Impact of sensor degradation on 4D CKF performance

Rodrigo Estevam de Paula
Marco Aurelio Lisboa Leite
Vitor Heloiz Nascimento

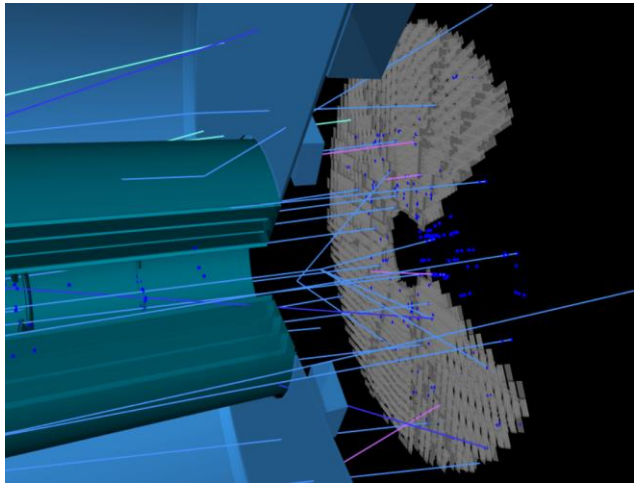
Universidade de São Paulo (BR)

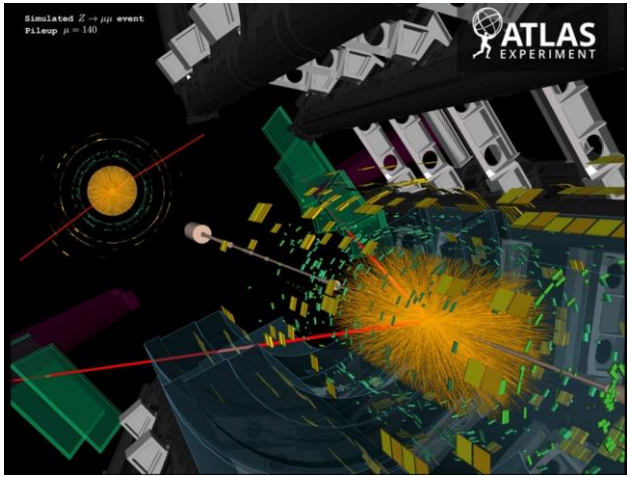
20 November, 2024

CNPq/MCTI INCT CERN/Brazil and FAPESP (2020/04867-2)

Objectives of this study

- Evaluate the performance of the Combinatorial Kalman Filter (CKF) reconstruction of a timing layer detector
  - Will be used as baseline performance for new reconstruction methods to be idealized High Luminosity (HL) environments
- Measure the impact of the radiation damage on the reconstruction performance
  - Important information for sensor commissioning

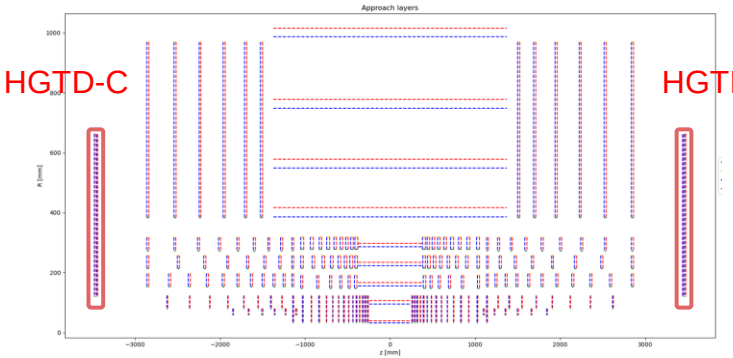




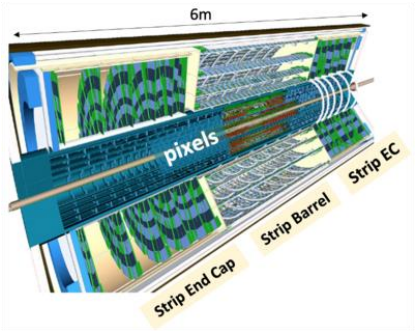
Images source: [1][3]

Detector geometry

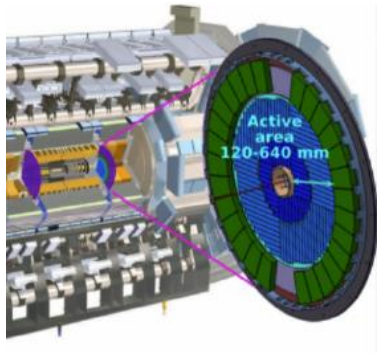
- Using ATLAS ITk and HGTD as a proxy for this study
- The High Granularity Timing Detector (HGTD) is the timing layer to be used on ATLAS for HL-LHC
  - Bult with special silicon detectors (LGAD) with high timing resolution (around 35 ps)
- Included smearing of time measurements on HGTD layer to simulate the timing layer
  - Used the nominal 35 ps resolution as described in the HGTD TDR [1] (Table 2.1)



ITk + HGTD sensor layout in ACTS



ITk visualization

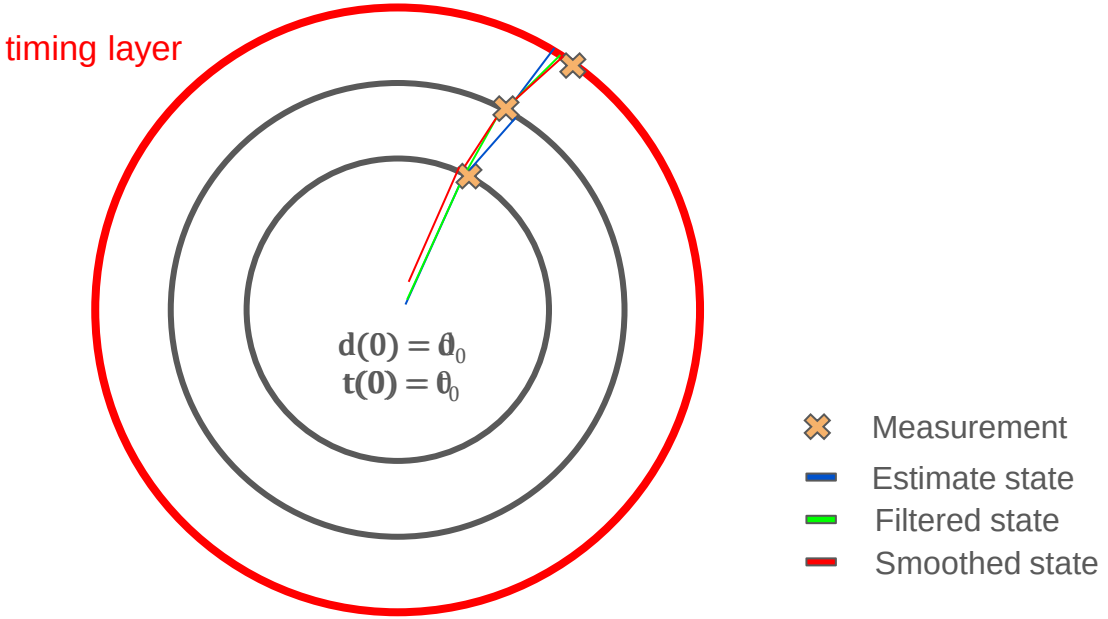


HGTD position within ATLAS

Images source: [4][5]

