Karlsruhe Institute of Technology

Development and Usage at Belle II

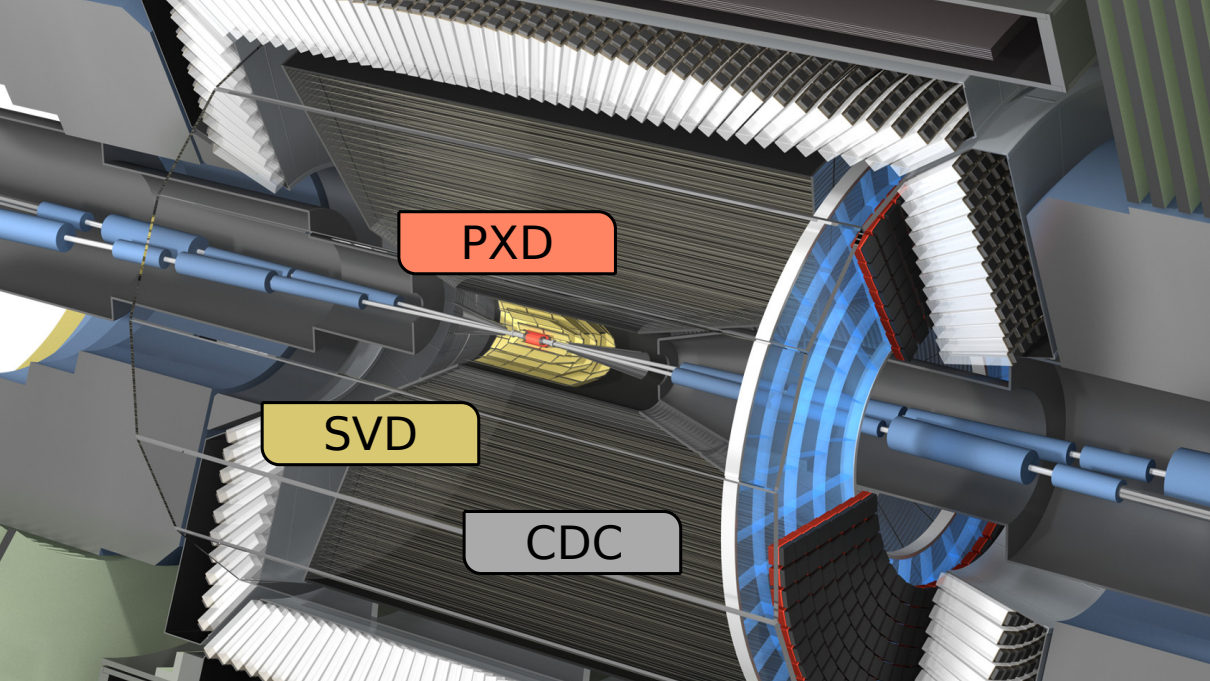
A COMMON TRACKING SOFTWARE

2019, September 30th

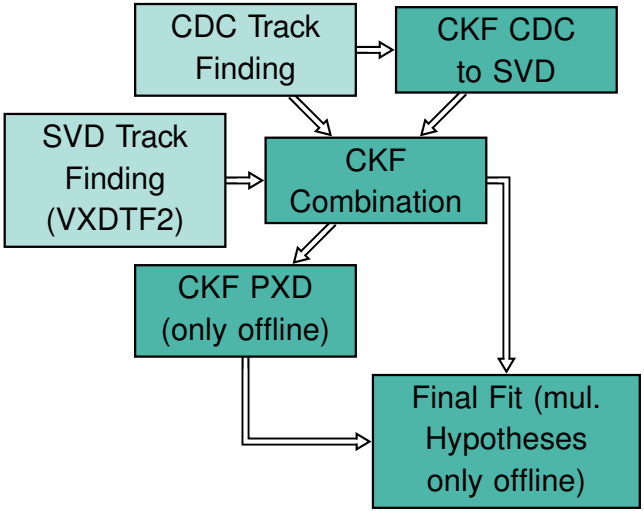
Nils Braun, Michael Eliachevitch | IETP - KIT

