

Towards an Implementation of ACTS for Track Fitting in the Belle II Experiment

Marc-Philipp Thomas

Masterthesis

28th August 2026

Institute of Experimental Particle Physics (ETP)

Reviewer:	Prof. Dr. Torben Ferber
Second Reviewer:	Prof. Dr. Jan Kieseler
Advisor:	Dr. Giacomo De Pietro

Editing time: 1st September 2025 – 28th August 2026

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

Karlsruhe, 28. August 2026

.....
(Marc-Philipp Thomas)

Disclaimer

Data analyses in high-energy physics such as the measurement presented in this master thesis are a collaborative effort. The SuperKEKB particle accelerator which provides the particle beams essential for all studies at Belle II was built and is operated and maintained by the SuperKEKB accelerator group. The Belle II detector was built and is maintained and operated by the Belle II collaboration. The Belle II collaboration also creates the simulated and recorded data sets and maintains the computing infrastructure necessary to process them. The software environment necessary for studies with Belle II data plays an important role and was created and is maintained by the collaboration. The framework used

This thesis incorporates the use of Artificial Intelligence (AI) tools to help with grammatical and stylistic improvement of text, and program code creation. Grammarly is used in this thesis for spelling and grammar checks and paraphrasing single sentences, to improve the style of writing. I have approved all suggested changes. Claude Code and Gemini Pro are used to aid the development of C++ and Python code, in particular code restructuring and optimisation that do not constitute the core scientific work of this thesis. I have approved and tested all suggestions to provide robust and reliable results.

- Grammarly¹ is used.
- Claude Code ² is used.
- Gemini Pro ³ is used.

¹<https://www.grammarly.com/> (Access date: 28th August 2026)

²<https://code.claude.com/> (Access date: 28th August 2024)

³<https://gemini.google.com/app> (Access date: 28th August 2024)

Contents

1. Introduction	1
2. The Belle II Experiment	3
2.1. The SuperKEKB Accelerator	3
2.2. The Belle II Detector	4
2.3. The Belle II Tracking Detectors	6
2.3.1. The Pixel Detector	6
2.3.2. The Silicon Vertex Detector	7
2.3.3. The Central Drift Chamber	8
3. Track Reconstruction	11
3.1. Charged Particles in a Magnetic Field	11
3.2. Track Parametrization	12
3.3. Track Propagation	12
3.4. Material Effects	14
3.5. Track Finding	15
3.5.1. Track Finding at Belle II	15
3.6. Kalman Filter	16
3.6.1. The Deterministic Annealing Filter	19
4. Software Frameworks	21
4.1. The Belle II Analysis Software Framework	21
4.1.1. GenFit2	22
4.2. ACTS	22
4.2.1. Geometry	23
4.2.2. Event Data Model	24
4.2.3. Propagation	25
4.2.4. Track Reconstruction Tools	25
4.2.5. traccc	26
5. Implementation of the Belle II Detector in ACTS	27
5.1. Beampipe and PXD	27
5.2. The Silicon Vertex Detector	28
5.3. The Central Drift Chamber	29
5.4. Material Mapping	30
5.5. Magnetic Field	32

5.6. Geometry ID Mapping	33
5.7. Navigation	35
6. Track Fitting with ACTS	37
6.1. Workflow	37
6.2. Data Object Conversion	38
6.3. Kalman Filter Configuration	40
6.4. Measurement Calibration	41
6.5. Track Fitting on GPUs with traccc	42
7. Performance Comparison ACTS vs. GenFit2	43
7.1. Setup	43
7.2. Tracking Performance Metrics	45
7.3. Silicon Vertex Detector	46
7.4. Central Drift Chamber	51
7.5. SVD and CDC Combined Fitting	55
7.6. Summary	61
8. Conclusion	63
A. Material mapping validation	65
B. Additional Performance Plots	67

1. Introduction

To date, the Standard Model of particle physics describes the known elementary particles and their interactions with very high accuracy. But although it is the best-fitting theory, it leaves open central questions: What is dark matter? What is the origin of the baryon asymmetry in the known universe? Why do neutrinos have mass? Some of the answers to the open questions are sought at particle colliders.

No new particle has been found since the discovery of the Higgs boson in 2012 [1]. This can have two reasons: The relevant energy scale for new particles lies above the energies of current detectors, or the precision of the measurements is not sufficient to find deviations from the Standard Model. Those reasons result in two complementary search strategies in collider physics. The energy frontier tries to produce new particles directly in high-energy collisions. On the other hand, the intensity frontier measures known particles and processes with high precision to find the smallest contribution of new particles that leads to a shift in the observed values. The Belle II experiment at the SuperKEKB collider searches with the second strategy. It studies the decays of B mesons, charmed hadrons, and τ leptons in e^+e^- collisions. The goal is to collect data corresponding to an integrated luminosity of 50 ab^{-1} , $50\times$ more than the predecessor Belle.

To reach this large amount of data, Belle II set the world record for instantaneous luminosity for any particle collider [2]. But this increase in collisions per unit time comes at a cost: an increase in beam-induced background, which leads to additional hits in the tracking detectors. This larger occupancy leads to a large increase in combinatorics for pattern recognition, especially in the central drift chamber. This translates directly into more tracks in the track fitting stage.

Track reconstruction in the Belle II experiment is split into pattern recognition and track fitting. Pattern recognition is performed by experiment-specific algorithms within the Belle II Analysis Software Framework (basf2). Track fitting is handled by the experiment-independent software GenFit2. GenFit2 has provided reliable and good results for Belle II, but was not designed for multi-threading and hardware acceleration. The software itself is only maintained and not developed anymore. At a time when the high-energy physics field has moved towards common software solutions, Belle II can participate in such projects and profit from the community effort.

One common solution is A Common Tracking Software (ACTS). It is an experiment-independent track reconstruction framework, thread-safe by design and actively developed.

The software is used by many experiments, e.g., ATLAS, sPHENIX, and FASER. The project `traccc` extends the algorithms and data structures from CPU to GPU architecture. However, ACTS was designed in the context of ATLAS with a full silicon-based tracking system and high momenta, whereas Belle II brings drift chambers and low momenta. Drift chambers in ACTS are not entirely new, implementations in planned detectors exist. For example, in the Super-Tau-Charm-Factory [3]. But no application compares ACTS to an established fitter on the same detector.

This thesis provides an implementation of ACTS for track fitting in the Belle II experiment. The geometry of the silicon vertex detector and the central drift chamber is described as an ACTS tracking geometry. The magnetic field is provided for ACTS, and the material distribution is mapped onto the tracking geometry. An interface in `basf2` is built to transfer tracking objects from `basf2`'s pattern recognition to ACTS for track fitting. A proof-of-concept of track fitting in the silicon vertex detector on GPUs in `traccc` is shown. A comparison of the track fitting performance of ACTS and GenFit2 is done in terms of track fitting efficiency, parameter resolution, and runtime. Differences in performance are discussed, and possible solutions are presented.

The thesis is structured as follows: Chapter 2 introduces the SuperKEKB accelerator, the Belle II experiment, and its subdetectors with a focus on the tracking system. Chapter 3 covers the theoretical background of track reconstruction, the pattern recognition chain in Belle II, and introduces the Kalman filter as the track fitting algorithm used in this work. Chapter 4 provides a brief introduction to the software frameworks used in this work: `basf2`, GenFit2, ACTS, and `traccc`, their structures, and the data objects required for track reconstruction. Chapter 5 presents the implementation of the Belle II detector in ACTS, covering the geometry, magnetic field, the process of material mapping, the mapping of the geometry identifiers, and the navigation. Chapter 6 describes the workflow used to fit tracks from `basf2` in ACTS, the configuration of the Kalman filter, and the calibration of measurements, and provides a proof-of-concept of track fitting in the silicon vertex detector on GPUs using `traccc`. Chapter 7 compares the track fitting performance of ACTS to GenFit2 on the same tracks. Chapter 8 summarizes the results and outlines possible next steps for integration into the `basf2` software stack.

2. The Belle II Experiment

The Belle II experiment at the SuperKEKB collider studies the decays of B -mesons and other particles produced in e^+e^- collisions. The physics program of Belle II requires a precise reconstruction of its decay products to search for deviations from the Standard Model [4]. In this chapter, the accelerator facility and the detector components are described, with a focus on the tracking system, which is the most important for this work.

2.1. The SuperKEKB Accelerator

The SuperKEKB accelerator is an asymmetric e^+e^- collider in Tsukuba, Japan, and the successor to the KEKB accelerator. It provides the electrons and positrons that collide inside the Belle II detector [5]. As a B -factory, its main purpose is to produce pairs of B mesons [4]. Figure 2.1 shows the accelerator with its two storage rings, the positron damping ring and the injector.

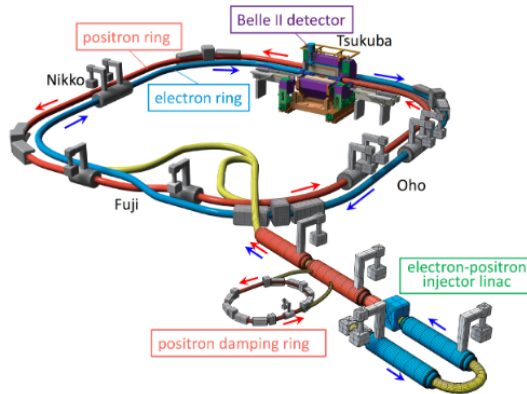


Figure 2.1.: Layout of the SuperKEKB accelerator. The electrons circulate in the high-energy ring, the positrons circulate in the low-energy ring. The beams collide inside the Belle II detector. Taken from [5].

A photocathode gun at the start of the injector produces the electrons. The positrons are produced by directing a part of the electron beam onto a tungsten target. The linear accelerator then accelerates the electrons to 7 GeV, and the positrons to 4 GeV, and they are then injected into the high-energy ring (HER) and the low-energy ring (LER), respectively.

The positron beam's emittance is reduced by a damping ring before the injection into the LER [5].

The resulting center-of-mass energy from the two beams is

$$\sqrt{s} = 2\sqrt{E_{\text{HER}} E_{\text{LER}}} \approx 10.58 \text{ GeV}. \quad (2.1)$$

E_{HER} is the energy of the electrons and E_{LER} is the energy of the positrons. This center-of-mass energy is tuned to hit the $\Upsilon(4S)$ resonance. The $\Upsilon(4S)$ is a bottomonium state whose mass is slightly above $2 \cdot m_B$ and decays into a pair of $B\bar{B}$ with a branching fraction of $> 96\%$ [6]. Since the two beams have different energies, the center-of-mass system is boosted with $\beta\gamma \approx 0.28$ along the z -axis, where β is the velocity of the center-of-mass system and γ is the corresponding Lorentz factor. The boost stretches the flight distance of the B -mesons and separates their decay vertices along the z -axis [5]. This allows time-dependent measurements of the CP -violation [4].

With a design instantaneous luminosity of $8 \times 10^{35} \text{ cm}^{-2} \text{ s}^{-1}$, the goal is to reach an integrated luminosity of 50 ab^{-1} [5]. The actual instantaneous luminosity, however, is one magnitude lower. This large instantaneous luminosity is achieved by the nano-beam scheme. Crossing the bunches at a large horizontal angle makes the overlap region much smaller than the bunches themselves. This allows the beam to be focused to a few tens of nanometers vertically in the interaction point. Additionally, the beam currents are doubled compared to the precursor accelerator KEKB [5]. The size of the dataset is needed to search for small deviations in the Standard Model's prediction on strongly suppressed decays [4].

The high beam currents, combined with the small beam size, also lead to a high beam-induced background. Scattering and radiation processes lead to particles that reach the detector without originating from the collision. This leads to increased occupancy and a worse signal-to-noise ratio, especially for tracking detectors near the beam axis [7].

2.2. The Belle II Detector

Unless stated otherwise, this section follows [8]. The Belle II detector surrounds the interaction point and detects particles produced in collisions between electrons and positrons. Belle II is arranged in multiple cylindrical layers. Each layer corresponds to one sub-detector with a different task. Figure 2.2 gives an overview of the detector components. An important difference to detectors at symmetric colliders is the asymmetric layout of the components of the Belle II detector. Because of the forward boost mentioned in section 2.1, the products of the collision are emitted in favor of the forward direction. The subdetectors extend further on the positive than on the negative z -axis.

The Belle II coordinate system is defined so that the interaction point is the origin. The positive z -axis is the symmetry axis of the solenoid magnet and points approximately in the same direction as the electron beam. The coordinate system is right-handed, the y -axis points upwards and the x -axis points outwards of the detector ring and lies horizontal. The angles θ and ϕ are defined as the polar angle with respect to the z -axis and the azimuthal angle in the x - y plane. Since the detector is approximately cylindrically symmetric,

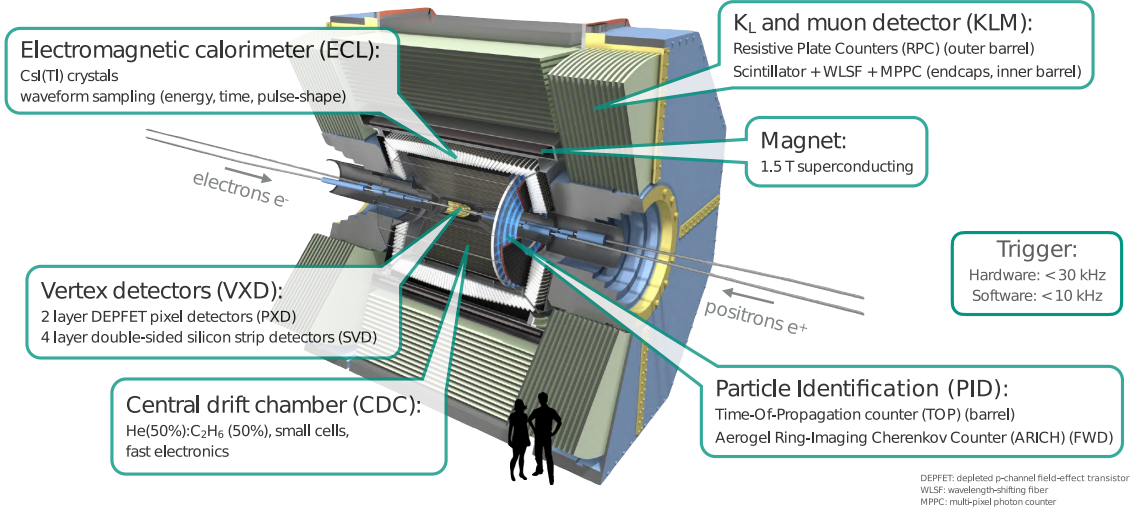


Figure 2.2.: Illustration of components of the Belle II detector. Taken from [9]

$r = \sqrt{x^2 + y^2}$ is the radial distance from the z -axis and is used more frequently than x and y in this work.

The superconducting solenoid provides a magnetic field of approximately 1.5 T along the z -axis for the inner subdetectors.

From the innermost to the outermost layer, the detector is built from the following components. The beampipe surrounds the beam axis and separates the vacuum of the accelerator from the detector. It is made of beryllium to keep the material budget low and therefore reduces multiple scattering.

The pixel detector (PXD) and the silicon vertex detector (SVD) form the vertex detectors. Together with the central drift chamber (CDC), they are the tracking system. Section 2.3 describes them in more detail since they are the foundation of this work.

The time-of-propagation counter (TOP) identifies charged particles in the barrel region. Cherenkov photons in quartz bars are used to determine the velocity of charged particles. Together with the momentum measured by the tracking system, particles are identified. The aerogel ring-imaging Cherenkov detector (ARICH) identifies particles in the forward region. The Cherenkov photons are emitted in an aerogel radiator. These photons form a ring whose radius determines the particle's velocity. By combining the measured velocity and momentum, the mass is determined and is used for particle identification. The TOP and the ARICH together form the particle identification (PID).

The electromagnetic calorimeter (ECL) measures the energy of photons and electrons using thallium-doped caesium iodide crystals. The particles produce electromagnetic showers and deposit energy in the crystal. The scintillation light is proportional to that deposited energy.

The K_L and muon detector (KLM) identifies long-lived kaons and muons. The design interleaves iron layers of the return yoke with active detector elements. K_L shower hadronically in the iron layers, the products of these showers are detected in the active layers. In contrast, the muons pass the iron layers and leave hits in the active layers.

2.3. The Belle II Tracking Detectors

The tracking detectors reconstruct the trajectories of charged particles in the magnetic field and estimate their momenta and origin. The Belle II tracking system consists of three components: the pixel detector, the silicon vertex detector, and the central drift chamber. The combined PXD and SVD is called the vertex detector (VXD). The large extent and therefore the most accurate momentum measurements make the CDC the main tracking detector.

2.3.1. The Pixel Detector

The PXD is the innermost tracking detector and is therefore exposed to the highest rate of beam-induced background and radiation. Strip sensors would have a too large occupancy, for this reason pixels are used. The PXD consists of two layers of DEPFET sensors (Depleted P-channel Field Effect Transistors) [8]. The DEPFET sensors have a thickness of only $75\text{ }\mu\text{m}$ and therefore use less material budget [10].

The two layers of the PXD sit at radii of 14 mm and 22 mm. The two layers consist of 8 and 12 ladders, respectively, with 16 and 24 sensors. During Run 1, only 2 out of the 12 PXD ladders on the second layer were installed. Figure 2.3a shows the layout of the PXD. To minimize material budget, the layers are attached to support structures which sit outside the acceptance of the PXD.

The inner layer sensors have a width of 15 mm and a length of 67.975 mm, the outer a width of 15 mm and a length of 94.975 mm. Each sensor has a sensitive region composed from 250×768 pixels on an area of 12.5 mm by 44.8 mm (inner) and 61.44 mm (outer). This gives a pixel size of $50\text{ }\mu\text{m} \times 50\text{ }\mu\text{m} - 85\text{ }\mu\text{m}$ [10]. The acceptance of the PXD is $17^\circ < \theta < 150^\circ$ [8]. Figure 2.3b illustrates the layout of the single sensors of the PXD.

The PXD has a hit efficiency of approximately 98 % even though about 1.5% of the pixels suffered inefficiencies through radiation damage [10].

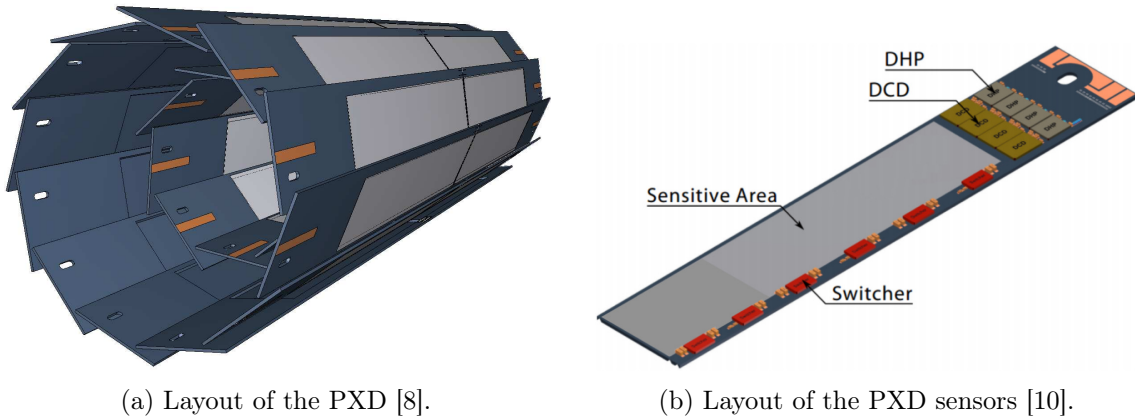


Figure 2.3.: Layout of the PXD and layout of the PXD sensors.

2.3.2. The Silicon Vertex Detector

Unless stated otherwise, this section follows [11]. The SVD consists of four layers of double-sided silicon strip sensors (DSSD). Since the PXD and the SVD together form the vertex detector (VXD), the SVD layers are numbered 3 to 6, continuing the numbering of the PXD layers 1 and 2. The main purpose of the SVD is vertex reconstruction. Because a helix can be described with three points, the four layers of the SVD are also enough for stand-alone track reconstruction. This is important for particles with $p_T < 50$ MeV that do not leave the SVD. The p_T resolution is typically around 7 %. To ensure coverage over the whole azimuthal angle ϕ , the sensors overlap in ϕ . The SVD covers the polar angle region of $17^\circ < \theta < 150^\circ$. A schematic layout of the SVD is illustrated in figure 2.4.

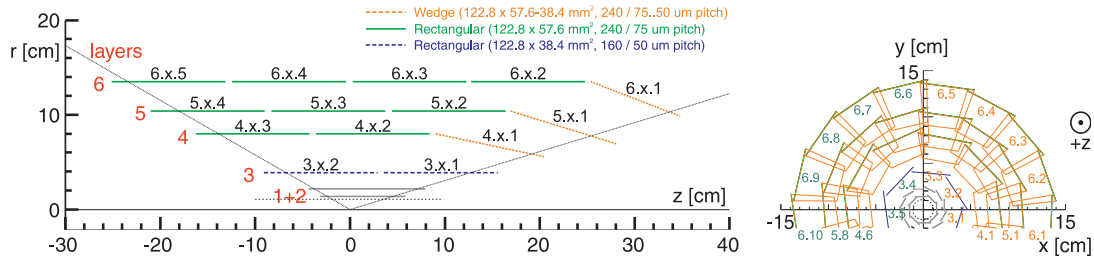


Figure 2.4.: Schematic view of the SVD layout in the r - z plane (left) and the x - y plane (right) [11].

The four layers sit at radii of 39 mm, 80 mm, 105 mm, and 135 mm. The number of ladders and sensors on these ladders increases with the layer and is summarized in table 2.1. The total number of sensors in the SVD is 172, and have an active area of 1.1 m^2 . In every layer except layer 3, the most forward sensor in each ladder is not parallel to the beam axis but slanted towards the interaction point. This design is based on two reasons: Due to the asymmetric energies of the SuperKEKB, the produced particles favor the forward region over the backward region. The second reason is material budget: A particle crossing the sensor at a small angle traverses more material of the sensor than when crossing orthogonally, and since the angle distribution of the produced particles is shifted towards small angles. Each SVD layer adds approximately 0.7 % of radiation length to the material budget.

Table 2.1.: Overview of the layer structure of the SVD.

Layer	Ladders	Sensors per ladder	Sensors total
3	7	2	14
4	10	3	30
5	12	4	48
6	16	5	80
Total			172

The strips on one sensor are rotated by 90° to measure two coordinates. The u -cluster is pointing in the $r\phi$ direction, and the v -cluster is parallel to the z -axis. Both axes form the local coordinate system of the sensor seen in figure 2.5. The strips are implanted into both

sides of an N -type bulk with a thickness of $320\text{ }\mu\text{m}$ for the rectangular sensors, and $300\text{ }\mu\text{m}$ for the trapezoidal sensors. The u -clusters form the acceptor side and are called the P -side, the v -clusters are the donor side, called the N -side. The created electron-hole pairs are separated in the N -type bulk and drift to the strips.