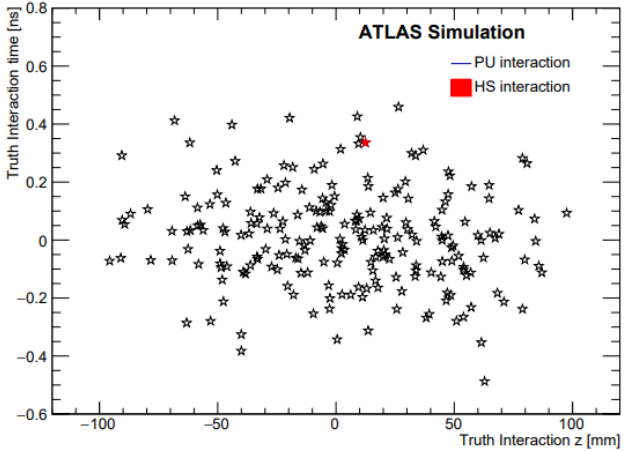
Adaptation of CKF for time reconstruction

- The present implementation of CKF already propagates correctly every parameter on the vector state
- The only adaption needed was to change the CKF's first time estimate to correspond to the travel time from origin to first hit point
    - Important for vertex reconstruction, as tracks from same vertex would be assigned different t0



Simulating HL-LHC beam with ACTS Pythia8

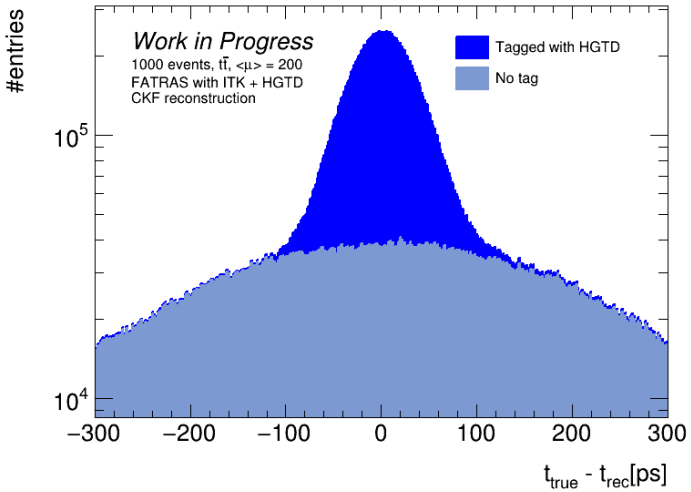
- Simulated HL-LHC Bunch Crossing as described on the HGTD TDR [1](Pag.5)
  - z0 ~ N(0,50 mm), t0 ~ N(0,175ps)
- Simulated 1000 events of pp → ttbar
- Using *Fatras* for detector simulation

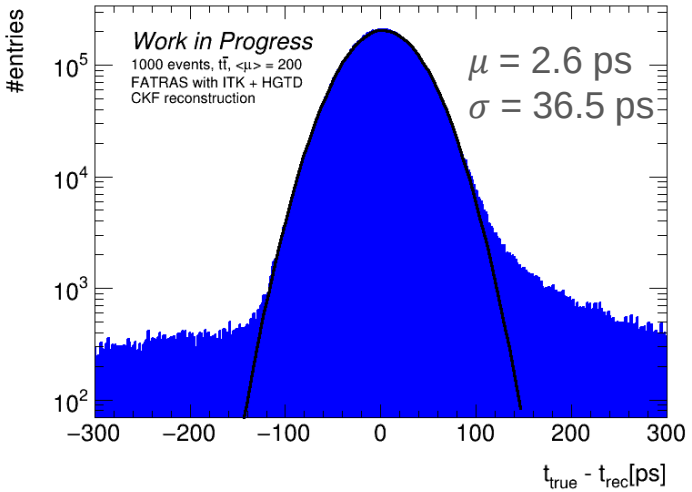


Images source: [1]

Performance evaluation: Residue plot

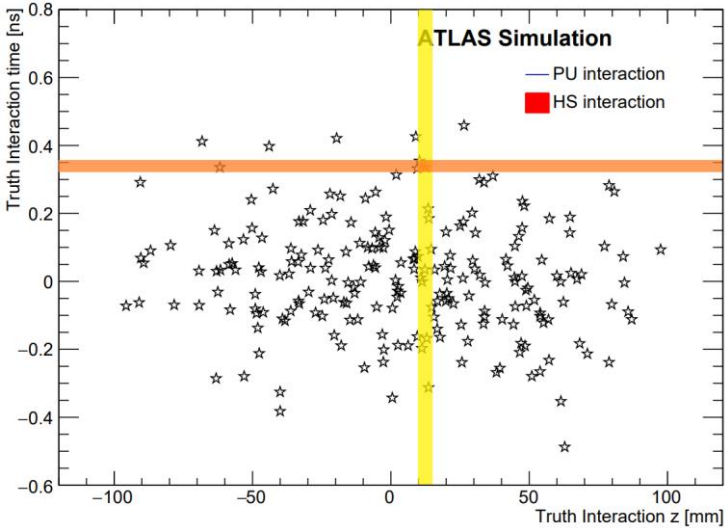
- Tracks without HGTD tagging rely on first estimate for time reconstruction, which has a high error associated with.
- When a track reaches HGTD, the error drops significantly because of the low error attributed to the measure
- In the plots below, the covariance threshold to separate the tracks (err_eT) was set to 1000





Global vertex time (t_0) reconstruction

- One of the main objectives of a timing layer detector is to provide an additional dimension for pileup rejection
- If we estimate (z_0,t_0) with enough resolution, it is possible to use this information to better isolate hard scatter from pileup
  - HGTD TDR proposes Jet reconstruction algorithms where the time information can be used



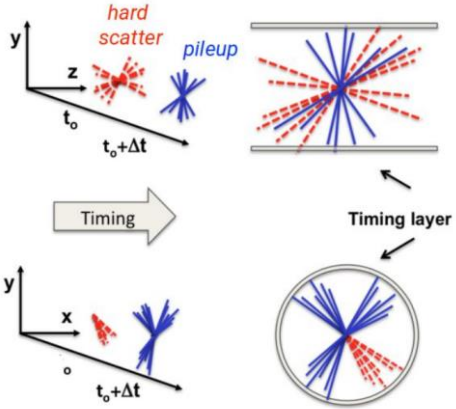
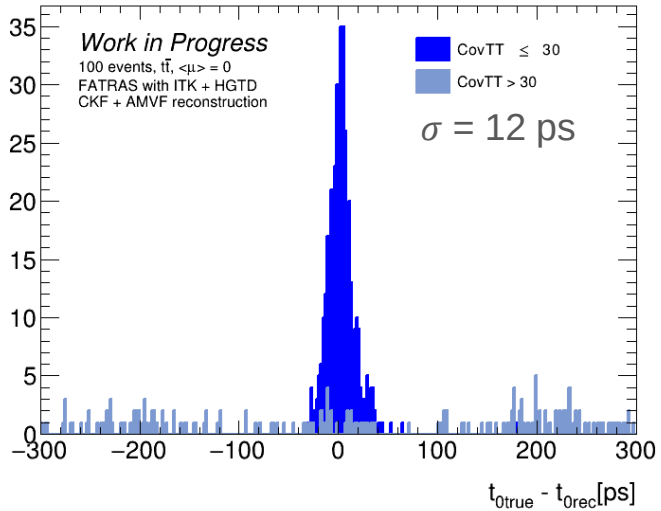
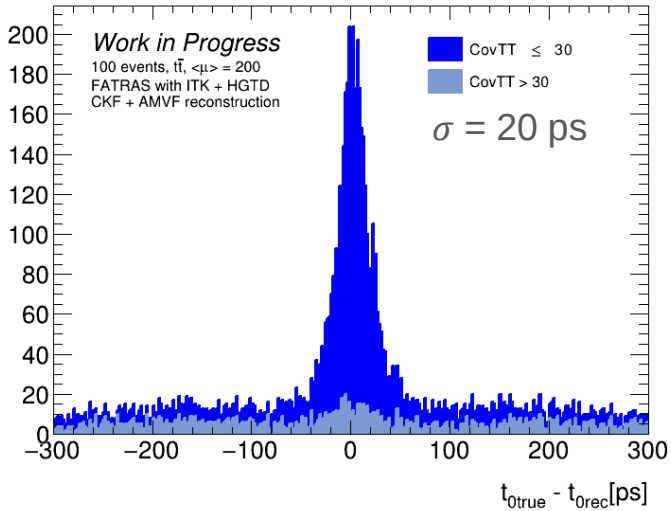


