10.11.2020

Track reconstruction updates
for Muon Collider

Conformal Tracking code developments

N. Bartosik

INFN Torino

Conformal Tracking: current state

Official Conformal Tracking package is not in perfect state

- contains some FIXMEs *(latest actual code update in Sep 2019)*
- crashes in certain configurations *(critical for Muon Collider)*
- has room for optimisations

Forked version of Conformal Tracking maintained in our repository

A lot of debugging information is used at current stage of studies, which needed some optimisation

- to understand where most of the CPU time is spent
- to decide which cuts to use at each step of pattern recognition

Debug plots: not with BIB

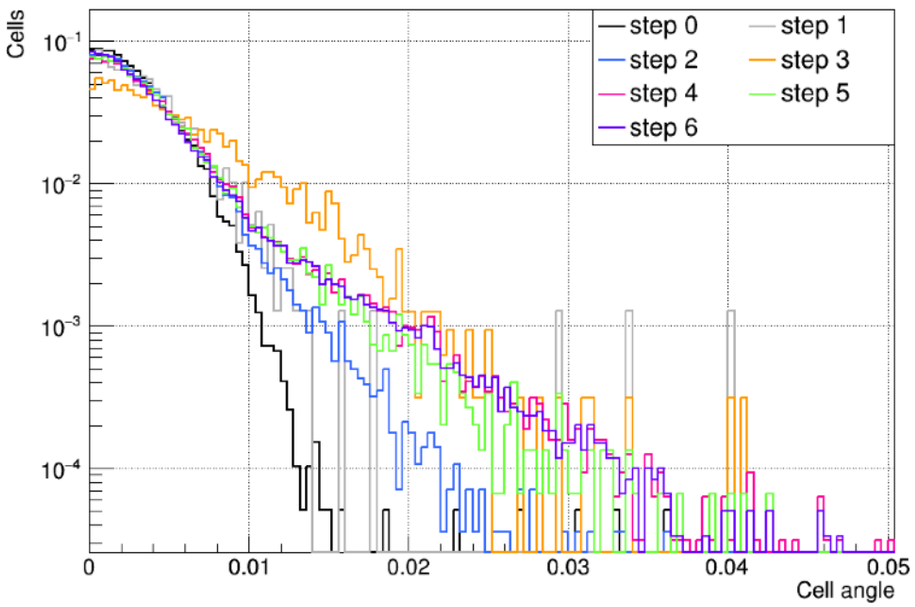
Debug plots production

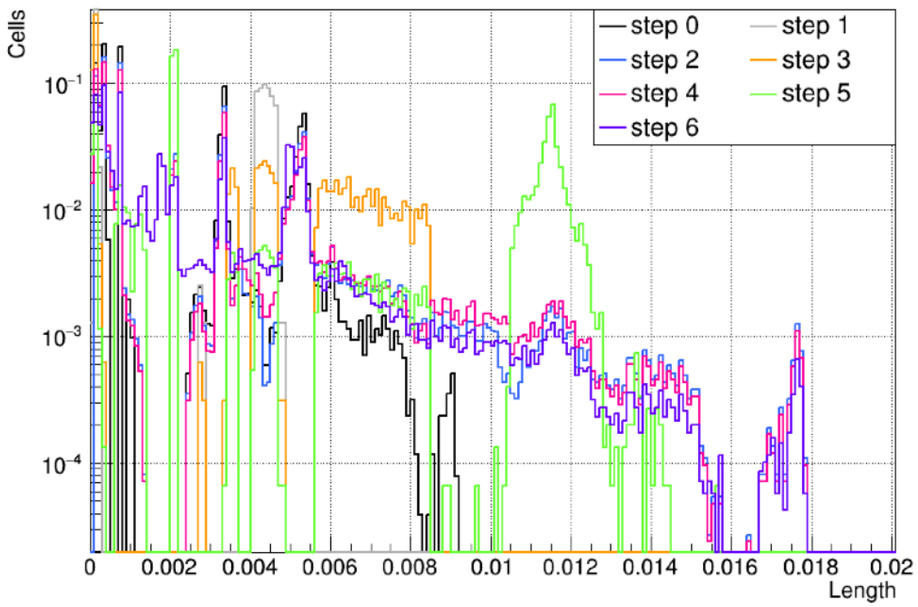
```
<!--enable debug plots -->
<parameter name="DebugPlots" type="bool"> true </parameter>
```

