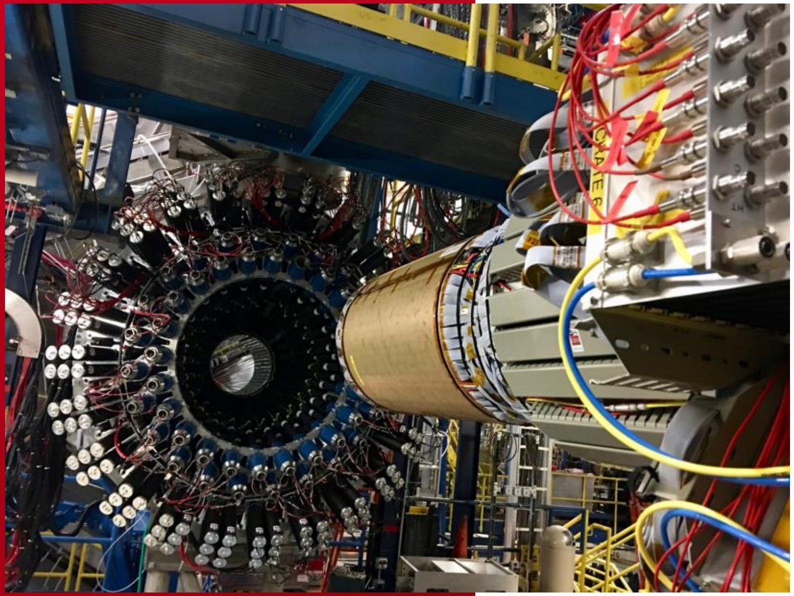
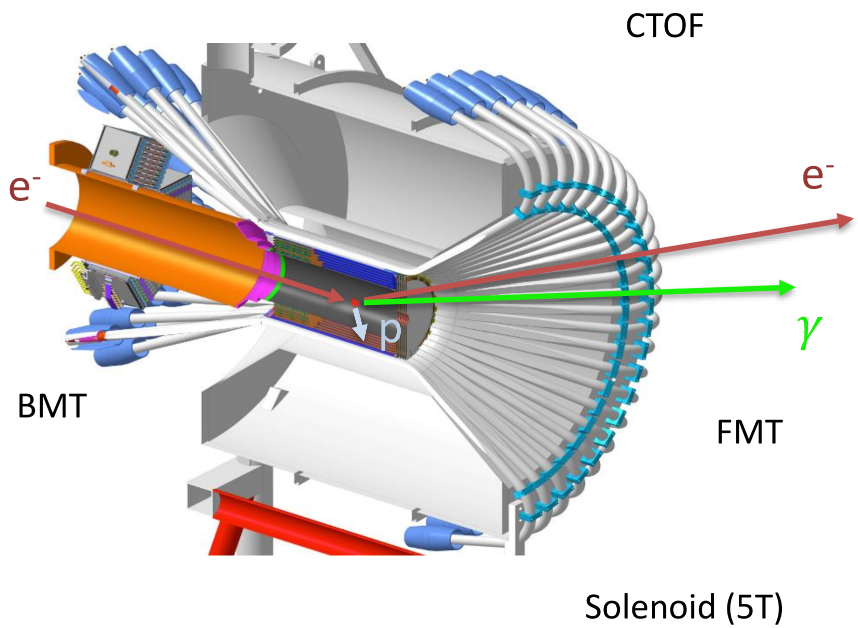
DE LA RECHERCHE À L'INDUSTRIE

CEA Saclay presents:

