# Kalman Filter inclusion in hps-java

PF, Robert

03/01/2020

# Introduction

- Robert’s Kalman Filter (KF) package merged in master branch
- Active branches on KF development:
  - iss659d : integration of KF in reconstruction pipeline
  - iss204e : original development branch - **now merged in iss659d**
- Status:
  - Robert’s KF has been integrated in the reconstruction pipeline
    - KalmanFullTracks are formed from StripClusters
    - TrackData object and relations (time and volume, **no isolation yet**) are formed
    - Extension to calorimeter is computed using TrackUtils + RK (from last hit in SVT)
    - Used tracks to form electrons and positrons and vertices
      - Disabled checks on Calo matching
    - Interfaced to hpstr

# Samples and track selection

- 2016 Tridents LCIOs:
  - location: /nfs/slac/g/hps3/mc/mc_2016/tri_trig/
- Steering file to run KF pattern reco + V0s from LCIO:
  - /nfs/slac/g/hps2/pbutti/kalman/hps-java/PhysicsRun2016FullReconMC.lcsim
- Track selection:

  - p_ele <1.75 GeV

  - ele/pos trkX2 < 6

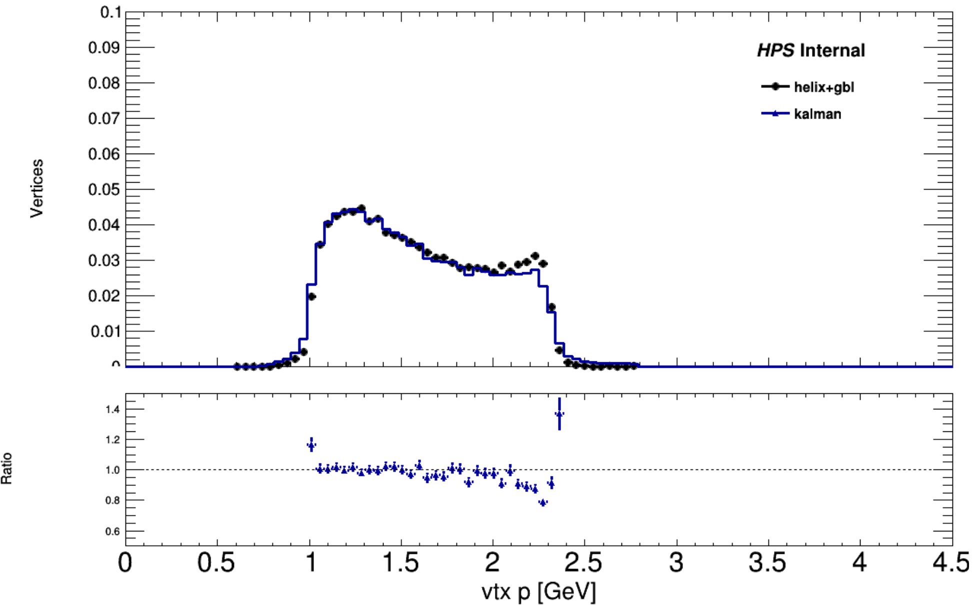
  - chi2 unc vtx < 20

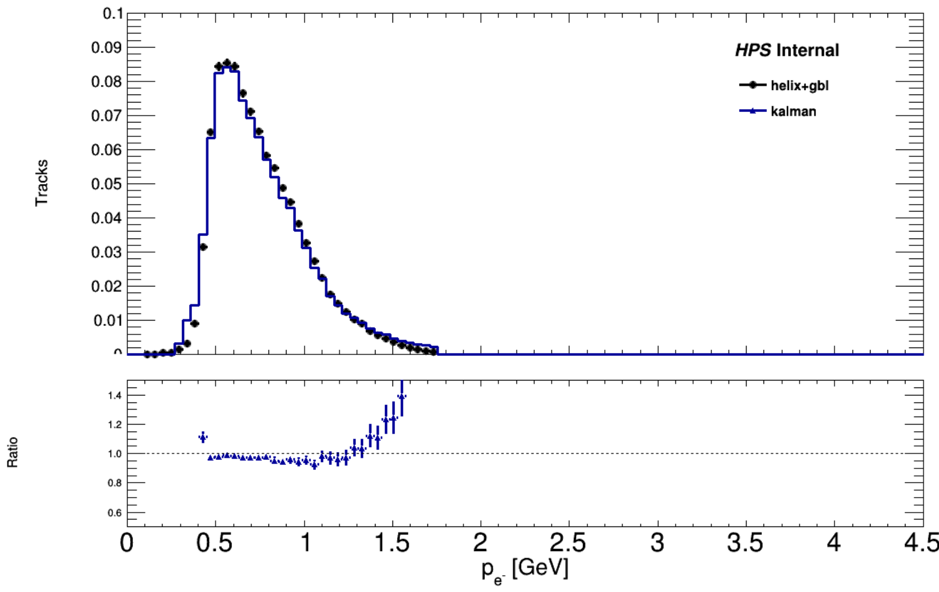
  - p_ele > 0.4

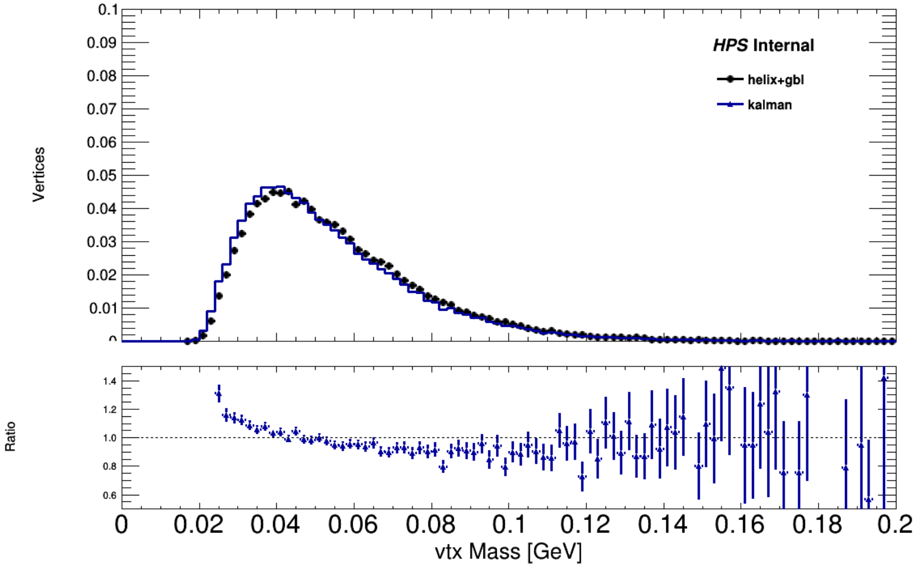
  - p_pos > 0.4

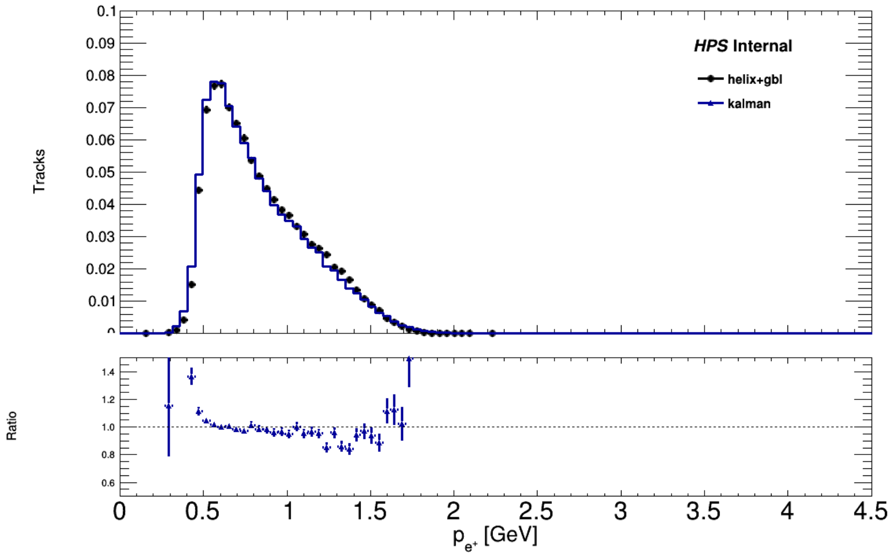
- Next slides show some track/vtx comparison from what comes out of reconstruction
- **KF and Helix+GBL follow different pattern reco procedures and cuts: timing and number of hits cuts aren’t harmonized yet**

