ACTS KF/CKF status

ACTS Workshop 2023

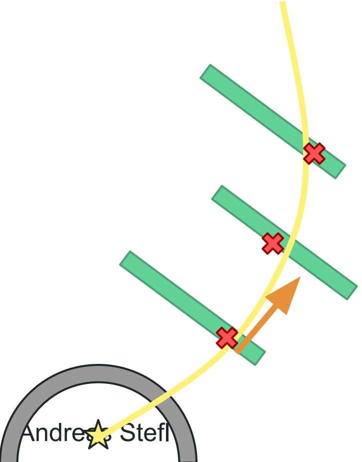
Outline

- Kalman fitter
    - Recap
    - State in ACTS
    - Plans
    - Discussions
- Combinatorial Kalman filter
    - Recap
    - State in ACTS
    - Plans
    - Discussions

Kalman fitter

KF: Recap

- Track fitting = combine measurements and estimates track parameters
- Kalman fitter does this in an iterative process
- Kalman filter updates track parameter with each measurement consecutively
- Acts like a fit when we add smoothing, another filter-like process in reverse
- Performing a fit requires an initial track state and the measurements



KF: State in ACTS

- Implemented as an actor, therefore part of the propagation
- Two smoothing modes: Linearized via Jacobian or reverse propagation
- Non-linear correction for global to local transformation
- Option to extrapolate parameters to a reference surface
- Material effects are handled by the KF actor and not by the propagator or another actor

KF: Recent changes

- Refactor and fix reference extrapolation
  - Start from measurement state https://github.com/acts-project/acts/pull/2470
  - Use material https://github.com/acts-project/acts/pull/2322
  - Refactor estimation for direction of reference extrapolation https://github.com/acts-project/acts/pull/2539

KF: Plans

- Remove reference surface extrapolation
  - This process seems too specific for a Core algorithm and does not serve all use cases
  - IMO it should therefore be removed and replaced by utilities that can also serve other fitters (GSF, GX2F, CKF)

KF: Discussion about Actor implementation

- Is it really more optimal to have fitters inside the propagation?
- IMO code could look nicer if the propagation is called from the fitter
- This could hide details of the propagation and constrain the interface to the important things

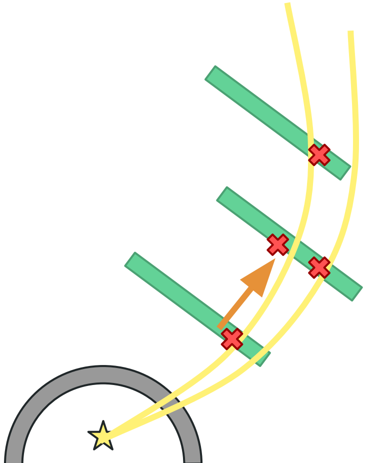
```
for (auto &surface : propagator.stepSurfaces()) {
     if (surface.isSensitive()) {
          auto params = filter(surface);
          propagator.setParams(params);
     }
     createTrackState(surface, ...);
}

// here could be an optional smoothing!
```

Combinatorial Kalman filter

CKF: Recap

1. Start propagation with initial parameters and covariance
2. Look for measurements on every sensitive surface encountered
3. Check if measurements are compatible and cut on #number and chi2
4. Create a new branch for each compatible measurement

