Track finding at Belle II

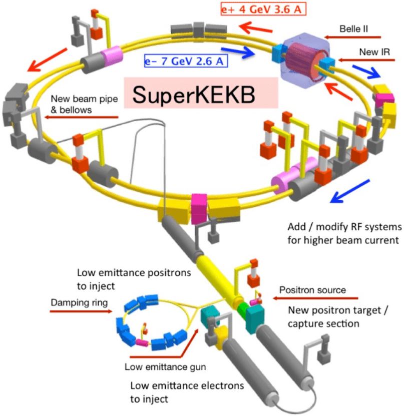
Thomas Lück

July 24, 2023

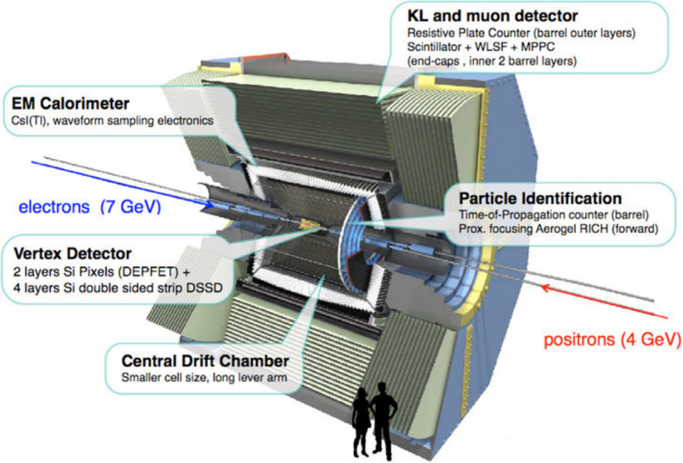
Table of contents

SuperKEKB

- asymmetric e^+e^- collider in Tsukuba (Japan)
- 7GeV electrons on 4GeV positrons
- B-factory:
  - center of mass energy of 10.58GeV
  - produce $e^+e^- \to \Upsilon(4S) \to B\bar{B}$
  - design luminosity $6 \times 10^{35} cm^{-2} s^{-1}$
  - recently broke world record luminosity

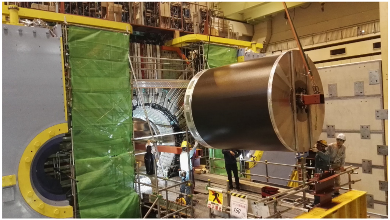


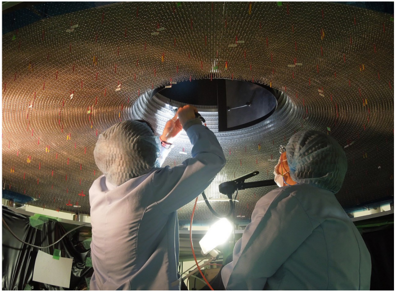
Belle II


KL and muon detector
Resistive Plate Counter (barrel outer layers)
Scintillator + WLSF + MPPC
(end-caps , inner 2 barrel layers)
EM Calorimeter
CsI(Tl), waveform sampling electronics
electrons (7 GeV)
Particle Identification
Time-of-Propagation counter (barrel)
Prox. focusing Aerogel RICH (forward)
Vertex Detector
2 layers Si Pixels (DEPFET) +
4 layers Si double sided strip DSSD
positrons (4 GeV)
Central Drift Chamber
Smaller cell size, long lever arm

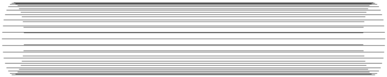
Central Drift Chamber (CDC)

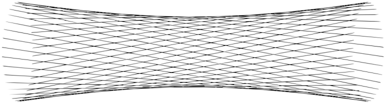
- gas filled chamber with wires providing high voltage E field
- 56 layers radius 168 - 1111 mm
- arranged into super layers of Axial (A) and stereo (U,V) wires

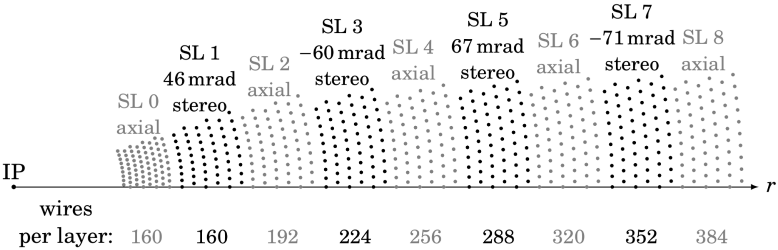




- parallel wires no way to tell where on the wire a track passed
- skewing part of the wires
  - position along wire determined from point where parallel and skewed wires cross
  - 3D reconstruction of tracks