# Vtxs comparison

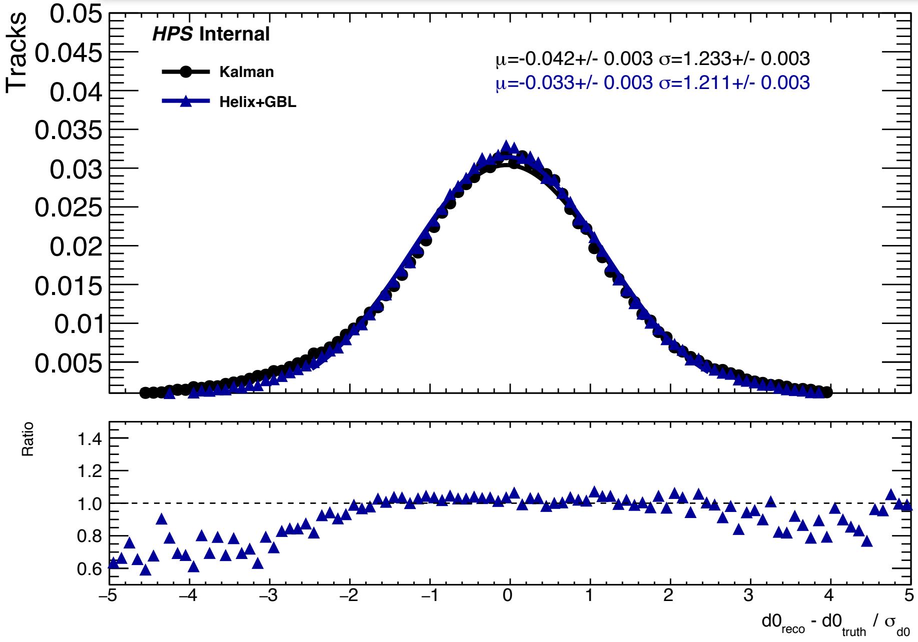


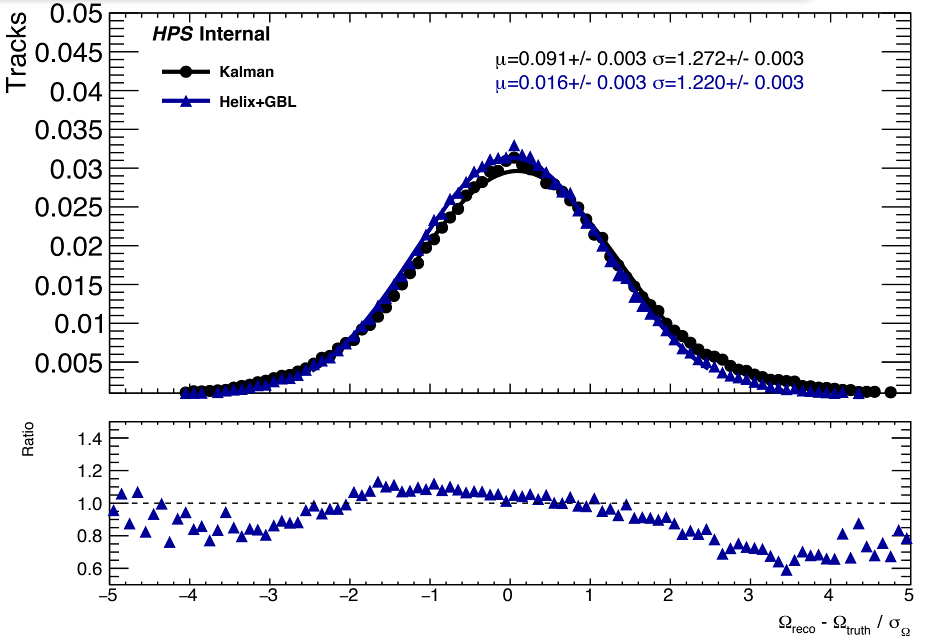


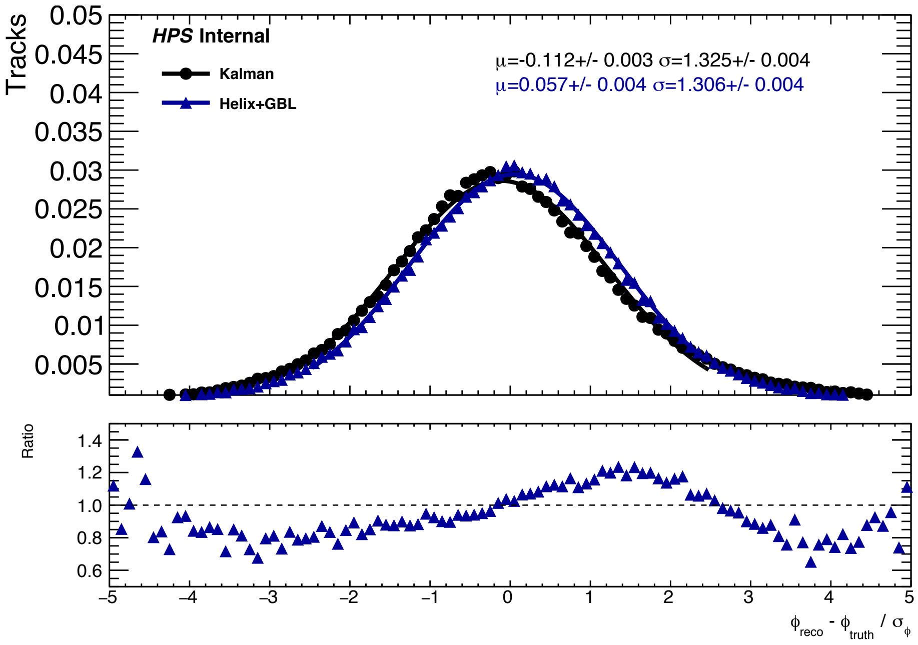


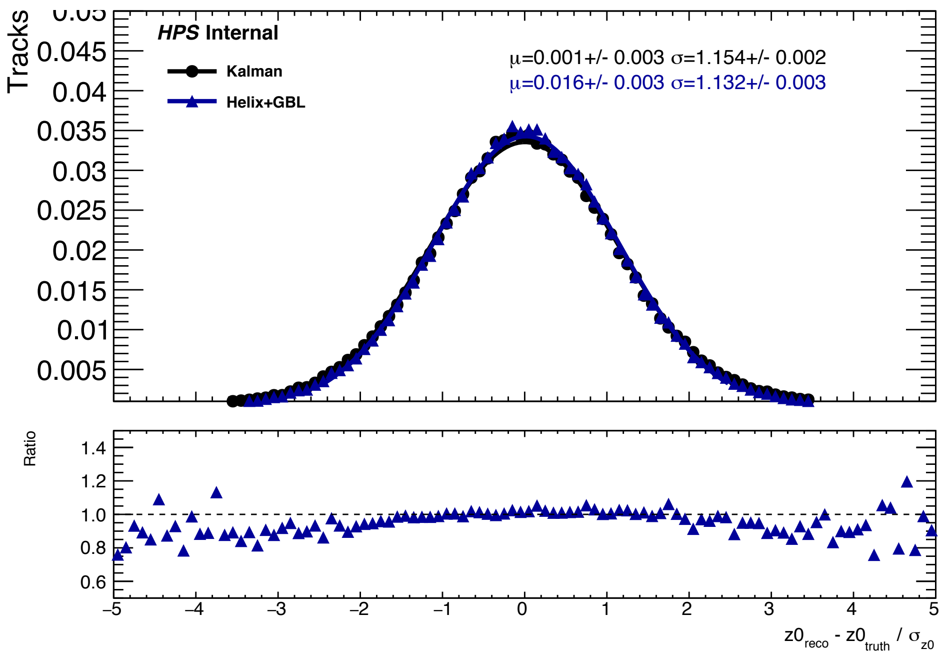


# Pulls wrt Truth Tracks (checking MCParticle with max number of hits on track)

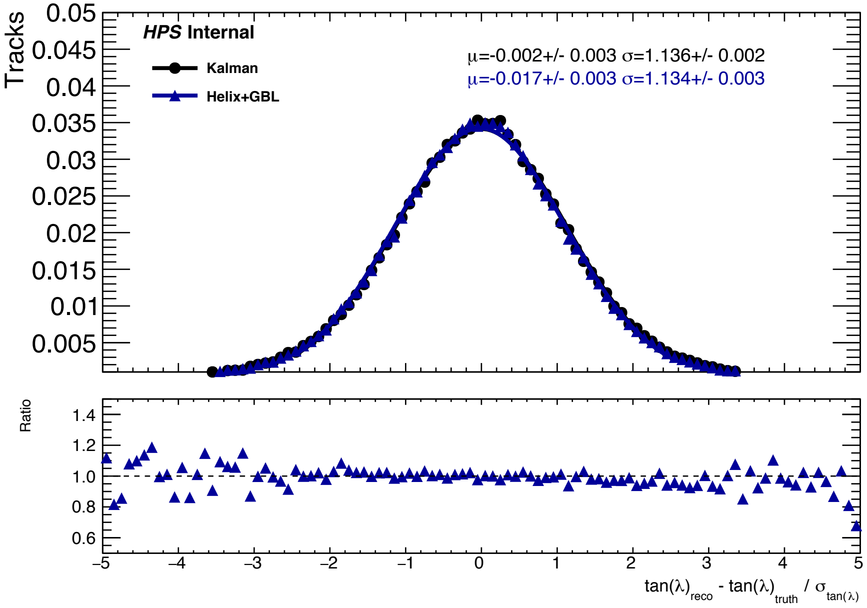


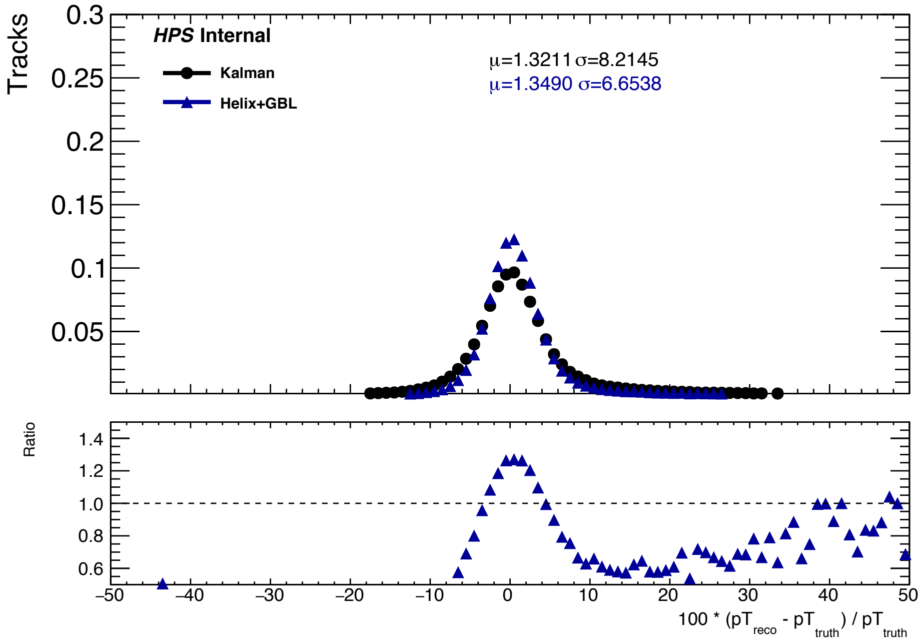




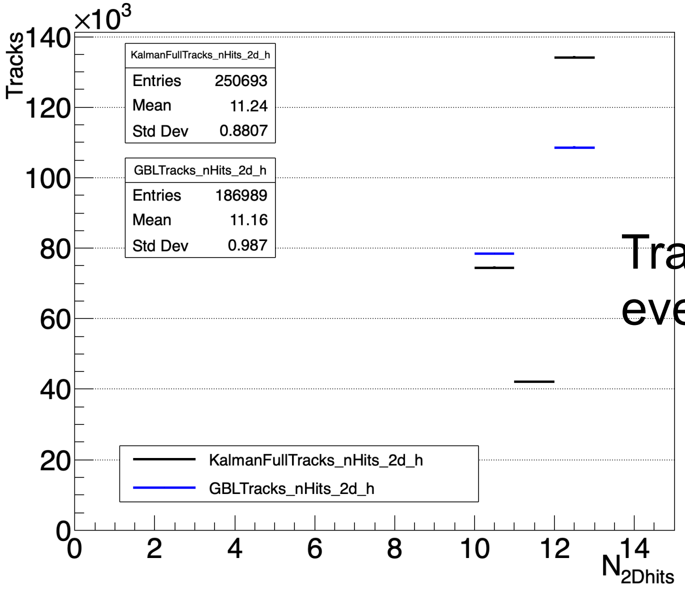


# Pulls comparison wrt Matched Truth





Slightly worse pulls in general in KF. Phi and Omega, mostly differ. Momentum ~1.3% off wrt truth track momentum. Slightly larger spread in KF.
However still to check if all quality cuts are the same.
Large gain in efficiency overall.



Tracks from same events

# Current Issues and to-do

- Current issues in particle reco:
  - Vertices have negative X2:
    - Investigating on this issue
- Crosscheck the truth matching for Kalman Tracks
- Add interface to GBL

# Open Pull Requests

- iss659d
  - This brings KF into hps-java reco chain in master.
  - Still some clean-up and developments to do but version runs on MC, provides vertices + final state particles + truth matching
  - Doesn't affect current reco
    - Only one change in V0s formation: top/bottom tracks are selected by checking tanLambda instead location of first hit [discussed and approved here at SLAC already]
    - For the rest completely orthogonal
- iss669
  - This adds the track momentum to the track Data structure
  - In principle not necessary but useful
    - Avoids recomputing track momentum in hpstr (bfield in input needed => error prone)
    - Can fix issue when forming FinalStateParticles momentum out of reconstructed lcio files [currently TrackStates.getMomentum() is called but not stored => (0.,0.,0.) momentum for ele/pos if tracking is not re-run