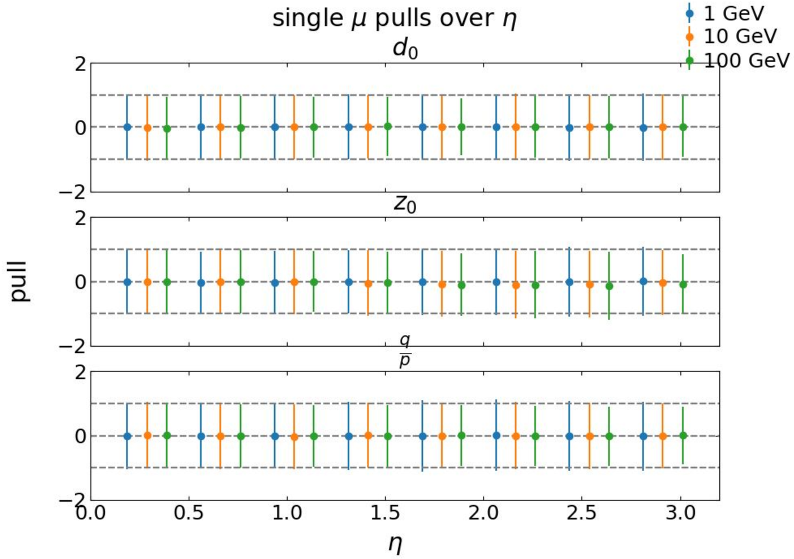


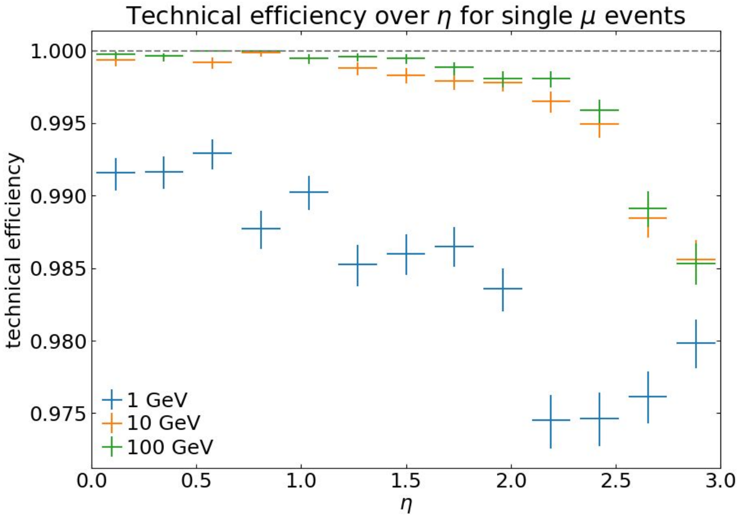
CKF: State in ACTS

- Implemented as an actor, therefore part of the propagation
- Single smoothing option: Linearized via jacobian
- Option to extrapolate parameters to a reference surface. This is coupled to the smoothing option.
- Material effects are handled by the CKF actor and not by the propagator or another actor
- **Limitation:** CKF can only discover measurements on surfaces encountered during propagation -> search window = trajectory
- **Limitation:** CKF can only discover measurements in one “direction”

CKF: State in ACTS

- Track finding and fitting performance study ACTS+ODD
  https://indico.cern.ch/event/1252748/contributions/5521502





CKF: Recent changes

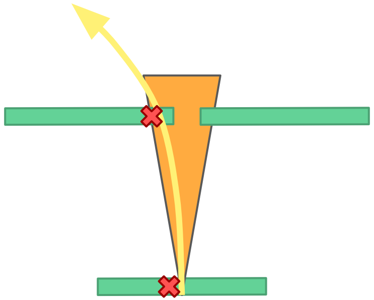
- Refactor and fix reference extrapolation
  - Start from measurement state https://github.com/acts-project/acts/pull/2470
  - Use material https://github.com/acts-project/acts/pull/2322
  - Refactor estimation for direction of reference extrapolation https://github.com/acts-project/acts/pull/2539
- Improve error handling https://github.com/acts-project/acts/pull/2540
  - Don’t fail seed if one branch fails
  - Introduce path limit for reference extrapolation
- Target surface for filtering phase https://github.com/acts-project/acts/pull/2319
- Fix finalization in case of multiple tracks per seed (big impact on ITk efficiency) https://github.com/acts-project/acts/pull/2299

CKF: Plans

- Remove reference surface extrapolation
- Explore possibilities to discover measurements on surfaces which do not intersect with the initial trajectory
- Explore possibilities to utilize the CKF in both “directions”

CKF: Discussion about measurement discovery

- Proposal A: With existing CKF and navigation
  - https://github.com/acts-project/acts/pull/2588
  - Increase search windows by making the navigator hit surfaces with extended boundaries
  - Caveat: We can be on multiple surfaces at the same time -> consequences for other actors/aborters?
- Proposal B: New track finding algorithm
  - New track finding algorithm could stear the propagation itself and target multiple surfaces
  - Road finding can be used to discover a sequence of relevant surfaces
  - Caveat: We could end up performing the same propagation steps multiple times