- additional printouts and plots with Conformal Track parameters
  ↳ assumes valid **RecHit → SimHit → MCParticle** relation for every hit
    ↳ crashes with BIB *(we are not storing BIB MCParticles)*
      this is not yet fixed *(don't use **DebugPlots** with BIB for now)*
- only 1 plot for each track parameter is created and filled by each MCParticle
  ↳ we need to tune conformal track cuts for each step individually

Sets of histograms created for each parameter at each step configured in XML

- run with relaxed step parameters on a **single μ±** sample without BIB





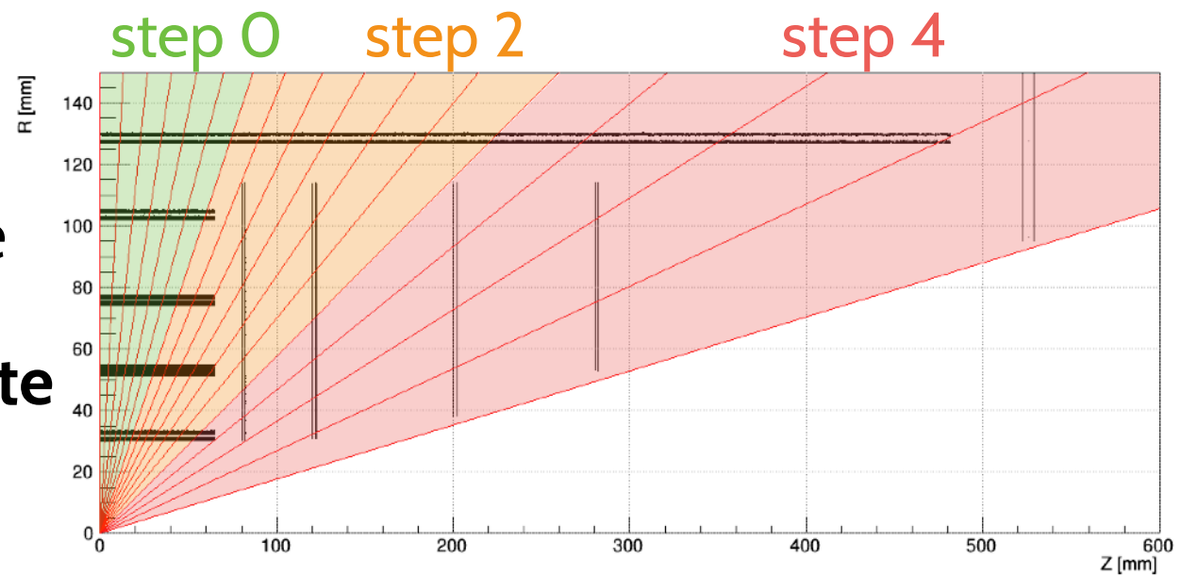
easy to define cuts for each step

Time monitoring: bigger picture

Time monitoring

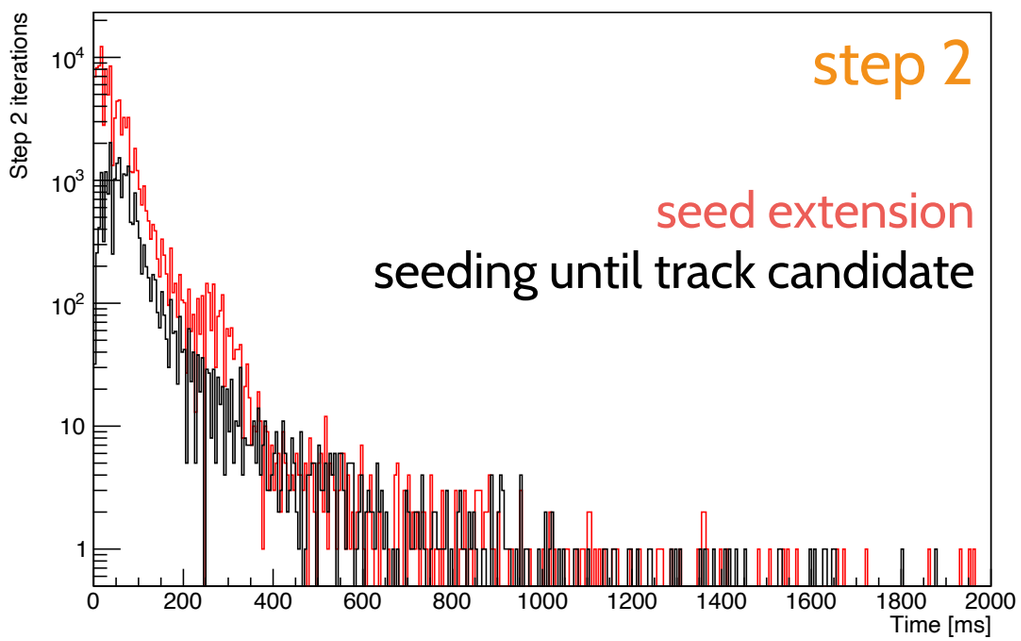
```
<!--enable debug timing -->
<parameter name="DebugTiming" type="bool"> true </parameter>
```