Central Tracking status For CLAS12

Francesco Bossu
Maxime Defurne





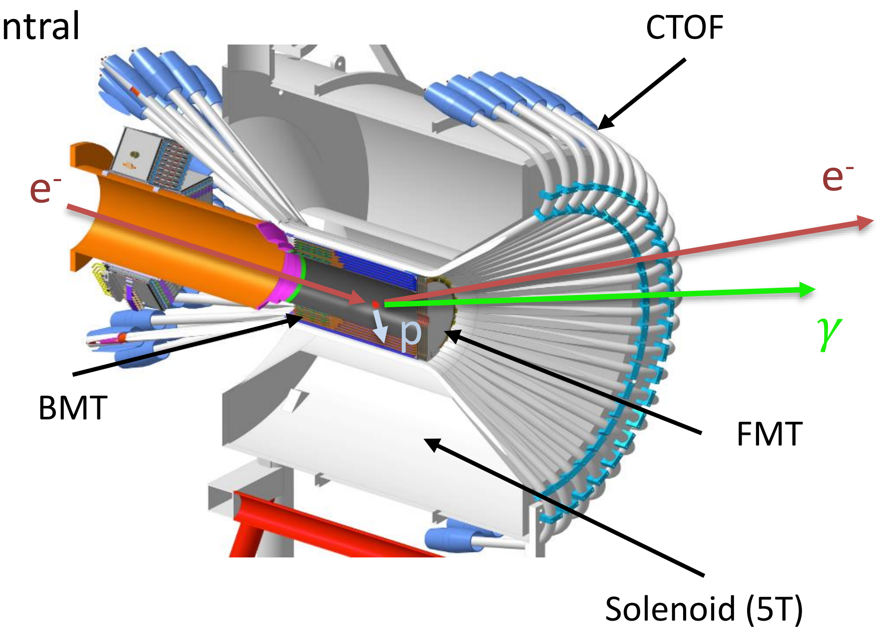
www.cea.fr

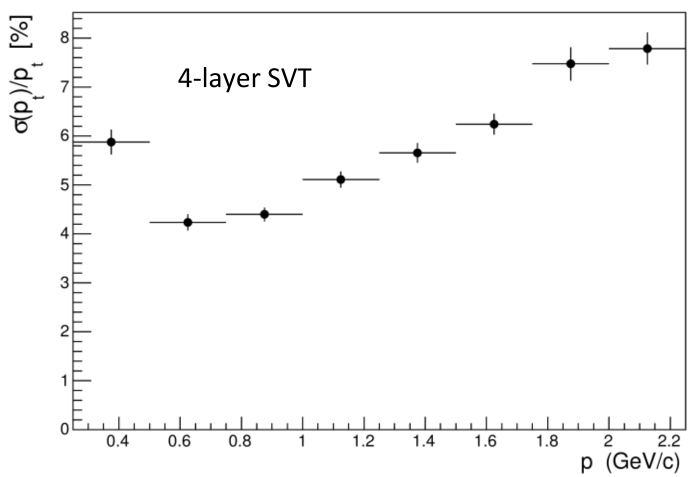
A central vertex tracker for CLAS12

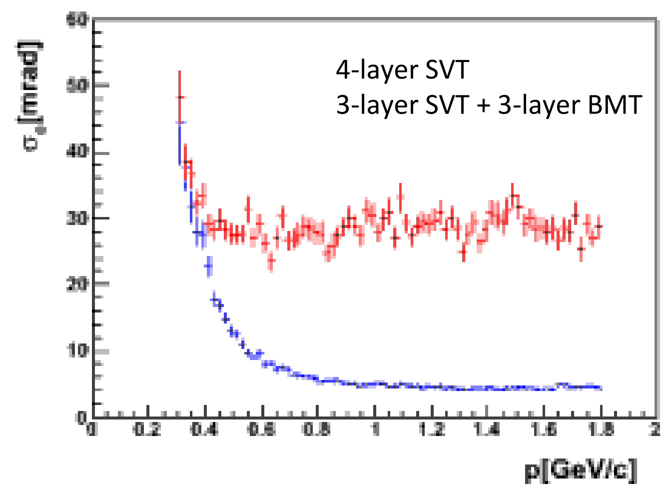
- In the 5T magnetc field of the solenoid, there is a central vertex tracker made of Micromegas and Silicon dectectors.

- The central vertex tracker must ensure:
  - 5% resolution on momentum.
  - 5 mrad in azimuthal angle.
  - 5 mrad in polar angle.
  - vx and vy resolution at 500 um.

- A central Time-of-flight completes the tracker for particle identification.



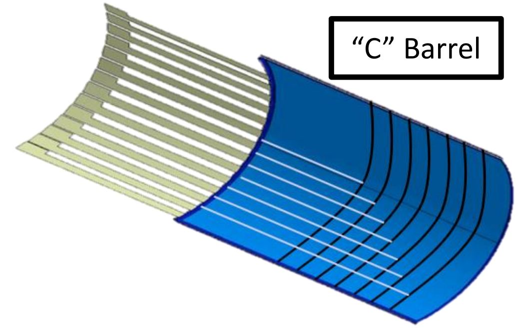




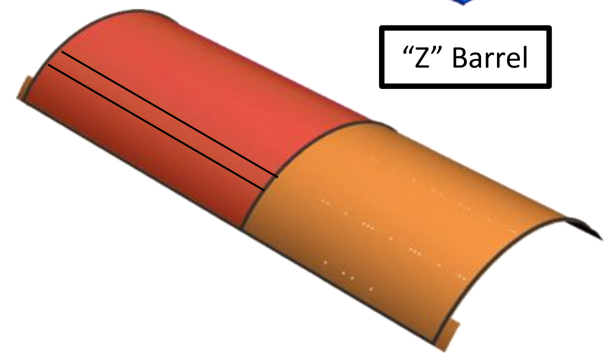
CVT description

- Barrel Micromegas tracker is made of 6 layers with **radii from 140 mm to 225 mm**.
- Each layer is made of 3 tiles covering approximately 120 degrees.
- There are two kind of tiles for the **azimuthal ("Z") and polar ("C") angle**.
- **Z-tiles** have a constant pitch of about **500 um**.
  C-tiles have a varying **pitch from 330 to 600um**.
- The Silicon Vertex Tracker is made of **three double layers**.
- Four regions were made but three are currently used in CVT.
- Strips on bottom and top bottom have a **stereo angle**.
- Spatial resolution is expected to be **50um.**
  (although current GEMC simulation gives 30um)