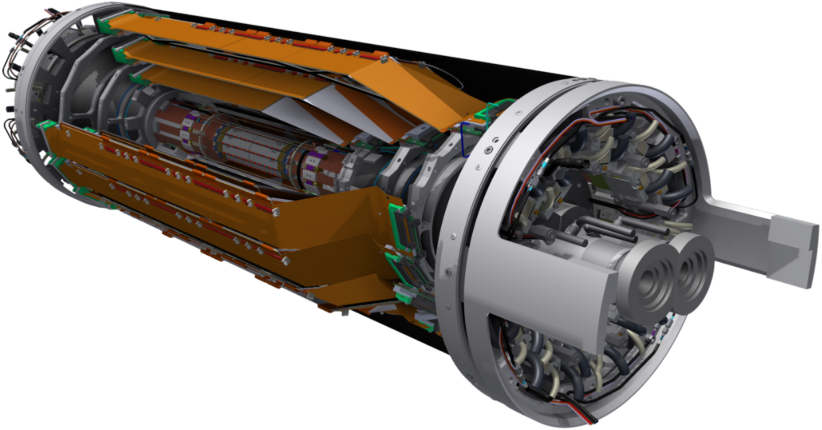






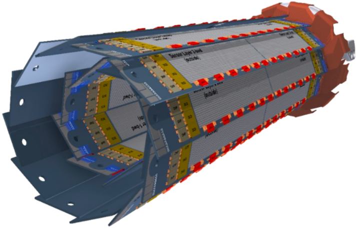
Silicon Vertex Detector (SVD)

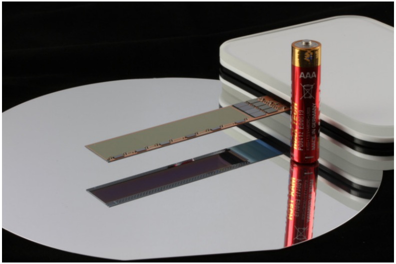
- 4 layers of double-sided silicon strip sensors (r=39, 80, 104, 135mm)
- 172 sensors, 220k readout strips
- strip distance between 50 and 160 μm
- strips are arranged perpendicular to get 2D information



Pixel Detector (PXD)

- 2 layers of DEPFET silicon pixel sensors (r = 14, 22 mm)
- pixel sizes 50x50 (75) μm
- 40 sensors, 7.7 million total pixels





What do we get from the detector

- CDC:
  - hit location: which wire has been hit (no information on where on the wire)
  - deposited charge (particle identification)
  - drift time: time between particle passing CDC and time charge arrives at wire
  - $\Rightarrow$ drift circle: distance particle passed wire
- SVD
  - usually two strips per hit (x and y direction)
  - combine x- and y- strip to get 2D position on sensor
- PXD:
  - pixel position on sensor

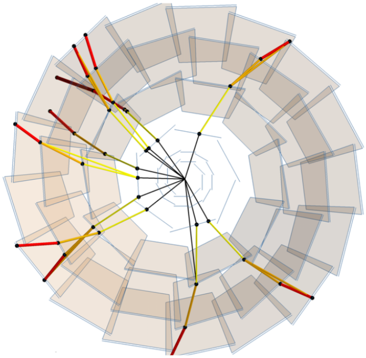
- local coordinates have to be translated into global coordinates:
  - use well known detector geometry for sensor position
  - precise alignment needed (precise position measurement)

Goals for tracking

- take detector information and reconstruct the tracks left by charged particles in the detector
- estimate the track parameter with high precision (5 parameter to describe a track $\Rightarrow \vec{p}(t), \vec{x}(t)$)
- high efficiency: single track inefficiency results in loss of events
- high purity: analysts dont like fake tracks (random combination of hits)
- speed:
  - Belle II has High Level Trigger (HLT):
    - software based trigger
    - reconstruct (includes tracking) events to perform preselection
    - decision which PXD hits are read out
  - event rate of 10kHz after hardware trigger
  - only limited time to reconstruct event
  - tracking most time consuming: Time = Money