Image source: [1]

AMVF Vertex time reconstruction

- To understand the impact of CKF on the global t_0 reconstruction we added a step to estimate primary vertex positions using AMVF
  - Need to include time information on the primary vertex seeder!
- Estimates with high residual can mostly be filtered by the time error parameter on the covariance matrix (internal parameter of AMVF)
  - Work point choice of purity x sensibility
- After filtering out high error estimates, a Gaussian fit was made to get t_{0rec} resolution

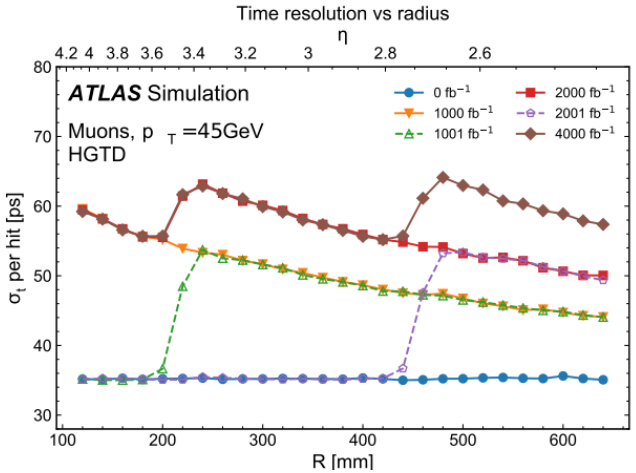




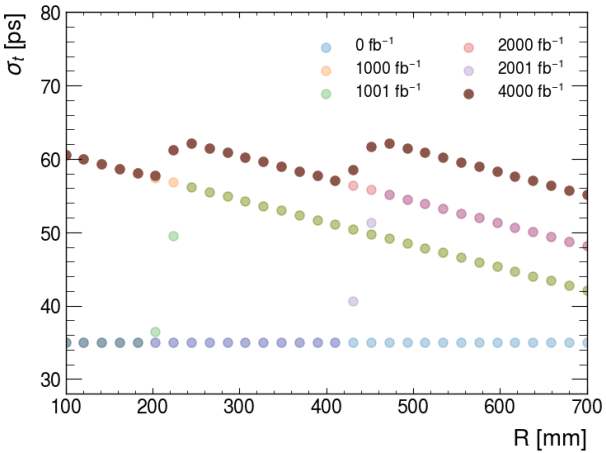
How to evaluate the impact of sensor degradation effects on ACTS?

Simulation of sensor deterioration

- HGTD sensors will degrade with integrated luminosity of LHC
  - The degradation is well characterized with test beams
- Asserting changes of sensor resolution is important for commissioning
- We simulated the sensor deterioration with increasing integrated luminosity
  - Using what is informed at the HGTD TDR [1] (Fig.2.13)



HGTD sensor resolution curves for various integrated luminosities



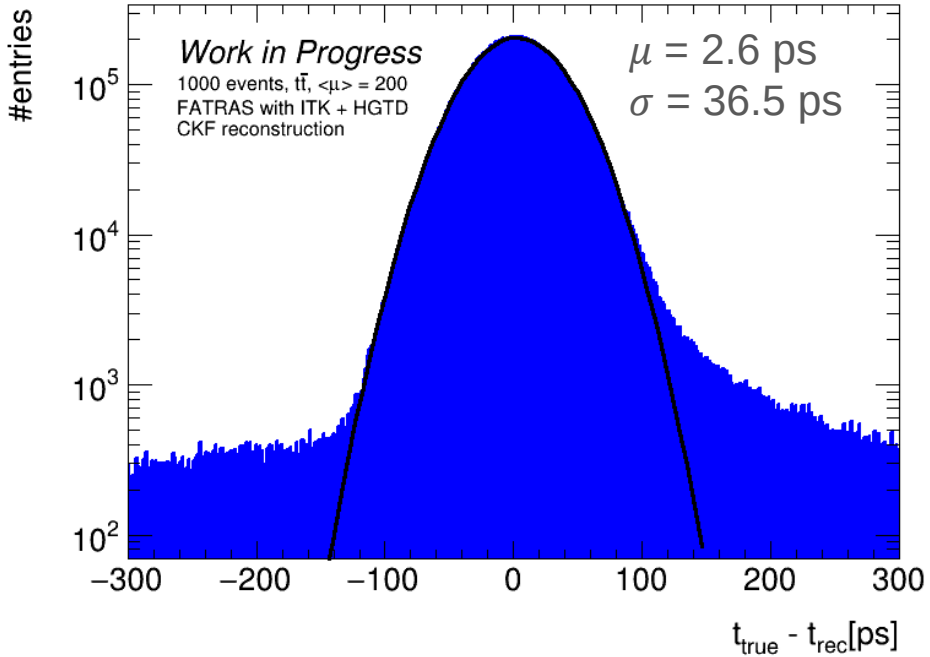
Analytical approximation of sensor resolution curves

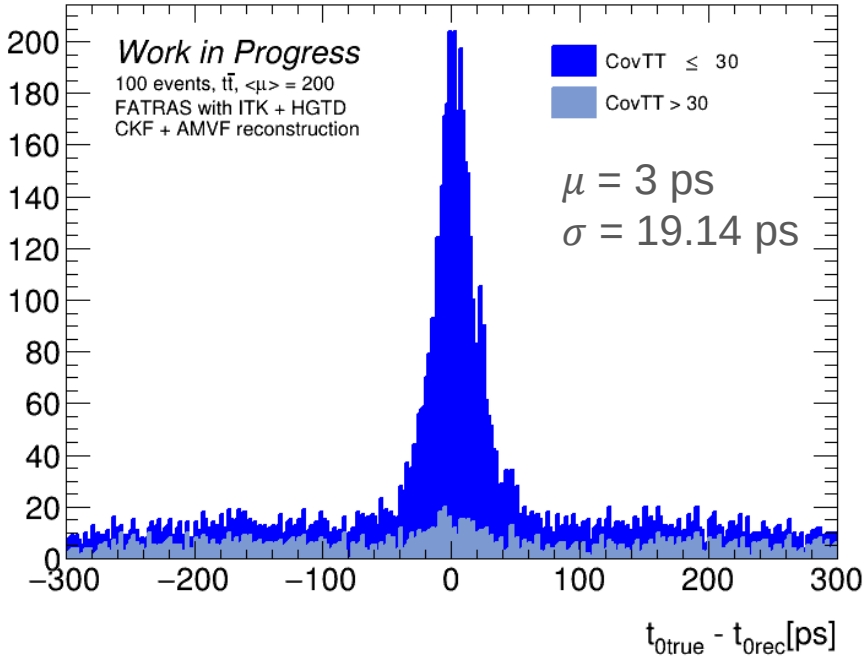
HGTD detector unit layers (TDR Fig.7.19)