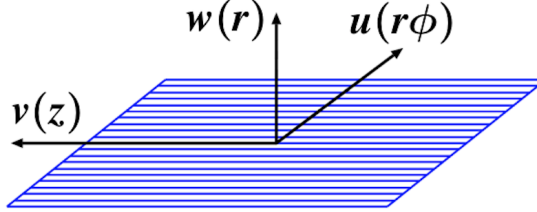


Figure 2.5.: Local coordinate system of the SVD DSSDs [11].

The SVD uses three different shapes of DSSDs: Small rectangular sensors for layer 3, big rectangular sensors for layers 4-6, and trapezoid sensors for the slanted parts in layers 4-6. The exact numbers are shown in table 2.2.

Table 2.2.: Summary of the properties of the different sensor types in the SVD.

Type	Layers	Active area (mm $\times$ mm)	Readout strips (u/P $\times$ v/N)	Readout pitch (μm)
Small	3	122.90×38.55	768×768	50 / 160
Large	4-6	122.90×57.72	768×512	75 / 240
Trapezoidal	4-6	$122.76 \times (57.59-38.42)$	768×512	50-75 / 240

2.3.3. The Central Drift Chamber

The CDC is the main tracking detector in Belle II. It plays a big role in the Belle II reconstruction chain because it not only reconstructs charged particles but is also involved in particle identification and providing trigger signals. Its large lever arm provides precise resolution measurements compared to the SVD. The typical p_T resolution is less approximately 1 %. The minimum required momentum a particle needs to reach the CDC and leave a track is approximately 50 MeV. The energy loss dE/dx provides information for particle identification, which is important for particles that do not reach the PID detectors [8].

The CDC has a cylindrical shape, reaching from $r = 160\text{ mm}$ to $r = 1130\text{ mm}$ and is 2416 mm long. The CDC contains 14 336 sense wires and 42 240 field wires. The length of the wires increases with higher radius to keep the acceptance angle θ constant. The wires are organized into 56 layers, which are again organized into 9 superlayers. The drift cells are the areas around the sense wires limited by the field wire. A schematic view is illustrated in figure 2.6a. The radial size of the drift cells varies between 10 mm for the innermost superlayer and 18.2 mm for the outermost superlayer.

The superlayers alternate between axial- and stereo-superlayers. In the axial-superlayers, the wires are parallel to the z -axis. The wires of the stereo-superlayers are slightly tilted

by the stereo angle. Table 2.3 summarizes the superlayer configuration. The axial layers only measure the trajectory in the $r\phi$ plane. Because of the tilt of the stereo-layers, the z information can be acquired, combining the information of both, a three-dimensional helix is measured.

The CDC is filled with a gas mixture of 50 % helium and 50 % ethane. A charged particle traversing the gas ionizes the gas in the drift cells. The freed electrons experience the electric field of the field wires and drift to the sense wire. The electrons reach the sense wire and create a measurable signal. The quantity of interest is the drift time, which can be translated into the drift length via the $x-t$ relation. This relation is not linear because the electric field in the cell is not homogeneous. The $x-t$ relation is also dependent on the angle and the side of the wire. The $x-t$ relation is shown in figure 2.6b.

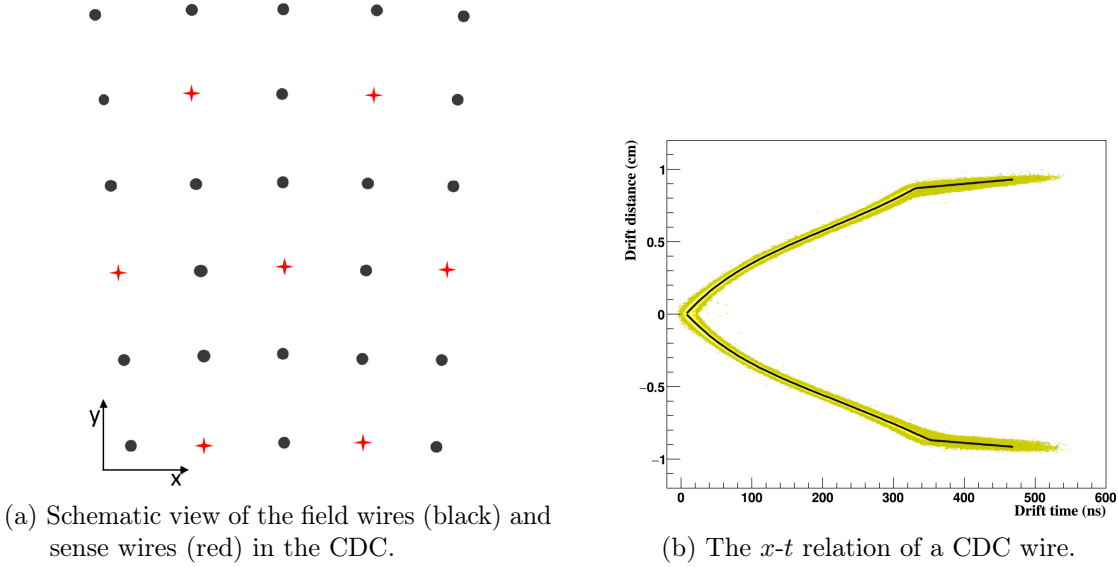
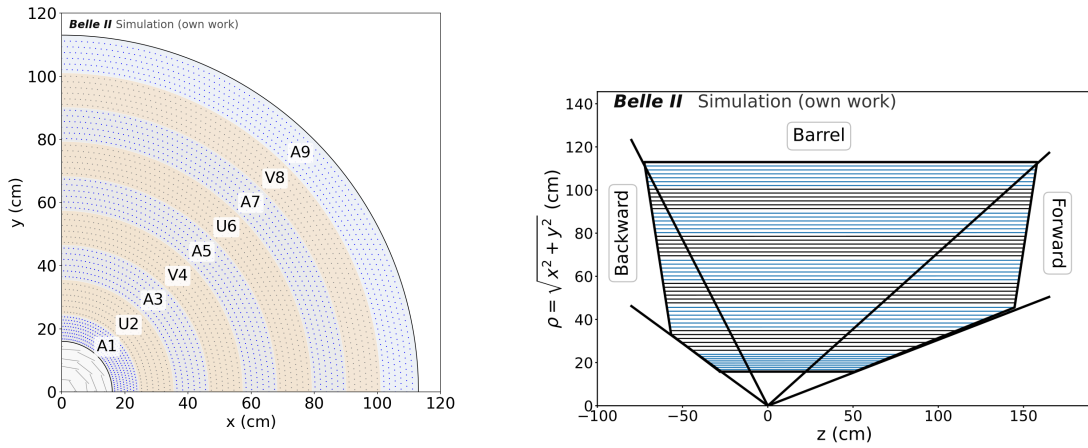


Figure 2.6.: Illustration of the CDC wire theme (left) and the $x-t$ relation for a drift cell (right). Both taken from [12].

Table 2.3.: Configuration of the superlayers of the CDC. Table taken from [8].

Superlayer	Layers	Cells per layer	Radius (mm)	Stereo angle (mrad)
Axial 1	8	160	168.0–238.0	0
Stereo U 2	6	160	257.0–348.0	45.4–45.8
Axial 3	6	192	365.2–455.7	0
Stereo V 4	6	224	476.9–566.9	–55.3––64.3
Axial 5	6	256	584.1–674.1	0
Stereo U 6	6	288	695.3–785.3	63.1–70.0
Axial 7	6	320	802.5–892.5	0
Stereo V 8	6	352	913.7–1003.7	–68.5––74.0
Axial 9	6	384	1020.9–1111.4	0



(a) Illustration of the CDC in the x, y plane. (b) CDC layout in the $r-z$ plane. Taken from [13]. The blue superlayers, labeled "A" are axial superlayers. The orange superlayers, labeled "U/V" are stereo superlayers.

Figure 2.7.: Layout of the central drift chamber in $x-y$ plane (left) and in $r-z$ plane (right).

3. Track Reconstruction

Particle collisions produce new particles that traverse the detector. Identifying the properties of these particles is crucial to investigate particle interactions. To reconstruct the track of a charged particle, a sensitive detector is needed. The particle can interact with the matter in the sensor and produce an electronic signal. By combining these signals into a set, one reconstructs the particle's path and properties, e.g., $\vec{p}$, q , etc. In the following chapter, I will give a brief overview of the fundamentals needed to reconstruct particles.

3.1. Charged Particles in a Magnetic Field

To reconstruct a particle's trajectory, its deflection in the detector's magnetic field is used. A particle with charge q and velocity $\vec{v}$ propagating through a magnetic field $\vec{B}$ experiences the Lorentz force

$$\vec{F} = q \vec{v} \times \vec{B}. \quad (3.1)$$

Since the Lorentz force acts orthogonal to the direction of motion, no work is done; the magnitude of the momentum $|\vec{p}|$ is conserved and only the direction changes.

The Belle II detector has a roughly homogeneous magnetic field of $B = 1.5$ T along the z axis [8]. In this field, the equation of motion separates into two parts: The circular motion in the transverse x - y plane and the uniform motion along the z -axis. The trajectory of the particle forms a helix. The transverse momentum p_T and the radius R have the relation

$$p_T = q B R. \quad (3.2)$$

When expressed in natural units, the relation becomes

$$p_T [\text{GeV}] \approx 0.3 \cdot q \cdot B [\text{T}] \cdot R [\text{m}]. \quad (3.3)$$

A particle with 1 GeV would bend with a radius of about 2.2 m in the magnetic field of the Belle II detector. The bending is expressed by the signed curvature

$$\kappa = \frac{1}{R} = \frac{0.3 q B}{p_T}, \quad (3.4)$$

the sign of κ carries the charge of the particle.

We can parametrize the momentum with the azimuthal angle ϕ and the polar angle θ

$$\hat{p} = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta), \quad (3.5)$$

the momentum splits into its transverse and longitudinal components as $p_T = p \sin \theta$ and $p_z = p \cos \theta$.

3.2. Track Parametrization

A helix has five degrees of freedom, so to describe it completely, we need a parametrization with at least five parameters. The usual choice is the perigee parametrization. It is defined by the point of closest approach (POCA) to a reference surface (for example, the beamline)

$$\vec{\tau} = (d_0, z_0, \phi, \theta, q/p). \quad (3.6)$$

In this parametrization, d_0 and z_0 are the transverse and longitudinal impact parameters, ϕ and θ are the direction of the momentum, and q/p is the charge-to-momentum ratio. The Belle II software uses ω instead of q/p and $\tan \lambda$ for θ . ω is the inverse of the radius of curvature, whereas $\tan \lambda$ is the tangent of the angle between the beamline and the helix in the longitudinal view.

The track fitting in this work is performed with ACTS, which represents trajectories in reference to a surface [14].

$$\vec{x}_{\text{bound}} = (l_0, l_1, \phi, \theta, q/p, t), \quad (3.7)$$

where l_0 and l_1 are the two local coordinates on the surface, and the time t at which the track crosses the surface. Finally, one can also parameterize the helix with free global parameters

$$\vec{x}_{\text{free}} = (x, y, z, t, d_x, d_y, d_z, q/p), \quad (3.8)$$

consisting of the global position x, y, z , the time t , the direction $d_{x,y,z}$.

3.3. Track Propagation

Track reconstruction algorithms require transporting the track parameters through the detector. In a magnetic field and without material effects, the equation of motion as a function of the path length s is

$$\frac{d^2 \vec{r}}{ds^2} = \frac{q}{p} \left(\frac{d\vec{r}}{ds} \times \vec{B}(\vec{r}) \right), \quad (3.9)$$

with the position $\vec{r}$, charge q , momentum $p = |\vec{p}|$ and the magnetic field $\vec{B}$, which follows from equation (3.1) and the conservation of p . Equation (3.9) can be solved analytically for a homogeneous magnetic field and the neglect of material effects [15].

For an inhomogeneous magnetic field, equation (3.9) has to be solved numerically. A common choice to solve second-order differential equations is the Runge-Kutta-Nyström method [16].

The Runge-Kutta-Nyström method of fourth order gives a numerical solution to the second-order differential equation f

$$\frac{d^2 y(x)}{dx^2} = f(x, y, y'), \quad (3.10)$$

by defining four stages in between the step size h at step n and gives the solution [17]

$$\begin{aligned}
\vec{k}_1 &= f(x_n, y_n, y'_n), \\
\vec{k}_2 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}y'_n + \frac{h^2}{8}k_1, y'_n + \frac{h}{2}k_1\right), \\
\vec{k}_3 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}y'_n + \frac{h^2}{8}k_1, y'_n + \frac{h}{2}k_2\right), \\
\vec{k}_4 &= f\left(x_n + h, y_n + hy'_n + \frac{h^2}{2}k_3, y'_n + hk_3\right), \\
y'_{n+1} &= y'_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \\
y_{n+1} &= y_n + hy'_n + \frac{h^2}{6}(k_1 + k_2 + k_3).
\end{aligned} \tag{3.11}$$

Adapting the equation of motion 3.9

$$\frac{d^2 \vec{r}}{ds^2} = \frac{q}{p} \left(\frac{d\vec{r}}{ds} \times \vec{B}(\vec{r}) \right) = f(s, \vec{r}, \vec{T}), \quad \vec{T} \equiv \frac{d\vec{r}}{ds}, \tag{3.12}$$

where $\vec{T}$ is the tangent vector to the track, gives the solutions

$$\begin{aligned}
\vec{T}_{n+1} &= \vec{T}_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \\
\vec{r}_{n+1} &= \vec{r}_n + h\vec{T}_n + \frac{h^2}{6}(k_1 + k_2 + k_3).
\end{aligned} \tag{3.13}$$

where the first solution of equation (3.11) gives the tangent vector $\vec{T}_{n+1}$ at the position of the particle $\vec{r}_{n+1}$, which is given by the second solution [15].

The main parameter is the step size h , which needs to be decreased to obtain accurate results when the particle has strong curvature. The adaptive Runge-Kutta-Nyström method of [17] uses the stages already calculated to estimate the local uncertainty of a step. The step size can be adjusted according to the local uncertainty.

Not only the track parameters have to be transported, but also their covariance. This is typically done by evaluating the Jacobian matrix at the current track state n

$$J = \frac{\partial \vec{x}_{n+1}}{\partial \vec{x}_n}, \tag{3.14}$$

and using it to transport the covariance matrix from track state n to the track state $n+1$ by linear error propagation [18].

$$C_{n+1} = J C_n J^T. \tag{3.15}$$

Figure 3.1 illustrates a step in the Runge-Kutta method.

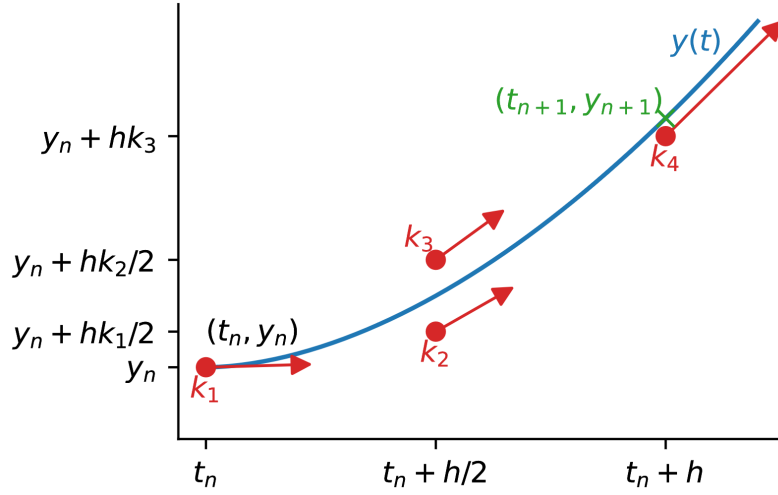


Figure 3.1.: A single step in the fourth-order Runge-Kutta method. The value y_{n+1} at the time t_{n+1} is estimated from the state y_n at the time t_n by calculating k_1, k_2, k_3 and k_4 . Taken from [15].

3.4. Material Effects

The helix from section 3.2 assumes no other physical processes but the interaction with the magnetic field. This does not hold for a particle traversing material in the detector like sensors, support structures or readout boards. The particle interacts with the matter in two relevant ways: through Coulomb scattering off the nuclei, which changes the direction of the trajectory, and energy loss, which only affects the magnitude of the momentum. The traversed material is measured in units of radiation lengths X_0 .

Multiple Scattering

The elastic scattering of the particle with the nuclei results in a large number of small-angle deflections. For those many scattering processes, the displacement and scattering distribution are Gaussian. The scattering distribution has a mean of zero, and the width is given by the Highland formula [19]

$$\theta_0 = \frac{13.6 \text{ MeV}}{\beta c p} z \sqrt{\frac{x}{X_0}} \left[1 + 0.038 \ln \left(\frac{x z^2}{X_0 \beta^2} \right) \right], \quad (3.16)$$

here p is the momentum, βc is the velocity, z is the charge number of the particle. x/X_0 is defined as the thickness of the material in radiation lengths [19]. Important to note is that $\theta_0 \propto 1/p$, so multiple scattering dominates the uncertainty at low p and becomes negligible at high p .

Energy Loss

When heavy charged particles like μ , π , K , etc. traverse through a medium, they also lose energy via ionization and excitation of the medium. The Bethe-Bloch formula [19] describes

the mean rate of energy loss

$$-\left\langle \frac{dE}{dx} \right\rangle = K z^2 \frac{Z}{A} \frac{1}{\beta^2} \left[\frac{1}{2} \ln \frac{2m_e c^2 \beta^2 \gamma^2 W_{\max}}{I^2} - \beta^2 - \frac{\delta(\beta\gamma)}{2} \right], \quad (3.17)$$

where $K \approx 0.307 \text{ MeV} \cdot \text{cm}^2 \cdot \text{mol}^{-1}$, Z is the atomic number, A is the mass number, I is the mean excitation energy, $W_{\max}$ is the maximum energy transfer in a single collision, and δ is the density-effect correction. Equation (3.17) holds for the region $0.1 < \beta\gamma < 1000$. The energy loss in a thin medium fluctuates according to an asymmetric Landau distribution [19].

3.5. Track Finding

The first step in track reconstruction is finding the hits that belong to a track and provide initial starting parameters (seed) for the fit. There is no general solution for this pattern recognition problem yet [20]. The goal of the track finding algorithms is clear: Find as many hits that belong to a particle, without attaching hits that do not originate from this particle. The complexity of this problem increases drastically with increasing background hits and is therefore one of the most challenging parts of track reconstruction.

Approaches to track finding can be separated into global and local methods. Global methods take the hits of a detector as a whole and process them all equally. Local methods group a small set of hits into a seed and attach hits to the track based on this seed [21]. The classes can be seen as complementary methods and are used in the Belle II track finding.

3.5.1. Track Finding at Belle II

The Belle II tracking system combines different detector types and measurements. Therefore, there is a complex chain of different algorithms for track finding illustrated in figure 3.2.

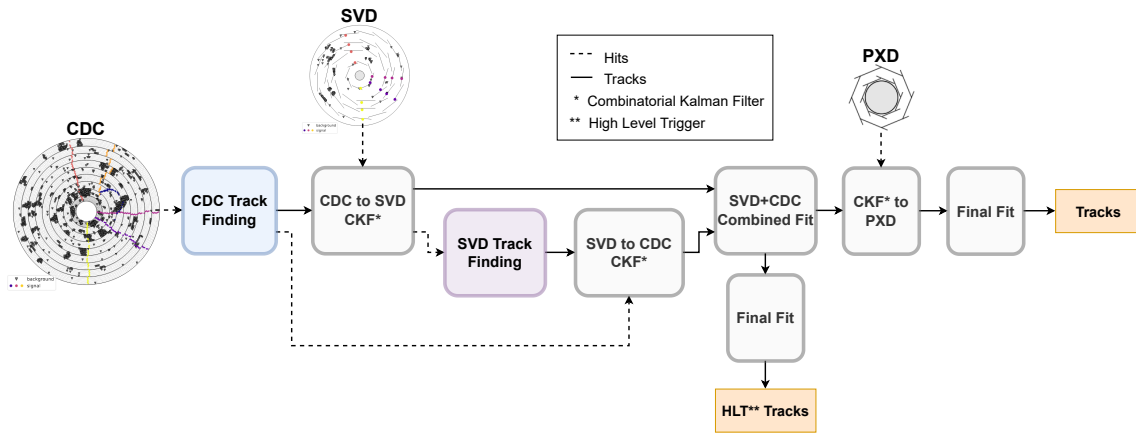


Figure 3.2.: Track reconstruction chain in Belle II for offline and online (HLT) reconstruction. Taken from [22].

The chain starts in the main tracking detector, the CDC: The CDC hits are processed by a global Legendre transformation and a local cellular automaton. Running both algorithms

combines the strengths of the methods: The global Legendre transformation can reconstruct tracks even though hits in several layers are missing. The local cellular automaton's advantage is that it manages to reconstruct displaced vertices or secondary particles with the same efficiency as tracks from the interaction point [23]. The results of both algorithms are merged into one set of track candidates.

Following the CDC standalone finding, the SVD finding is performed. The CDC tracks are extrapolated to the SVD and used as seeds to attach SVD space points using the combinatorial Kalman filter (CKF). SVD spacepoints that are not attached to a track after the CKF are processed in the SVD standalone track finding, the VXD Track Finder 2 (VXDTF2) [23].

After the finding step, the SVD, SVD+CDC, and CDC tracks are fitted using a deterministic annealing filter (DAF) [24]. The fitted track parameters are then used as a seed to attach PXD hits in regions of interest, defined during online reconstruction. The track candidates with the combined hits from the PXD SVD and CDC are again fitted using the DAF [23].

GNN Track Finding

In addition to conventional track finding, there is an alternative approach to substitute this complex chain of track-finding algorithms with a graph neural network (GNN). The CAT Finder [25] reconstructs tracks directly from CDC hits using object condensation [26]. The output of the GNN is the found tracks and an estimation of the track parameters.

The BAT Finder [22] extends the CDC-only track finding of the CAT Finder to SVD hits. Both SVD and CDC hits are processed together by the GNN. The predicted tracks and parameters can then be fitted by a conventional fitter.

Especially for displaced vertices, the CAT and BAT finder perform well above the baseline algorithms. The CAT Finder is available in basf2 since release-11. In all cases, the track fitting still relies on the DAF by GenFit2 [27].