CKF: Discussion about both “directions”

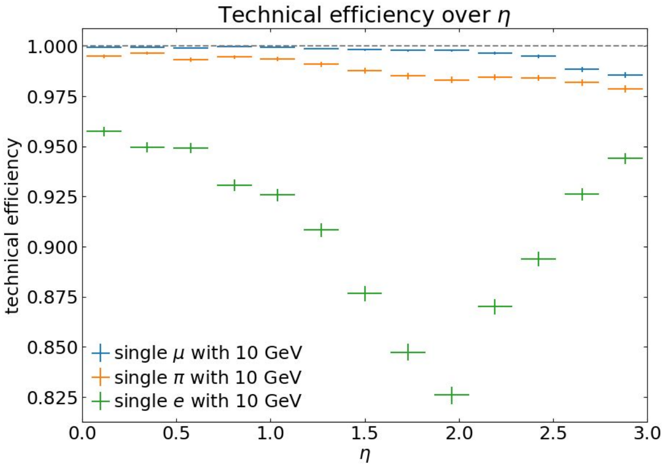
- Proposal A: Use the CKF twice
  - First call CKF to find tracks outwards; smooth them
  - Use the smoothed parameters to search inwards
  - This may require an additional smoothing or refitting at the end

Summary

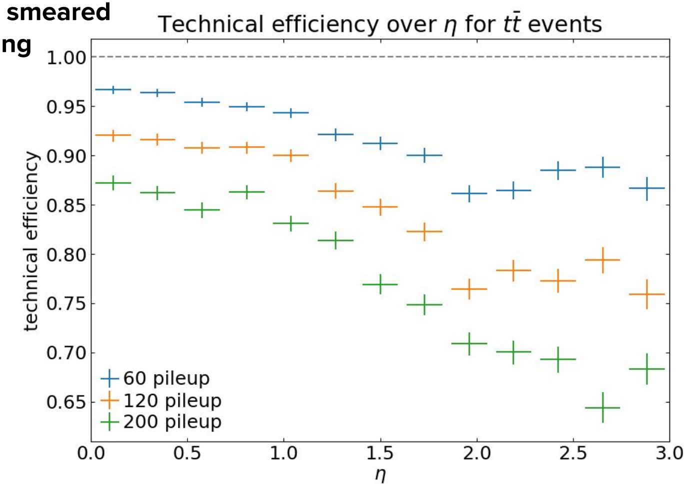
- KF seems to be in a good state
  - Small improvements can be made
  - Refactoring the tight coupling to the propagation my not be worth dev time as it works as is
  - Fitting performance is validated an

- CKF / track finding needs attention
  - Efficiencies are too low -> Navigation issue / surface discovery too constrained?

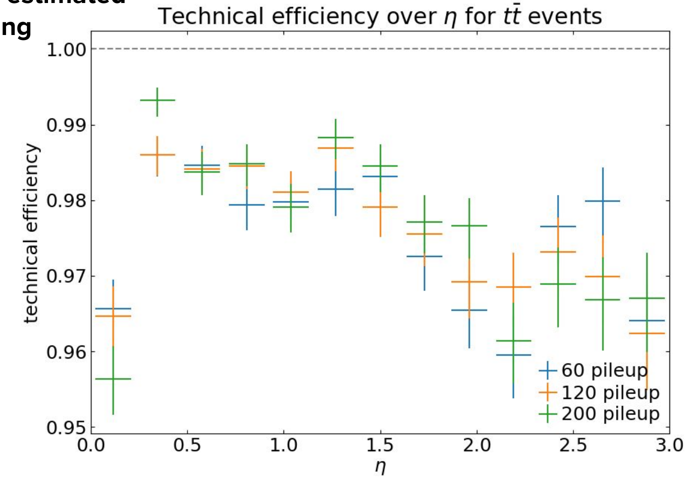
Backup



Truth smeared seeding



Truth estimated seeding


Technical efficiency over η for $t\bar{t}$ events
technical efficiency
η
60 pileup
120 pileup
200 pileup