Images source: [1]

Reconstruction performance at 0 fb^{-1}

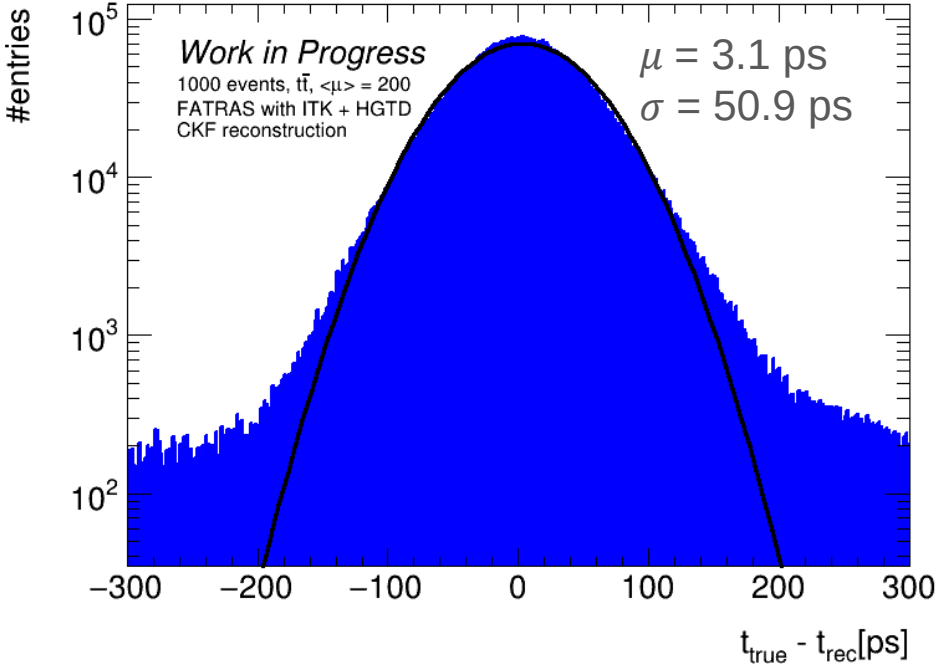
0 fb^{-1}: 36.5 ps

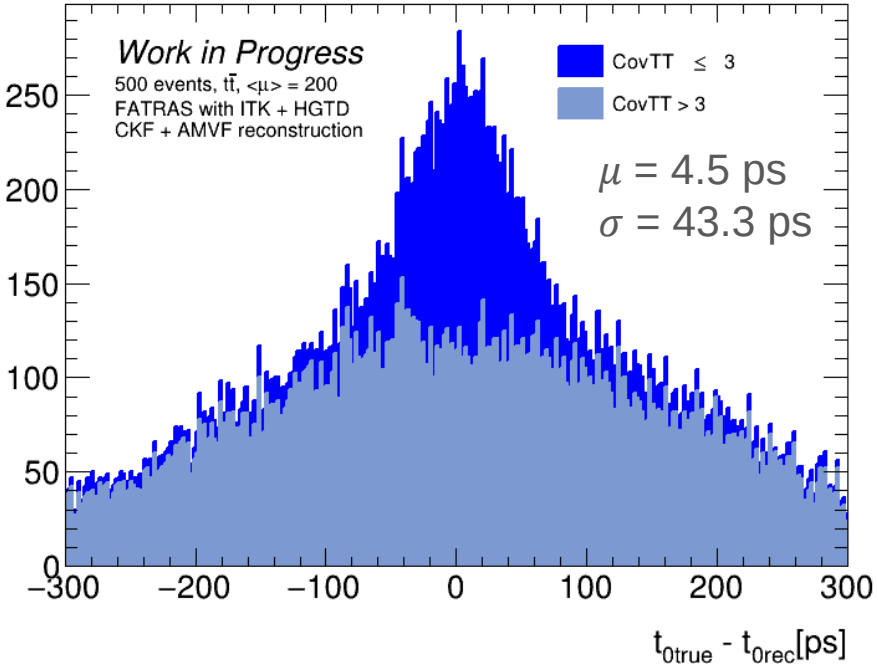




Reconstruction performance at 1000 fb^{-1}

0 fb^{-1}: 36.5 ps → 1000 fb^{-1}: 50.9 ps

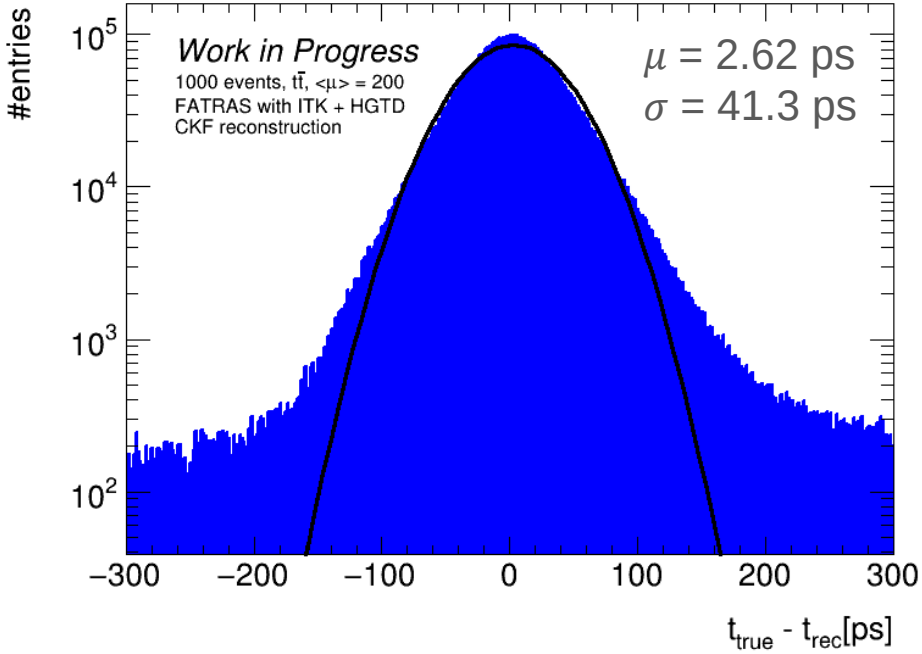


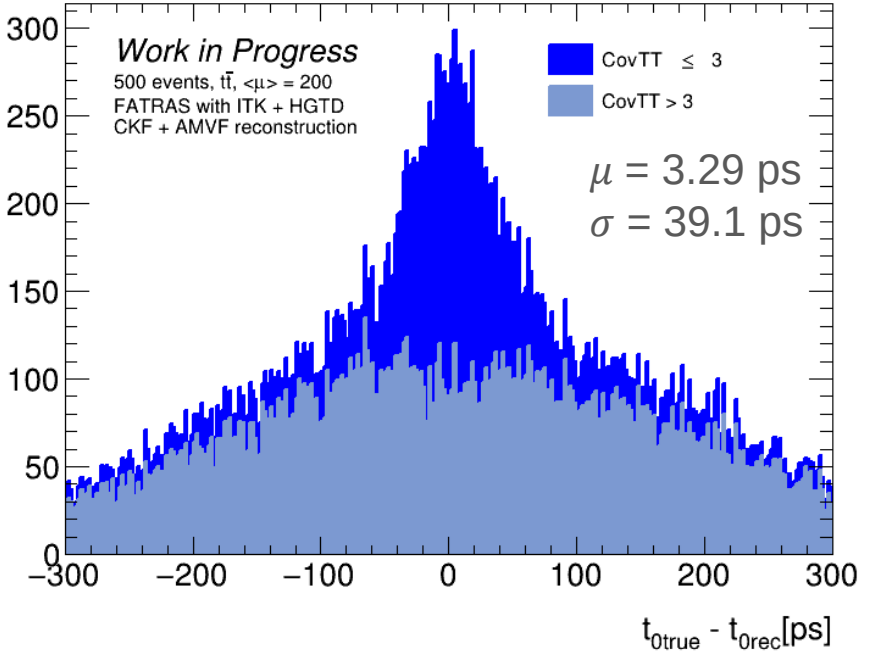