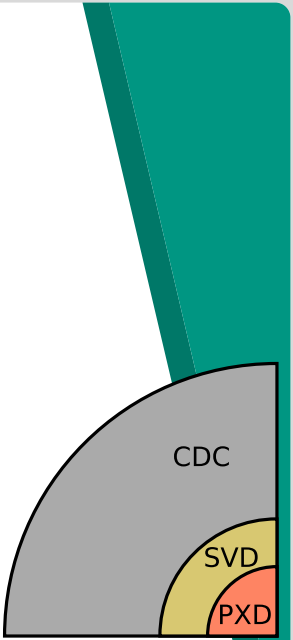
1.

OVERVIEW BELLE II


PXD
SVD
CDC

TRACKING WORKFLOW AT BELLE II





Things are currently changing...

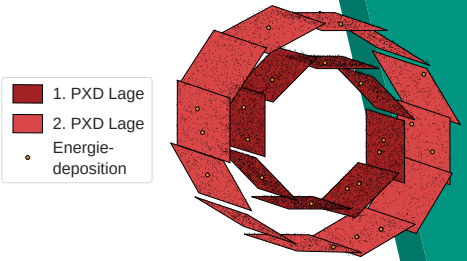
THE CHALLENGES OF TRACKING AT BELLE II

On average 11 tracks per event...
We want all of those, but not a single fake!

Low momentum particles

- Non-negligible fraction
- Multiple scattering

Beam-induced background High occupancy due to background hits:
11 tracks $\rightarrow$ 10^2 signal hits
vs. 10^4 background hits



- Synchrotron radiation
- Touschek scattering
- Beam-gas scattering
- Radiative Bhabha scattering
- e^+e^- pair production

GOALS WITH ACTS

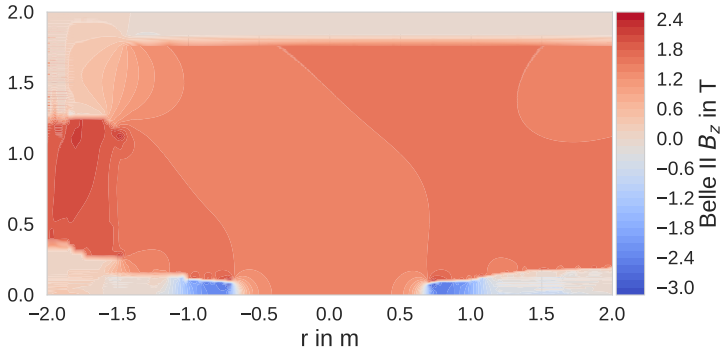
- Replacing the full tracking stack of Belle II with ACTS is impossible
  - Genfit is deeply integrated into `basf2` and especially our alignment procedures
  - But we are (now) also heavily relying on CKFs
  - In the end, it works already
- The fit (especially in the CDC) is definitely slower than needed
  - We hope to gain speed in using ACTS for extrapolation and material handling
- Some open issues with our tracking/fitting (e.g. Bremsstrahlung/Electron fit)
  - We hope to benefit from the developments in ACTS
- Belle II has a modern framework with many already implemented tracking algorithms which we can give back to ACTS
  - Hough/Legendre, SectorMap, CKF, background filters, MVA framework
  - Real data, real HLT

2.

MAGNETIC FIELD

MAGNETIC FIELD

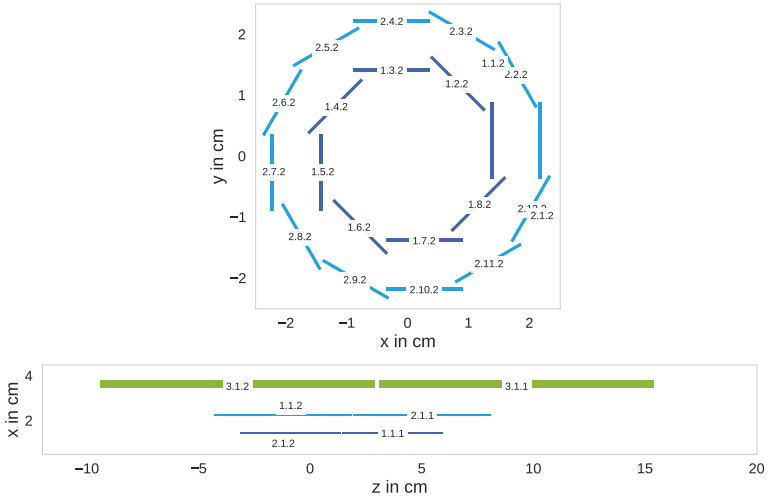
- ACTS includes a mechanism for field maps and interpolation
  - But Belle II already stores the magnetic field in such a format
  - Simplest (and probably fastest) solution: just use it
- Worked out of the box



