On-the-fly measurement calibration with ACTS

Louis-Guillaume Gagnon (UC Berkeley)

Connecting The Dots 2023
2023/10/11

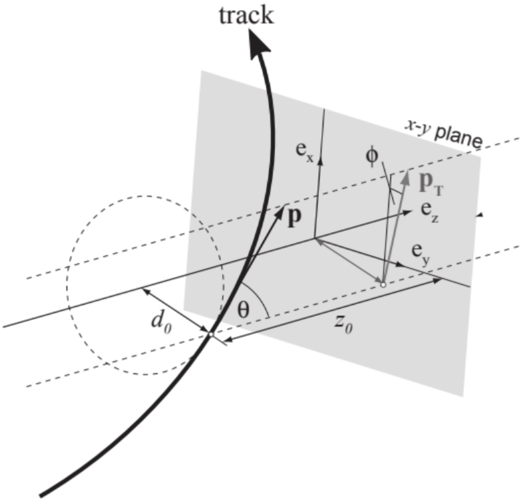
ACTS: A Common Tracking Software project

- Experiment-independent toolkit for tracking
- Free software (Mozilla Public License v2.0)
- Considered for use by ATLAS, FASER, Belle II, CEPC, sPHENIX, PANDA, EIC, . . .
- Three overarching goals:
  1. Preserve current tracking approaches while enabling development for HL-LHC
  2. Serve as an algorithmic test bed incl. ML-based algorithms and accelerated hardware
  3. Enable rapid and realistic development of new tracking detectors
- Includes an ONNX plugin, to enable import of various ML models anywhere in the tracking workflow
- Ongoing R&D for GPU tracking (traccc)

- Overview paper: [2106.13593]
- Project webpage: acts.readthedocs.io
- Code repository: github.com/acts-project/acts

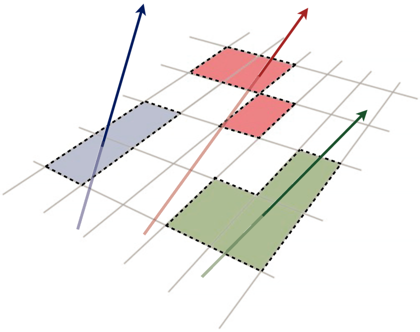
Introduction: Kalman Filter

- ACTS Track state model: $(d_0, z_0, \theta, \phi, q/p, t)$
  - with associated covariance
- Estimated with measurements from detector
- E.g. for pixel detector: $m = (x, y)$
  - with associated covariance, usually diagonal
- Track state incorporates measurements via Kalman Filter formalism
  - Start from track seed parameters
  - Predict parameters at next surface
  - Search for matching measurements
  - Kalman update stage: Update track state using matching measurement
  - Repeat until no more surfaces
- Usually followed by Kalman "Smoothing"
  - Replace each local state estimate with an optimal estimate given a complete set of measurements
- Nucl.Instrum.Meth.A 262 (1987) 444-450



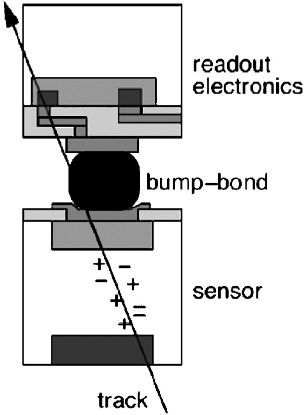
Introduction: Measurements

- From pixel detector, need measurements $m = (x, y)$
- However, only get individual pixels from readout
- $\rightarrow$ Use connected component analysis to obtain clusters
- Then can estimate measurement position:
  - $(x, y) =$ charge-weighted cluster center
  - $(\sigma_x, \sigma_y) =$ pix. width / $\sqrt{12}$
- Possible to improve:
  - Take direction into account
  - Do fancier shape analysis
  - ...
- Measurement calibration paradigm: Apply corrections to estimated measurements during Kalman update stage
  - Simple scale-and-offset schemes
  - ATLAS: "Analogue clustering", NN-based clustering
  - ... many other possibilities