Reconstruction performance at 1001 fb^{-1}

0 fb^{-1}: 36.5 ps → 1000 fb^{-1}: 50.9 ps → 1001 fb^{-1}: 41.3 ps

inner ring replacement

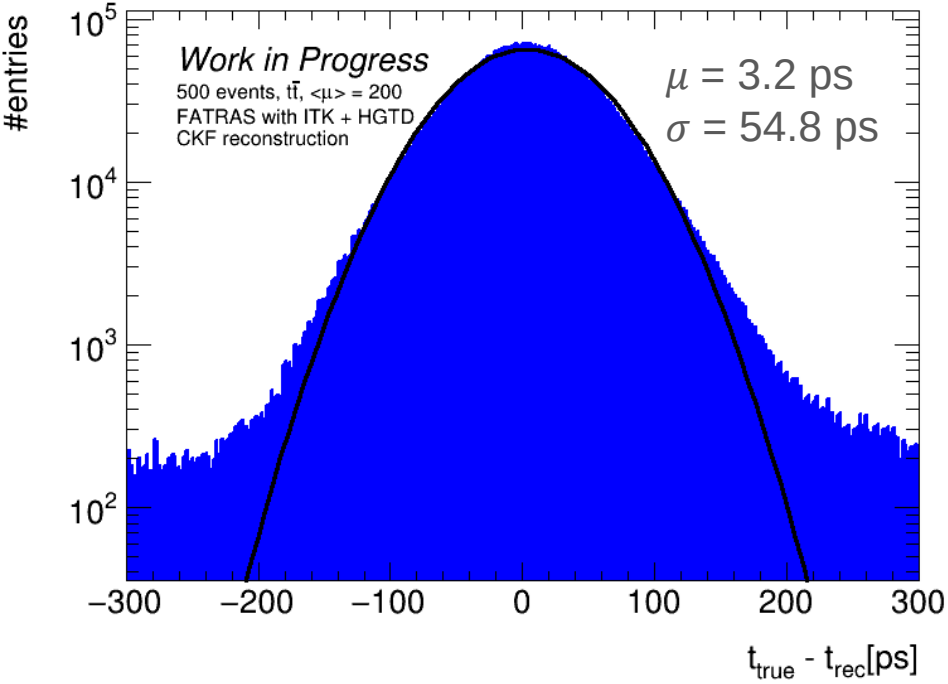


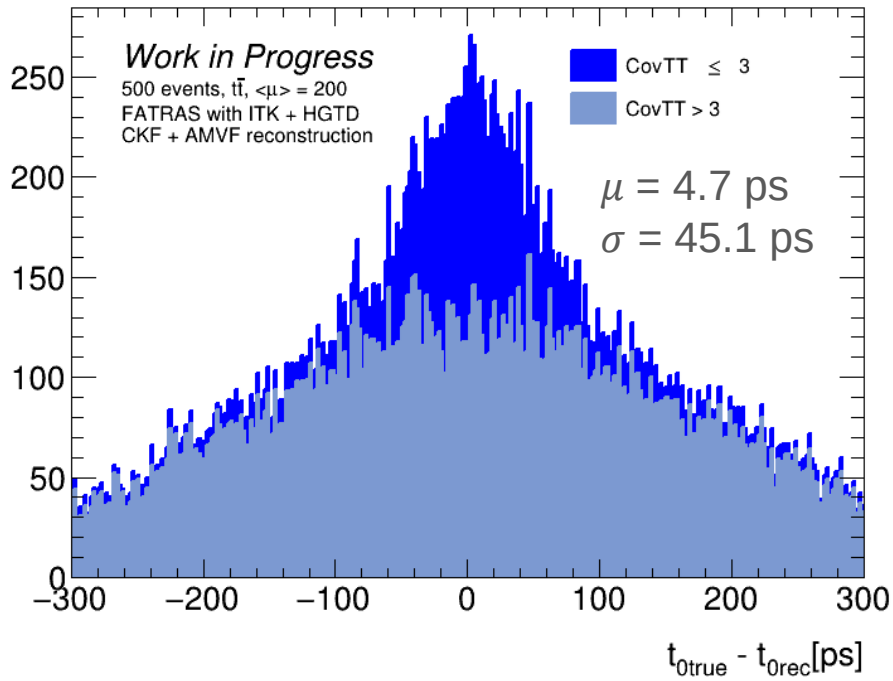


Reconstruction performance at 2000 fb^{-1}

0 fb^{-1}: 36.5 ps → 1000 fb^{-1}: 50.9 ps → 1001 fb^{-1}: 41.3 ps → 2000 fb^{-1}: 54.8 ps