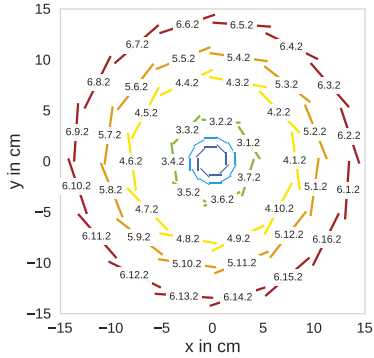
3.

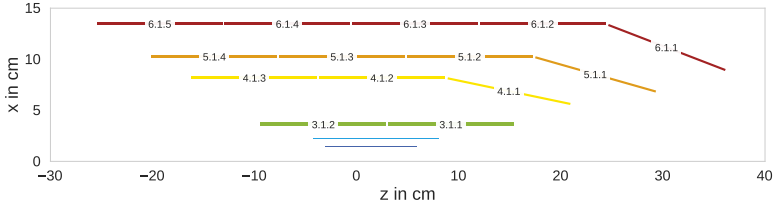
GEOMETRY AND SENSORS

VERTEX DETECTOR: PXD ORIGINAL



VERTEX DETECTOR: SVD ORIGINAL

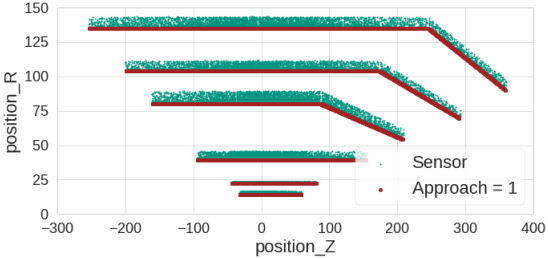


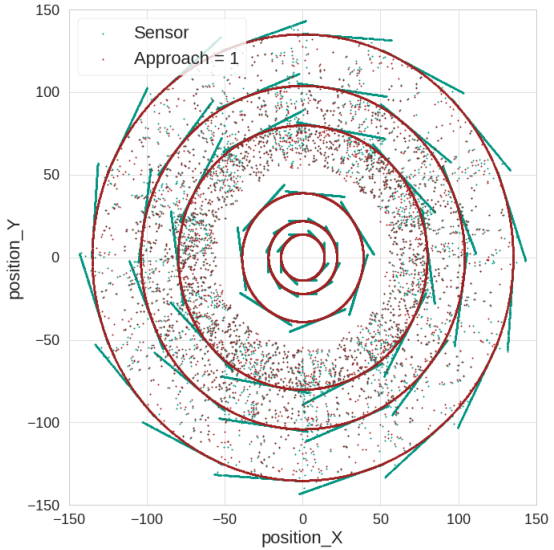


CHANGES IN ACTS

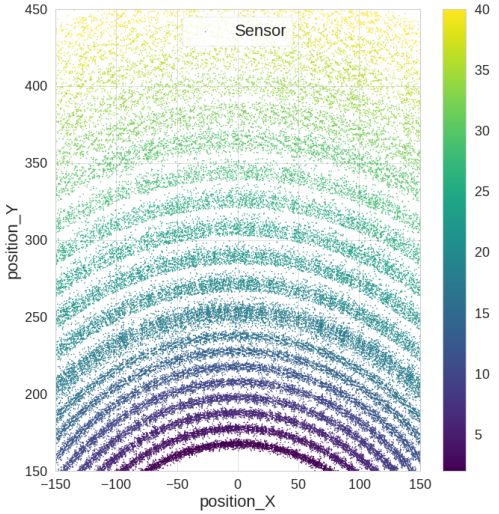
- Asymmetric Detectors needed some fixes
- Slanted Design of SVD needs new surface, bounds, layer and implementation in all helper functions
  - Is a bit hacky (some positions are undefined for the norm etc.)
  - So far it works, probably needs more tests
- Now all VXD and CDC sensors are implemented at their nominal positions and properly connected
  - I also started adding some proto material layers (see below)

VXD IMPLEMENTED





CDC IMPLEMENTED



- Non-sagged wires (axial and stereo) works as expected
- Some open issues with wire sag and interpolation to POCA (discussion with ACTS developers)

4.

ALIGNMENT AND CDC WIRES

NOMINAL POSITION

- For the pure extrapolation and material effects handling alignment does not play a major role
- Therefore my plan:
  - Only implement the nominal positions (or simple deformations like translations) into the sensors/surfaces
  - Handle more complex deformations only during Kalman filter
  - Can reuse some code we already have
- Benefit: extrapolation fast and simple (can reuse the current implemented classes in ACTS)