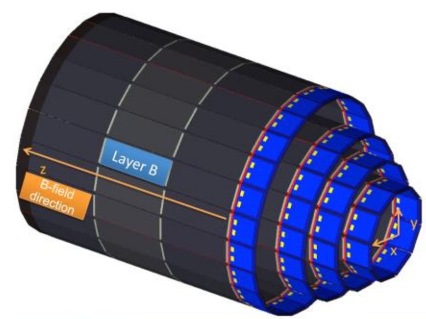
CVT = A TOTAL OF 92 ELEMENTS with 3 different geometries



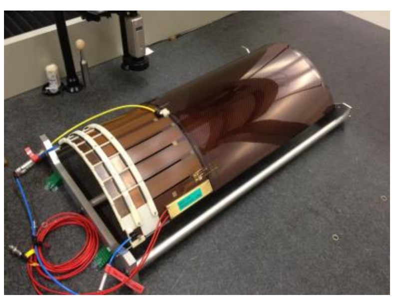
"C" Barrel

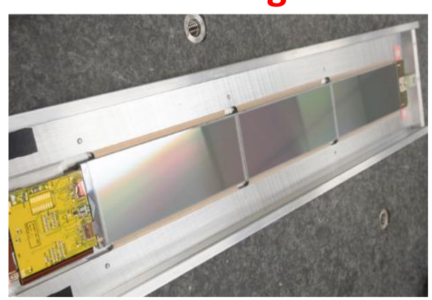


"Z" Barrel



LAYER A | LAYER B | Intersection of Strips

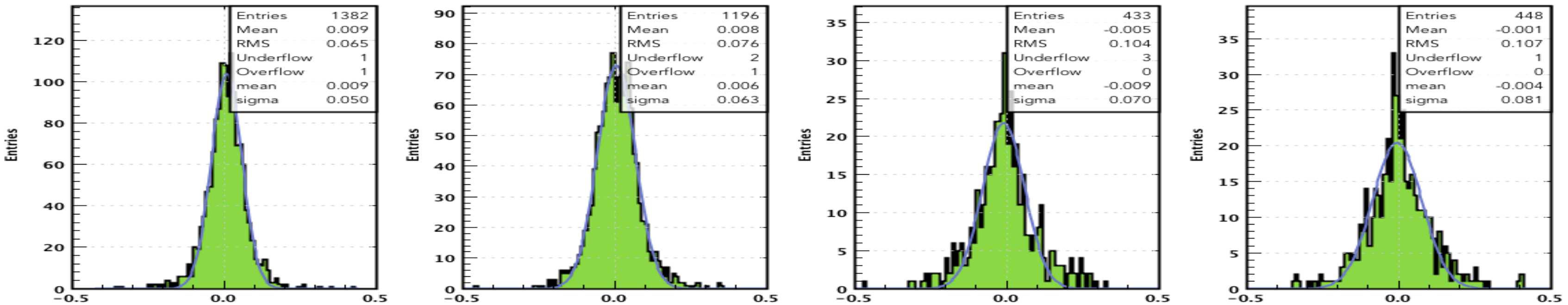




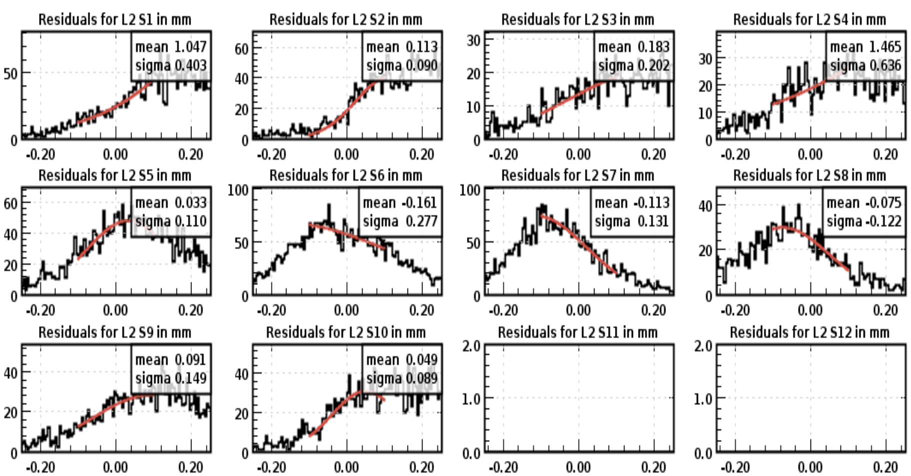
Summary of last meetings: Alignment done