Track finding

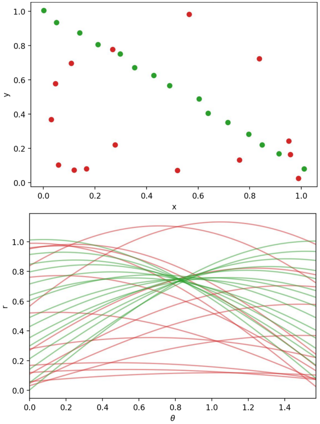
- on average 11 charged particles in B-events
- SVD: 11 hits per layer not counting background hits $\Rightarrow \approx 14000$ different combinations (for 4 layers)
- CDC has 56 layers $\Rightarrow 11^{56}$
- smarter approaches:
  - Local approach: search for neighboring hits starting from a seed
  - Global approach: take all hits and look for patterns



DOI: 10.1051/epjconf/201715000007

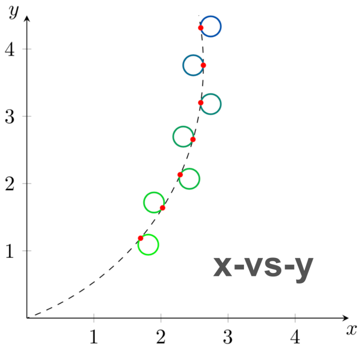
Hough transformation

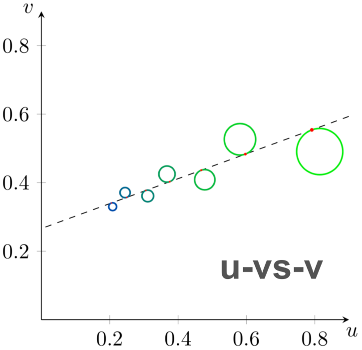
- global finding: start with all position measurements
- How to find a pattern if you know its shape (functional form)?
- Simple example: straight line $r = x\cos(\theta) + y\sin(\theta)$
- Go into parameter space (Hough space) by inverting function: For each measurement (x,y) look at all parameters resulting in a line through (x,y).
- Points with high density in Hough space correspond to parameters which fit for many measurements $\Rightarrow$ the parameters for our line



Conformal mapping

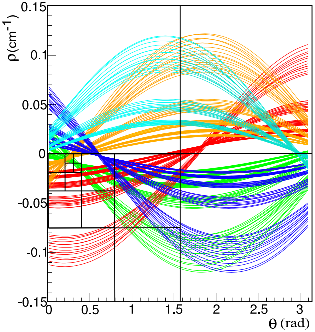
- problems at Belle II:
  - B-field $\Rightarrow$ helix (circle in x-y-plane)
  - in CDC only know distance to hit wire (drift circles) not exact position
- conformal mapping $u = \frac{x}{x^2+y^2}$; $v = \frac{y}{x^2+y^2}$
- properties
  - circles through the origin become straight lines in u - v
  - circles not going through origin stay circles (e.g. drift circles)





CDC case

- use Legendre transformation for Hough space:
  - parameter space representing all tangents to a drift circle
  - $\rho = x_0 \sin(\theta) + y_0 \cos(\theta) \pm R_{Drift}$
- employ fast Quad-Tree-Search for finding high density regions:
  - subdivide parameter space in 4 quadrants
  - pick quadrant with highest density
  - repeat for this quadrant until convergence



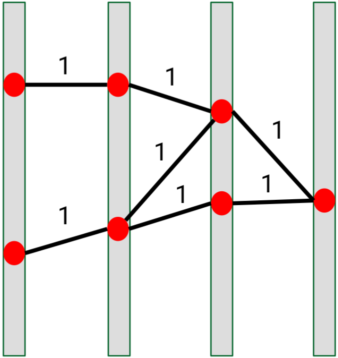
Cellular Automaton

- used for local finding algorithms
- Definition in Literature: "A cellular automaton is a collection of cells on a grid ... evolves through a number of discrete time steps according to a set of rules based on the states of neighboring cells."
- for example "Conways game of life"
  https://youtu.be/C2vgICfQawE
- a bit different usage for track finding:
  - solve longest path problem on a directed acyclic graph
  - fast: $\mathcal{O}(n)$ instead of $\mathcal{O}(n!)$ (for general graphs)

Cellular Automaton

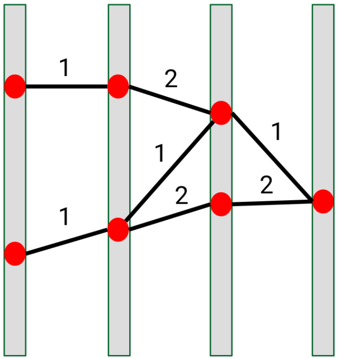
Initialize

- create physical meaningful cells (e.g. connection of hits on neighboring sensors)
- initialize all to state 1