3.6. Kalman Filter

Once a track candidate is found and its hits are associated with it, track fitting is needed to improve the estimation of the track parameters. This can be done through multiple algorithms, but in this work the Kalman filter [28] is used. The Kalman filter is a recursive method of track parameter estimation. The algorithm processes layer after layer, using the least-squares method at each measurement. Material effects can be easily added where it is needed. The notation in this section follows [15].

The Kalman formalism describes a track by its state vector $\vec{x}_k$, which corresponds to the track parameters at k . The matrix C_k is the corresponding covariance matrix at k . The propagation from step $k - 1$ to step k

$$\vec{x}_k = F_{k-1} \vec{x}_{k-1} + \vec{w}_{k-1}, \quad (3.18)$$

F_{k-1} is the transport matrix that propagates the state vector from $k - 1$ through the magnetic field to k . $\vec{w}_{k-1}$ is called the process noise and accounts for multiple scattering.

Measurements in the Kalman formalism are linear functions. The relation between a measurement vector $\vec{m}_k$ and the state vector is

$$\vec{m}_k = H_k \vec{x}_k + \vec{\epsilon}_k, \quad (3.19)$$

where H_k is the measurement mapping function, which projects the state vector into the measurement space. The measurement noise $\vec{\epsilon}_k$ accounts for the specific measurement uncertainties. The associated covariance of $\vec{\epsilon}_k$ is V_k .

The filter performs three steps: prediction, filtering, and smoothing. In the Kalman filter implementation of ACTS, the prediction and filtering are performed at every step of the algorithm. The smoothing is performed once the last measurement is processed by the filtering step.

Prediction

The state vector $\vec{x}_{k+1}^k$ denotes the predicted state vector at step $k + 1$, based on the information at step k .

$$\vec{x}_{k+1}^k = F_k \vec{x}_k, \quad (3.20)$$

$$C_{k+1}^k = F_k C_k F_k^T + Q_k, \quad (3.21)$$

The covariance matrix C_{k+1}^k follows the same notation as the state vector. Q_k is the covariance at step k of the process noise $\vec{w}_k$.

Filtering

The filtering describes the update of the predicted state vector $\vec{x}_k$ with the measurement m_k encountered at step k . The notation $\vec{x}_k = \vec{x}_k^k$ is used for the filtering.

$$K_k = C_k^{k-1} H_k^T \left(V_k + H_k C_k^{k-1} H_k^T \right)^{-1}, \quad (3.22)$$

$$\vec{x}_k = \vec{x}_k^{k-1} + K_k \left(\vec{m}_k - H_k \vec{x}_k^{k-1} \right), \quad (3.23)$$

$$C_k = (I - K_k H_k) C_k^{k-1}. \quad (3.24)$$

The Kalman gain K_k at step k is computed from the covariance of the state vector at step k , which contains all information from previous steps, the measurement mapping function for the measurement at step k , and its covariance at step k . The state vector at step k is a composition of the state vector at $k - 1$ and the residual of the measurement and the projection of the predicted track state into the measurement space, scaled by the Kalman gain. The covariance of the track state at k is also updated with the Kalman gain and the measurement mapping function.

Smoothing

The smoothing step is performed and improves the state vector k with the information of the state vector $k + 1$. This step is performed from the final track state to the first track

state.

$$\vec{x}_k^n = \vec{x}_k^k + A_k \left(\vec{x}_{k+1}^n - \vec{x}_{k+1}^k \right), \quad (3.25)$$

$$A_k = C_k^k F_k^T \left(C_{k+1}^k \right)^{-1}, \quad (3.26)$$

$$C_k^n = C_k^k + A_k \left(C_{k+1}^n - C_{k+1}^k \right) A_k^T, \quad (3.27)$$

$\vec{x}_k^n$ notes the smoothed state vector at step k . The smoother gain matrix A_k with the track state covariance at k and $k + 1$. The covariance is smoothed accordingly.

Outlier Rejection

Outlier detection can be easily applied after the prediction step, before filtering is performed, to quantify the compatibility of the measurement with the prediction. The residual is defined as

$$\vec{r}_k = \vec{m}_k - H_k \vec{x}_k^{k-1}, \quad (3.28)$$

and is the difference between the measurement and the predicted track state projected into the measurement space. The covariance of $\vec{r}_k$

$$R_k = V_k + H_k C_k^{k-1} H_k^T, \quad (3.29)$$

adds the covariance of the measurement and the covariance of the track state projected into the measurement space. The χ^2 -value is obtained through

$$\chi_k^2 = \vec{r}_k^T R_k^{-1} \vec{r}_k. \quad (3.30)$$

The value of χ^2 is used to detect outliers and remove them before the filtering step. This could be useful for detecting incorrectly assigned hits by the track finding step. The sum of all χ^2 of one track also serves as a sign of the quality of the track [28]. This cut is hard: everything above a threshold is removed. So a measurement either enters the filter with its full weight or no weight. Another limitation is that the χ^2 cut is performed only on information from past measurements, leading to an unequal distribution of information.

Figure 3.3 gives a visual explanation of the prediction step of the Kalman filter.

Remarks

The Kalman filter originally presented by Kalman in 1960 is formulated for linear systems [30]. As described in section 3.3, the propagation of track parameters is not linear. The predicted state is therefore not obtained by matrix multiplications, but by numerical integration. F_k is used only for the transportation of the covariance. The algorithm applied in track fitting is therefore called the extended Kalman filter. In track fitting, this extended Kalman filter is commonly called the standard Kalman filter to separate it from the deterministic annealing filter or the Kalman filter with a linearisation around a reference track.

After the smoothing step is complete, all the track states have the information every other track state has. After the forward pass of the predict/filter steps, every track state only

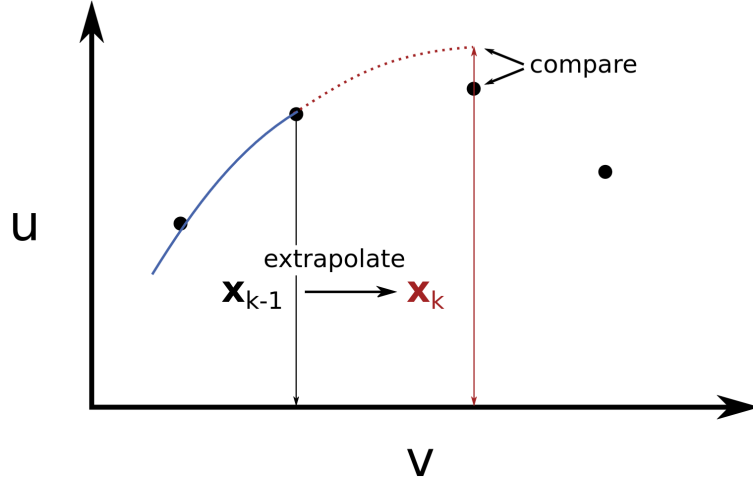


Figure 3.3.: Illustration of the principle of the Kalman filter. The track state x_{k-1} is extrapolated to x_k and the prediction is compared with the measurement. Taken from [29].

has the information of the track states before it. The smoother equalizes these information differences.

The initial track state $\vec{x}_0$ is obtained as the estimated parameters after the pattern recognition; see section 3.5. The initial covariance matrix C_0 is obtained from pattern recognition or created by hand. A common strategy is to inflate the matrix or some elements to lower the impact of a bad set of initial parameters after track finding.

The process noise $\vec{w}_k$ corresponds to the effect on the track state of multiple scattering. Since the expectation value of multiple scattering is zero, this term does not affect the track state directly during the prediction step. The covariance of the process noise term is not zero and therefore enters the covariance of the track state during prediction. The effect of process noise on the track state is mediated by the Kalman gain matrix during the filtering step. The Kalman gain matrix can be interpreted as the degree of certainty in favor of the track state or the measurement. The Kalman gain increases with the covariance of the track state. A large covariance of the track state, for example due to multiple scattering, also inflates the Kalman gain and increases the effect of the measurement on the track state during filtering.

3.6.1. The Deterministic Annealing Filter

The deterministic annealing filter is an extension of the standard Kalman filter for track fitting. It solves the problem of the hard χ^2 cuts by assigning weights to measurements: Measurements with a big residual are weighted lower than those with a small residual. The weights are limited to the interval $[0, 1]$. The DAF is also able to process multiple hits per layer assigned to the same track. The hits are weighted depending on the smoothed track state on the layer. This is particularly useful for hits in drift chambers, where the drift radius defines two points to the left and right of the wire. The DAF then takes

both of them as measurements and distributes weights depending on the residual. The DAF performs multiple iterations: Each iteration is a Kalman filter and a smoother. The annealing corresponds to the temperature T_I at iteration I , which decreases with increasing I . This mechanism determines which of multiple concurrent measurements at one layer is weighted high and which is weighted low [20].

The description of the algorithm follows [20]. An initial set of parameters has to be chosen: ε is the minimal weight a measurement can get assigned, a threshold χ_c^2 that defines if a measurement is declared incompatible with the track, an initial temperature T_0 , and a scheme to lower the temperature. The covariances of the measurements of the first iteration are set to the covariances of the measurements a standard Kalman filter would use: $V_k^j(0) = V_k^j$, where k denotes the layer, and j denotes the j -th measurement in this layer. The first iteration sets all measurement weights to the same initial weight and performs the Kalman filter.

The weights for measurement j in layer k is calculated for each iteration I

$$w_k^j = \max \left[\varepsilon, \frac{\exp(-X_k^j/2T_I)}{\exp(-\chi_c^2/2T_I) + \sum_{\ell=1}^{M_k} \exp(-X_k^\ell/2T_I)} \right], \quad (3.31)$$

$$X_k^j = \left[\vec{m}_k^j - \vec{h}_k(\vec{q}_{k|n}) \right]^T \left[V_k^j(0) \right]^{-1} \left[\vec{m}_k^j - \vec{h}_k(\vec{q}_{k|n}) \right], \quad (3.32)$$

X_k^j denotes the χ^2 -value of measurement j in layer k using the smoothed state vector $q_{k|n}$ and the measurement covariance of the first iteration $V_k^j(0)$. The structure of two residuals and an inverse covariance is similar to equation (3.30). But the definition of the χ^2 -value for the DAF uses the measurement covariance instead of the residual covariance and the smoothed state instead of the predicted state.

The first iteration is finished. The iteration count I is incremented, the covariance of the measurement is set to $V_k^j(I) = V_k^j(0)/w_k^j$, a new temperature is set according to the scheme, and the Kalman filter iterates again. These steps are repeated until the final temperature is reached and the fit converges.

4. Software Frameworks

4.1. The Belle II Analysis Software Framework

The Belle II Analysis Software Framework (basf2) [31] is the accumulated software for the Belle II experiment. It is the successor to basf, used for the Belle experiment, and combines software for simulation, unpacking real data, reconstruction, and analysis. The framework is mostly written in C++ for efficiency, but the user interface is provided via Python bindings. basf2 relies on several external libraries, such as ROOT [32] for data handling, Geant4 [33] for simulation, and GenFit2 for track fitting.

Modules and Paths

basf2 is highly modular. One module performs one well-defined task, for example event simulation or track fitting. Modules are packed and added to the so-called path, which is the sequence of modules that are executed for each event. This path gets assembled in a steering file. Modules do not communicate directly, but instead read and write to the Data Store. This makes it easy to skip or add steps in the reconstruction.

Data Store and Object Relations

The Data Store holds objects or arrays of objects and provides a globally accessible interface for modules to access and modify them. The entries in the Data Store can have a permanent or event-based lifetime.

Besides providing its entries, the Data Store also holds relations between objects. A relation between objects is a direct link that can be used to jump between them and is always weighted. A simple example is the relation between an MCParticle and a SVDSimHit. Relations can be used in either direction and form the foundation for truth comparison in studies with simulated data.

Conditions Data

Information about run conditions, for example alignment or calibration data, is stored globally in a conditions database as payloads. The payload is accessed at runtime via a global tag and has a validity interval.

4.1.1. GenFit2

GenFit2 is an experiment-independent toolkit for track fitting. It was originally developed for the PANDA experiment and is the successor of GenFit. The framework is written in C++ and uses ROOT for geometry navigation and material effects. It is the standard track fitting tool for Belle II and many other experiments [34].

GenFit2 defines three pillars of track fitting: Measurements describing detector hits, track representation describing the parametrization and extrapolation and track fitting algorithms [34].

Measurements

Measurements are represented by reconstruction hit objects containing the raw measurement quantity and their covariance matrix, the detector plane, and the measurement projection matrix for the algorithm. The detector plane can be the actual sensor plane for silicon sensors or virtual detector planes for drift chambers or time-projection chambers [27].

Track Representation

GenFit2 parametrizes tracks by five parameters on a detector plane. This plane is represented by two orthogonal vectors $\vec{u}$ and $\vec{v}$ and the parameters are

$$\vec{x} = (q/p, du/dw, dv/dw, u, v) \quad (4.1)$$

q/p represents the charge over momentum, and the local coordinates on the plane u and v . du/dw and dv/dw define slopes on the plane, w is the normal vector of the plane [27].

The equation of motion is solved by the Runge-Kutta method as described in section 3.3. The transport of the covariance matrix is done by the corresponding Jacobian. Material is looked up from ROOT's TGeoManager, and the effects of multiple scattering and energy loss are applied to the track [?].

Fitter Algorithms

GenFit2 provides several different algorithms for track fitting. Two versions of the Kalman filter are provided: one where the linearization is performed around the track state prediction, and one where it is performed around a reference track. On top of that, an implementation of the deterministic annealing filter and a general broken lines fitter [34].

4.2. ACTS

A Common Tracking Software is an experiment-independent tracking software written in modern C++. It was developed out of the ATLAS tracking software. The original spirit is to provide a framework with the necessary tools and algorithms for track reconstruction, and to be extended to meet any experiment's needs.

The components of ACTS are thread-safe by design. Algorithms that depend on time-dependent detector information use a context object (GeometryContext, MagneticFieldContext, etc.) rather than global states. Where possible, compile-time calculations are used to run faster at runtime.

ACTS is highly modular and allows the user to utilize and extend the components to individual needs. The geometry module allows users to convert detectors from common detector description languages, such as TGeo or DD4Hep, into the ACTS tracking geometry format. Plugins are used to outsource additional dependencies [14].

4.2.1. Geometry

Running track reconstruction with the full detailed geometry is very expensive. ACTS uses a simplified detector geometry called a tracking geometry, consisting of the sensitive surfaces and material surfaces in a detector. Surfaces are the central element in the geometry model of ACTS: every measurement is defined on a surface, and the navigator propagates the track state from one surface to the next one.

Surfaces can have different shapes, e.g., planes, discs, cylinders, and line surfaces. The shapes are combined with bounds that define their extent and placed through transformations. Surfaces are placed in volumes to organize them. Besides organizing surfaces, the main use of volumes is navigation. Each volume has portals that connect it to other volumes. The propagator can only switch volumes through these portals.

Passive material can be included by material mapping. This is done by averaging the passive material in the detector onto dedicated surfaces, so the particle sees an effective amount of material as it propagates through the detector. Surfaces can be marked mappable by assigning a proto material to them. The proto material is replaced with the mapped material after mapping. The material representation in ACTS is illustrated in figure 4.1. The mapped material is used to calculate the energy loss and multiple scattering [35].

ACTS provides different generations of geometries: Gen1 (*layer-based*), Gen2 (*layerless*), and Gen3 (*adaptive*). Gen1 and Gen2 can be seen as opposite philosophies, where Gen1 geometry is fully based on layers as volume containers, Gen2 completely got rid of it and placed volumes freely in other volumes. Gen3 is a mix of both, combining the advantages of layer-based geometries with the flexibility of layerless geometries [36]. In the future, Gen3 will become ACTS' standard, because it combines the advantages of accelerated layer structures with the flexibility of layerless geometries and provides volume-based navigation policies [36].

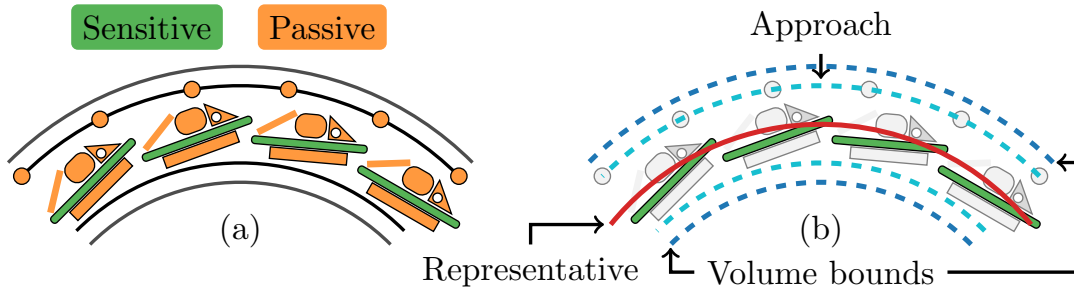


Figure 4.1.: Illustration of the simplification of material in ACTS. Taken from [14].

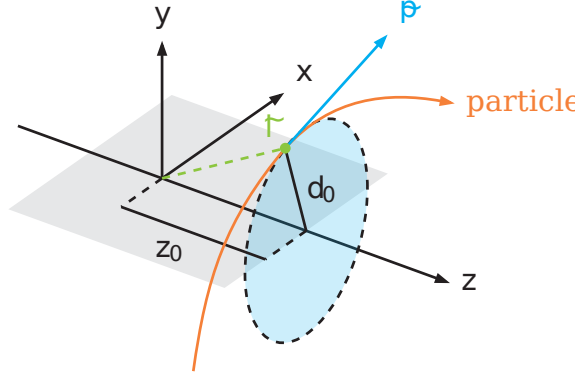


Figure 4.2.: Definition of the perigee surface. The local coordinates of the perigee surface are the impact parameters d_0 and z_0 . Taken from [15].

4.2.2. Event Data Model

Main track parameters in ACTS are bound parameters

$$\vec{x}_{\text{bound}} = (l_0, l_1, \phi, \theta, q/p, t), \quad (4.2)$$

defined with respect to a reference surface. This surface can be a sensor surface, the perigee surface shown in figure 4.2, or any other surface. l_0 and l_1 are the two local coordinates on that surface. ϕ and θ are the azimuthal and polar angles. q/p is the charge over momentum, and t notes the time. An illustration of the bound track parameters is shown in figure 4.3. The free parametrization

$$\vec{x}_{\text{free}} = (x, y, z, t, d_x, d_y, d_z, q/p), \quad (4.3)$$

with the global position, time, direction, and q/p is used to describe the track without a reference surface. Both parameterizations carry a covariance matrix [14].

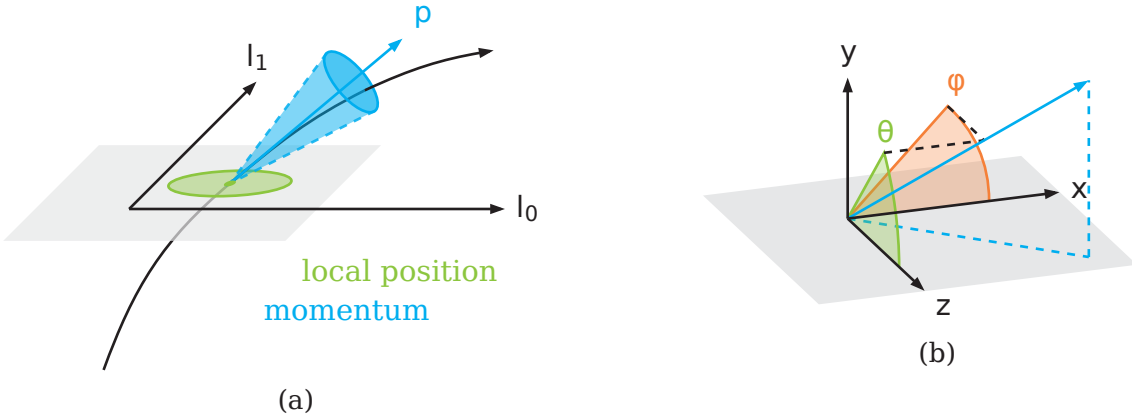


Figure 4.3.: Illustration of the bound parameters l_0 , l_1 , ϕ , θ . Taken from [15]

Measurements

A measurement in ACTS is associated with a surface and contains information about the measurement and its covariance. The measurement is a subset of the bound parameters. What parameter is measured is declared by an 8-bit number called `local_key`. The bits represent parameters and are set corresponding to the measurement. A typical measurement of a pixel sensor contains the parameters `loc0` and `loc1`. A wire in a drift chamber could use `loc0` as the drift length [14].

Track States and Trajectories

When the propagator reaches a surface with a measurement or material, a track state is created. A track state can store the predicted, filtered, and smoothed parameters, their covariances, the calibrated and uncalibrated measurements, and many more. The track states together are held by a container called `MultiTrajectory` [35].

4.2.3. Propagation

The propagation transports the track state through the detector and its magnetic field from surface to surface. The propagator is built from two parts: the stepper and the navigator. The stepper performs the actual propagation, and the navigator decides which surfaces are visited. Actors and aborters are passed to the propagator before propagation begins to handle algorithm-specific tasks. Actors are called at every propagation step and have access to the propagation state. Aborters work similarly and decide whether a propagation should end. The Kalman filter itself is an actor that updates the track state at surfaces.

For a homogeneous magnetic field, the equation of motion can be solved analytically. But since most magnetic fields are inhomogeneous, the equation of motion has to be solved numerically. ACTS uses the Runge-Kutta-Nyström method with an adaptive step size [14].

The navigator gives the propagator the sequence of surfaces along the trajectory. By default, the navigator determines the surface candidates from the volume structure. The navigator enters a volume and finds all the portals leading to other volumes. In cases where the pattern recognition is already done, for example for only track fitting, the surfaces the propagator needs to visit are already known; in this case, the `DirectNavigator` can be used. A surface list with all material and measurement surfaces on the trajectory is created and handed over to the propagator.

4.2.4. Track Reconstruction Tools

ACTS also contains a set of algorithms used for track reconstruction. They are highly modular and can be easily exchanged, so an experiment can adapt the necessary ones while keeping its own implementations. The algorithms cover the whole reconstruction chain: Seeding, track finding, track fitting, and vertexing. For track finding, ACTS offers the combinatorial Kalman filter and machine-learning-based finders. For track fitting, an implementation of the standard Kalman filter, the Gaussian sum filter, and a global χ^2 fitter are available [14].

4.2.5. **traccc**

traccc was developed in the ACTS project as an effort for track reconstruction on heterogeneous hardware. It extends and combines previous results and moves the full tracking chain to accelerators to minimize data transfer between the CPU and the GPU. traccc also follows the experiment-independent design of ACTS and provides features for clusterization, seeding, track finding, and track fitting. The building of the tracking geometry is the only part done on the host system [37]. The tracking geometry for traccc is provided through detrack, a specialized library to provide the detector geometry in a GPU-friendly form. It uses a flat memory layout without virtual functions to maximize efficiency on GPUs. detrack is also part of the ACTS project and follows the same concepts for navigation and propagation. Besides the geometry, traccc also provides track state propagation for GPUs [38].

traccc was successfully tested on the Open Data Detector and achieved good performance in terms of tracking efficiency. The GPU setup even surpassed the computing performance of the CPUs in high pileup conditions. Nevertheless, despite the promising results, the project is still in a rather early stage of development. The studies are limited to the Open Data Detector, and a broader physics performance evaluation is needed [39].