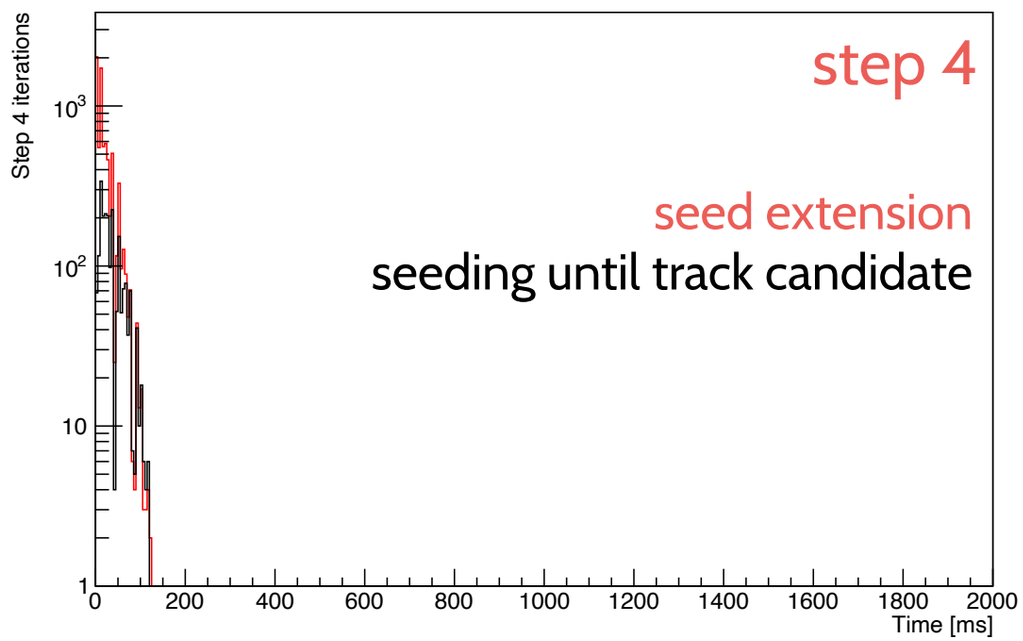
- prints **real** time used by each operation during track reconstruction
  ↳ no easy way of seeing a big picture of where most time is spent
- sets of timing histograms were added for **each operation** of **each step**
- filled with **CPU** time for real performance