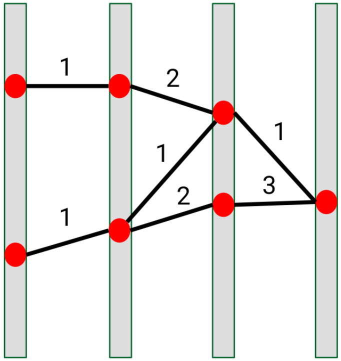
Update if compatible left neighbor with same state

- check e.g. angle between cells for compatibility
- update all cells with compatible neighbor at same time by increasing state by one



Repeat until convergence

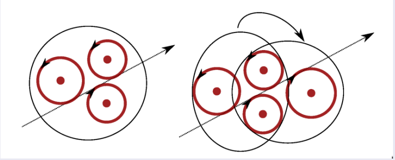
- now state number indicates the length of the path to the left
- select the longest path



CDC track finding

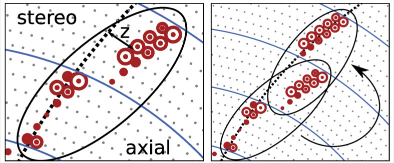
Cellular automaton for segment building in CDC

- segments: shorter track pieces (usually within one super layer)
- start combining triplets of hits assuming straight trajectory



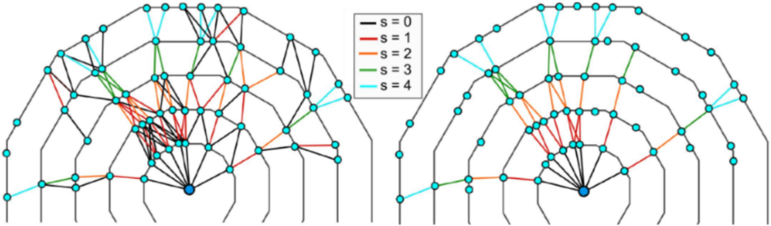
Cellular automaton for track building in CDC

- cell: pair of axial + stereo wire segments
- combining cells into tracks, by selecting longest path



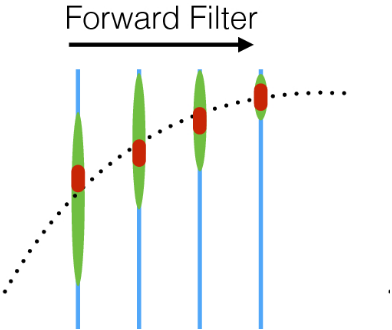
SVD Standalone track finding

- segments (cell): connection between hits on neighboring sensors
- connections of segments are filtered with simple geometric properties: distance, angle, etc.
- Cellular automaton collects longest path
- start from outer - most hits due to less background



Principle of Kalman filtering

- we know how trajectory looks in detector
- starting with a "seed" (initial estimate for trajectory)
- use trajectory to extrapolate to next measured hit location
- use difference to found hit to update our trajectory
- use updated trajectory to repeat with next hit



From: A GPU-Based Kalman Filter for Track Fitting

Use of Kalman filtering in Belle II tracking

Track/Hit Finding

- use track found by one track finder as seed track
- extrapolate track to plane with expected next hit (usually detector plane)
- look for hit in the vicinity of the extrapolation
- if hit found attach it to track, and update track
- repeat
- e.g. PXD hits solely attached through this method
- referred to as Combinatorial Kalman Filter (CKF)

Track Merging

- merge two tracks from different track finders (or detectors)
- use one track as seed
- Kalman Filter to update seed track with hits from other track

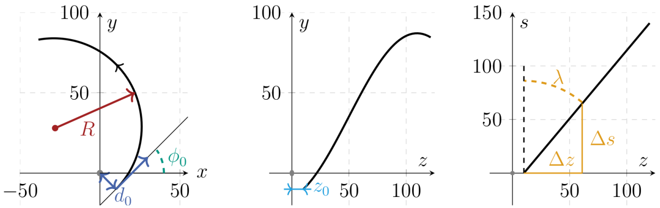
Track fitting

- for analysis only track parameter of interest $\Rightarrow \vec{p}; \vec{x}$ for particle
- track finding only gives you track candidates: "hits (measurements) assumed to belong to a track"
- track fitting tries to find optimal set of parameters for corresponding track hypothesis for given measurements (hits)
- different methods: least squares; Kalman filter; Gaussian sum filter
- fitting tends to be CPU intensive, as material effects (energy loss; Bremsstrahlung) have to be taken into account

Belle II: Deterministic annealing filter (DAF)

- specialized form of Kalman Filter
- iterate Kalman filtering for track candidate
- reject hits farthest away from track in each iteration
- stop when track quality does not improve anymore

What are track parameter?