5. Implementation of the Belle II Detector in ACTS

In the following, I will give a detailed description of the implementation of the Belle II geometry in ACTS. Due to reasons described in section 5.2, I chose the Gen3 geometry, since it gives the most flexibility in terms of organizing sensors into volumes.

The tracking geometry consists of four components: Beampipe, PXD, SVD, and CDC, which are described in detail in chapter 2. The beampipe only consists of passive layers to account for material. In this work, the PXD is also implemented as material only and does not carry sensitive surfaces. The following sections describe the tracking geometry from the inside out, followed by material mapping, the magnetic field, the geometry identifier mapping, and the navigation.

5.1. Beampipe and PXD

Neither the beampipe nor the PXD contribute measurements to the fits performed in this work. But both are part of the tracking geometry and contribute to the material budget a particle has to traverse on the way to the first SVD layer. To determine multiple scattering and energy loss correctly, both the beampipe and the PXD have to be implemented into ACTS as passive surfaces with the only purpose of carrying material.

Beampipe

The beampipe is the innermost volume of the detector container. The cylinder volume extends from $r = 0$ mm to $r = 13$ mm in radial direction and from $z = -3000$ mm to $z = 3000$ mm along the z -axis. The large extent in the longitudinal direction is a necessity of ACTS: all subvolumes on one hierarchy level have to have the same z bounds to allow a gap- and overlap-free navigation through the detector.

The passive material surface is a cylindrical surface at $r = 10.5$ mm with a half length of 100 mm in the z -direction. The surface is deliberately shorter than the volume. If it were to span over the whole volume, the binning would be expanded to $|z| < 3000$ mm. The restriction to 100 mm is sufficient, since the tracks considered in this work pass the beampipe at much smaller z because of the limited acceptance.

For the material mapping, the surface gets assigned a proto material with a binning of 72 bins in ϕ and 300 bins in z .

PXD

The PXD has two layers of sensors together with support and cooling structures. The PXD is represented by a cylinder volume extending from $r = 13$ mm to $r = 30$ mm in radial direction and from $z = -130$ mm to $z = 130$ mm. Inside the volume, three cylinder surfaces are placed at 15, 20 and 25 mm and two disc surfaces with $15 \text{ mm} < r < 25 \text{ mm}$ at $z = \pm 150$ mm.

These surfaces do not represent the actual layers in the PXD and are only receivers for the material mapping. The layers are placed close to where the actual material from the sensors, cooling, and support sits. The cylindrical surfaces are binned in 72 bins in ϕ and 400 bins in z , the disks have 72 bins in ϕ and 10 bins in r .

5.2. The Silicon Vertex Detector

As described before, the SVD consists of four layers of double-sided silicon strip sensors, numbered 3, 4, 5, 6 following the Belle II convention. Layer 4, 5 and 6 are not purely parallel to the beam axis, the forward-most sensors are tilted towards the beam axis and cannot be put in a large enough cylinder layer without overlapping. This slanted part is the reason for choosing the Gen3 geometry, because it allows us to directly attach sensors to a volume without a layer.

The position of these sensors comes from a TGeo file exported from basf2. This file contains every element of the detector with its position and material, such as sensors, cooling parts, and structures. Reading the file exported by basf2 guarantees that ACTS and basf2 use the exact same detector geometry, any mismatch between basf2 and ACTS would leave the measurements at different positions and bias the fitted parameters.

From this file, only the sensitive elements are used by ACTS to build the tracking geometry. The nodes are selected by their names, which follow the convention `SVD.<layer>.<ladder>.<sensor>`. Each selected node is then turned into a detector element that owns a plane surface. The surface bounds are then chosen automatically by ACTS. The rectangular barrel sensors are described by rectangular bounds in ACTS, whereas the slanted trapezoid sensors are described through trapezoid bounds. The slant is defined by the rotation of the sensor with respect to the beam axis.

A SVD sensor measures two coordinates independently. One strip measures the u coordinate ($r\phi$ -direction), the other strip measures the v coordinate (z -direction). Each sensor is therefore represented by two plane surfaces, where the back-side sensor is rotated 90° . so that the local coordinate system corresponds to the strips on the back side. This rotation is realized by changing the local axis.

The two surfaces are placed with a distance of $10 \mu\text{m}$. The DSSD has a thickness of $320 \mu\text{m}$, but since both strip sides collect charge from the same ionization trail, the placement of the two surfaces has to be at the position of the sensor and not $320 \mu\text{m}$ apart. Since placing the surfaces at the same position will lead to navigation issues, the gap is chosen as small as possible, but still large enough to allow the navigation to reach the back-side surface.

The sensors of the SVD carry proto material with a binning of 15×15 bins per sensor, this makes it possible to collect the silicon of the sensor and nearby structures and map it on the sensor without the need for additional passive layers for material mapping.

Since the barrel sensors of a layer lie on the same radius, these are grouped into one cylindrical layer, and the Gen3 builder builds a volume from them. The slanted sensors cannot be grouped into cylindrical layers like this, because the volumes of different layers would overlap. For this reason, the slanted sensors are attached directly to the SVD volume. Since the fits in this work use direct navigation, this does not affect the performance.

All four barrel layers and the slanted sensors are contained by the cylindrical SVD mother volume `SVD_Container`, shown in figure 5.1. It extends from $30 \text{ mm} < r < 150 \text{ mm}$ and $|z| < 600 \text{ mm}$. To map material from support structures and cooling, the binning is 120 bins in $r\phi$ and 1200 bins in z . The fine binning of z is a consequence of the large extent of the volume. A 3D model of the SVD in ACTS is shown in figure 5.2.

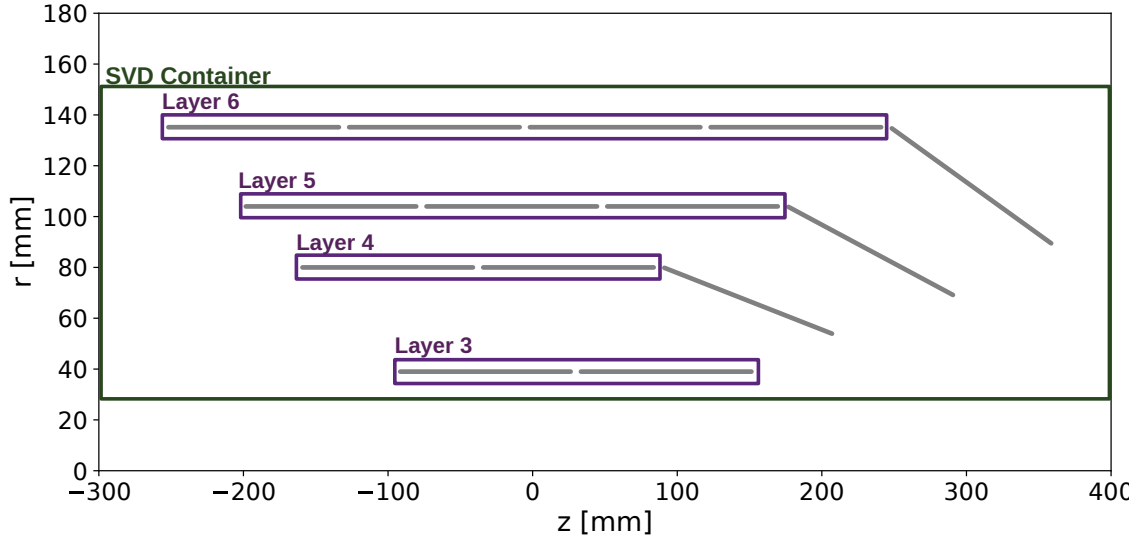
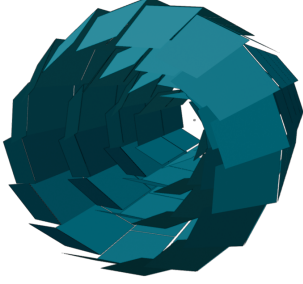


Figure 5.1.: View of the SVD volume structure in ACTS in the r - z plane. The barrel sensors of layer 3 to 6 are grouped into layers shown by the purple boxes. The slanted sensors in the forward part are directly attached to the SVD container volume displayed by the green box.

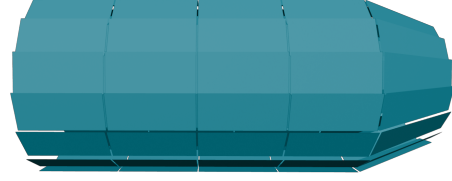
5.3. The Central Drift Chamber

The CDC, as the main tracking detector of Belle II, consists of 14,336 sense wires, organized in 9 superlayers with 56 layers in total, alternating between axial and stereo superlayers.

For the implementation of the central drift chamber, a CSV file is generated with the information from basf2. It contains the wire position and rotation, drift cell size, and indices of the wire, layer, and superlayer, as well as the information if the wire is an axial or stereo layer. Each row in this file corresponds to one sense wire.



(a) Front view of the SVD



(b) Side view of the SVD

Figure 5.2.: Three-dimensional model of the sensitive surfaces of the SVD built by ACTS. Only the sensitive detector elements are shown, the volumes, layers and passive volumes are not included.

From this information, straw surfaces with line bounds are built in ACTS. The radius of the line bounds corresponds to the size of the drift cell. Stereo wires are handled just like axial wires, the rotation matrix takes care of the tilt. A geometry identifier is created and assigned to the wire as described in section 5.6.

The wires are put into 56 cylindrical layers with increasing radius. These layers are contained by the `CDC_Container`. The volume extends from $150 \text{ mm} < r < 3000 \text{ mm}$ and $|z| < 3000 \text{ mm}$. The cylinder carries the material with a binning of 120 bins in $r\phi$ and 100 bins in z .

The material in the CDC is not carried by the wires, but by the portal surfaces of the layers. Each of the 56 layers carries the material of the layer on its inner surface and on the disk portals. The inner cylinder surfaces receive 20 bins in $r\phi$ and 100 bins in z , the disks receive 2 bins in r and 20 bins in ϕ .

A straw surface does not measure a position but the distance of closest approach between the trajectory and the wire. This distance is reconstructed through the $x-t$ relation, already in `basf2`. In this work, the $x-t$ relation is evaluated with the estimated track parameters after finding. An evaluation of this relation for every measurement with the updated parameters during the fit is not performed. Since the drift length leaves a right/left ambiguity, this has to be solved before or during the fit with the estimated parameters after the finding or with the updated parameters during the fit, as described in section 6.4.

5.4. Material Mapping

The tracking geometry described earlier consists of sensitive surfaces, some passive surfaces, and the volumes that contain them. The real Belle II detector, on the other hand, is built from thousands of small volumes: sensors, support structures, cooling parts, cables, etc. Particles traversing these volumes can scatter within them and lose energy, as described in section 3.4. In ACTS, the way to represent this passive material is to map it onto dedicated surfaces, so that a track crossing a surface picks up the material it would traverse in the

real detector. This procedure is called material mapping and is done only once for the detector, the resulting file can be loaded into the geometry before the fitting.

Material Tracks

The first step is to create a set of material tracks. These tracks are propagated from the interaction point through the full detailed Geant4 description of the Belle II detector. The particles in these tracks are called Geantinos, and they record the propagation step, radiation, and interaction length in the volume they cross, but do not interact with the detector. In this work, 10 000 events with every 200 tracks are simulated in the range of $17^\circ < \theta < 150^\circ$, which corresponds to the acceptance of the CDC.

Mapping Surfaces and Binning

While building the detector, a proto material can be assigned to surfaces. This proto material declares that this surface can carry material. For the SVD, these are mostly sensitive surfaces, in the beampipe and PXD, they are passive surfaces. It is also possible to map on the portal surfaces of layers and volumes. This is done mostly for the CDC, but also for the containing volume of the SVD. Binning is used to adjust the granularity. A high number of bins will prevent the material from being smeared over a larger area, but an excessively high number of bins will lead to statistical instability.

The detector can be built in two configurations: one with proto material for material mapping, and one in which the surfaces are decorated with the material. This is controlled by the `buildProto` flag. The flag is set to true once for material mapping. After that, the flag is disabled, and ACTS loads a material map and decorates the surfaces with material.

Mapping

For each material track recorded in the first step, the recorded steps are matched to the closest surface carrying proto material. The radiation and interaction length are then attached to the corresponding bin on this surface. After processing all tracks, the bins are filled with the average accumulated radiation and interaction length of the tracks that crossed it.

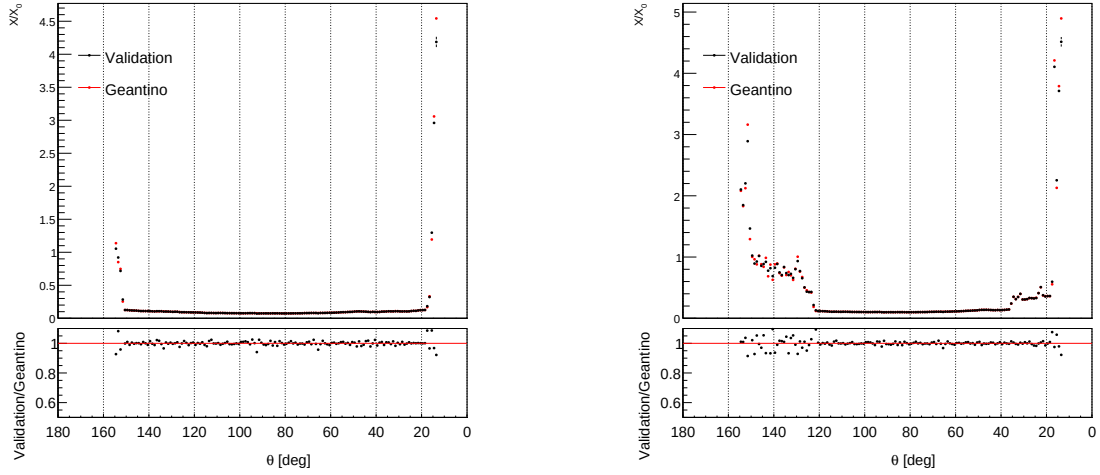
The result is written to a JSON file and is then used to decorate the detector in the next building step.

Validation

To validate this procedure, another set of tracks is generated, but this time with the decorated tracking geometry instead of the detailed Geant4 detector. The material along the tracks is written in the same format as in the tracks before. The number of events is also set to 10 000 with 200 tracks each in $17^\circ < \theta < 150^\circ$.

The resulting X/X_0 of the validation tracks is compared to recorded tracks as a function of θ . In this comparison, it is easy to spot areas where too little or too much material is assigned. For the whole tracking system including the beampipe, this is shown in figure 5.3. Separate plots for single detector components are provided in chapter A. Along with the

validation through the ratio over θ , another method is used: The integrated radiation length over the radius, shown in figure 5.4. This ensures that the material also sits at the correct radius, which is very important for material effects. The bottom row of all four plots shows the difference in the amount of material a particle traverses between the detailed geometry and its approximation in the ACTS tracking geometry. A ratio of 1 means the interaction length a particle experiences is the same for the full geometry and the ACTS approximation, which is the goal of material mapping.



(a) Validation plot for the SVD: X/X_0 over θ . The tracks through the detailed Geant4 geometry are red, the tracks through the decorated ACTS geometry are black.

(b) Validation plot for the CDC: X/X_0 over θ . The tracks through the detailed Geant4 geometry are red, the tracks through the decorated ACTS geometry are black.

Figure 5.3.: Ratio of the radiation length X/X_0 obtained from the ACTS geometry to the one obtained by the detailed Geant4 geometry as a function of θ .

5.5. Magnetic Field

To propagate through a detector in ACTS, it's required to know the magnetic field at every step of the propagator. Any differences in the magnetic field of basf2 and ACTS would lead to a bias in the fitted momentum of a track. For this reason, the field is directly extracted from basf2. A basf2 module writes the magnetic field vector on a grid into a CSV file, which is then read back into ACTS.

The magnetic field in the Belle II detector is, to a good approximation, symmetric to the z-axis. The field can therefore be described with the radial component B_r and the longitudinal component B_z on a two-dimensional grid in r and z .

The grid covers $0 \leq r \leq 3100$ mm and $|z| \leq 3100$ mm with a step size of 10 mm in both directions. The step size is chosen as a compromise between the size of the map and the accuracy of the interpolated values between the grid points. The range extends 100 mm beyond the tracking volume, so that it is not possible for the propagation to leave the grid, which would cause ACTS to crash.

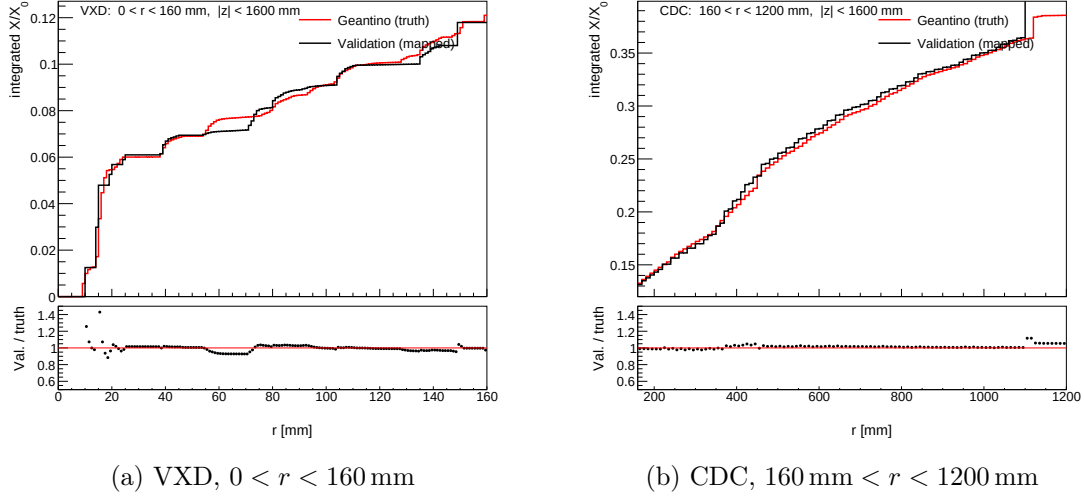


Figure 5.4.: Radiation length X/X_0 integrated along the radius, for the full Geant4 description (red) and from the decorated ACTS geometry (black). The lower panels show the ratio of the two.

The field is then read into ACTS and converted to an interpolated field map. The constructed map is then passed to the stepper.

Validation

To verify the approximation, the interpolated values are compared directly to the basf2 values of the magnetic field. Figure 5.5 shows the verification of the assumptions made.

The interpolation is tested at random points between the grid. The mean deviation of the interpolated map and the basf2 values is below $1.3 \cdot 10^{-4}$ mT in the tracking region. In relation to the field of roughly 1.5 T, this is negligible.

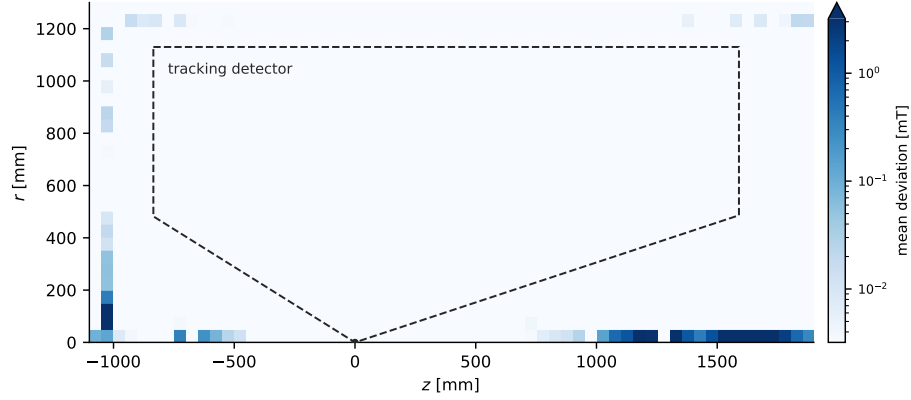
The azimuthal symmetry is probed by evaluating the basf2 field at 16 azimuthal positions for fixed (r, z) and taking the spread of the highest and lowest values of $\vec{B}$. Inside the tracking detector, the spread stays below 0.96 mT. Compared to the magnitude of the magnetic field, this leads to deviations in the order of 10^{-4} .

5.6. Geometry ID Mapping

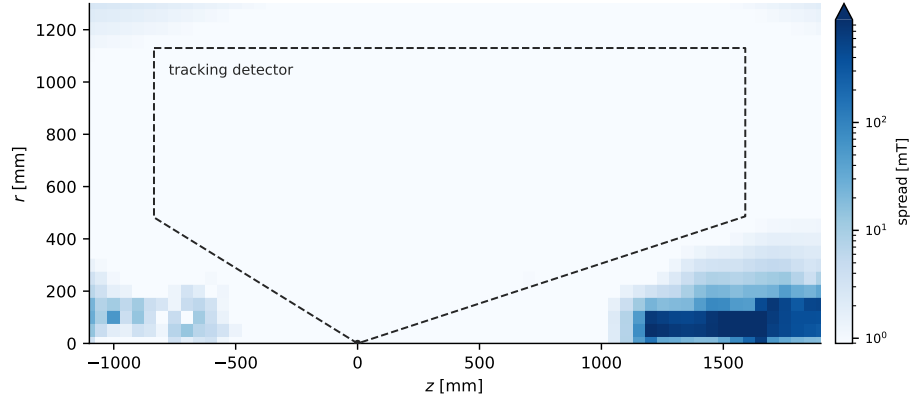
The measurements produced by basf2 and fitted by ACTS are always connected to a surface. Both frameworks identify their surfaces in a different way, mapping them is necessary to allow ACTS to use basf2 generated measurements.

In basf2, an SVD sensor is identified by a `VxdID`, which packs the layer, ladder, and sensor number into a 16-bit integer.

$$\text{id} = (\text{layer} \ll 13) \mid (\text{ladder} \ll 8) \mid (\text{sensor} \ll 5). \quad (5.1)$$



(a) Interpolation accuracy. Mean deviation $\langle |\vec{B}_{\text{ACTS}} - \vec{B}_{\text{basf2}}| \rangle$ between the interpolated map and the basf2 field, evaluated at random positions in between the 10 mm grid.



(b) Azimuthal symmetry. Peak-to-peak spread of $\vec{B}$ over 16 azimuthal positions at fixed (r, z) .

