Track finding at the EIC

Yue Shi Lai

LBNL NSD

UC Consortium Meeting, Jul 19 2022

Seeding for EIC

- Essential ingredient of a tracking pipeline
- Even in detector study, needed for realistic performance
  - Efficiency, fake rate, multiple reconstruction, etc.
  - Beam condition (synchrotron radiation, beam gas interaction)
- Code now included in DD4hep/Juggler grew out of ATHENA
  - Had its share of ACTS issues and bugs due to the high field strength and tracker size
  - Only recently discovered all ACTS issues resulting in a Juggler plugin with the needed workarounds
  - Now works for efficiency studies with $p > 100$ MeV$/c$ and $|\eta| < 4$
- Parallel effort with ECCE
  - By Sebastian Tapia (Iowa State Univ.), supervised by Joe Osborn (ORNL)
  - Never committed to ECCE repository (*private* GitHub)
  - The ACTS issues discovered with ATHENA/DD4Hep/Juggler remained/likely never discovered
  - The lower field and tracker size allowed a set of parameters for $p > 1$ GeV$/c$ and $|\eta| < 3$, for the ECCE performance plot
  - “used [...] huge search windows”

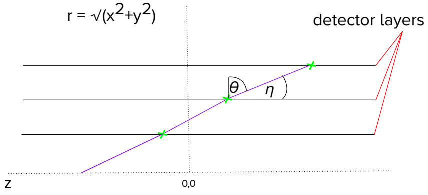
Seeding Code Landscape

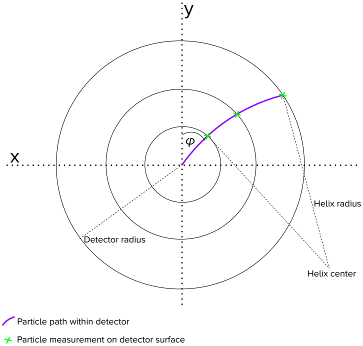
- ACTS (ATLAS)
- Several CMS-derived, public code
  - FKDTree `https://github.com/felicepantaleo/FKDTree`
    - ⇒ Abandoned (last commit March 2018)
    - ⇒ Undocumented
  - TrickTrack `https://github.com/HEP-SF/TrickTrack`)
    - ⇒ Semi-abandoned (last commit Feb. 2020)
    - ⇒ Obscure (hard to find via Google)
  - Close to what is used with CMS Run-3
- Documentation, development, responsiveness of developer clearly favor choosing ACTS
- From TrackML: Not trivial to beat traditional seeder, possibly even less so without the LHC-like multiplicity
  - ⇒ Unlikely to be priority until further in the detector development

ACTS Seeding

`https://acts.readthedocs.io/en/latest/core/seeding.html`

- Triplet generation (not limited to 3 layers, and works for forward configuration)





y
x
φ
Helix radius
Detector radius
Helix center
Particle path within detector
Particle measurement on detector surface

- Triple-loop filtering based on reasonable curvature, compatible η, backprojection to the vertex z range, etc.
- Double-loop filtering, weighting seeds based on mergeability of seed groups (ideally merge to seeds with 5 space points), backprojection to the lowest vertex z and ρ possible
- Single-loop filtering to only retain the highest quality N seeds per middle space point

Implementation Detail

- A plug-in named TrackParamACTSSeeding, near drop-in replacement for TrackParamTruthInit
- Input is the raw hits (seeding is done mostly in global coordinate, with one exception)
  - Local coordinate “measurements” currently do not work, there seem to be issues in global-to-local transform (material description or geometry issues?)
- Reuses the source links from Kalman filter, and will reorder internally (source links and hits can have different order)
- Differences vs. how ATLAS runs seeding:
  - Seeds are pre-filtered to 1 seed/track
  - ATLAS would reconstruct (using CKF) all seeds and then filter

Bugs, etc. with ACTS

- The code underwent 4 significant versions to track compatibility-breaking ACTS interface changes around versions 12, 15, 19
- Studies made with this plugin with ATHENA also contributed to the discovery or fix of at least two bugs in the ACTS code
  - For very forward tracking, the number of η bins can go to 0, then a difficult-to-trace segfault (masked by other code before 12, then "suddenly appeared")