- Drawback: need additional steps during Kalman filter (see below for the different mindsets)

CDC WIRES AND WIRE SAG

Multiple Possibilities for implementing the CDC Wires

- Stop the extrapolation once the track reaches the wire cell
  - Fast, but needs re-extrapolation for Kalman filter
  - Wire sag could be implemented in this second step (as a refiner)
  - Currently preferred solution
- Stop the extrapolation once the track reaches the POCA
  - Still has some problems in the implementation, needs more iterations
  - To get wire sag one would need to implement a sagged line (difficult)
- Do not stop at all
  - Not needed for material effects (so fine)
  - But ”hole-counting” is a problem
  - Would need another solution for the Kalman filter (1 or 2)

5.

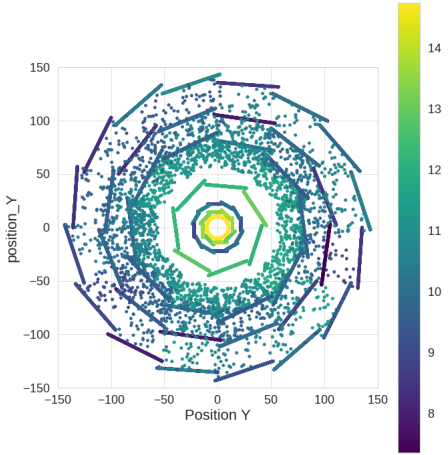
MATERIAL EFFECTS

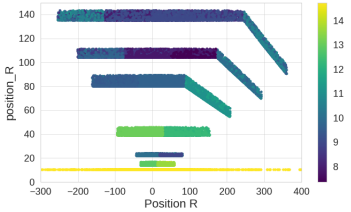
GENFIT → ACTS

- Genfit uses the Geant4 geometry model to request the material properties at a specific position
- Mapped to ACTS using the aggregated material mappers already in ACTS
- Workflow, Belle II database connection and validation implemented

Currently ongoing.

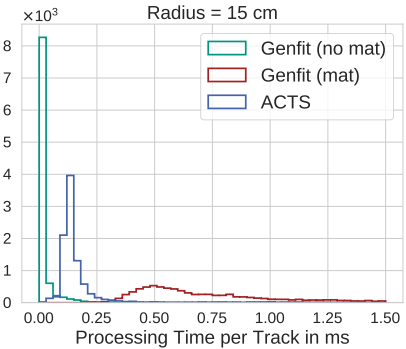
EXAMPLE: AVERAGE Z PER SENSOR





Promising first results, still some work to understand all details. . .

PROCESSING TIME



Attention: ACTS uses material effects (processing time should be correct) without applying any correction (as material effects do not make sense already)

6.

