Track Seed Finding in ACTS

Tomohiro Yamazaki (UC Berkeley) for the ACTS developers

IRIS-HEP topical meeting for HL-LHC R&D

March 24, 2021
Challenges at HL-LHC

- Many interesting physics programs, such as measurements of Higgs rare decays and searches for BSM with small xsec, are expected at HL-LHC from 2026.
- Tracking performance needs to be retained/improved under the high pile-up environment.
- For HL-LHC tracking, concurrent algorithms exploiting modern computing architectures with many cores are necessary.

The ATLAS tracking detector will be replaced with the full silicon detector for HL-LHC.

→ ITk

Tracking software also needs to be replaced with a new accurate, efficient, and fast software.

→ ACTS
A Common Tracking Software (ACTS)

- ACTS is an experiment-independent tracking for future detectors
- Based on ATLAS tracking algorithms
- Modern C++17 and minimal dependencies
- Facilitates new R&D project for new tracking techniques, e.g. ML, GPU
- Increases maintainability and usability
- Allows parallelization/vectorization
- ACTS PV reconstruction has been implemented in athena for ATLAS Run 3
- GitHub, documentation
- Previous ACTS talks by Paul, Xiaocong 1, 2, Irina
Track Reconstruction

- Starts with **Space Point** formation from local measurements on silicon layers.

- **Seeds** are then formed from triplets of hits.

- **Combinatorial Kalman Filter** is used to extend the seeds and build track candidates (**Track Finding**).

- Track candidates are scored according to track quality to remove duplicate tracks (**Ambiguity Solving**).
Seed finding in ACTS

The **ACTS seeding** implementation is based on the ATLAS track seeding. Space Points on pixel layers are used for seeding.

Seed finding algorithm iterates over middle SP. For each middle SP, top-middle and bottom-middle doublets are formed. Apply some cuts (eta, origin along z-axis, distance in r between SP) Top-middle and bottom-middle doublets are tested for compatibility. Seed triplet is created by combining the compatible doublets.
Seed finding in Generic Detector

Examples of ACTS tracking reconstruction chains are available in ACTS

With the generic detector geometry,
- Hits on the pixel detector are used for seed finding.
- Only a simple seed finding configuration is available.
- The same parameters for the barrel and endcap for now.
- Detector specific weights are not used.

Fatras simulation with a 2 T magnetic field

Optimization for the ATLAS ITk layout is ongoing, but this is not shown today.
Seed finding has been added in ACTS Example for validation
- The tracking efficiency can not be higher than seeding efficiency -> High efficiency required
- Validated with Generic detector
- Event generation, fatras simulation, and seeding/tracking examples in ACTS
- 99.7% efficiency for single muon events

\[
\text{Efficiency} = \frac{N_{\text{truth}} \text{(selected, matched to reco)}}{N_{\text{truth}} \text{(selected)}}
\]

Truth particles with \( p_T > 1 \text{ GeV} \) and \( \geq 9 \text{ hits} \) are considered
ACTS seeding under the HL-LHC environment

- Seed finding performance for ttbar events with $\mu=200$ at $\sqrt{s} = 14$ TeV
  - ~5k particles/event
- 91.4% efficiency
- The current configuration is not optimal for the barrel-endcap gap
  - No detector specific weights are used.
  - Need configurations for the barrel and endcap separately
# Seed finding performance

- The number of seeds affects the track finding CPU time and performance
  - Avoid creating seeds where there is no tracks (fake)
  - Should not provide many seeds for the same track (duplicated seeds)

- 58% fake rate for ttbar $\mu=200$ with the Generic detector

\[
\text{Fake Rate} = \frac{N_{\text{reco (selected, unmatched)}}}{N_{\text{reco (selected)}}}
\]

- Number of duplicated seeds:

![Plot of number of duplicated seeds vs. Truth $p_T$ (GeV)]

![Plot of number of duplicated seeds vs. Truth $\eta$]
Initial track parameters are estimated from seeds (triplets of space points), and used as proto-tracks.

Combinatorial Kalman Filter (CKF) algorithm is used to extend the proto-tracks and reconstruct tracks.

CKF Tracking efficiency for ttbar
- Reconstructed seeds as initial parameters
- Smeared truth particles as initial parameters
Track finding fake rate and duplication

- Fake rate $< 10^{-3}$ for CKF tracking from reco and truth seeds

\[
\text{Fake Rate} = \frac{N_{\text{reco}} \text{(selected, unmatched)}}{N_{\text{reco}} \text{(selected)}}
\]

- Duplication rate $< 0.2$ for CKF tracks with truth seeds
  ~ 0.6 for CKF tracks with reco seeds

\[
\text{Duplication Rate} = \frac{N_{\text{reco}} \text{(selected, matched, duplicated)}}{N_{\text{reco}} \text{(selected, matched)}}
\]

- Ambiguity solver is not used yet. The duplication rate could be suppressed by the ambiguity solver.
Integration of ACTS into athena is one of the main projects in this year.
- Develop wrapper to interface ACTS in athena
- Optimize athena/ITk specific configurations

For seeding,
- Optimize layer selection, seeding configurations for the barrel and endcap regions for ITk
- Detector specific weights to reduce duplicated seeds

Efforts are being made to run ACTS algorithms on GPU (Xiaocong’s talk)
- Long term goal is to port more modules to GPU to have a GPU-based algorithm chain
High performance tracking software is necessary for HL-LHC

ACTS is an open-source toolkit for track reconstruction
  - Retain/improve tracking performance for various detectors

Seeds provide Initial track parameters for the CKF tracking.
  - Tracking efficiency cannot be higher than seeding efficiency.
  - Suppression of fake/duplication seeds is important to improve the tracking performance

Good performance for the generic detector examples. Further optimization is ongoing.

Working on ACTS integration into athena for HL-LHC
  - Need ITk specific optimization.
Backup
Track Finding

**Kalman Filter**

- start with transport of track parameters (and covariances) to measurement surface, create predicted parameters (“predicted state”)
- combine/update predicted parameters with measurement to updated parameters (“filtered state”)

**Combinatorial Kalman Filter (CKF)**

- Simultaneous track fitting and finding
- Allows track branching if more than one compatible measurements found on a surface
- Allows stopping of bad quality branch

CKF results for ttbar events with \( \mu = 200 \) (~7k particles, ~80k hits)