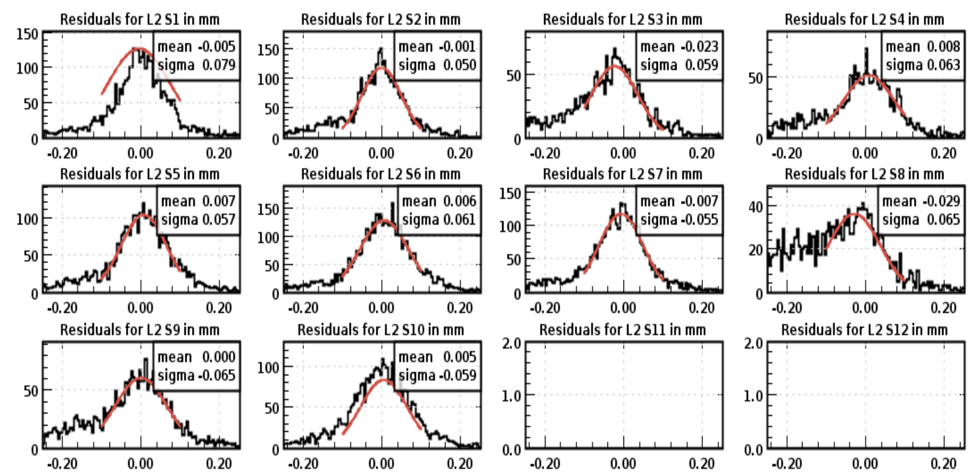
- Tracker.jar: A fully validated tracking code for straight track (cosmics and no-field beam data).
- Alignator.jar: A script finding best rotations and translations of SVT modules and MVT tiles, using Tracker.jar.

SVT residuals with ideal simulation



- The codes were run on Spring 18 data.

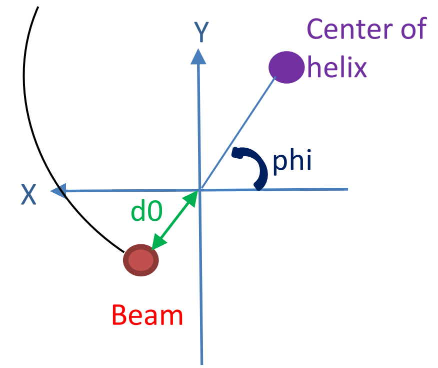




- CCDB tables have been created to store the constants… But current COATJAVA CVT codes is not designed to handle misalignment corrections.

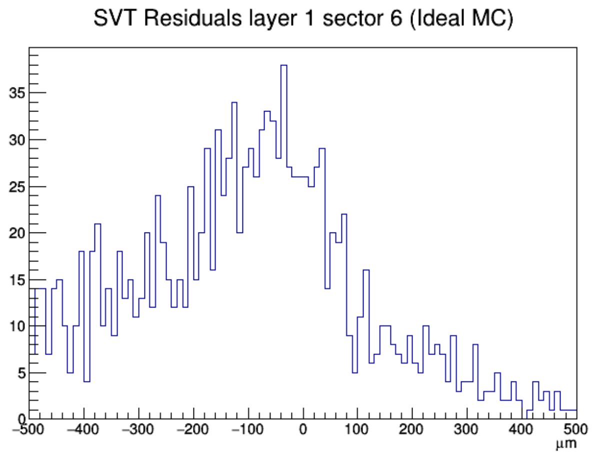
Track parameterization and prerequisites

- First, let's introduce the track parameters:
  - d0 = distance of closest approach to ref point in XY-plane
  - phi = angular position of helix center wrt to ref point
  - z = distance of closest approach to ref point along z-axis
  - tandip = tan(Pz/Pt)
  - kappa=Q/Pt



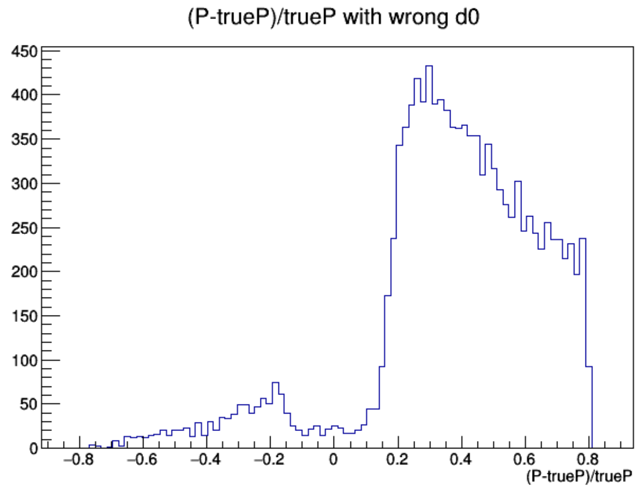
This ref point is (0,0,0) by default. It will be beam position once calibrated.