inner ring replacement



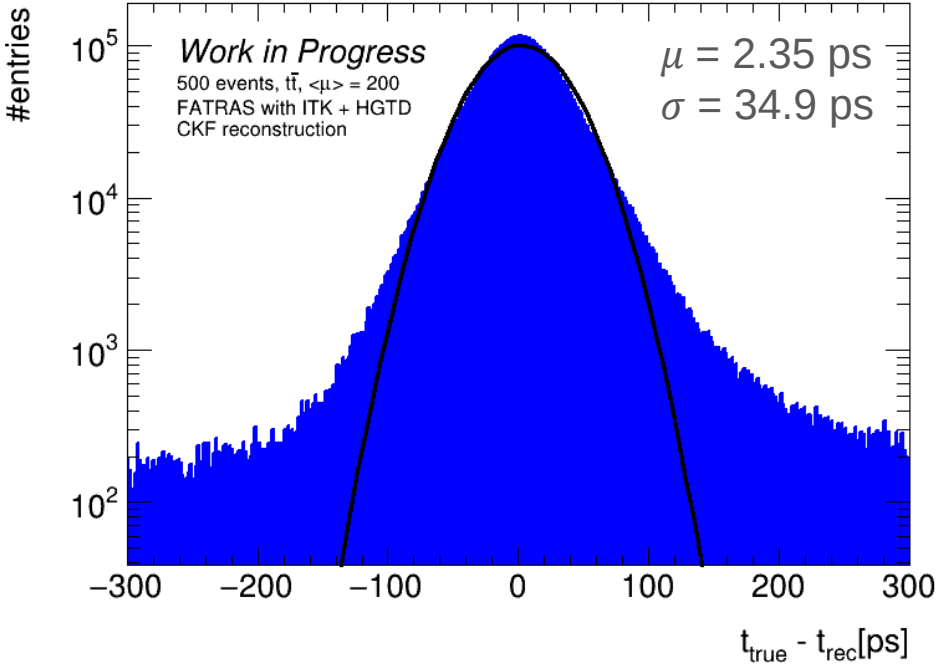


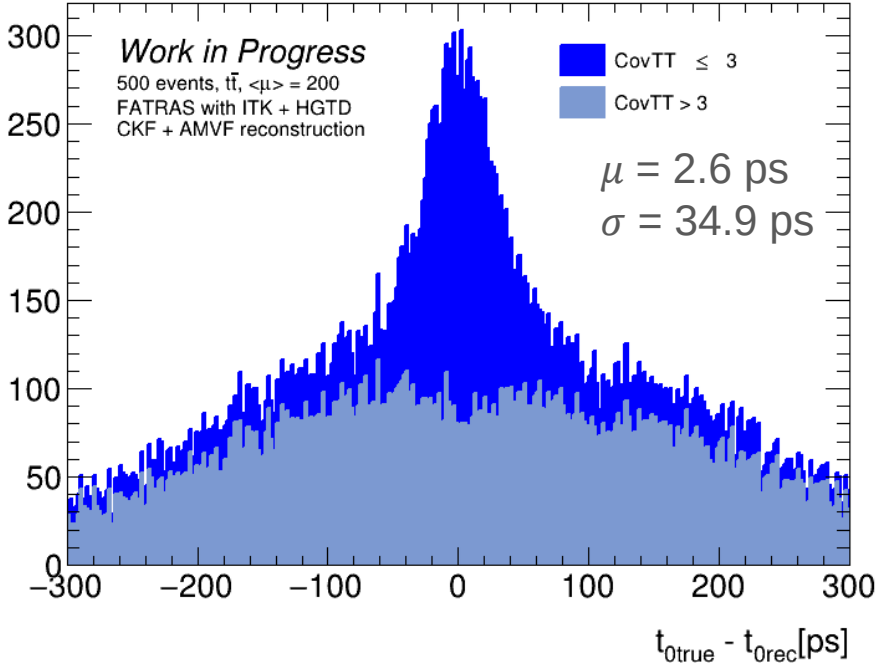
Reconstruction performance at 2001 fb^{-1}

0 fb^{-1}: 36.5 ps → 1000 fb^{-1}: 50.9 ps → 1001 fb^{-1}: 41.3 ps → 2000 fb^{-1}: 54.8 ps → 2001 fb^{-1}: 34.9 ps

inner ring replacement

middle and inner ring replacement



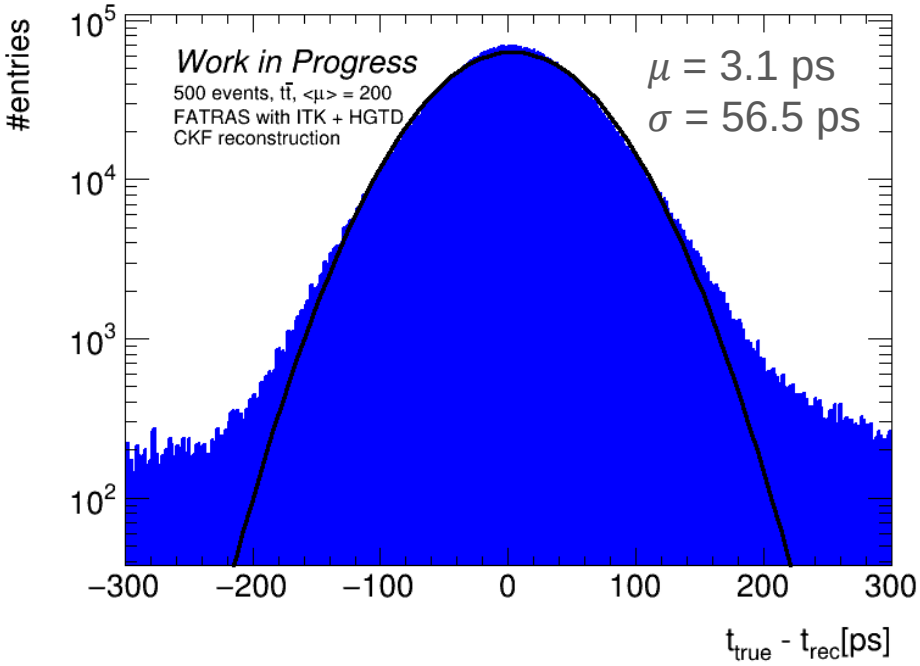


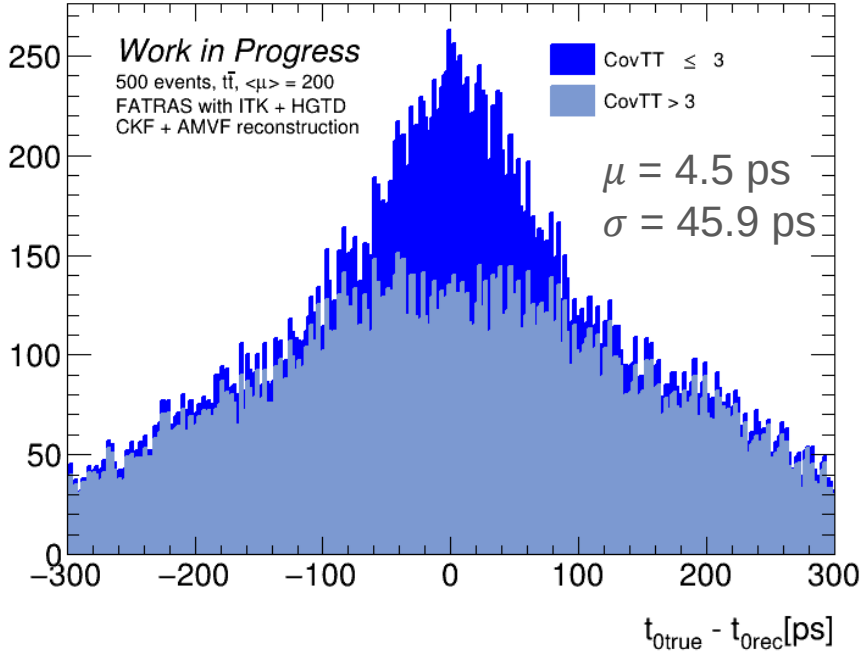
Reconstruction performance at 4000 fb^{-1}

0 fb^{-1}: 36.5 ps → 1000 fb^{-1}: 50.9 ps → 1001 fb^{-1}: 41.3 ps → 2000 fb^{-1}: 54.8 ps → 2001 fb^{-1}: 34.9 ps → 4000 fb^{-1}: 56.5 ps