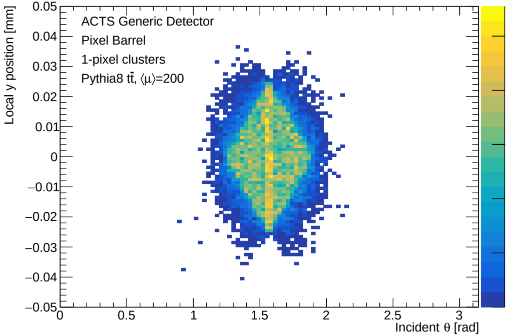
Example 1: Single pixel clusters

- Primarily rely on shape analysis to constrain position
- Edge case: 1-pixel clusters, “no” shape information
- However: Angles of incidence give some constraint!
  - $\approx 90^\circ$ crossing: Anywhere on surface
  - $\rightarrow 0^\circ$ crossing: Near center (else, ≥ 2 pixel)
- N.B. position defined at middle of Si bulk, by convention

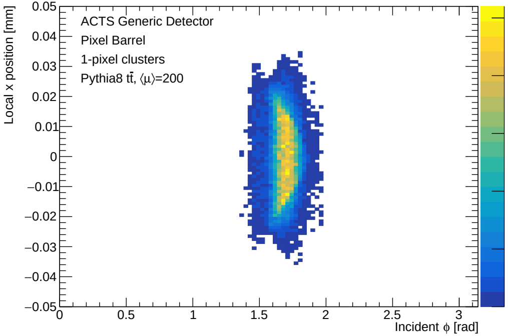


1-pixel cluster: Local positions vs Angles of incidence

- Clear relationship between $\sigma(\mathrm{pos})$ and angle



- Transverse direction: Shift due to B field



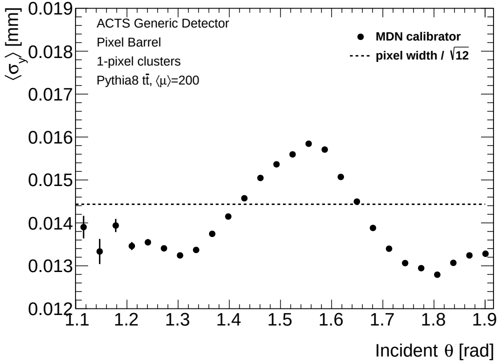
Measurement Calibration with Neural Networks

- MDN $\equiv$ Mixture Density Network
- i.e. any neural network trained to output parameters of a gaussian mixture
- Model output: parameters π_i, μ_i, σ_i such that:

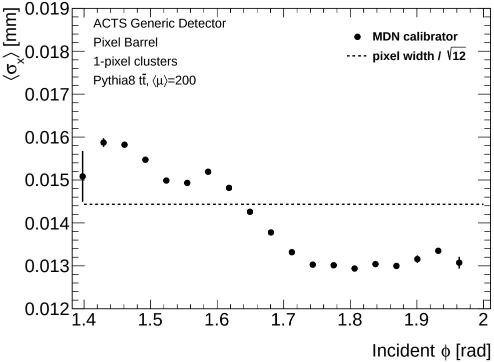
$$P(Y|X) \sim \sum_i \pi_i(X)\mathcal{N}(Y|\mu_i(X), \sigma_i(X))$$

  - X is set of variables describing a measurement (e.g. charge, volume/layer, angles of incidence)
  - Y is true crossing position in Si bulk (ground truth)
  - $\pi_i(X)$: Prior probability for i-th component (if using ≥ 2 components)
  - $\mu_i(X)$: Calibrated position estimate (Supervised learning)
  - $\sigma_i(X)$: Uncertainty estimate (Unsupervised learning)
- If using single component, model is a simple normal distribution
- Trained using probabilistic programming paradigm: loss is directly $-\log P(Y|X)$
- At runtime, use $\mu_i \pm \sigma_i$ corresponding to highest π_i as position estimate
- This method naturally generalizes to clusters with ≥ 2 particles
- Method used by ATLAS collaboration for pixel measurement calibration
  - See e.g. ATL-PHYS-PROC-2019-082