Only ~25% of seeds lead to a track candidate

- most time spent on seed cell extension to the neighbouring hits

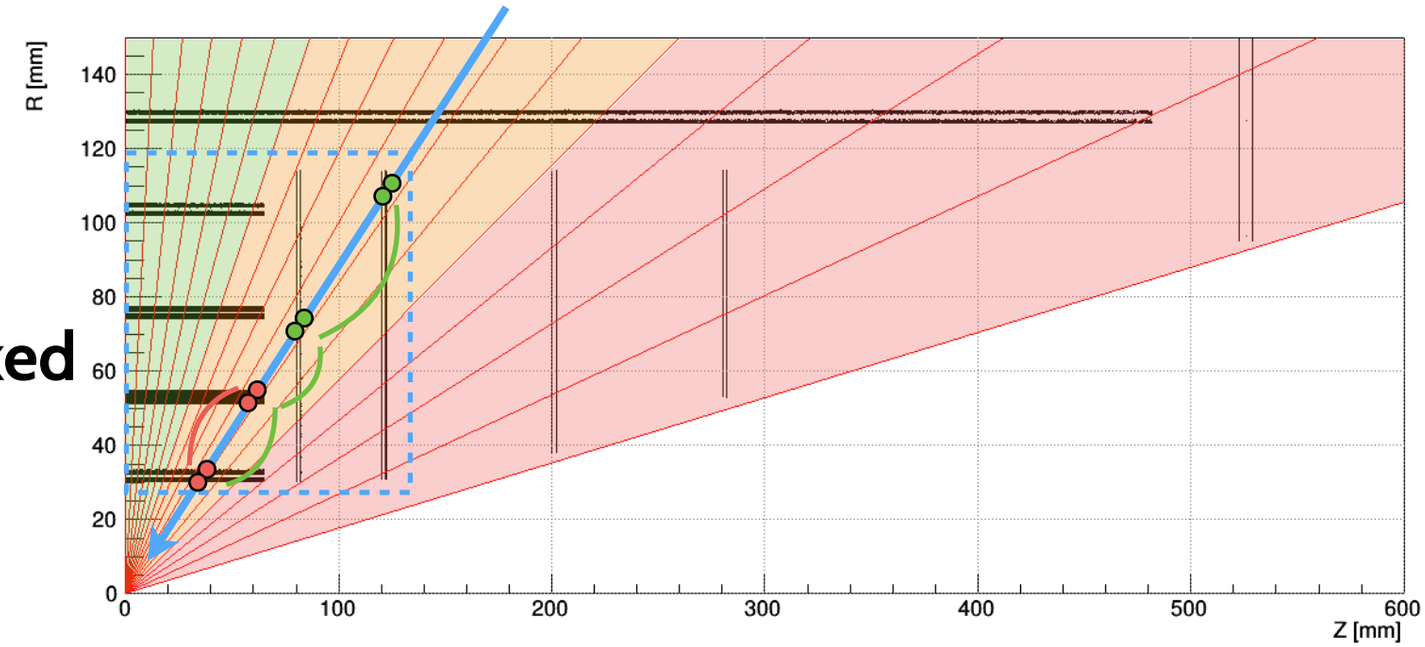




Seeding logics: performance improvement

Original CT algorithm uses every single hit in the input collection as a seed

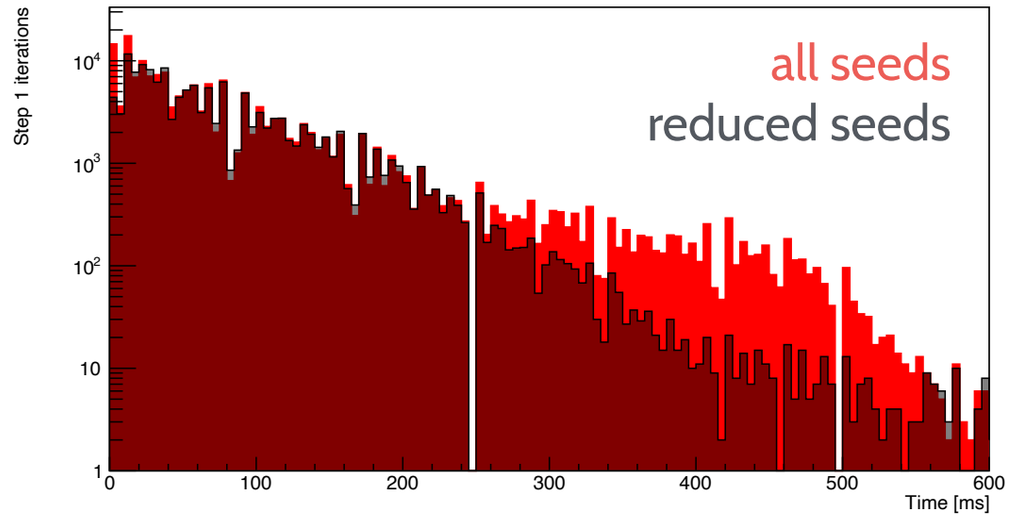
- hits at certain layers can't be seeds for good track candidates
  e.g. with a minimum N_{hits} = 5



Now input collections can be marked with "-" to not be used as seeds

- but will be used for neighbour search and cell extensions like before

```
[VXDMiddle]
@Collections : -VXDBTrackerHitsInner, -VXDBTrackerHitsMiddle,
VXDBTrackerHitsOuter, VXDETrackerHitsInner,
VXDETrackerHitsMiddle
@Parameters : MaxCellAngle : 0.015; MaxCellAngleRZ : 0.014;
Chi2Cut : 60; MinClustersOnTrack : 4; MaxDistance : 0.02;
SlopeZRange: 3.0; HighPTCut: 1.0;
@Flags : HighPTFit
@Functions : CombineCollections, BuildNewTracks
```



Tracking speed in 1 event with 50% of BIB improved by 20% (4.6 h → 3.8 h)

- with 30% of BIB improvement is 12% (16 m → 14 m)
- even greater effect expected with 100% of BIB

Summary

A few improvements to the Conformal Tracking code have been implemented

1. aggregated view of the computation time in each step
2. clear view of the pattern recognition parameters to be used in each step
3. improved performance of track seeding

In the current state the code can be used for more detailed studies and parameter tuning for less-perfect scenarios: electrons, displaced tracks, etc.