

New Kalman Filter Event Data Model

Paul Gessinger

CERN / JGU Mainz

Acts Developers Meeting - 10.07.2019

Introduction

- New KF EDM as a replacement for current `std::vector<Acts::TrackState>`
- [MR !559](#) open
- Currently pending:
 - ▶ Will do performance profiling once [acts-framework!126](#) (Xiaocong's Fitting example) is in, and FW is updated to v0.10.X
 - ▶ Investigate usage for the GSF, since GSF has specific requirements
 - ▶ After discussion with Jin and Moritz, using dedicated EDM for GSF probably makes more sense
 - ▶ Can still share new EDM for KF and CKF (and maybe Global- χ^2 ?)

Old EDM

- `std::vector<Acts::TrackState>` is *array of structs*
- TrackState has optionals inside, sequencing is external (vector)
- No support for branching at all (we probably need this for a CKF)

New EDM

- Use `Eigen::Array` with one dynamic and one static size for storage
- Keep track of capacity and size, provide read-write views (Maps) into storage

```
using Coefficients = Eigen::Matrix<Scalar, Size, 1, Flags>;  
using Covariance = Eigen::Matrix<Scalar, Size, Size, Flags>;  
using CoefficientsMap = Eigen::Map<ConstIf<Coefficients, ReadOnlyMaps>>;  
using CovarianceMap = Eigen::Map<ConstIf<Covariance, ReadOnlyMaps>>;
```

- Stores index-tuples, parent relationship allows branching

New EDM: IndexData

```
struct IndexData {  
    using IndexType = uint16_t;  
    IndexType irefsurface = kInvalid;  
    IndexType iprevious = kInvalid;  
    // ...  
    IndexType ijacobian = kInvalid;  
    IndexType iprojector = kInvalid;  
    double chi2; // some other info as well  
    double pathLength;  
    IndexType iuncalibrated = kInvalid;  
    IndexType icalibrated = kInvalid;  
    IndexType icalibratedsourcelink = kInvalid;  
    IndexType measdim = 0;  
};
```

New EDM: TrackStateProxy

- TrackStateProxy is a class that retains a link to the track state collection, and knows its index

```
class TrackStateProxy {
    size_t index() const { return m_istate; }
    size_t previous() const { return data().iprevious; }
    const Surface& referenceSurface() const { /* ... */ }
    Parameters predicted() const; // read-write map for pred parameters
    Covariance predictedCovariance() const; // same for covariance
    // components can optionally be "unset", no storage is allocated
    bool hasPredicted() const { /* ... */ }
    // ...
};
```

New EDM: TrackStateProxy

- TrackStateProxy can share data between multiple instances
- This could be useful for CKF where we don't need to duplicate prediction, calibrated measurement...

Example: Kalman Gain Matrix updater

- Before:

```
std::visit([&](const auto& calibrated) { // visit on autogenerated meas
    // work with concrete local measurement dimension
}, *ts.measurement.calibrated);
```

- After:

```
visit_measurement(ts.calibrated(), ts.calibratedCovariance(),
    ts.calibratedSize(), [&](const auto calibrated,
        const auto calibrated_covariance) {
    // storage in EDM is overallocated, so at max meas dim
    // *calibrated* and *calibrated_covariance* are static sized mats
});
```