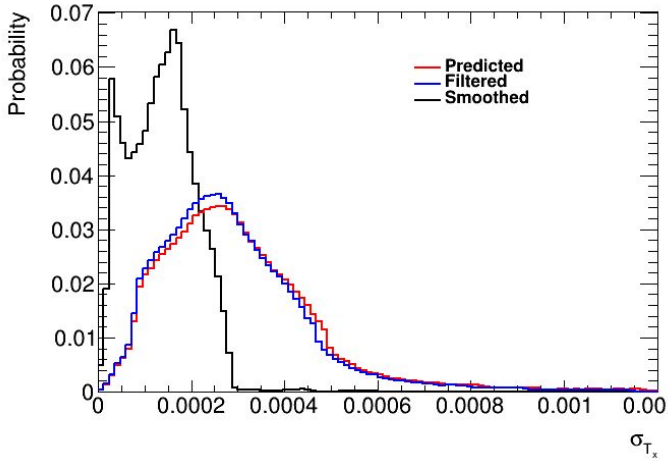


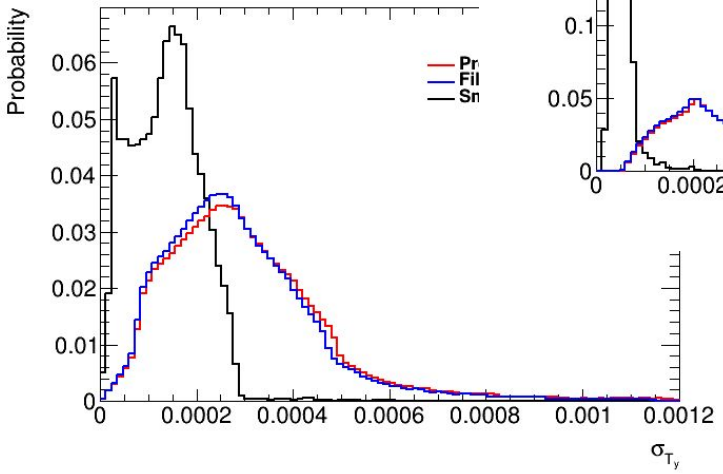
TPC errors


Probability
Predicted
Filtered
Smoothed
σ_y [mm]


Probability
Predicted
Filtered
Smoothed
σ_z [mm]


Probability
Predicted
Filtered
Smoothed
σ_{T_z}


Probability
Predicted
Filtered
Smoothed
σ_{T_x}


Probability
σ_{T_y}