Karlsruhe Institute of Technology

Having Multiple Candidates
in Parallel

NEW CDC CKF

26.10.2018

Nils Braun | IETP - KIT

BASIC PRINCIPLE

```
for (CDCCKFPath& path : paths) {
    m_stateCreator.apply(nextStates, path, wireHits); // 1
    m_stateFilter.apply(path, nextStates);            // 2
    if(nextStates.empty()) {
        resultPaths.push_back(path);
    }
    for (const auto& nextState : nextStates) {
        path.push_back(nextState);
        newPaths.push_back(path);
        path.pop_back();
    }
    nextStates.clear();
}
m_pathMerger.apply(newPaths);                         // 3
m_pathSelector.apply(newPaths);                       // 4
paths.clear();
apply(newPaths, wireHits, resultPaths);               // afterwards: 5
```

DETAILS

State Creation (1)

Create a state for all hits, which are

- maximally 2 layers apart
- not already used in the candidate
- maximally $\pi/8$ away in ϕ

State Filter (2)

1. pre-filter
2. helix extrapolation without uncertainties
3. basic filter (z, s, Δd; may be MVA at some point?)
4. genfit extrapolation without material effects and Kalman update
5. final filter (Δd; χ^2?; may also be MVA?) and select best $\mathtt{N}$ (currently 4)

DETAILS II

Path Merging (3)

This step will (at some point) remove doubled candidates which just differ by single hits to save computing time and increase the number of ”tryouts”.

Path Selection (4)

We compare each path pair and select the best `N` candidates (currently 3).

- $\Sigma\Delta d^2$, $s_{\max}$
- MVA?
- MC?

Result Selection (5)

For a single seed, there could be multiple results. Currently, the longest candidates is selected (probably fine).