Figure 5.5.: Validation of the magnetic field map in the r - z plane. The dashed area marks the tracking volume. The color scales are logarithmic.

The segment field is not used for SVD sensors and is always zero in this case. A CDC hit is identified by its layer and wire index.

ACTS identifies a surface through a **geometryID**, a 64-bit integer, packing the volume, boundary, layer, approach, and sensitive surface. The value is set automatically by ACTS when creating the geometry, but it is possible to set a custom **geometryID** through a hook. When ACTS closes the geometry after the creation, the hook is called and can change the values.

The hook used for the Belle II SVD accesses the detector element behind the surface, which holds the TGeo node. The naming convention of the TGeo nodes is

SVD.<layer>.<ladder>.<sensor>.Active

so the layer, ladder, and sensor can be obtained by splitting the TGeo node name. The

information is then packed into a 16-bit integer following equation (5.1). Adding four bits set to zero, this 20-bit integer is placed as the sensitive field in the `geometryID`. The remaining fields of the ACTS ID remain untouched and keep the values ACTS assigned. It is important to note that the ACTS layer does not correspond to the SVD layer. Since every SVD sensor consists of two ACTS surfaces, the lowest bit of the sensitive surface is set according to the front or back of the sensor. After this step, the creation of the `geometryID` of the SVD sensors is complete. Figure 5.6 illustrates the two different identifiers and how they are mapped.

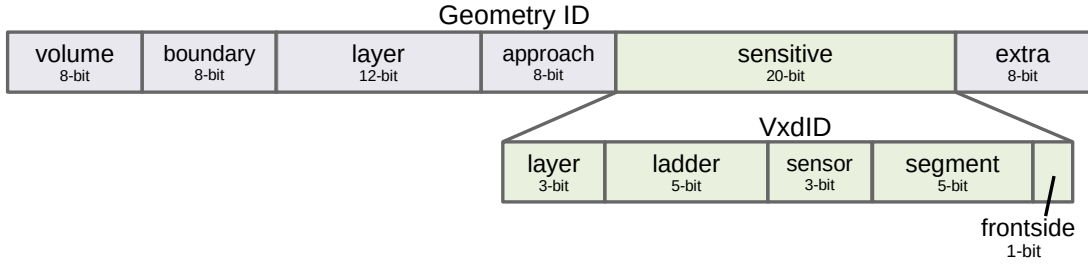


Figure 5.6.: Construction of the geometry identifier of SVD surfaces. The upper bar shows the 64-bit `GeometryID` of ACTS with six fields. The lower shows the `VxdID` of basf2 with 4 fields plus the 1-bit setting the frontside/backside bit.

The CDC is handled differently. The straw surfaces are not built from TGeo nodes and therefore have no node name. The identifier is directly set through the layer and wire index from the CSV file when creating the detector element, so no hook is needed for the CDC `geometryID`.

5.7. Navigation

To propagate the track state through the detector, the propagator must know what surfaces are on its way through the detector. The navigator's task is to provide a set of the surfaces the trajectory is expected to cross.

Standard Navigation

The standard ACTS navigation takes advantage of the hierarchy structure of the geometry. Starting from the current volume, it collects the surfaces of the layers that are intersected by the trajectory and orders them by path length. If the trajectory leaves a volume through its portal, it enters the neighboring volume. For this to work, the volumes must not leave gaps or overlaps.

For good estimated parameters, this navigation method works well. But if the estimated parameters are bad, it is possible that the predicted trajectory does not cross the actual measurement surfaces. Especially in low p_T , this leads to a significant loss in track fitting efficiency, due to missed measurements. Another disadvantage of this navigator is runtime: checking many more surfaces and sorting them by path length require many operations and thus increase runtime.

Direct Navigation

When ACTS is only used for track fitting, it is not necessary to navigate through the whole detector volume for a single track. It is enough to pass ACTS the surfaces with measurements and material intersected by the track. The surfaces are then visited one by one in the order of the surface list. This method removes the dependency of the estimated parameters' quality and is faster than the standard navigation.

The sequence of the surfaces has to be handed over to the direct navigator at the call. The sequence must include all the measurement surfaces and material surfaces the track traverses, in the correct order. The measurement surfaces are provided by the prototrack, which is a container storing references to the measurements that are part of the track. The material surfaces are picked by propagating a straight ray from the vertex of the particle in the direction of the momentum. Every material surface the ray crosses, with a smaller radius than the last measurement surface of the track, is added. This prevents wrong material assignment in the case of curlers. Material surfaces that also carry a measurement, like the SVD sensors, are only kept once.

The measurement surfaces are kept in the same order as the track finder hands them over via the prototracks, because they are already sorted. The material surfaces are placed depending on their path length along the helix of the estimated parameters.

The straight ray approximation for the selection of the surfaces only holds for tracks with a large enough p_T . Tracks with a strong curvature could in principle get the wrong material surfaces assigned to them. But since the material in the CDC sits on the layer portal surfaces, which are cylindrical, it is irrelevant since the surface is reached by every track with sufficient momentum, no matter the direction. The position on the surface, and thus the bin for the material, is resolved by the direct navigator at the moment it reaches the surface. This does not hold for curlers. The material surfaces added to the track are only the ones the track traverses on the way to the furthest point of the interaction point. The material surfaces traversed on the way back into the detector will not be added.

6. Track Fitting with ACTS

6.1. Workflow

In this work, simulation, digitization, and track finding are done in basf2, ACTS is only used for track fitting. Since the two frameworks are not yet connected through wrappers, the exchange is done via CSV files. Figure 6.1 shows the overall workflow.

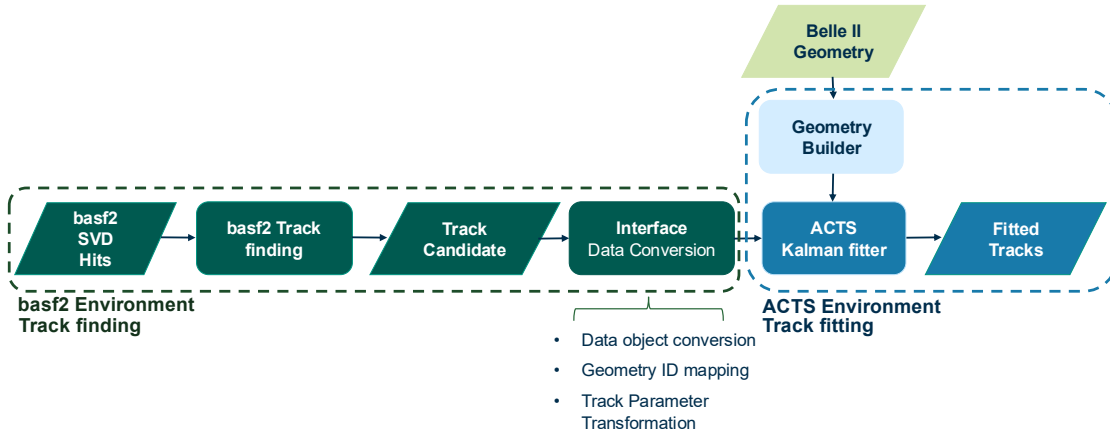


Figure 6.1.: Workflow of the project. The green space marks the basf2 environment, and the blue space marks the ACTS environment. The track finding is done in basf2, and the fitting is done in ACTS.

basf2-ACTS Interface

A basf2 module writes the objects needed for fitting in ACTS into CSV files. Each event produces six files with the naming convention `event<number>-<object>.csv`.

The *measurements* contain the local coordinates and their covariances, the **geometryID** of the measurement surface, and the global coordinates of the measurement. The *prototracks* contain the list of measurements for each track found by basf2, along with the global position of those measurements. The *estimated parameters* contain the track parameters and their covariance after track finding. Also the *particles*, *simhits* and *measurement-simhit-map* are exported, ACTS does not need them for fitting, but they are used to match the tracks and hits to their truth for use in the evaluation of the track performance.

Exporting to CSV files rather than implementing ACTS into the basf2 externals and calling ACTS through wrappers allows a considerable simplification of debugging in ACTS. The price of this setup is performance across almost every part of the workflow: Writing from basf2, reading into ACTS, and writing from ACTS and then reading the files for evaluation are time-consuming I/O operations.

ACTS Side

The ACTS sequence contains a series of algorithms that are executed for each event. The algorithms read data from the whiteboard, perform their task, and then write the results to the whiteboard. The whiteboard is ACTS' event data storage. The tracking geometry, including the material decoration and the magnetic field, is constructed only once before the event loop and is shared by all events.

The event loop first reads in the data from the CSV files. The proto tracks, measurements, and the estimated parameters are passed to the Kalman Fitter, whose configuration is described in section 6.3. After the fit, the fitted tracks are selected to require a minimum number of hits. For a good comparison, the number in ACTS is the same as the number in the Kalman Filter in basf2. The resulting tracks are written into a CSV file. The written tracks can then be analyzed and compared between ACTS and GenFit2.

Since the readers and writers for the estimated parameters and the fitted track parameters are very specific for this workflow, they were implemented into ACTS as a part of this work.

6.2. Data Object Conversion

The data formats used by ACTS do not exist in basf2. The data objects mentioned above need to be assembled from one or multiple data objects in basf2. This section describes how the conversion of these objects is handled in a basf2 module. Figure 6.2 shows which basf2 data objects contribute to which ACTS data objects.

Measurements

Measurements in ACTS contain the measurement identifier, the **GeometryID** of the surface, a set of local coordinates and their covariances, and a key that declares which parameter was measured through a bit mask.

For the SVD measurements, the key is set to 2, which means only the l_0 parameter is measured. An important note: The key does not change for u or v clusters. It stays the same because the detector element and its local coordinate system is rotated by 90° . Each **SVDCluster** gets converted into one measurement, which gives two one-dimensional measurements per sensor. The **GeometryID** is set by converting the **VxdID** as described in section 5.6, with the lowest bit set according to the side of the sensor.

For the CDC, the measured quantity is the drift length. It is obtained from the drift time of a **CDCHit** through the $x-t$ relation in basf2. The key is also set to 2, since a straw surface only measures the distance of closest approach to the wire. The drift length is unsigned, so at this point it is not known on which side of the wire the track passed. The R/L ambiguity is solved during the fit as described in section 6.4.

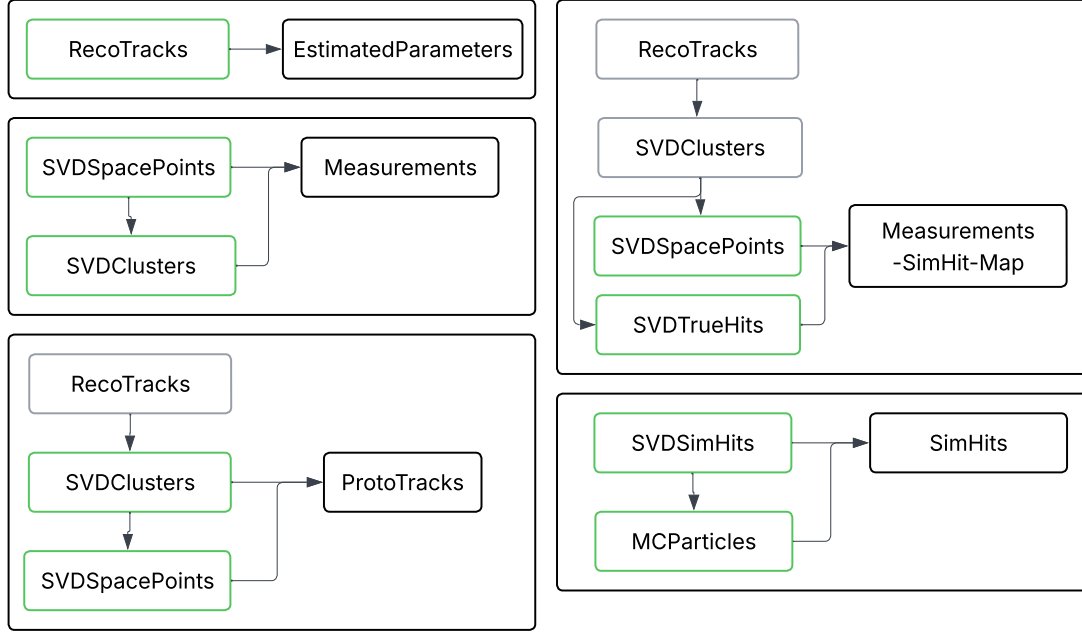


Figure 6.2.: Overview of the connection between basf2 data objects (left side in the boxes) and ACTS data objects (right side in the boxes) for the SVD. Each box represents one object in ACTS.

basf2 quantities use cm, ACTS uses mm. This has to be respected by scaling the coordinates by a factor of 10 and the covariance by a factor of 100.

Prototracks

A proto track is the list of measurements that belong to a track candidate. For each **RecoTrack** after track finding, the hits are collected and attached to a prototrack. So a prototrack is converted one-to-one from the **RecoTrack**.

Estimated Parameters

The estimated parameters are the starting point for the fit. They are obtained after track finding from the **RecoTrack** object. ACTS and basf2 use different parametrizations, ACTS uses bound parameters, whereas basf2 uses the helix parameterization. The parameters and their covariances thus have to be converted. The exact transformation is given in table 6.1

The covariance matrix is transformed in the same way. The covariance matrix in ACTS parameters is given by

$$C_{\text{ACTS}} = J C_{\text{basf2}} J^T \quad (6.1)$$

with J as the Jacobian of the transformation of table 6.1

$$J_{ij} = \frac{\partial p_{\text{ACTS},i}}{\partial p_{\text{basf2},j}} = \begin{pmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\sin^2 \theta & 0 \\ 0 & 0 & 0 & \frac{-10\omega \sin^2 \theta \cos \theta}{0.3B_z} & \frac{100 \sin \theta}{0.3B_z} \end{pmatrix}, \quad (6.2)$$

Truth Information

The truth information is not needed for the fit, but for the evaluation after the fit. The `SimHits` are converted from the `SVDSimHits`/`CDCSimHits` together with the `MCParticles`, so every simulated hit carries the relation to the particle. A map is created relating every measurement to its simulated hit. Through this map, it is possible to link a measurement to a particle in the evaluation after the fit.

Table 6.1.: Correspondence between the helix parameters used by basf2 and the bound track parameters used by ACTS, which are defined with respect to the perigee surface. The first three rows are identities up to the unit conversion, the last two require a computation.

basf2	ACTS	Relation	Unit
d_0	loc_0	$\text{loc}_0 = d_0$	cm $\rightarrow$ mm
z_0	loc_1	$\text{loc}_1 = z_0$	cm $\rightarrow$ mm
ϕ_0	ϕ	$\phi = \phi_0$	rad
$\tan \lambda$	θ	$\theta = \text{atan2}(1, \tan \lambda)$	rad
ω	q/p	$q/p = \omega \sin \theta / (0.3 B_z)$	GeV $^{-1}$

6.3. Kalman Filter Configuration

After assembling the geometry and supporting the data, the Kalman filter itself needs configuration. This work uses the standard Kalman filter in ACTS. The measurements are attached to the tracks in basf2 and ACTS only uses these measurements, no combinatorial work is done.

Stepper, Propagator and Navigator

The track state propagation between the surfaces is done by the standard ACTS stepper. The navigator decides which surfaces the propagator encounters. Since the set of surfaces is known before the fit, the direct navigator is chosen over the standard geometric navigator. In case the list of surfaces built for the direct navigation contains a surface that cannot be resolved, the standard navigation is used.

Multiple scattering and energy loss are both enabled for the fit. Multiple scattering enters the fit through the material surfaces in the surface list. In this work, all tracks are fitted with the pion hypothesis. A hypothesis is required to translate p for the Bethe-Bloch formula equation (3.17) and account for the energy loss during the fit.

A path limit is set dynamically to prevent a track from leaving the magnetic field at the boundary $r = 3100$ mm: The path limit scales to the track as $1.5 \cdot s_{\max}$, with $s_{\max}$ being the path to the outermost surface. The minimal path length is set to 1500 mm and the maximum is set to 2800 mm.

Starting State

The starting track parameters are obtained from basf2 after track finding. Converted as described in section 6.2 and defined on the perigee surface. The covariance is taken from basf2 and transformed with the Jacobian as shown in equation (6.1).

Measurement Calibration and Outlier Rejection

After the prediction, the measurement on the surface is passed through a measurement calibrator. The calibrator distinguishes between straw surfaces and plane surfaces. Measurements on a plane surface get passed through the calibrator unchanged, measurements on a straw surface get calibrated to solve the R/L ambiguity. The exact calibration is described in section 6.4.

After the calibration but before the update, the outlier finder is called. The predicted state and the calibrated measurement are compared, and the χ^2 -value is calculated. A measurement with a χ^2 -value greater than the configured cutoff is rejected from the fit.

Smoothing

After reaching the last measurement and thus completing the forward pass, the track is smoothed so that every state has the information of all measurements attached to the track.

ACTS offers reverse filtering as an alternative for low p_T tracks, but it is not used in this work.

6.4. Measurement Calibration

The measurements in the SVD and the CDC differ in nature. A SVD cluster states where a particle hit the sensor. A CDC hit states how far away the particle passed the wire, but not on which side. Solving this L/R ambiguity is essential for CDC fitting and the reason why they need to be calibrated.

The Measurement

The CDC does not measure the drift length directly. Instead, the CDC measures the drift time, the time electrons take to reach the wire. The drift time can be translated into the drift length via the $x-t$ relation. The relation is evaluated in basf2 before the fit using the seed parameters. It would be preferable to evaluate this relation during the fit with the updated parameters, but this is not implemented in this work. The drift length exported by basf2 is unsigned, so it seeds calibration to solve the right-left ambiguity.

Present Handling

The `DriftChamberCalibrator` is called at every surface before the update and receives the measurement and track state. The uncalibrated measurement is the drift length without a sign. The predicted local coordinate on the straw surface is signed and states on which side of the wire the prediction passes. This sign is transferred to the drift length of the measurement.

The calibrator only applies this mechanism to straw surfaces, so the SVD measurements remain untouched by this calibration.

Robustness

This mechanism is only good as long as the prediction of the track. Especially for the first wire in the track, the result depends only on the seed, with no measurements supporting it. The result is that about 8% of the first measurements are assigned to the wrong side.

Once the trajectory gets pulled to the wrong side, this effect is self-reinforcing since the track covariance gets smaller over time, but the measurement covariance stays the same, so the fit trusts the track more. Since no outlier rejection is used in this work, a measurement that contradicts the trajectory is not rejected, but pulls the trajectory further.

The effects vary in severity. For wires with a small drift length, the two possibilities are close together, so it is more likely to be miscalibrated. But since the effect on the trajectory is also small in this case, the next measurement can recover the track. The expected effect is only a decrease in resolution. For a large drift length, the sign is less likely to be wrong, but the effect on the trajectory is stronger, which makes further wires more vulnerable to miscalibration.

6.5. Track Fitting on GPUs with `traccc`

The described implementation runs fully on CPUs. The ACTS tracking geometry can also be used to convert it to `detray` and run the track fitting with the Kalman filter on GPUs. This section describes a proof of concept for SVD-only track fitting on GPUs using `traccc`.

Although ACTS and `traccc` are very closely related, they do not share the same geometry description. The description used for ACTS is optimized for CPUs, whereas the description used by `traccc` is optimized for GPUs. `traccc` uses `detray`, which stores the detector information in a flat container of surfaces. The first step towards track fitting on GPUs is converting the tracking geometry to the `detray` geometry. Because the SVD consists of planar sensors and the measurements don't need calibration, the SVD is the most suitable for such a test. The tracking geometry is built in ACTS and converted to `detray` by a short script. The converted geometry is then checked against the corresponding tracking geometry in ACTS. With the SVD geometry in `detray`, the Kalman filter is executed on the GPU using example data. The fit runs successfully without problems.

The scope of this test is very limited to the SVD and does not include quantitative evaluation of the track fitting efficiency or resolution. The test showed that the conversion from a tracking geometry to `detray` works and the obtained detector geometry can be used for track fitting with the Kalman filter on GPUs.

7. Performance Comparison ACTS vs. GenFit2

7.1. Setup

The setup for ACTS was described in chapter 6. To compare the performance of ACTS track fitting to GenFit2, both fitters run on identical input: Simulation, digitization, and track finding are performed only once in basf2. The results are handed over to the fitters. GenFit2 is called directly in basf2, whereas the data for ACTS are exported from basf2 and imported into ACTS. Therefore, both fitters process the exact same hits and tracks. The fit results are saved and then evaluated. The workflow of the comparison is illustrated in figure 7.1.

Datasets

The comparison is done with two different datasets. The first set generated is a particle gun example with 10 000 events with five tracks in each event. The particles used are pions (π^+). The momentum range of the pions is $p_T = [0 - 4]\text{GeV}$, the momentum is uniformly distributed. The particles are generated in the acceptance range of the tracking detectors $17^\circ < \theta < 150^\circ$ and are uniformly distributed. The vertex of the particles is set to the interaction point (0,0,0). No background is added. The properties of the distribution are summarized in table 7.1.

The second dataset used in this work is 10 000 mixed B -events. The conditions are run 0, experiment 1003, which correspond to Run 1. Beam background is overlaid to the dataset. This dataset provides more realistic settings for reconstruction: The average

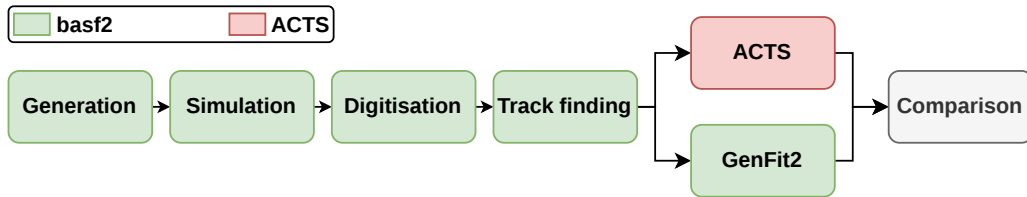


Figure 7.1.: Workflow used for the comparison of ACTS and GenFit2. Generation and track finding are done in basf2, the same data is fitted with ACTS and GenFit2.

number of tracks is roughly 11 per event, the momentum distribution is strongly shifted to low momentum. The θ distribution is not limited to the acceptance range and is right-tailed. The properties of the distribution can be seen in table 7.1. Figure 7.2 shows the momentum and polar angle distributions for both samples.

Table 7.1.: Comparison of the two datasets. Only particles with at least one hit are considered in this table.