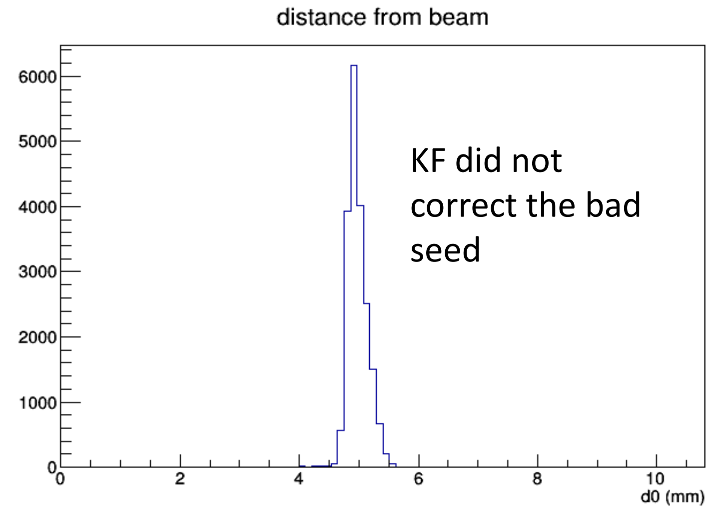
- With current tracking code, residuals look weird with ideal simulation:
  Not centered at 0 with an asymmetric shape => Must be corrected prior alignment work.



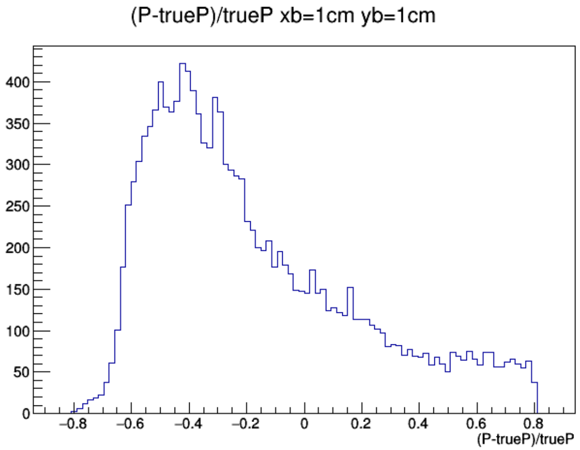
- Kalman filter must be resilient to bad seeding (First guess of track parameters).

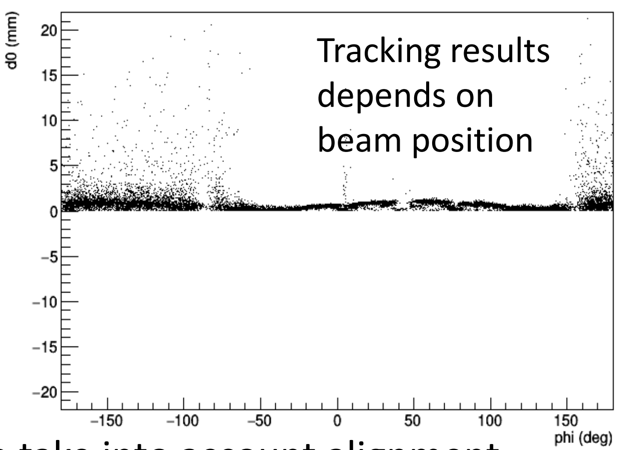
- **First test**: Shifting by -5 mm d0 of the seed.
  Here it is ideal simulation (vertex at x=0 and y=0)… so d0 must be 0 after Kalman filter.





- **Second test**: Resilience of tracking code with wrong beam position (After all, you do not know the beam position before doing tracking.)

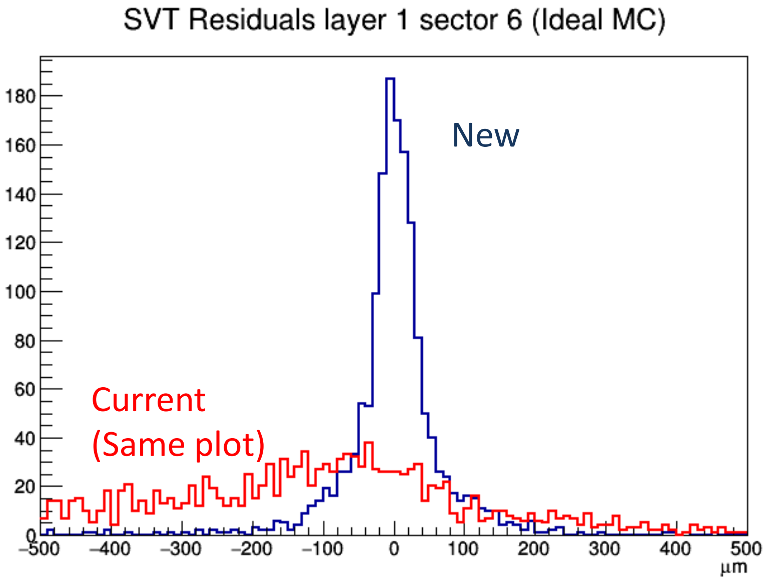




- **Conclusion:** Tracking algorithm not flexible enough to take into account alignment.