- Clear relationship between $\sigma(\mathrm{pos})$ and angle
  - Stronger constraint at large angles
  - Weaker constraint for head-on particles



- Asymmetry due to Lorentz Angle in transverse direction (known constant shift due to B-field and Si thickness $\rightarrow$ easy to correct)



N.B. $\sigma_{x/y}$ are model-estimated uncertainties, not residuals

Calibration interface in ACTS

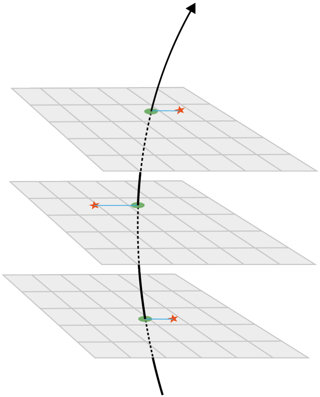
- The ACTS tracking toolkit contains Kalman Filter-based track finding & fitting algorithms
- Calibrations can be applied on-the-fly during track finding / track fitting
- Interface implemented using template-based delegation:

```
class KalmanFitterExtensions {
  using Calibrator = Delegate<void(const GeometryContext&, const CalibrationContext&,
                                   const SourceLink&, TrackStateProxy)>;

  /// The Calibrator is a dedicated calibration algorithm that allows
  /// to calibrate measurements using track information, this could be
  /// e.g. sagging for wires, module deformations, etc.
  Calibrator calibrator;

  ...
};
```

- Calibrator class acts directly on track state proxy, which holds the current measurement
- Dynamic geometry effects and intra-run calibration changes encapsulated via contextualization
- `ONNX` plugin: NN-based calibration methods are supported!
  - See NeuralCalibrator in ACTS Examples

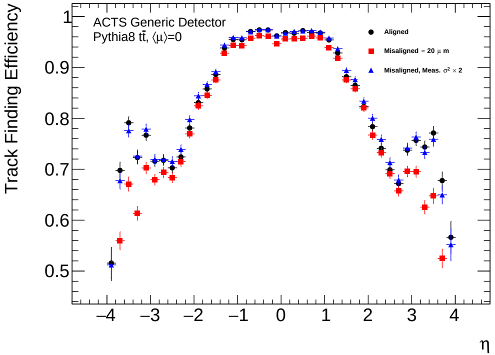
Example 2: Detector alignment

- Alignment $\equiv$ deriving corrections to geometry description for shifts in element positions
- Work ongoing in ACTS on Kalman Filter-based alignment algorithm
- If detector is very misaligned, performance is degraded:
  - Track efficiency drops
  - measurements are lost
- Can "bootstrap" the alignment procedure with measurement calibration:
  - Scale measurement errors up $\rightarrow$ recover tracking efficiency
  - Perform alignment with unscaled errors



- Alignment minimizes track–hit χ^2
- c.f. [2007.07624]

- Simulated "large" misalignment: linear shift $\sim \mathcal{N}(0, 20\mu m)$ for each module
- Clear effect on efficiency & measurements per track
- Using the ACTS ScalingCalibrator, apply a $\times 2$ factor to the measurement variances
  - Emulates artificially-large clusters
  - Efficiency is recovered



Conclusion

- Measurement Calibration: Correcting the measurement positions & errors on-the-fly during track finding & track fitting
- The ACTS Kalman Filter includes efficient template-based interface to measurement calibration
- Different examples are provided: Simple ScalingCalibrator, Fancy MDN-based NeuralCalibrator
- Future plans:
  - Provide documentation and tutorials for the interface and the examples
  - Explore more calibration methods (e.g. ATLAS "Analogue Clustering")
  - Implement ATLAS-inspired dense environment calibration (Cluster splitting, positions for ≥ 2 particles, ...)