inner ring replacement

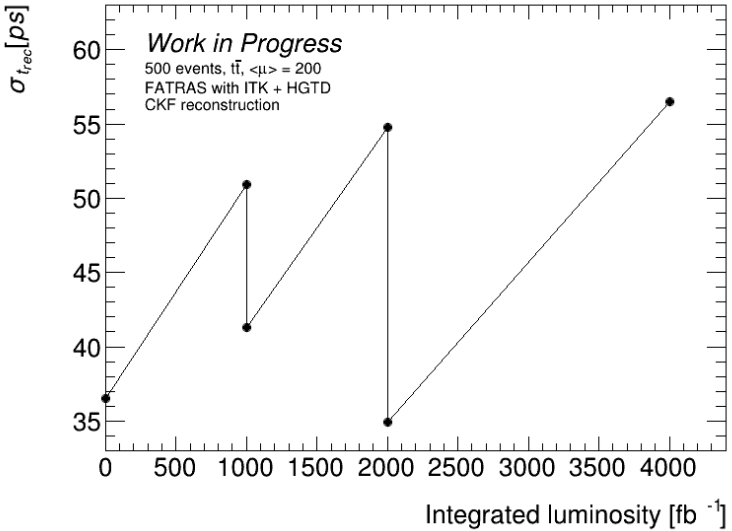
middle and inner ring replacement



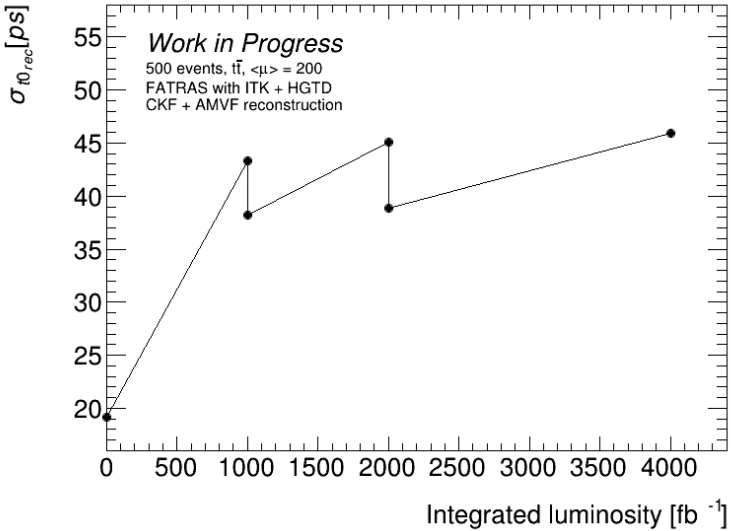


Time reconstruction x integrated luminosity

- Errors when performing the Gaussian fit to estimate the standard deviation can explain the “weird” results 2000 fb^{-1}
  - hit time residual curve is more non-Gaussian at 2000 fb^{-1}, but the fit estimates a lower std than the original one
- Replacing inner and middle layer at 2000 fb^{-1} has low impact on the t0 resolution
  - Need to better understand



hit time resolution



t0 resolution

Outlook

- We managed to include HGTD time information in the CKF track reconstruction
  - Still need to evaluate (and improve) track efficiency
- This reconstruction chain can be used to assert sensor performance in different phases of its lifetime
  - We could include the custom smearer (resolution vs int. lumi) in the main repository
    - could use an input file with a table resolution vs sensor position vs int. luminosity
    - or an analytical function (as it is implemented today)
- The time reconstruction performance of the CKF can also be used as a baseline for new methods

Thank you for your attention!

Questions?

References

[1] ATLAS Collaboration (2020). *Technical Design Report: A High-Granularity Timing Detector for the ATLAS Phase-II Upgrade* [White paper]. CERN.

[2] Gessinger-Befurt, Paul, 30.04.2021. *Development and improvement of track reconstruction software and search for disappearing tracks with the ATLAS experiment*. 10.25358/openscience-5901

[3] M. Gullstrand, and S.Maraš. “Using Graph Neural Networks for Track Classification and Time Determination of Primary Vertices in the ATLAS Experiment” (Dissertation). Available at: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-288505

[4] A. Garcia.ATLAS ITk project at Nikhef. Available at: https://wiki.nikhef.nl/atlas/ITk_Endcap_Homepage

[5] S. Malik. *Detector upgrade at Run3 and HL-LHC.* Available at: https://indico.cern.ch/event/783977/contributions/3467002/attachments/1893181/3123312/Malik_DMatLHC_DetectorUpgradeAtRun3andHL-LHC1.pdf

Backup

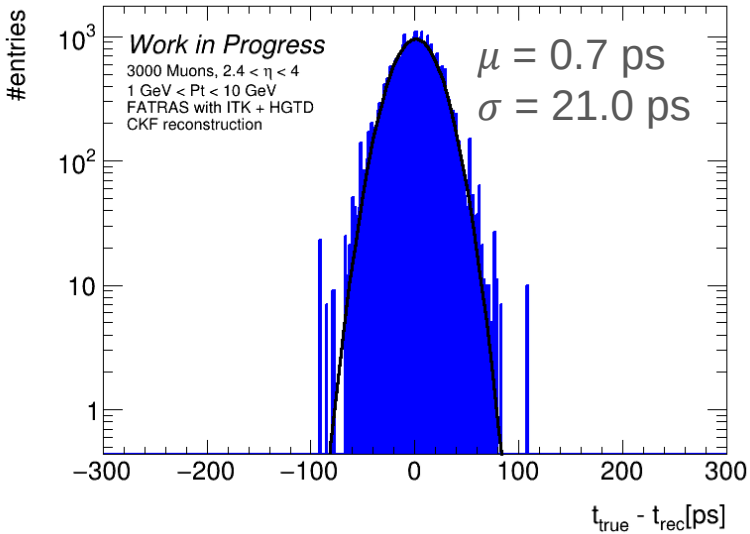
CKF Muon time reconstruction performance

Setup:

- Particle gun of single muon
- Direction $2.4 < \eta < 4.0$
- Primary vertex of following distribution:
  - z0 ~ N(0,50 mm), t0 ~ N(0,175ps)

Performance:

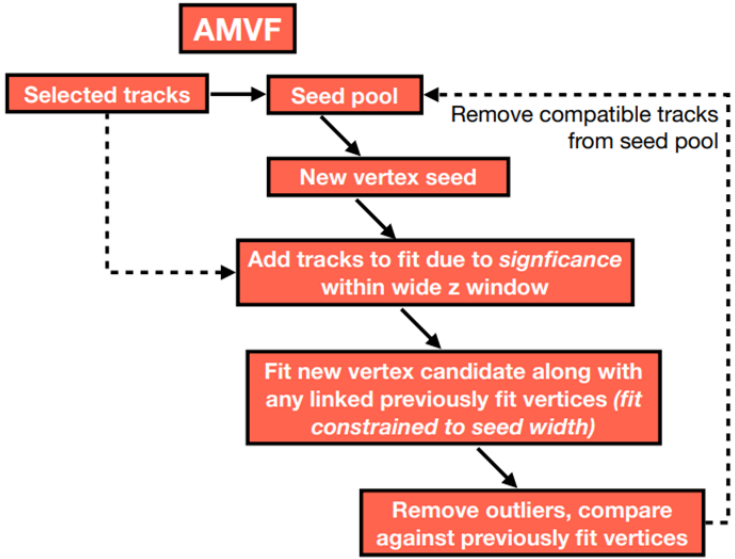
- Better resolution than ttbar events
- Distribution still shows slight deviation from zero
  - Even if smaller than ttbar events