Dataset	Tracks/event	Mean p_T [GeV]	Median p_T [GeV]	Mean θ [$^\circ$]	Median θ [$^\circ$]
Particle gun	5.05	1.495	1.344	81.5	81.4
Mixed B	11.03	0.405	0.319	74.8	72.3

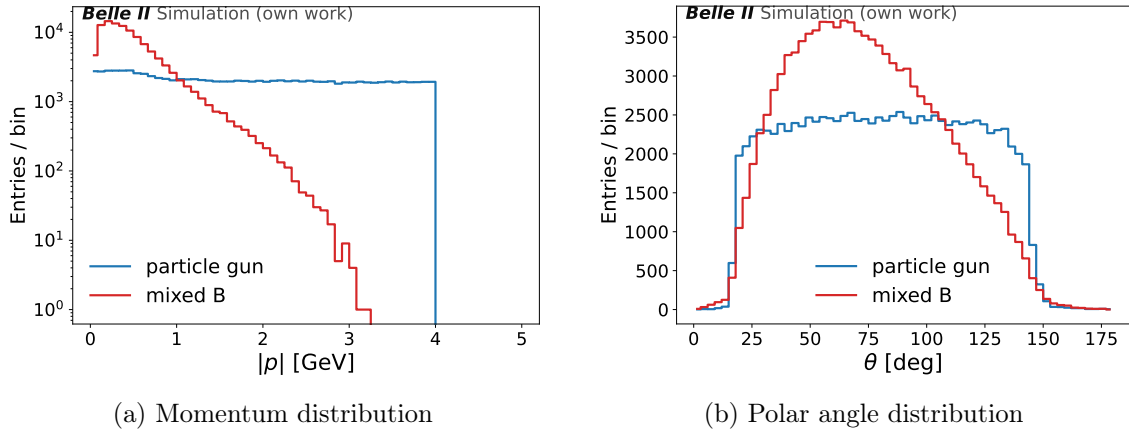


Figure 7.2.: Distribution of the momentum and the polar angle for both samples. The particle gun sample is shown in blue, the mixed B -events are shown in red. Only particles with at least one hit are included.

Detector Setup

This work uses three detector configurations: SVD-only, CDC-only, and SVD+CDC tracking. The separations provide a clear view of the performance of the isolated detectors. The SVD+CDC fitting corresponds to the standard reconstruction used for Belle II analysis. The PXD is not part of this work and does not contribute any measurements. Its material budget is represented in ACTS and GenFit2 to account for multiple scattering and energy loss.

Fitter Configuration

The fitter configuration for ACTS is described in detail in section 6.3. All tracks are fitted with the pion hypothesis, multiple scattering and energy loss are taken into account. The configuration is adapted for each detector setup to achieve optimal track-fitting efficiency and momentum resolution, as GenFit2 has been fine-tuned for the Belle II detector. The minimum number of required measurements is set to 3, according to GenFit2's threshold.

The fitting algorithms themselves are not equal, which limits the reach of the comparison. The DAF used in GenFit2 runs multiple iterations, each iteration consists of a full Kalman filter with one forward pass and one smoothing pass. ACTS' extended Kalman filter performs one forward pass and one backward pass. Every iteration of a Kalman filter improves the linearisation points and the energy loss. Outlier rejection in the DAF is done in a soft way, where measurements that are incompatible with the trajectory lose their influence over the iterations. In ACTS, measurements are rejected by a hard χ^2 -cutoff, which is influenced by the covariance of the track. Measurements at the start of the track have a minimal χ^2 value because of the large covariance of the track in the beginning, the cut therefore cannot remove early hits. The two fitters also solve the right-left ambiguity in the drift chamber differently. The DAF takes both sides of the wire as competing measurements and decides the side over the iterations. The calibrator used in this work assigns the sign depending on the side of the predicted state.

These differences are differences in algorithms, not in frameworks, and have to be taken into account when interpreting the results.

Runtime Measurements

The runtime is measured by performing both fits in comparable technical environments (same machine, similar time). The time for other operations like I/O operations is not included, only the time spent in the fitting algorithm. For each framework, 10 runs with 1000 events are used.

7.2. Tracking Performance Metrics

For the comparison between GenFit2 and ACTS, this work uses three quantities: the track fitting efficiency, the transverse momentum resolution, and the runtime. For the evaluation, the truth particle data is used to match measurements to the true hits and tracks to the true particles.

Truth Matching

The measurements of a track are matched to their true hits through the measurement-simhit-map. To match the track to the simulated particle, a majority vote selects the particle that contributes the most hits. After the matching, the tracks are characterized by two quantities: The purity and the hit efficiency:

$$p = \frac{N_{\text{matched}}}{N_{\text{hits}}}, \quad \varepsilon_{\text{hit}} = \frac{N_{\text{matched}}}{N_{\text{sim}}}, \quad (7.1)$$

The purity is defined by the fraction of the number of hits assigned to the particle N_{matched} and the total number of hits N_{hits} of a track. The hit efficiency is the fraction of the number of hits of the assigned particle and the number of hits the simulated particle produced, N_{sim} . A track is called a fake track if its purity is below 0.49 or its hit efficiency is below 0.05. In the event that more than one track is assigned to a simulated particle, the track with the highest purity is kept, and the other tracks are declared as clone tracks. Fake and clone tracks are removed from further evaluation.

Track Fitting Efficiency

The track fitting efficiency, instead of the tracking efficiency, is used in this work to isolate the fitting performance. Inefficiency in the track finding will not affect the track fitting efficiency. The track fitting efficiency sets the number of fitted tracks in relation to the number of found tracks.

$$\varepsilon_{\text{fit}} = \frac{N_{\text{fitted}}}{N_{\text{found}}}, \quad (7.2)$$

with N_{found} the number of found tracks and N_{fitted} the number of fitted tracks. A track is called fitted if the fitter fits it and is not flagged as a fake or clone track. A track is found if a prototrack exists and the purity is above 49 %.

In this work, the efficiency is evaluated as one number and as bins of p_{T}^{MC} and the polar angle θ^{MC} . The binned plot allows a deep insight of the conditions and areas in which differences in performance occur.

Momentum Resolution

The resolution of the transverse momentum is

$$\frac{\Delta p_{\text{T}}}{p_{\text{T}}} = \frac{p_{\text{T}}^{\text{fit}} - p_{\text{T}}^{\text{MC}}}{p_{\text{T}}^{\text{MC}}}. \quad (7.3)$$

with the fitted transverse momentum $p_{\text{T}}^{\text{fit}}$ and the true transverse momentum p_{T}^{MC} .

Another important quantity is the 68 % quantile of the absolute deviation from the median of the distribution equation (7.3)

$$r(p_{\text{T}}) = Q_{68\%} \left(\left| \frac{\Delta p_{\text{T}}}{p_{\text{T}}} - Q_{50\%} \left(\frac{\Delta p_{\text{T}}}{p_{\text{T}}} \right) \right| \right), \quad (7.4)$$

where Q is the quantile.

$r(p_{\text{T}})$ is referred to as resolution in this work.

The same evaluation is done for the longitudinal momentum p_z . The p_z resolution for ACTS and GenFit2 for all three detector setups and datasets. Since they all follow the same qualitative behavior as their respective p_{T} resolution, they are put into figure B.1.

Pulls

The pull is

$$\text{Pull}(x_i) = \frac{x_i^{\text{fit}} - x_i^{\text{MC}}}{\sigma_i^{\text{fit}}}. \quad (7.5)$$

with x_i being a track parameter and σ_i the corresponding diagonal element in the covariance after the fit. It is compared to a unit Gaussian to find systematic biases or problems in the covariance.

7.3. Silicon Vertex Detector

In this section, only the four SVD layers are used for reconstruction.

Setup

Different configurations regarding initial covariance scaling and χ^2 -cutoff thresholds were tested. It was found that applying a χ^2 -cutoff in a four-layer setup is not practical, since the value of χ^2 is scaled by the covariance, which is large in the first layers and only drops to reasonable values at later layers. This results in the cut only being applied to outer layers, so it was decided not to use a χ^2 -cutoff in the SVD-only setup.

The covariance can be evaluated by looking at the pulls. A too-small covariance matrix results in too-wide pulls for the parameter, indicating the fit is too confident in the fitted parameter. Inflating the covariance matrix allows the Kalman filter to down-weight the initial parameters and place more weight on the measurements than on the seed. It was found that inflating the covariance of q/p by 100 leads to good results. The ACTS pulls for no inflation and different inflation configurations can be seen in figure B.2. The corresponding differences in the track fitting efficiency over true p_T and the p_T resolution over true p_T are shown in figure B.3a and figure B.3b.

The track parameter pull plots for the chosen configuration are shown in figure 7.3.

Particle Gun Sample

With the particle gun dataset, the track fitting efficiency for ACTS is 99.88 %. For GenFit2 the track fitting efficiency is 99.48 %. Figure 7.4a shows the track fitting efficiency as a function of the transverse momentum for both frameworks. No p_T dependency is visible for both frameworks.

The map in figure 7.5a shows the difference between ACTS and GenFit2 over true p_T and true θ . The difference shows no structure. All bins are positive, indicating ACTS has a higher track fitting efficiency than GenFit2 in these bins.

Figure 7.4b shows the resolution of ACTS and GenFit2 in the SVD over true p_T . The values lie between 6 % and 10 % for both frameworks, with a minimum around 0.7 GeV. The relative difference between ACTS and GenFit2 is around 3 %, absolute about 0.2 %.

Mixed B -Events

For the mixed B -events, the track fitting efficiency for ACTS reaches 99.01 % and for GenFit2 97.89 %. Both values are lower than for the particle gun sample, and the overall difference is larger. The loss in track fitting efficiency is driven by the momentum spectrum of the B -events, which contains many more low- p_T tracks. Figure 7.6a shows that the efficiency in both frameworks drop for small p_T^{MC} . Above 0.5 GeV the frameworks perform almost similarly, with ACTS' efficiency being slightly higher for most transverse momenta.

The two-dimensional track fitting efficiency in figure 7.5b differs from the particle gun example in that ACTS outperforms GenFit2 across all bins except for three in the forward region. The largest deviation between the two frameworks of 0.9 percentage points is found in a central high p_T bin. The forward bin in which ACTS performs worse than GenFit2 contains ACTS tracks where the extrapolation after the fit fails. The same plot including the tracks that are fitted but not extrapolated can be seen in figure B.4a.

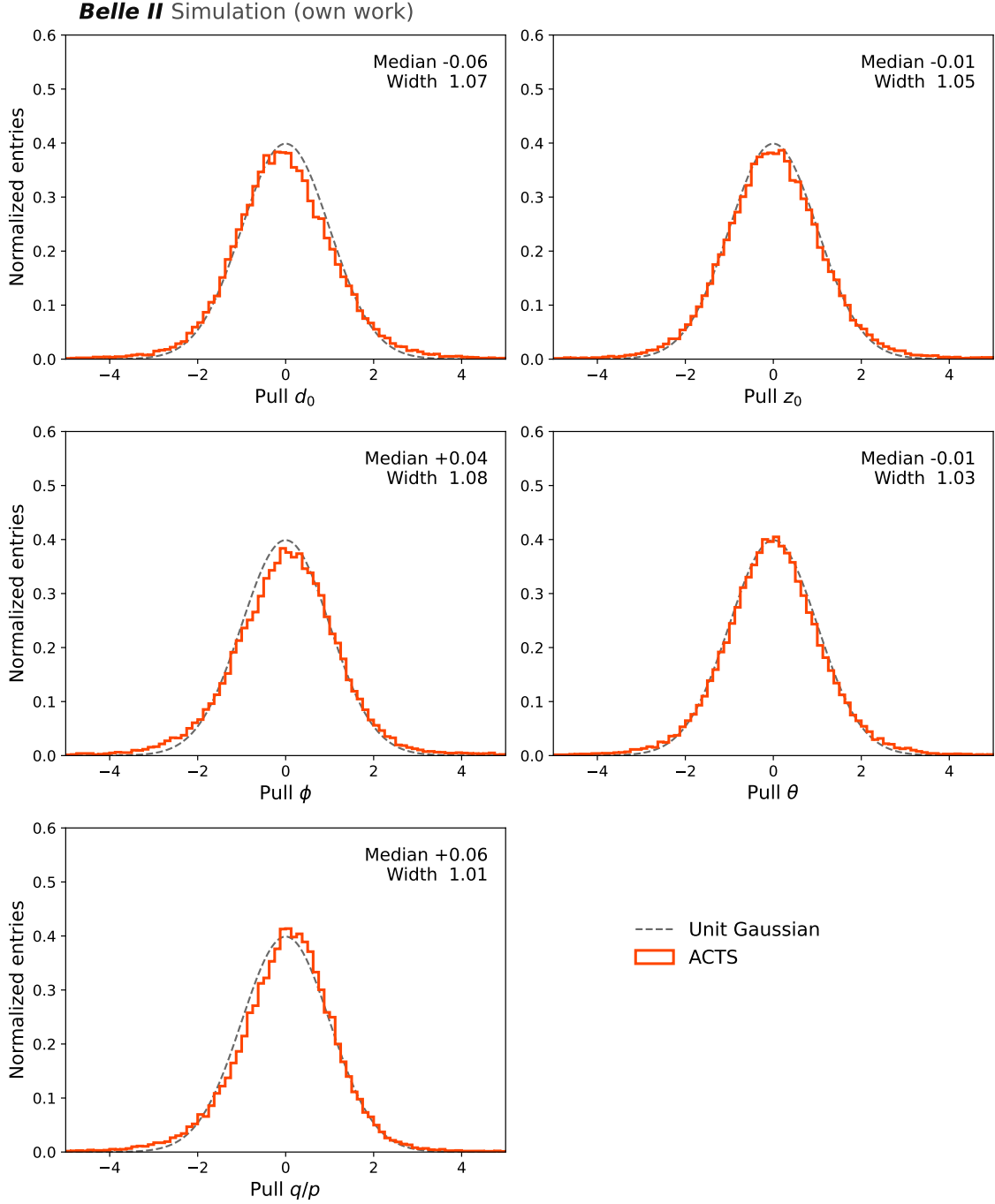
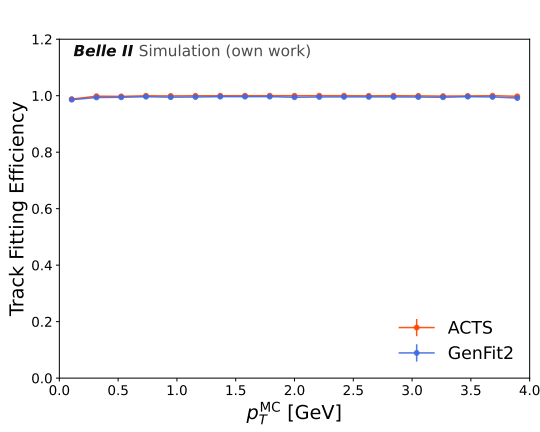
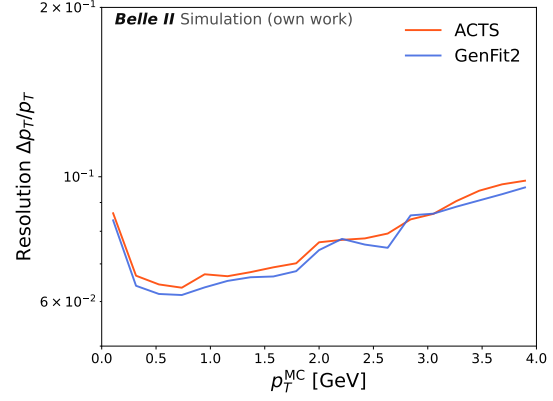


Figure 7.3.: Track parameter pulls for the SVD-only configuration. In case the uncertainty after the fit is estimated correctly, the width of the distribution is 1, and the median is 0.

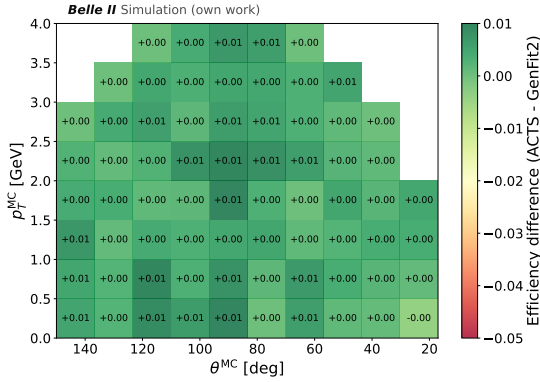


(a) Track fitting efficiency in the SVD over true p_T .

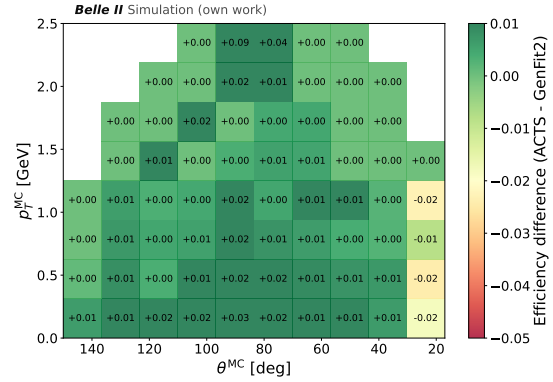


(b) Resolution of the transverse momentum in the SVD over true p_T . Only tracks fitted from both frameworks contribute.

Figure 7.4.: Track fitting efficiency and transverse momentum resolution in the SVD for the particle gun example. The orange line represents ACTS and the blue line represents GenFit2.



(a) Particle gun sample



(b) Mixed B -events

Figure 7.5.: Difference of the track fitting efficiency between ACTS and GenFit2 for the SVD in bins of true p_T and true θ . A positive value indicates that ACTS performs better, a negative value that GenFit2 performs better. Both maps use the same color scale.

Figure 7.6b behaves similarly to the particle gun example, the difference is the largest for small momenta and vanishes at higher momenta. The values move around 10 % with GenFit2 performing slightly better, especially in the lowest p_T bin. The average difference is 12 %. The number of entries per bin decreases with the momentum over 1 GeV and thus creates spikes.

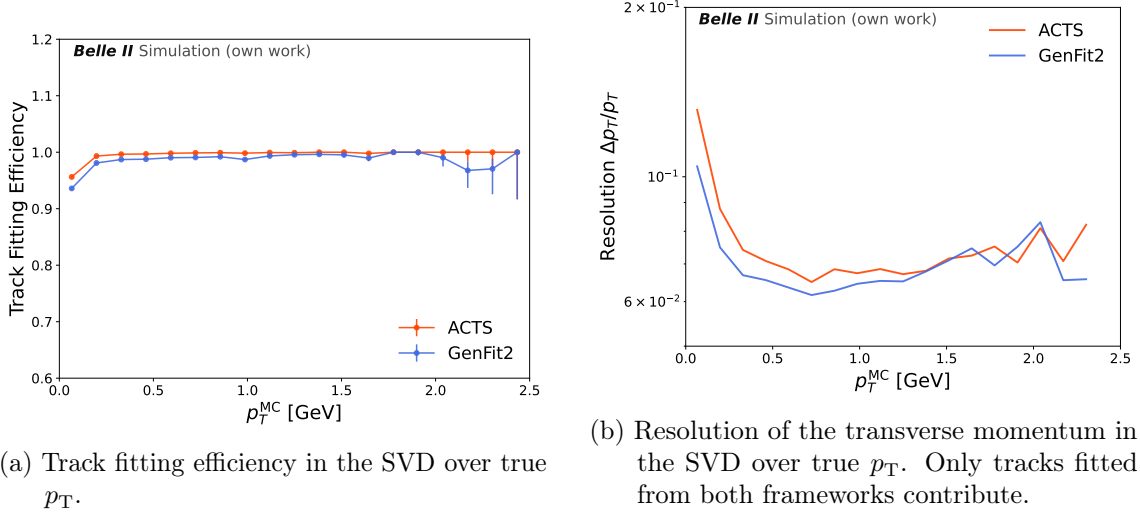


Figure 7.6.: Track fitting efficiency and transverse momentum resolution in the SVD for the mixed B -events. The orange line represents ACTS, and the blue line represents GenFit2.

Runtime

The results of the runtime measurement are summarized in table 7.2. The ACTS' standard Kalman filter is ≈ 3.47 times faster than the DAF in GenFit2.

Table 7.2.: Average runtime for the fitter algorithms in each framework. The data is averaged over 10 runs with 1000 particle gun events each

Fitter	t (ms/event)	Speedup
GenFit2 (DAF)	3.655 ± 0.020	1.00
ACTS (KF) 1 iteration	1.052 ± 0.023	3.47

Interpretation

The SVD-only runs show that ACTS can reproduce the performance achieved with GenFit2. In both samples, the track fitting efficiency is higher, with a larger difference in the mixed B -events, in which the tracks are lower in momentum and are closer to the reality of Belle II events.

The transverse momentum resolution of ACTS performs minimally worse in both samples. The differences between the DAF and the standard Kalman filter are a possible explanation, although this reason is not evaluated. The biggest differences in both algorithms are the number of iterations: whereas ACTS only performs one iteration for each track, GenFit2 performs multiple iterations. Each iteration improves the momentum-dependent material corrections. For low momentum, where material effects are large, these effects would have the most impact. The observed behavior for low momentum is consistent with this. Another effect that would be very present at low momentum is the material interactions. The

validation in chapter 5 is performed regarding the interaction length X_0 as a representation for the multiple scattering and energy loss. The quantities responsible for the energy loss are not the radiation length but the material thickness and atomic number. Those numbers were not validated.

This difference in iterations also has to be taken into account in the runtime differences. ACTS takes only about a third of the time for fitting compared to the DAF of GenFit2. Performing multiple iterations would make these differences smaller.

7.4. Central Drift Chamber

In this section, the CDC is used for reconstruction.

Setup

Different configurations for initial covariance scaling and χ^2 -cutoff values are also tested for the CDC-only runs. Introducing a χ^2 -cutoff is found not to be beneficial compared to no χ^2 . Although the resolution could benefit minimally for very high values of 100 or higher, the cutoff lowers the track fitting efficiency in the low momentum area. Applying a hard χ^2 cut was found to always remove at least 2 correct calibrated hits for 1 wrongly calibrated hit.

Factors for scaling the initial covariance have been evaluated. The optimal value was chosen to be scaling all elements by 5 and the covariance of q/p by 100. The pulls for ACTS for no inflation and different inflation configurations can be seen in figure B.5. The corresponding differences in the track fitting efficiency over true p_T and the p_T resolution over true p_T are shown in figure B.3a and figure B.3b.

The track parameter pulls can be seen in figure 7.7

Particle Gun Sample

The CDC-only tracking fitting efficiency for the particle gun sample is 98.23 % for ACTS and 98.07 % for GenFit2. Figure 7.8a shows the track fitting efficiency as a function of the true transverse momentum. Both frameworks show a clear loss in the region below 0.5 GeV. ACTS performs worse in the first two bins until 0.5 GeV but better than GenFit2 in the remaining momentum reach.