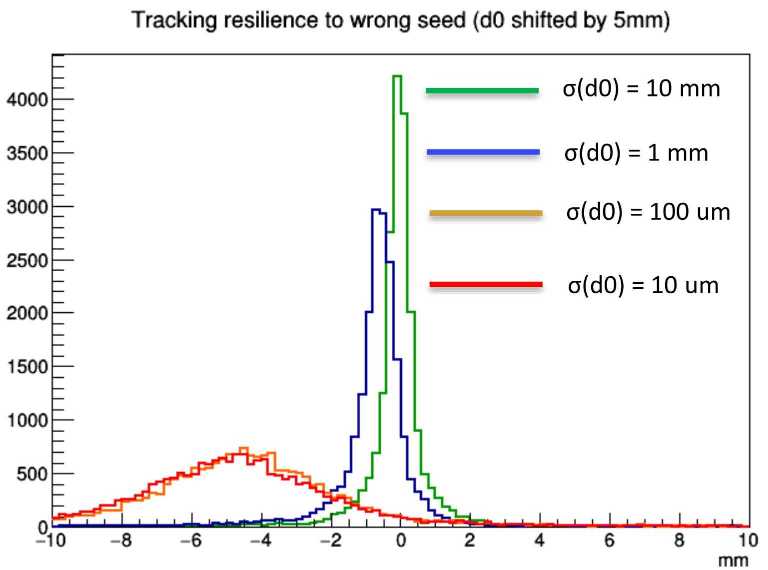
Improvements of actual tracking code

- For the last months, we worked on improving the tracking code:
  - Now, on ideal simulation, all residuals are as expected:
  Centered at 0 with accuracy between 30-50 um.

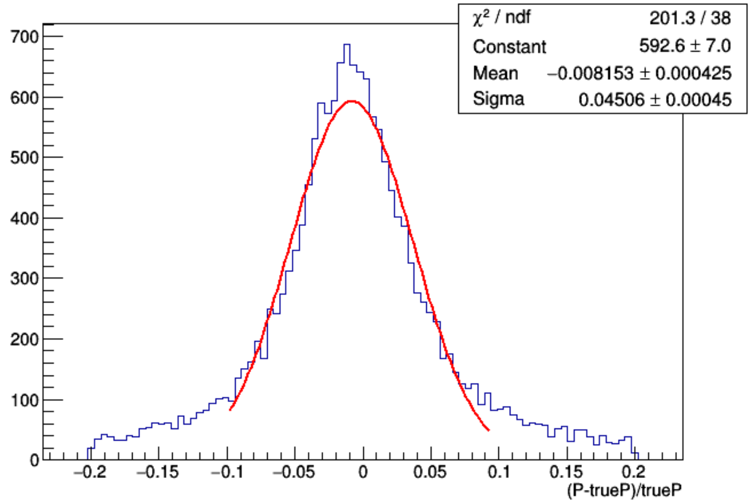


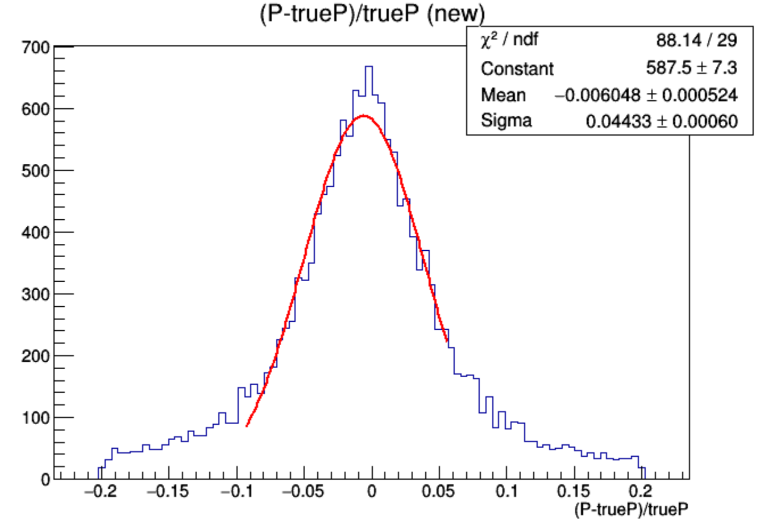
- The new tracking algorithm is resilient to un-accurate seeding.