  ⇒ Fixed around ACTS 15
  - For very forward tracking, low-ish min. p_T, and high magnetic field, the number of ϕ bins can become NaN (NaN converts on x86 to $2^{64} - 1$ for size_t/uint64_t, causing STL bad_alloc)

  ⇒ How to fix is not very clear, related to prior performance issues, but ACTS developer notified
- Some of problems took long time and require intimate knowledge of ACTS to track down, e.g. forward low p inefficiency was caused by an

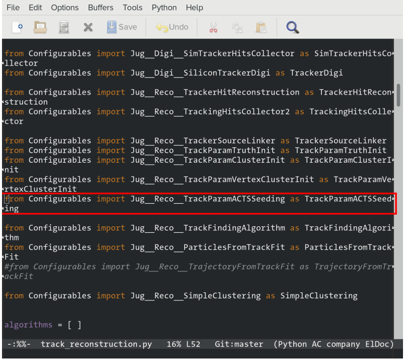
```
if ((bCoef * bCoef) > (1 + aCoef * aCoef) /
m_config.minHelixDiameter2) {...}
```

Status

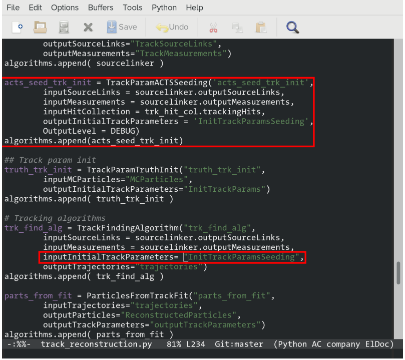
- There is an older version already merged into Juggler
  - Good performance for $p > 2$ GeV/c throughout the acceptance
  - Good performance for $p < 2$ GeV/c up to $|\eta| < 2$
- A version with good *efficiency* for all p and $|\eta| < 4$ is being tested and will be submitted to Juggler
  - Performance is reasonable for efficiency, but not Δp for $p < 2$ GeV/c (offset towards $+\Delta p$)
  - Also the $|z|$ vertex compatibility has to be a few 100 mm (otherwise inefficient)

  ⇒ Hints at potential material description/geometry issues
- Not really optimized (yet) for user experience, none of the parameters are not Python configurable, requires editing of the C++ code and recompile

How to Use



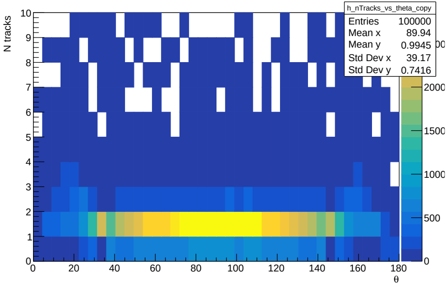
```
from Configurables import Jug__Digi__SimTrackerHitsCollector as SimTrackerHitsCo
llector
from Configurables import Jug__Digi__SiliconTrackerDigi as TrackerDigi

from Configurables import Jug__Reco__TrackerHitReconstruction as TrackerHitRecon
struction
from Configurables import Jug__Reco__TrackingHitsCollector2 as TrackingHitsColle
ctor

from Configurables import Jug__Reco__TrackerSourceLinker as TrackerSourceLinker
from Configurables import Jug__Reco__TrackParamTruthInit as TrackParamTruthInit
from Configurables import Jug__Reco__TrackParamClusterInit as TrackParamClusterI
nit
from Configurables import Jug__Reco__TrackParamVertexClusterInit as TrackParamVe
rtexClusterInit
from Configurables import Jug__Reco__TrackParamACTSSeeding as TrackParamACTSSeed
ing

from Configurables import Jug__Reco__TrackFindingAlgorithm as TrackFindingAlgori
thm
from Configurables import Jug__Reco__ParticlesFromTrackFit as ParticlesFromTrack
Fit
#from Configurables import Jug__Reco__TrajectoryFromTrackFit as TrajectoryFromTr
ackFit

from Configurables import Jug__Reco__SimpleClustering as SimpleClustering


algorithms = [ ]

-:%%-  track_reconstruction.py   16% L52   Git:master  (Python AC company ElDoc)
```