The difference in track fitting efficiency as a function of true p_T and true θ in figure 7.9a confirms the behavior of ACTS in low p_T . ACTS performs worse than GenFit2 in almost every bin below 0.5 GeV. In higher momentum, ACTS outperforms GenFit2 in almost every bin. Two columns stand out: All bins at $\theta^{\text{MC}} \approx 140^\circ$ and almost all bins at $\theta^{\text{MC}} \approx 20^\circ$. In these columns, ACTS performs significantly better than GenFit2. These areas correspond to the most forward and the most backward parts. The difference comes from a drop in efficiency in GenFit2.

Figure 7.8b shows the transverse momentum resolution as a function of true p_T . The resolution for both frameworks improves from 0 GeV to 0.75 GeV and slowly degrades with higher momentum. Both curves move around 1 %. GenFit2 performs better than ACTS at every momentum. The largest deviation of ACTS from the GenFit2 baseline happens above 1.0 GeV. The relative difference integrated over the momentum is 38 % and the absolute difference 0.3 %.

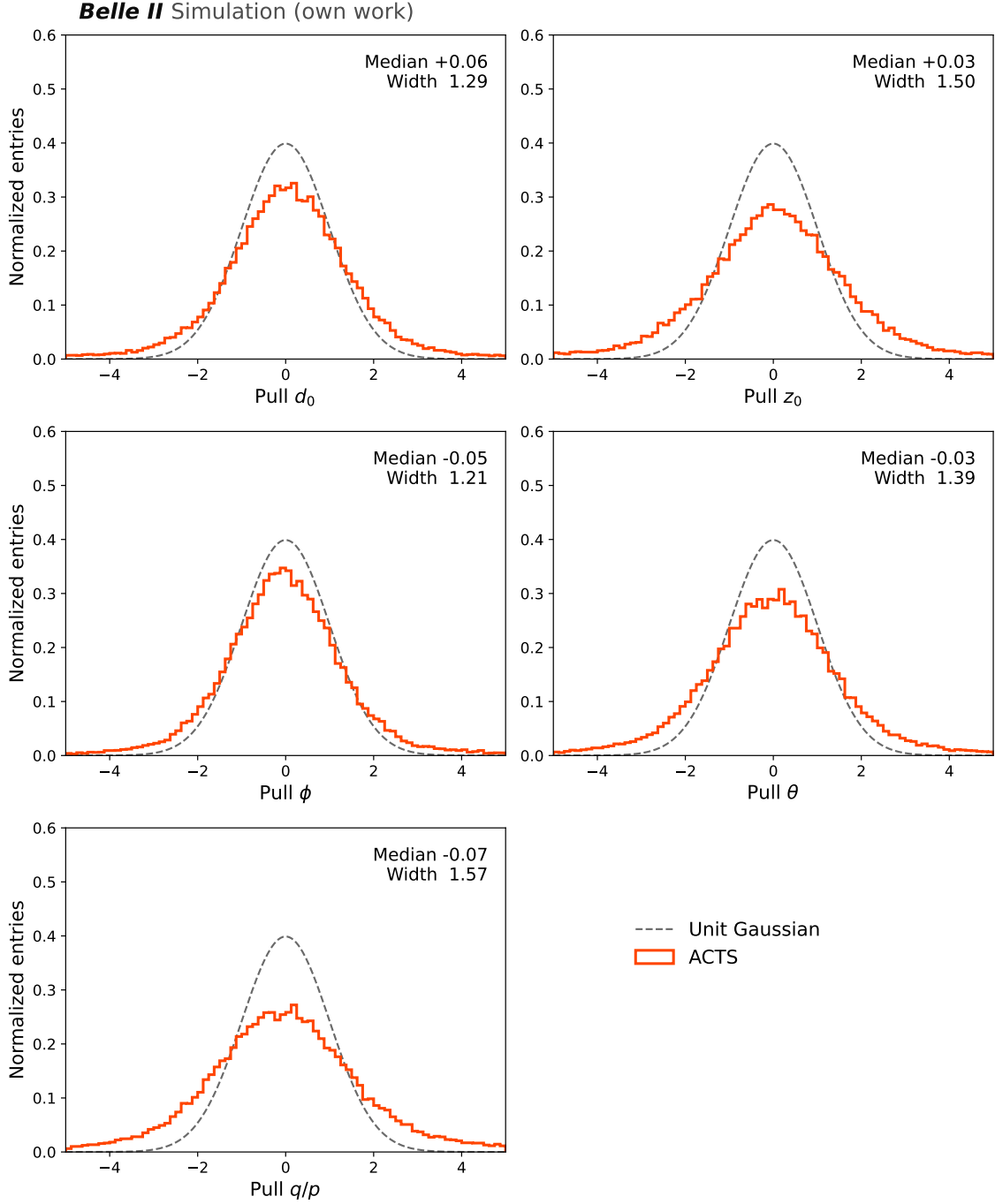


Figure 7.7.: Track parameter pulls for the CDC-only configuration. In case the uncertainty after the fit is estimated correctly, the width of the distribution is 1, and the median is 0. All pulls show a broader width than in the optimal case, and all medians are slightly displaced.

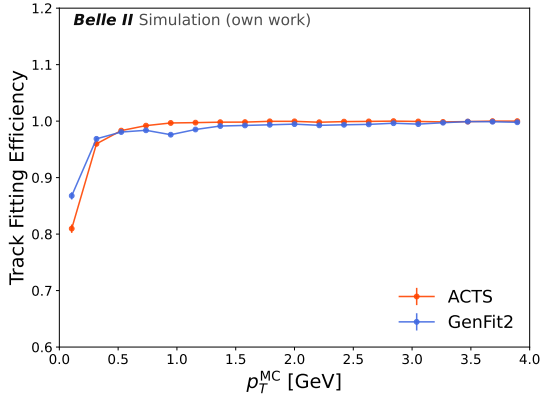
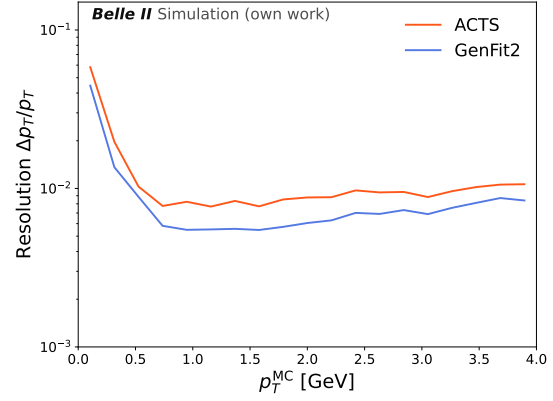
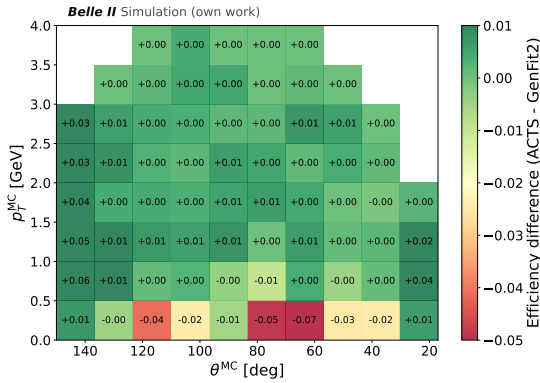
(a) Track fitting efficiency in the CDC over true p_T .(b) Resolution of the transverse momentum in the CDC over true p_T . Only tracks fitted from both frameworks contribute.

Figure 7.8.: Track fitting efficiency and transverse momentum resolution in the CDC for the particle gun example. The orange line represents ACTS, and the blue line represents GenFit2.



(a) Particle gun sample

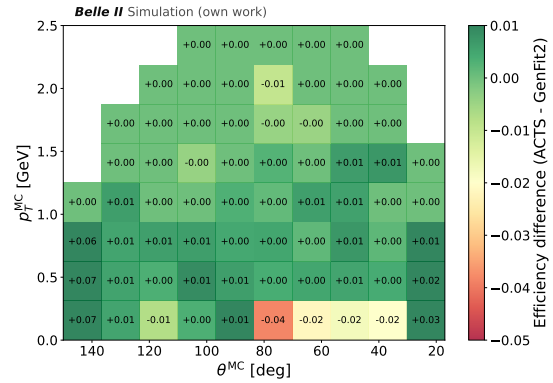
(b) Mixed B -events

Figure 7.9.: Difference of the track fitting efficiency between ACTS and GenFit2 for the CDC in bins of the transverse momentum and the polar angle. A positive value indicates that ACTS performs better, a negative value that GenFit2 performs better. Both maps use the same color scale.

Mixed B -Events

The difference in track fitting efficiency for the mixed B sample remains: ACTS reaches 95.02 %, whereas GenFit2 comes up to 94.99 %. Figure 7.9b shows that the difference in low p_T drops for the B -events.

The two-dimensional fitting efficiency in figure 7.9b shows the increased performance of ACTS in this sample compared to the low- p_T loss in the particle gun sample: Very few bins in the region under 0.5 GeV show a weaker performance of ACTS. The better performance in the most forward region remains, whereas the performance in the backward region changed

slightly.

The p_T resolution shown in figure 7.10b paints a similar picture as in the particle gun sample: GenFit2 achieves a better resolution at every bin except for the lowest p_T bin. The average relative difference increased to 50 %.

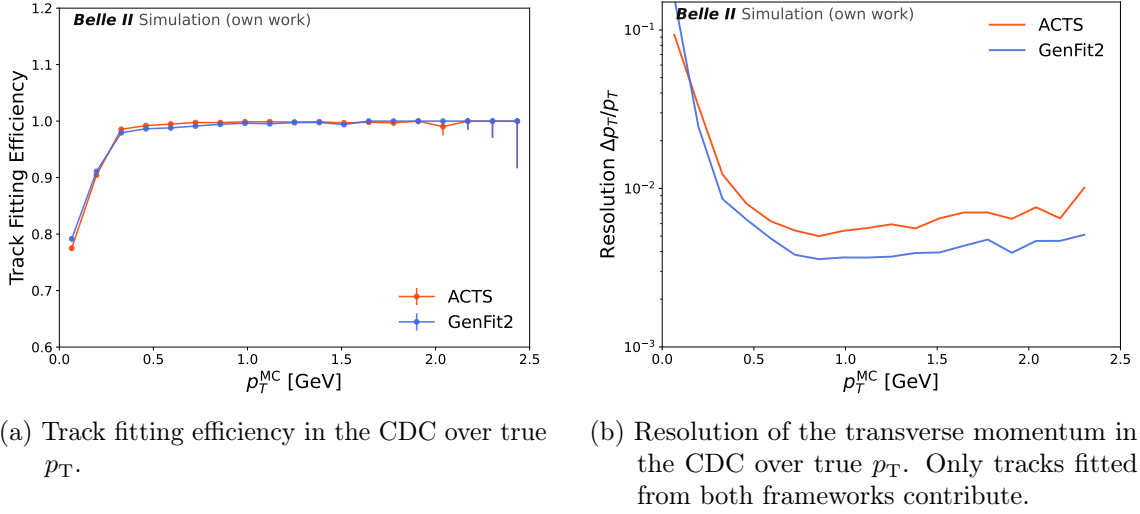


Figure 7.10.: Track fitting efficiency and transverse momentum resolution in the CDC for the mixed B -event example. The orange line represents ACTS, and the blue line represents GenFit2.

Runtime

The results of the runtime measurement are summarized in table 7.3. The speedup of ACTS compared to GenFit2 is 6.08.

Table 7.3.: Average runtime for the fitter algorithms in each framework. The data is averaged over 10 runs with 1000 particle gun events each

Fitter	t (ms/event)	Speedup
GenFit2 (DAF)	15.77 ± 0.05	1.00
ACTS (KF) 1 iteration	2.60 ± 0.03	6.08

Interpretation

The loss in track fitting efficiency in both the particle gun and the B sample has one common origin. Although the fit succeeded, the extrapolation to the perigee surface after the fit fails. This mechanism is described in section 6.3. This problem is technical in nature, and there are solutions that are more robust for the low p_T secondary particles in Belle II. This effect is observed more often with the particle gun due to the momentum distribution and the production of more secondary particles farther from the interaction point. The track fitting efficiency for a run in which the track parameters on the first measurement surface are taken in case the extrapolation fails is given in figure B.4b.

The differences in resolution remain in both samples and have multiple reasons: In this setup, the track states information at the first CDC layer is only the initial parameters. Solving the right-left ambiguity only depends on the initial parameters of a track traversing material for 16 cm and often results in the wrong decision. But the right-left ambiguity is not the only quantity that gets degraded by a poor prediction on the first CDC layer. The prediction also decides the linearisation point, if the prediction at the first layer is bad, the linearisation is performed at the wrong point. For the particle gun CDC-only sample, 4.07 % of the measurements are assigned to the wrong side of the wire. The rate is higher for drift radii below 1 mm, then drops to the minimum, until it spikes again for radii of 8 mm to 9 mm, where 15 % of the measurements are calibrated to the wrong side. Table 7.4 summarizes the rates for different drift lengths. Hits with a drift length d pull the track $2d$ to the wrong side. This effect becomes more severe for high drift length. In the current setup, ACTS cannot handle these hits at all because no χ^2 cut is applied. Wrongly assigned hits contribute fully to the track. The DAF can handle those hits much better. Although the rate of wrongly assigned hits is only 10 % lower than in ACTS, the dynamic reweighting of the hits at every iteration can remove bad hits in a soft way, leading to a better resolution. Comparing only tracks that have zero wrongly assigned hits in both frameworks, the relative resolution difference shrank by almost 50 %.

Figure 7.11 shows an increase in the absolute resolution difference between the two frameworks. The wrong right-left decision and the wrong linearisation point are compatible with this behavior. A wrongly assigned measurement displaces the measurement and affects the momentum resolution through the measurement uncertainty. Following the Gluckstern formula, this yields a relative momentum error that increases with momentum. An error in the linearisation point enters the Gluckstern formula in the same way. Both effects cannot be separated with the available measurements. The track parameter pulls figure 7.7 are consistent with this picture, since the fit is too confident in the fitted parameters.

The evaluation of the $x-t$ relation before the fit with the seed does not account for a large resolution difference, but not none. By evaluating it before the fit with the seed and not during the fit with the updated track parameters and the correct side, another uncertainty is added to the measurement. It was found that the deviation of the drift radius through the seed from the true value increases with increasing drift radius, yet the residual is small. Figure 7.12 show the residuals of the drift length for ACTS and GenFit2. Comparing the width of the residuals of ACTS and GenFit2, the width of the residuals of ACTS is 4 % broader than GenFit2's. Recalling the Gluckstern formula, this translates directly to a 4 % decrease in momentum resolution.

7.5. SVD and CDC Combined Fitting

In this section, the SVD and CDC are fitted together. This removes some of the disadvantages of SVD/CDC -only fitting.

Setup

For the SVD and CDC combined configuration, multiple scaling factors for the initial covariance and χ^2 -cutoff thresholds have been tested. This time, a χ^2 cutoff improved the

Table 7.4.: Comparison of the wrong side assignments in ACTS and GenFit2.

r [mm]	ACTS wrong	GenFit2 wrong
0–1	6.18 %	3.20 %
1–2	4.05 %	3.01 %
2–3	3.09 %	2.90 %
3–4	3.17 %	3.32 %
4–5	3.28 %	3.45 %
5–6	3.44 %	3.75 %
6–7	4.21 %	4.67 %
7–8	6.57 %	7.04 %
8–9	15.11 %	16.71 %
Total	4.07 %	3.78 %

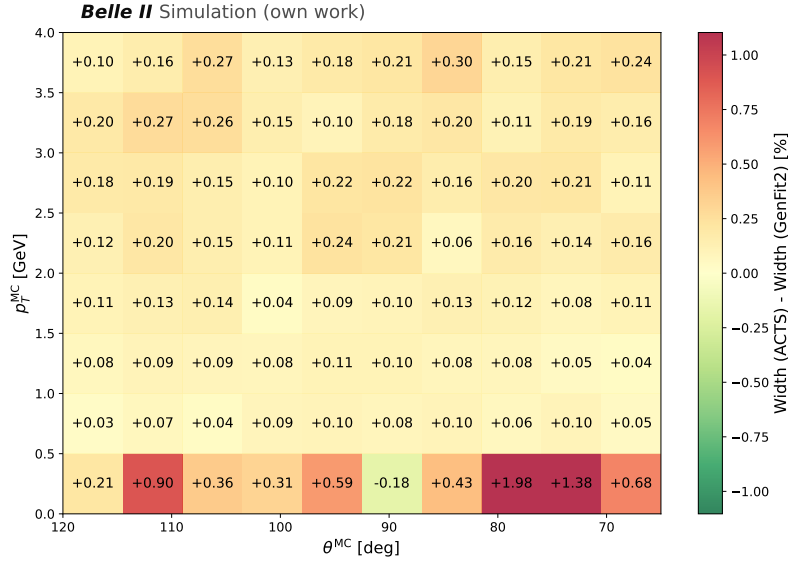
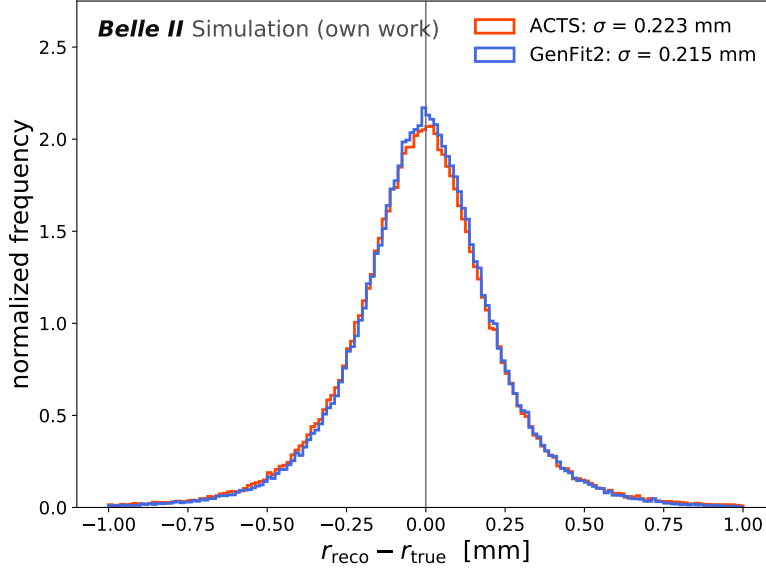


Figure 7.11.: Difference of the transverse momentum resolution for ACTS and GenFit2 in bins of the transverse momentum and the polar angle. The polar angle θ is restricted to exclude tracks that do not reach the last layer of the CDC.

Figure 7.12.: Residuals of r_{drift} for ACTS and GenFit2

p_T resolution while almost remaining the track fitting efficiency. The optimal value was found to be 50, a measurement above this value will not influence the track.

Scaling only q/p by a factor of 100 was found to be optimal among the tested configurations. The pulls for ACTS for no inflation and different inflation settings can be seen in figure B.7. The corresponding differences in the track fitting efficiency over true p_T and the p_T resolution over true p_T are shown in figure B.8a and figure B.8b.

Figure 7.13 shows the track parameter pulls for the chosen configuration.

Particle Gun Sample

The track fitting efficiency for the SVD and CDC combined fit is 98.80 % for ACTS and 99.23 % for GenFit2. Figure 7.14a illustrates the track fitting efficiency over p_T for both frameworks. GenFit2 outperforms ACTS for the momentum range below 1 GeV. Above 1 GeV ACTS and GenFit2 perform almost similar.

As seen in figure 7.15a, this drop in efficiency of ACTS compared to GenFit2 originates from the central low-momentum bins. In these bins, ACTS fitting efficiency is up to 6 % worse than GenFit2. ACTS performs at least as well as GenFit2 in the bins above 1 GeV.

The p_T resolution shown in figure 7.14b shows that the difference in resolution is the most in intermediate momentum between 1 GeV and 2.5 GeV. Outside this interval, ACTS and GenFit2's resolution comes very close. The average difference between the two frameworks is 20 %.

Mixed B -Events

Figure 7.16a shows the track fitting efficiency of ACTS and GenFit2 over true p_T . Both frameworks lose very little efficiency below 0.5 GeV. The higher-momentum bins have a

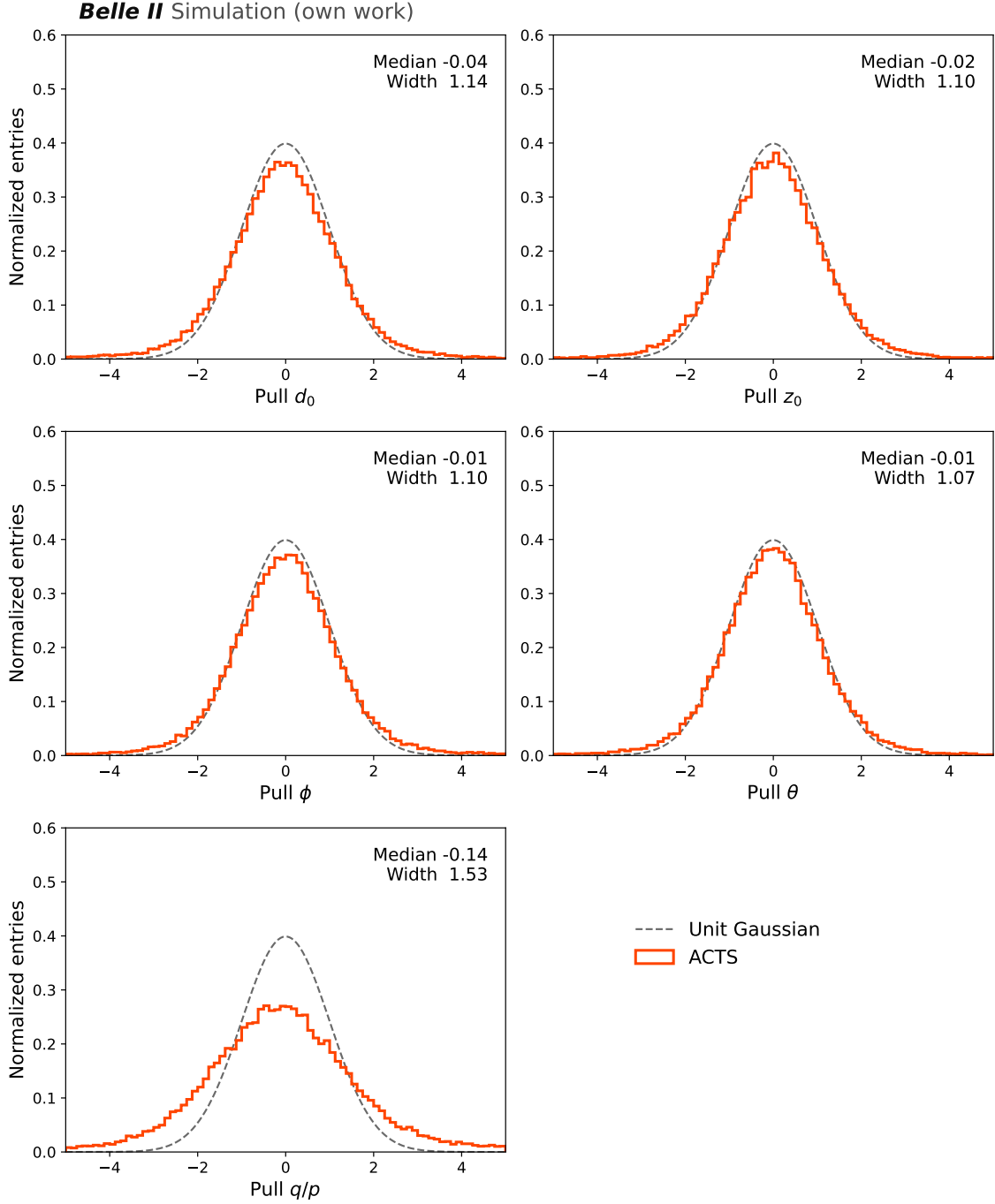


Figure 7.13.: Track parameter pulls for the SVD and CDC combined configuration. In case the uncertainty after the fit is estimated correctly, the width of the distribution is 1, and the median is 0. Especially the q/p pull shows a broader width than in the optimal case, and the median is displaced.