  On ideal simulation, with beam at (0,0), d0 must be found at 0.
  Here, by hand, d0 is shifted by -5 m before KF:
  - KF recovers as long as uncertainties on initial parameters are reasonable (not too small).

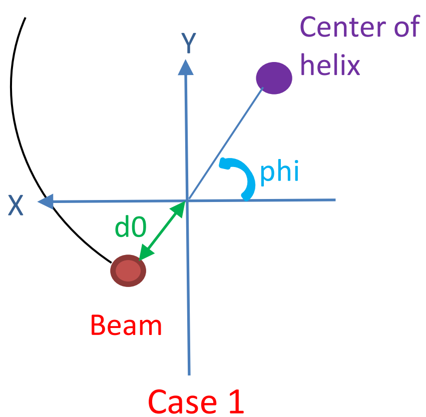


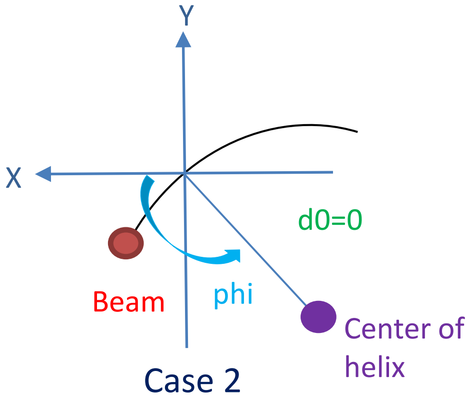
- Tracking results does not depend anymore on beam position:
  - One plot is with shifted beam (x = 1cm , y=-1 cm), the other is with beam at (0,0).
  Guess which is which?

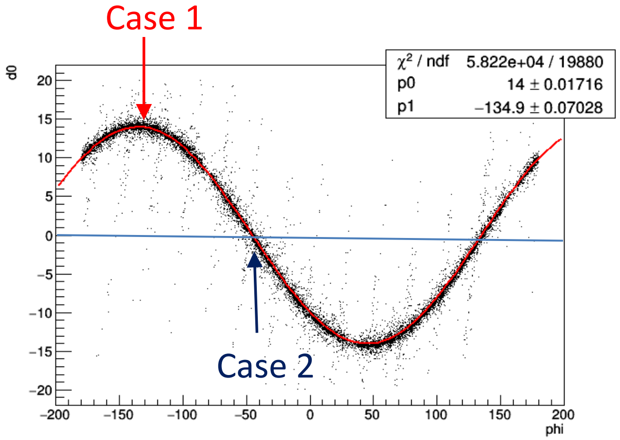




- We can even locate the beam position to refine vertex determination:
  d0 = p0*Sin(phi+p1) with p0 = distance of beam to (0,0) and p1 = angular position of beam

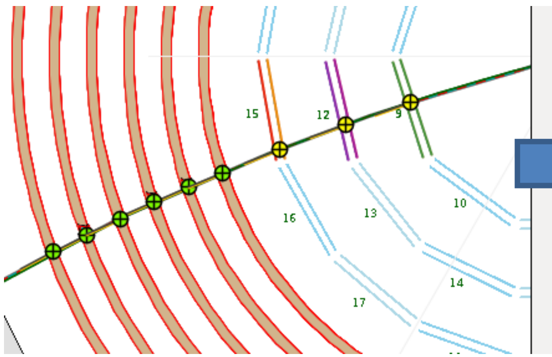




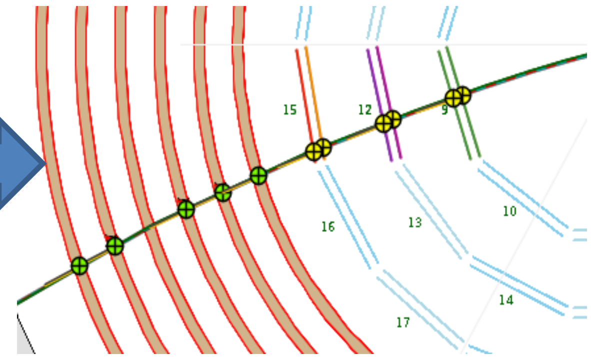


Ongoing work

- All these improvements revealed imperfections in the pattern recognition algorithm:
  - SVT crosses not always associated to track.
  - Bad clusters from background are quite often associated to a track, degrading the performances of the code.
- Doubling SVT information for pattern recognition: from one cross per region to one cross per layer.