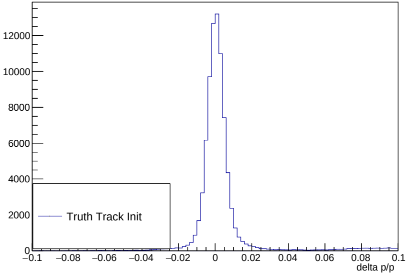


```
        outputSourceLinks="TrackSourceLinks",
        outputMeasurements="TrackMeasurements")
algorithms.append( sourcelinker )

acts_seed_trk_init = TrackParamACTSSeeding('acts_seed_trk_init',
        inputSourceLinks = sourcelinker.outputSourceLinks,
        inputMeasurements = sourcelinker.outputMeasurements,
        inputHitCollection = trk_hit_col.trackingHits,
        outputInitialTrackParameters = 'InitTrackParamsSeeding',
        OutputLevel = DEBUG)
algorithms.append(acts_seed_trk_init)

## Track param init
truth_trk_init = TrackParamTruthInit("truth_trk_init",
        inputMCParticles="MCParticles",
        outputInitialTrackParameters="InitTrackParams")
algorithms.append( truth_trk_init )

# Tracking algorithms
trk_find_alg = TrackFindingAlgorithm("trk_find_alg",
        inputSourceLinks = sourcelinker.outputSourceLinks,
        inputMeasurements = sourcelinker.outputMeasurements,
        inputInitialTrackParameters= "InitTrackParamsSeeding",
        outputTrajectories="trajectories")
algorithms.append( trk_find_alg )

parts_from_fit = ParticlesFromTrackFit("parts_from_fit",
        inputTrajectories="trajectories",
        outputParticles="ReconstructedParticles",
        outputTrackParameters="outputTrackParameters")
algorithms.append( parts_from_fit )
-:%%-  track_reconstruction.py   81% L234  Git:master  (Python AC company ElDoc)
```

- Edit e.g. `reconstruction_benchmarks/benchmarks/tracking/options/` `/track_reconstruction.py`
- Import `Configurables.Jug__Reco__TrackParamACTSSeeding` as something more weildy
- Connect input and name output as e.g. `’InitTrackParamsSeeding’`
- Change the input to the (inappropriately named) `TrackFindingAlgorithm` (fitting and not really track *finding*) to `’InitTrackParamsSeeding’`

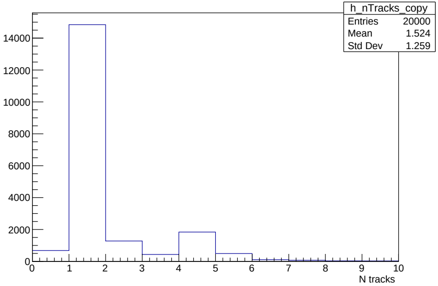
Performance

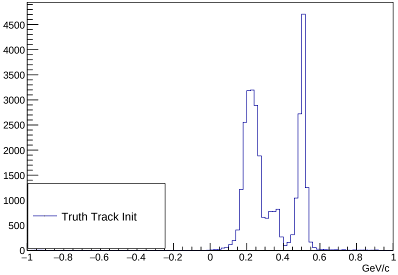




- ATHENA tracking reconstruction benchmark (slightly) modified to give all θ and a 2D N_{track} vs. θ (175° is $\eta \approx 3.1$)
- $N_{\text{track}} = 0$ is inefficient, $N_{\text{track}} = 1$ is single seed, $N_{\text{track}} > 1$ is multiple seed
- Still ATHENA/with a 3 T field, 1–2 GeV$/c$ π^+ (the most critical region)
- Ignore "Truth Track init" (someone hardcoded it into the ATHENA tracking benchmark)
- ACTS recommendation is to use cuts on fully reconstructed tracks to remove multiple reconstruction, but performance appear good enough to be without

Performance





- Select on $2 < \eta < 2.5$
- Again, ignore “Truth Track init” (someone hardcoded it into the ATHENA tracking benchmark)
- Highly efficient, $\varepsilon > 0.95$
- Issues remain with Δp

Way Forward

- Track Δp performance hints at potential material description or geometry in DD4hep/Juggler, that are not fixable from the seeding code side
  - ⇒ Likely not a show-stopper for background studies with efficiency/fake rate only
- ACTS developers encourages move towards a new orthogonal range search seed finder
  - Main advantage is bin-less, and avoids all the ACTS bugs discovered so far with binned spacepoint group
  - But not clear how well-tested for low p_T and forward, as we experienced sofar with the ACTS test coverage
  - ⇒ Likely not a priority until after background studies