AMVF Time reconstruction

AMVF overview

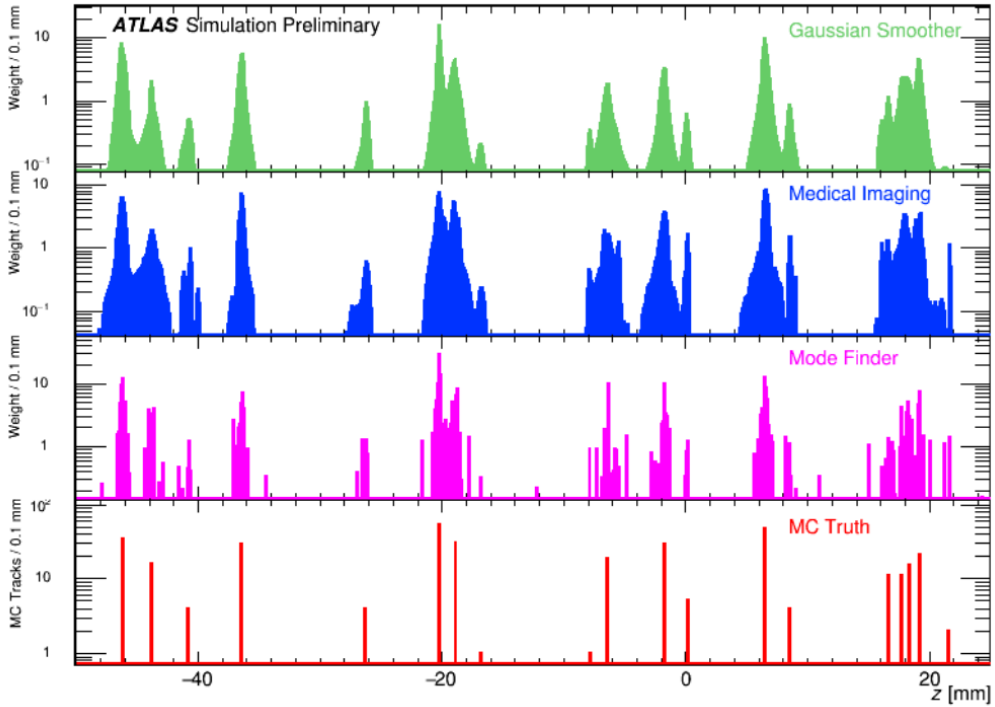
- I managed to include time information in the AMVF reconstruction
- The following slides will be an overview of the method in order to understand its performance
- The process can be divided into three steps:
  - Vertex seeding: Gaussian Track Density
  - Vertex finding: AMVF
  - Vertex fitting: Kalman Filter updater
- The steps are looped until all (valid) tracks are assigned to a vertex



- Reference for this section:
  - ATLAS Collaboration (2019). Development of ATLAS Primary Vertex Reconstruction for LHC Run 3 [White paper]. CERN.

Vertex seeder

- The seeding step establishes first estimates for primary vertices positions



- Most of the vertex seeders project tracks to origin and evaluate density distribution
  - The peaks of the distribution will be the seeds for vertices

Gaussian Track Density

- The density of track origin can be represented by a multi-variate Gaussian distribution:

$$P(r,z) = \frac{1}{2\pi\sqrt{|\Sigma|}} e^{-\frac{1}{2}((r-d_0),(z-z_0))^T \Sigma^{-1}((r-d_0),(z-z_0))}. \qquad \Sigma = \begin{pmatrix} \sigma^2(d_0) & \sigma(d_0,z_0) \\ \sigma(d_0,z_0) & \sigma^2(z_0) \end{pmatrix}.$$

- If we project it to the z-axis, we have the density at that axis. Furthermore, If we consider the distribution to be locally Gaussian, it's possible to do a peak search with steps of size:

$$W(z) = \sum_{i \in \text{tracks}} P_i(0,z). \qquad \Delta z = \frac{W(z)W'(z)}{W'^2(z) - W''(z)W(z)}.$$

- The output will be the position of the peak and the width of the distribution around it

$$z_{max} = \max_z W(z) \qquad \sigma(z) = \sqrt{-\frac{W(z_{\max})}{W''(z_{\max})}}$$

Adaptive Multi Vertex Finding (AMVF)

- Given a collection of reconstructed tracks and estimates of vertexes, establishes a “compatibility” value for each track-vertex
- The algorithm is adaptive in a sense that vertexes compete for the same track (multiple vertex-tracks weights)
- Iterate over association weights until convergence
- Short paper explaining

Fitting procedure

- Fit all vertexes using the assignment probability as track weights
- Recompute the assignment probabilities using the most recent vertex positions

Weight function

- Having n tracks to be fitted to m vertexes

The weight of vertex j to track i is:

$$w_{ij} = \frac{\exp(-\chi^2_{ij}/2\,T)}{\exp(-\chi^2_{\text{cut}}/2\,T) + \sum_{k=1}^{m} \exp(-\chi^2_{ik}/2\,T)}$$

T is a temperature parameter
X^2cut is a cut-off to suppress not assigned tracks
X^2ij is the chi2 distance between track and vertex

Kalman Filter Updater

- From the collection of tracks assigned to a vertex originated from the previous step, a Kalman Filter is used to fit the vertex position
- The position of the first deposition (measurement) is used to evaluate the vertex position and momentum of the track (vector state)
- The measurement equation would be:

$$\boldsymbol{q}_i = \boldsymbol{h}_i(\boldsymbol{v}, \boldsymbol{p}_i),\ i = 1 \ldots, n.$$

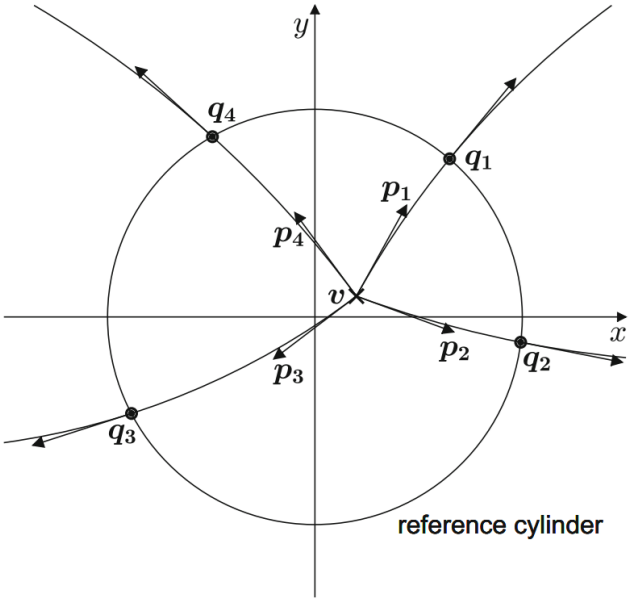


Fig. 8.1 A vertex fit with four tracks. The parameters of the fit are the vertex $\boldsymbol{v}$ and the momentum vectors $\boldsymbol{p}_i$; the observations are the estimated track parameters $\boldsymbol{q}_i$

Kalman Filter updater equations

- The measurement equation is linearized so the Kalman Filter can be used
- Check on the book for detailed explanation
- The AMVF weights multiply the inverse of the covariance matrix as to imitate a “significance” of the measurement

$$\boldsymbol{v}_i = \boldsymbol{C}_i \left[\boldsymbol{C}_{i-1}^{-1} \boldsymbol{v}_{i-1} + \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i^{\mathrm{B}} (\boldsymbol{q}_i - \boldsymbol{c}_i) \right],$$

$$\boldsymbol{p}_i = \boldsymbol{W}_i \boldsymbol{B}_i^{\mathsf{T}} \boldsymbol{G}_i (\boldsymbol{q}_i - \boldsymbol{c}_i - \boldsymbol{A}_i \boldsymbol{v}_i),$$

update equation for both vertex position and particle momentum

$$\mathrm{Var}\left[\boldsymbol{v}_i\right] = \boldsymbol{C}_i = \left(\boldsymbol{C}_{i-1}^{-1} + \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i^{\mathrm{B}} \boldsymbol{A}_i \right)^{-1},$$

$$\mathrm{Var}\left[\boldsymbol{p}_i\right] = \boldsymbol{W}_i + \boldsymbol{W}_i \boldsymbol{B}_i^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{A}_i \boldsymbol{C}_i \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{B}_i \boldsymbol{W}_i,$$

$$\mathrm{Cov}\left[\boldsymbol{v}_i, \boldsymbol{p}_i\right] = -\boldsymbol{C}_i \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{B}_i \boldsymbol{W}_i.$$

update equation for both vertex position and particle momentum

AMVF overview (again)

- The process can be divided into three steps:
  - Vertex seeding: Gaussian Track Density
  - Vertex finding: AMVF
  - Vertex fitting: Kalman Filter updater
- The steps are looped until all (valid) tracks are assigned to a vertex