INTEGRATION AND DIFFERENT MINDSETS

HOW TO GET ACTS INTO BASF2

There are multiple possibilities

- Replace the EDM and all genfit fitters
  - Unfeasible
- Use ACTS only during fitting and write converters
  - Possible, but how about extrapolation during CKF?
  - Maybe re-implementation of CKF possible (see below)
  - Kalman Filter works differently in ACTS and Genfit
  - Highly dependent on developments in ACTS
- Use ACTS as track representation only during extrapolation
  - Simplest solution for code development, but see drawbacks below
  - Reuse as much as possible (also alignment etc.)
- Only replace material effects handling
  - Quite complicated to merge different mind sets (if not impossible)

DIFFERENCE: KALMAN FILTER

Genfit

1. Define plane of next hit
   - Genfit always needs planes, this means this step might imply an extrapolation already
2. Extrapolate to plane
   - If not hit, abort
3. Kalman update
4. Repeat

ACTS

1. Extrapolate into direction of end surface
2. If surface hit, check if hit is given and do Kalman update
   - Surface definition may need to be larger than usual for this

Problem 1: will definitely give different results if full fitter replaced
Problem 2: if only extrapolation replaced, can not use all benefits of ACTS (especially processing speed)

DIFFERENCE: BASE OBJECT

Genfit

Base object to extrapolate to is always a plane

- lines, circles, points etc. are always mapped to planes by extrapolating until POCA is reached without covariance transport, then plane definition, then extrapolation with full covariance
- Might not be the best solution, but maps easily to the Genfit EDM

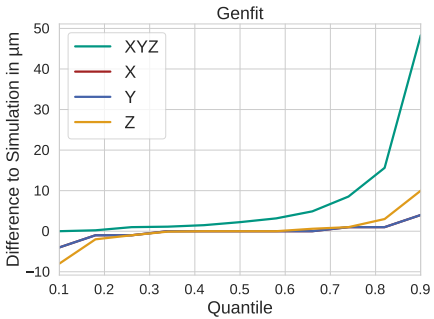
ACTS

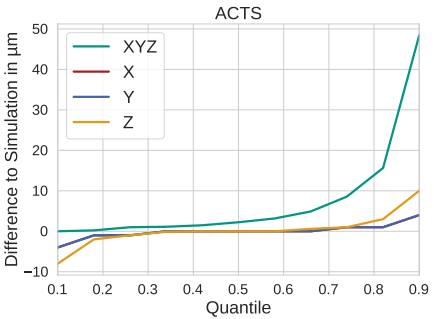
- Knows how to extrapolate to every single surface type (but is it faster?)
- Would need some translation work to get the benefits of very fast extrapolation

How to not do it: naïve replacement of extrapolation

$\Rightarrow$ ACTS is actually slower...

RESULTS WITHOUT MATERIAL EFFECTS MATCH





Difference between ACTS and Genfit is basically 0.

DIFFERENCE: CKF

Genfit and `basf2`

- define next possible hits without complex extrapolation
- extrapolate to each hit (planes) separately, Kalman update etc.
- Extrapolation might be done multiple times

ACTS (would be)

- Extrapolate until a surface is hit
- definition of boundaries might be different (larger) than usual
- Kalman update etc.

TODO LIST

1. Only replace extrapolation for checking conversion and feasibility
   - Done, works
2. Possibility 1: smart base plane object in genfit enabling ACTS to use faster (and more correct?) extrapolation
   - Needs some very careful thinking, because the planes are part of the public API of Genfit
3. Possibility 2: replace full fitter with ACTS
   - Conversion of results and start parameters is simple (and halfway implemented)
   - Measurement coordinate transformation not done yet, so not testable so far :-/

7.

SUMMARY

SUMMARY

- Core functionalities are implemented in `basf2`
  - Geometry, Surfaces, (Material), Magnetic Field
- Emphasis on software usability und simpler development (in the future)
  - Many utilities and helper classes, integration into `basf2` module schema
  - Interface with database (still needs some additional development)
  - many validations already in place, jupyter notebooks for analysis
- Next step involve heavier rework of current software to use the benefits of ACTS properly
  - Think about planes or replace full fitter
  - Rewrite CKF