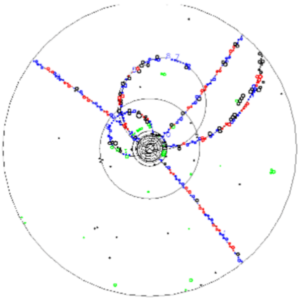


- mdst: track parameters are stored at point of closest approach to IP (POCA), and corresponding covariance matrix (*TrackFitResult*)
- tracks move on helix through detector: described by 5 parameters
  - d_0: signed transverse distance to IP at POCA (sign defined by sign of angular momentum)
  - ω : signed inverse radius of circle in x-y-plane (sign defined by charge): $\omega = 1/R$ (basically $1/p_T$ modulo constant)
  - z_0: distance in z to IP at POCA
  - tan λ: tangents lambda ($\lambda = \pi/2 - \theta$)
  - ϕ_0: azimuthal angle at POCA
- Note: POCA w.r.t. (0,0,0) and NOT measured IP (use *dr* and *dz* for analyses)

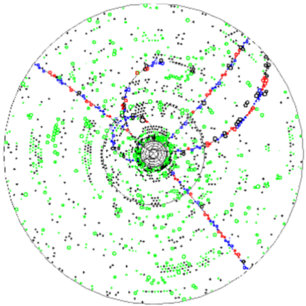
Track Quality

- due to high luminosity sizable machine background:
  - Touschek, beam-gas events, two-photon-events, radiative BhaBha scattering
- Typical event: order of 100 signal hits vs 10000 background hits
- using Multivariate Methods (AI)
  - filter background hits before finding step
  - track quality estimation: reject fake tracks

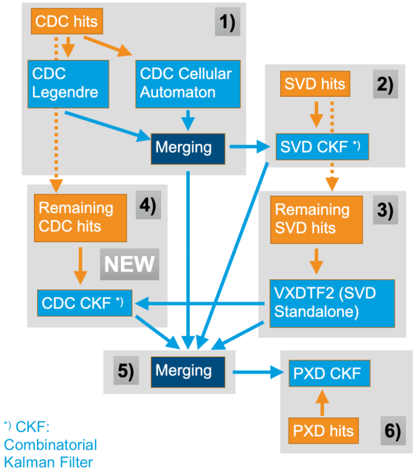
Belle



Belle II



Bringing it all together



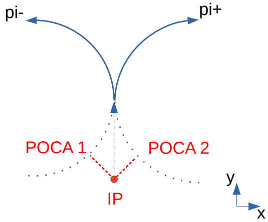
- 2 different tracking algorithms for CDC
- one stand alone algorithm for SVD
- have to combine tracks found in different detectors
- attach PXD hits to tracks

Hit based truth matching

- Monte Carlo simulation (MC) allows to check true nature of reconstructed objects
- tracks are matched based on amount of true hits contained
  - matched: pattern recognition (PR) track contains at least 5% of all true hits, and the fraction of true hits over all hits in track is at least 66%
  - clone: matched tracks, but multiple PR tracks matched to same MC particle. PR track with highest hit efficiency assigned matched, all others as clones
  - fake: tracks failing the above requirements
- Note: also tracks from beam background are classified as fake due to missing MC information (potential real tracks)
- looper:
  - tracks trapped in B-field perform many cycles in detector
  - different cycles often reconstructed as different track candidates (clones)
  - typically have similar track parameter

V0 Reconstruction

- V0 are neutral particles which decay into two charged particles ($K^0_{short} \rightarrow \pi^+\pi^-$; $\gamma \rightarrow e^+e^-$; $\Lambda \rightarrow p^+\pi^-$)
- relatively long lifetime: typically decay after flying several cm
- we store track parameter at IP, problematic for V0 reconstruction:
  - correction of energy loss: momentum not the same at IP and vertex
  - direction of momentum differs between IP and vertex
- dedicated module in reconstruction chain searching for V0
  - parameters of V0-daughters at decay vertex stored (in addition)
  - for analysis use dedicated V0 lists



Summary

- track reconstruction in HEP is a complex topic
- concepts used in track finding presented
- some basic nomenclature introduced
- good (or bad) news: most of the things presented here you dont need to know
- as analyst you will only use final product: track parameter already translated into position and momentum
- some tracking related topics left out:
  - detector alignment
  - PXD data reduction
  - event time and track time estimation
  - vertex fitting
  - tracking related systematics