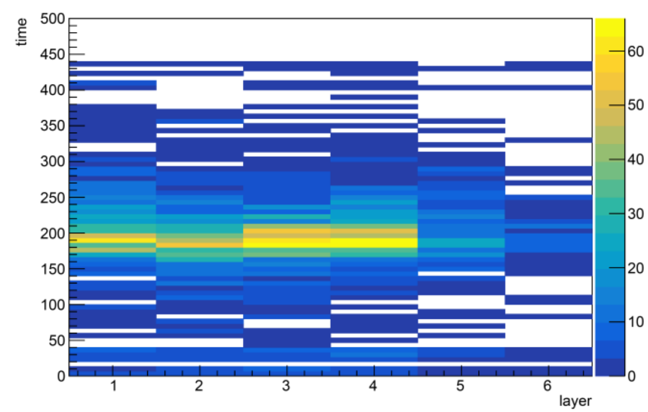


One cross per region



One cross per layer

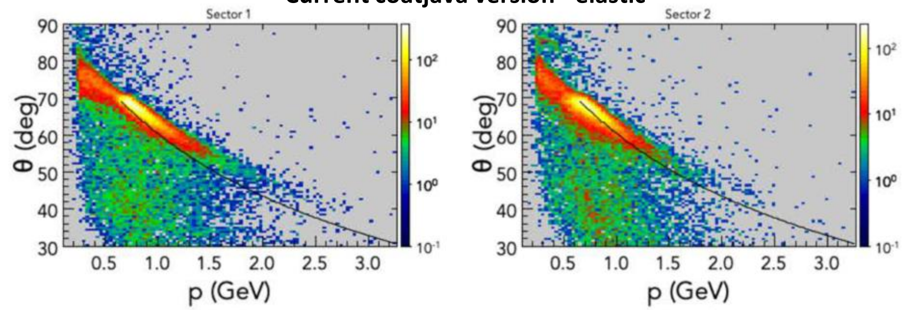
- Time information of Micromegas will be used to associate BMT-C and BMT-Z tiles:
  => Cleaner seeds



Conclusion

- Stress tests showed rigidity and biases in the current code, although they are extreme cases.
- Recent cookings with current code showed some “physics”, which is good enough for preliminary studies but not for publication.

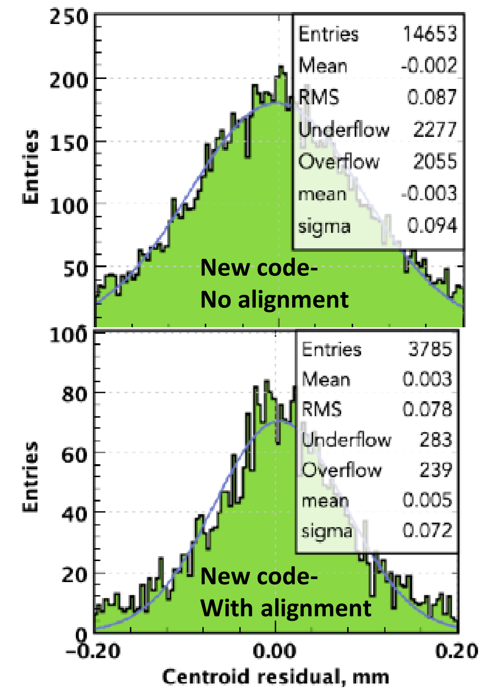
Current coatjava version - elastic



- Recent developments triggered because of incompatibilities between existing code and alignment corrections.
- Kalman filter has been greatly improved over the last months:
  - Resilient to bad seeding,
  - Resilient to beam position uncertainty,
  - Ready to use alignment constant

  (Encouraging results but need validation).

  => Able to handle the extreme cases of stress tests, useful for secondary vertices for instance.



- Ongoing work to increase efficiency (At 50 nA, 40%):
  - Pattern recognition improvement to increase efficiency of new tracking code.
  - Implementation of track-following algorithm to increase resilience to pattern reco mistake.

Kalman filter

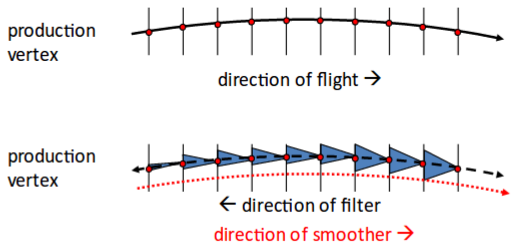
- Kalman Filter (KF) is not a global fitting method (distance minimization to all points at once) but a best estimator of state vector (position+momentum).

- First you start with a “guess”, the seed, of position and momentum of your particle and corresponding uncertainties at a given measurement site k.

- Based on this guess, you do a prediction of the next measurement k+1 and compare it to real next measurement (residual).

- Depending on accuracy of the prediction and actual measurement, a correction is applied to position and momentum of particle predicted at site k+1 from k-state vector. Corresponding uncertainties is also reduced with the information gain from measurement k+1 => Then you repeat process until last measurement.

- For CVT, you have the best knowledge of momentum and position of the particle at the very last layer you consider in the Kalman Filter… then you swim back to the vertex.



- Residual is the key quantity to quantify the goodness of a tracking algorithm.
Good accuracy => Need for alignment.