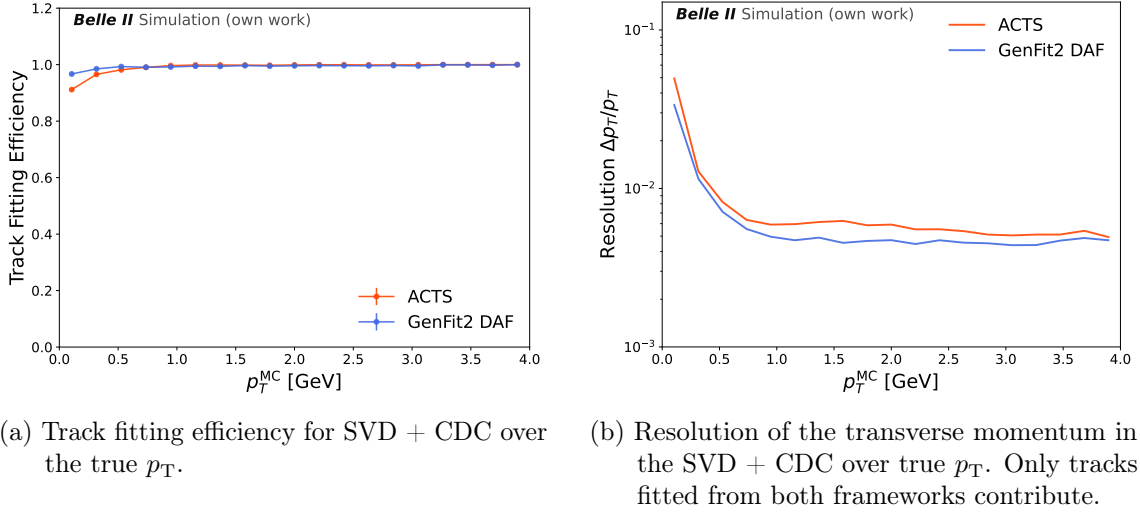


Figure 7.14.: Track fitting efficiency and transverse momentum resolution in the SVD + CDC for the particle gun example. The orange line represents ACTS, and the blue line represents GenFit2.

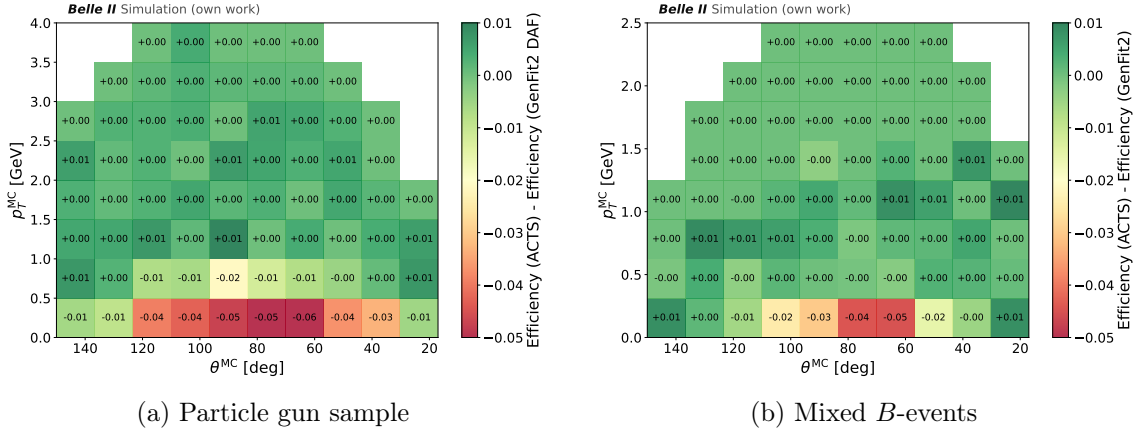


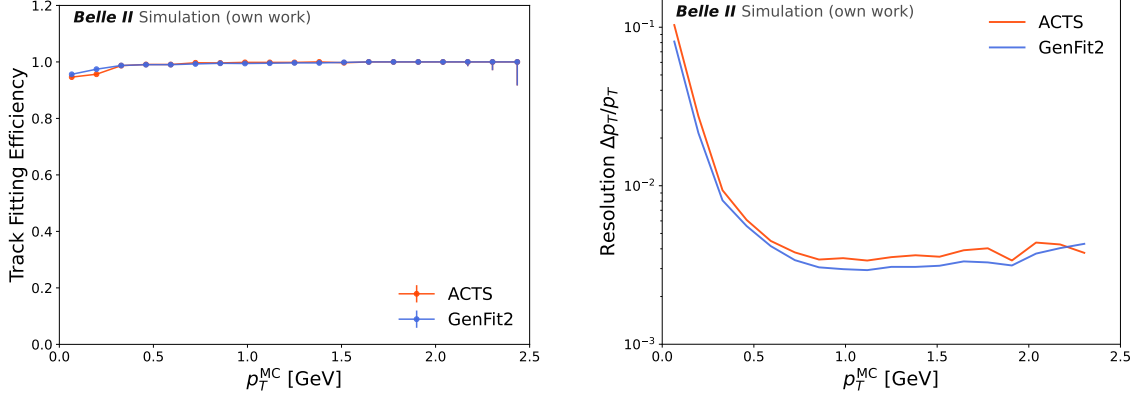
Figure 7.15.: Difference of the track fitting efficiency between ACTS and GenFit2 for the combined SVD and CDC fit in bins of the transverse momentum and the polar angle. A positive value indicates that ACTS performs better, a negative value that GenFit2 performs better. Both maps use the same color scale.

large uncertainty because of the small number of tracks in these bins.

The comparison of the track fitting efficiency over true θ and p_T in figure 7.15b shows a small improvement in the low momentum bins compared to the particle gun sample. The number of bins in which ACTS performs worse than GenFit2 shrinks. Especially for low p_T tracks in the backward and forward region there is a visible improvement compared to the particle gun sample. In every bin above 0.5 GeV ACTS performs better than GenFit2.

The resolution difference also drops significantly compared to the CDC-only configuration. The remaining relative difference is 30 % over all momentum bins. Figure 7.16b shows the

typical behaviour for the p_T resolution. The resolution drops fast for low momenta and then remains stable.



(a) Track fitting efficiency for the SVD and CDC combined setup over true p_T .

(b) Resolution of the transverse momentum in the SVD and CDC combined setup over true p_T . Only tracks fitted from both frameworks contribute.

Figure 7.16.: Track fitting efficiency and transverse momentum resolution in the SVD and CDC combined setup for the mixed B -event example. The orange line represents ACTS, and the blue line represents GenFit2.

Runtime

The results of the runtime measurement are summarized in table 7.5.

Table 7.5.: Average runtime for the fitter algorithms in each framework. The data is averaged over 10 runs with 1000 particle gun events each

Fitter	t (ms/event)	Speedup
GenFit2 (DAF)	22.50 ± 0.33	1.00
ACTS (KF) 1 iteration	2.18 ± 0.17	10.3

Interpretation

The loss in track fitting efficiency below 0.5 GeV has the same origin as in the CDC-only runs. Even though the fit succeeds, the extrapolation to the perigee surface fails. GenFit2 gains 1.16 percentage points, whereas ACTS only gains 0.57. The remaining differences concentrate in the same central low-momentum bins as in the CDC-only runs. This result is consistent with the mechanism, since the SVD does not affect the extrapolation back to the perigee surface.

Adding the SVD and therefore four more layers of measurements between the interaction point and the CDC increases the resolution as expected. The momentum-dependent growth disappears. The absolute difference between the frameworks decreases to 0.12 percent

points in the particle gun sample. The relative difference drops to 20 % in the particle gun sample and 30 % in the mixed B -sample.

The improved prediction is visible in the rate of miscalibrated hits: The rates of 4.07 % (ACTS) and 3.78 % (GenFit2) in the CDC-only run fall to 3.15 % for ACTS and 3.35 % for GenFit2. This shows that the right-left ambiguity resolution in ACTS itself is no worse than the DAF's decision.

A relative p_T resolution difference of 20 % remains between ACTS and GenFit2. The pulls figure 7.13 give a hint where the remaining difference is located. While all parameters have widths close to 1 and a median close to 0, q/p stands out with a width of 1.53 and a median of -0.14 . The fit is overconfident about the momentum by about 50 % and carries a small bias.

A part of the remaining difference is known. The $x-t$ relation evaluated before the fit with the seed accounts for roughly 4 % of the difference in resolution. Two possible contributions cannot be distinguished. The remaining 3.15 % of wrongly assigned hits that the extended Kalman filter cannot weight down like the DAF, and the multiple iterations of the DAF. With each iteration, the linearisation point improves, and the energy loss is re-evaluated. Distinguishing the effects of the iterations from the right-left solving is not possible without running multiple iterations.

7.6. Summary

Table 7.6 summarizes the results of the three detector configurations and the two datasets. ACTS is able to compete with GenFit2 in terms of track fitting efficiency. For the setups involving the CDC the differences in low momentum originate from the failed extrapolation. The p_T resolution $r(p_T)$ includes the full momentum range. The differences are suspected in the evaluation of the $x-t$ relation, the right-left ambiguity solving, and the difference in the number of iterations.

Table 7.6.: Track fitting efficiency, relative transverse momentum resolution and runtime of ACTS and GenFit2 in the three detector configurations.

Configuration	Sample	ε_{fit} (%)	$r(p_T)$ (%)	Speed-up
		ACTS / GenFit2	ACTS / GenFit2	
SVD	Particle gun	99.88 / 99.48	7.36/7.14	3.47
	Mixed B	99.01 / 97.89	8.11/7.21	
CDC	Particle gun	98.23 / 98.07	1.03/0.74	6.08
	Mixed B	95.02 / 94.99	1.48/0.99	
SVD + CDC	Particle gun	98.80 / 99.23	0.69/0.57	10.3
	Mixed B	97.59 / 98.18	1.50/1.11	

8. Conclusion

The SuperKEKB collider will further increase its luminosity during the next operating periods. This will increase computation time for track reconstruction, which already accounts for a large share of the total reconstruction time. The current framework for track fitting, GenFit2, served the Belle II experiment reliably for years but is no longer being developed and is not designed for multi-threading or porting to heterogeneous hardware. ACTS, on the other hand, is an actively developed tracking framework with a large group of contributors, designed for multi-threading and research and development of heterogeneous hardware adaptations, but, other than GenFit2, is not specifically designed for drift chambers.

This thesis examined the use of ACTS for track fitting in the Belle II experiment. ACTS is a suitable option for track fitting in Belle II in the future and provided promising results, but is not yet able to reach its full potential.

Summary

This thesis detailed the most important steps for implementing ACTS for track fitting in the Belle II experiment. The core work of this thesis is the implementation of the complex geometry of the SVD and CDC, the magnetic field, and the material representation. The geometry of the SVD and the CDC is represented in the tracking geometry. The beampipe and PXD are included as passive material. The magnetic field is provided by an interpolated field map and validated against basf2. To account for material interactions, the detector material is mapped onto dedicated surfaces. The material-decorated detector is validated against the detailed Geant4 truth. A custom geometry identifier was created to identify sensors in both frameworks. An interface for tracking objects between basf2 and ACTS was realized through file transfer. The basf2 side is performed by a dedicated basf2 module, which serves the purpose of writing the information about the tracks after pattern recognition through the Belle II track finding chain. The tracking objects are converted into event-wise data in the ACTS data format. The Kalman filter is used for track fitting, with a dedicated measurement calibrator, to resolve the left-right ambiguity in the drift chamber at every step. An implementation of a direct navigator was built, using the surface list of the pattern recognition together with material surfaces along the path. The track fitting efficiency and momentum resolution were compared to GenFit2 in three different detector configurations with the same tracks from two different datasets. A proof of concept for track fitting with the Kalman filter in the SVD on GPUs was carried out using traccc.

Further improvements can be achieved by performing multiple iterations and implementing the DAF. The implementation of other features like the evaluation of the $x-t$ relation will improve the resolution further, and implementing a more robust extrapolation method will improve the track fitting efficiency in low p_T .

Results

ACTS reached the track fitting efficiency of GenFit2 in every detector configuration outside the low-momentum region. In the SVD, ACTS surpassed the track fitting efficiency of GenFit2 in both datasets over the full momentum range. ACTS achieved a track fitting efficiency of 99.88 % compared to 99.48 % in the particle gun sample for SVD-only fitting. In the CDC and SVD+CDC setups, ACTS reached the track fitting efficiency of GenFit2 for $p_T > 0.5$ GeV. Overall, ACTS achieved a track fitting efficiency of 98.23 % and 98.80 % in the CDC and SVD+CDC particle gun setup, compared to GenFit2's 98.07 % and 99.23 %. The differences for $p_T < 0.5$ GeV originate from an extrapolation issue for secondary particles with an origin well in the CDC.

The momentum resolution in all three configurations could not yet reach the baseline of GenFit2's deterministic annealing filter. For the SVD-only run ACTS achieved a p_T resolution of 7.36 %, compared to 7.14 % for GenFit2. The CDC-only run resulted in a p_T resolution of 1.03 % for ACTS and 0.74 % for GenFit2. For the SVD and CDC combined setup the p_T resolution of ACTS is 0.69 %, GenFit2 achieved 0.57 %. The difference in the SVD and CDC combined is due to multiple reasons: The evaluation of the $x-t$ relation with the seed instead of the current track parameters, the hard impact of hits assigned to the wrong side, and only performing one iteration. A quantitative estimate of the improvement in implementing these features is provided for the evaluation of the $x-t$ relation and correct L/R ambiguity solving.

ACTS is able to fit the same tracks as GenFit2 in a fraction of the time. Reaching from a speed-up factor of 3.5 in the SVD up to 10 in the SVD and CDC combined configuration. The difference in the number of iterations has to be considered when looking at this result. It was shown that the tracking geometry of the SVD is suitable for conversion to detray format and for track fitting on GPUs. A quantitative evaluation of the track fitting efficiency, resolution, and runtime was not performed.

This work provides a comparison of ACTS with an established fitter. The results show that ACTS performs well in track fitting for the Belle II experiment and that low momenta and drift chambers are no obstacles for ACTS. The runtime and fitting efficiency reach the baseline of GenFit2, and the origin of the remaining difference in resolution is pinpointed as closely as possible.

A. Material mapping validation

This appendix shows the validation of the material mapping for the individual subdetectors. The ratio is defined as in figure 5.3. Figure A.1 covers the beampipe and the PXD, which carry material but do not contribute measurements to the fits performed in this work.

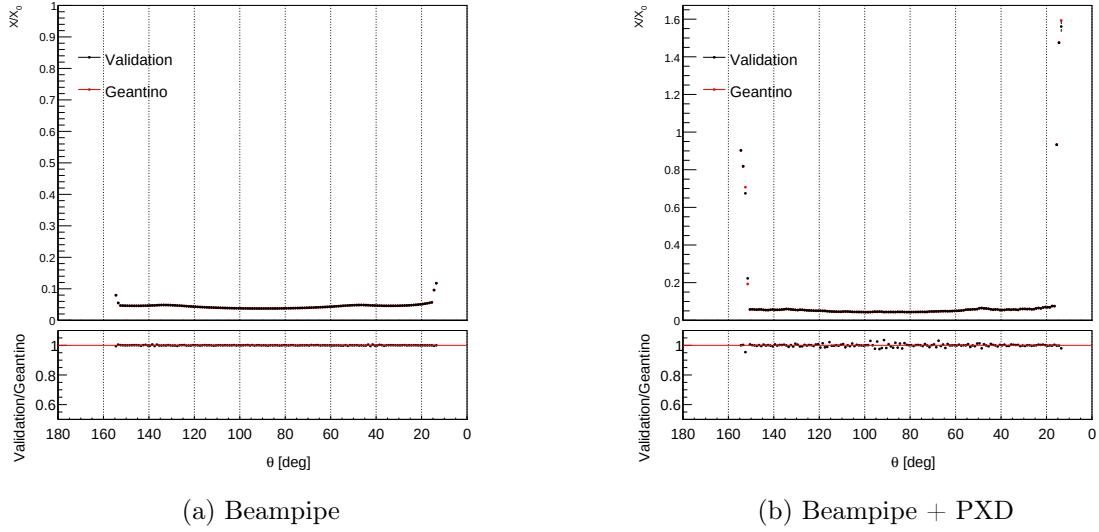
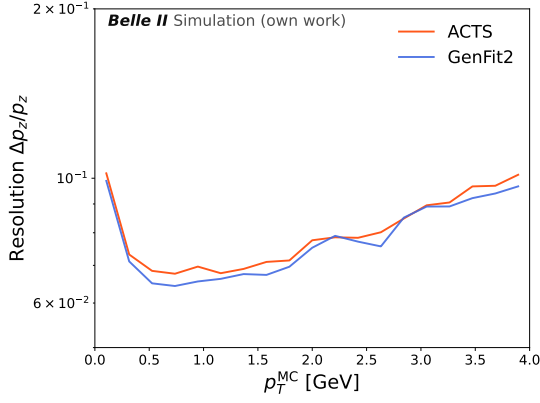


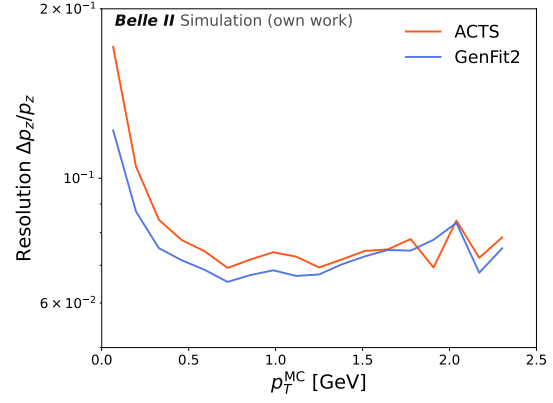
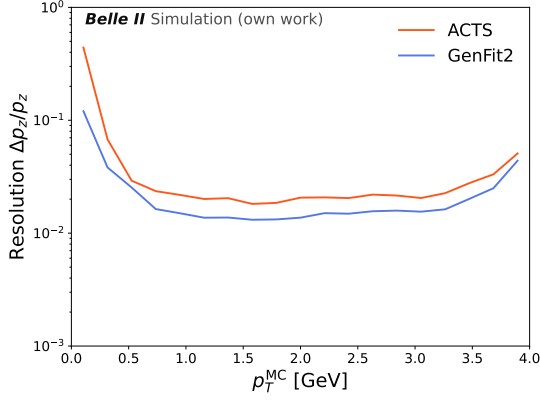
Figure A.1.: Ratio of the radiation length t/X_0 from the decorated ACTS geometry to the one from the full Geant4 description, as a function of θ , separately for the beampipe and the beampipe + PXD.

B. Additional Performance Plots

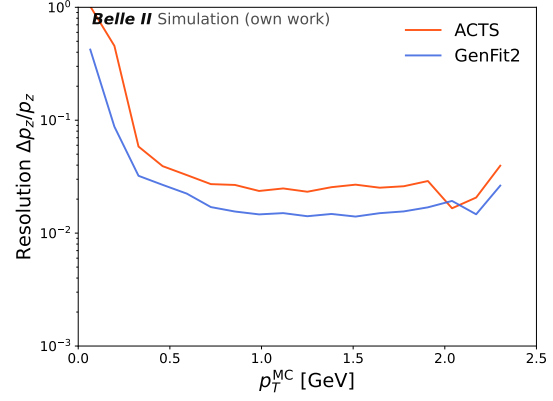
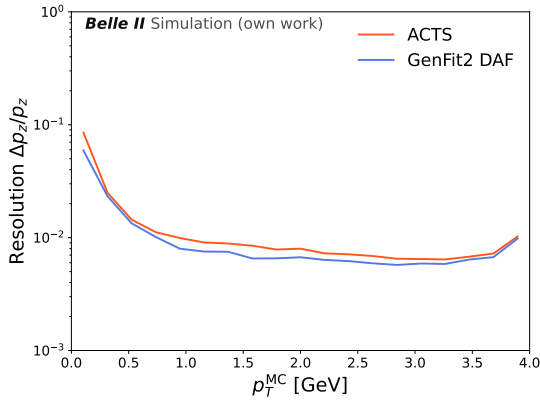
This appendix collects the plots referenced in chapter 7. The p_z resolution, the track parameter pulls for different scalings of the initial covariance, the corresponding track fitting efficiencies and p_T resolutions, and the track fitting efficiency obtained when tracks succeed, but the extrapolation fails.



(a) SVD, particle gun sample.

(b) SVD, mixed B -events.

(c) CDC, particle gun sample.

(d) CDC, mixed B -events.

(e) SVD + CDC, particle gun sample.

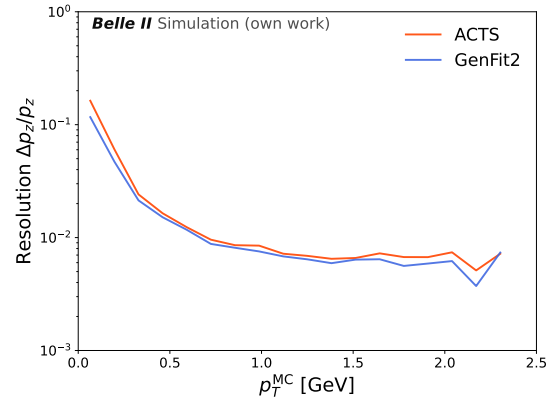
(f) SVD + CDC, mixed B -events.

Figure B.1.: Resolution of the longitudinal momentum p_z as a function of the true transverse momentum, for the three detector configurations and both samples. The orange line represents ACTS, the blue line GenFit2.

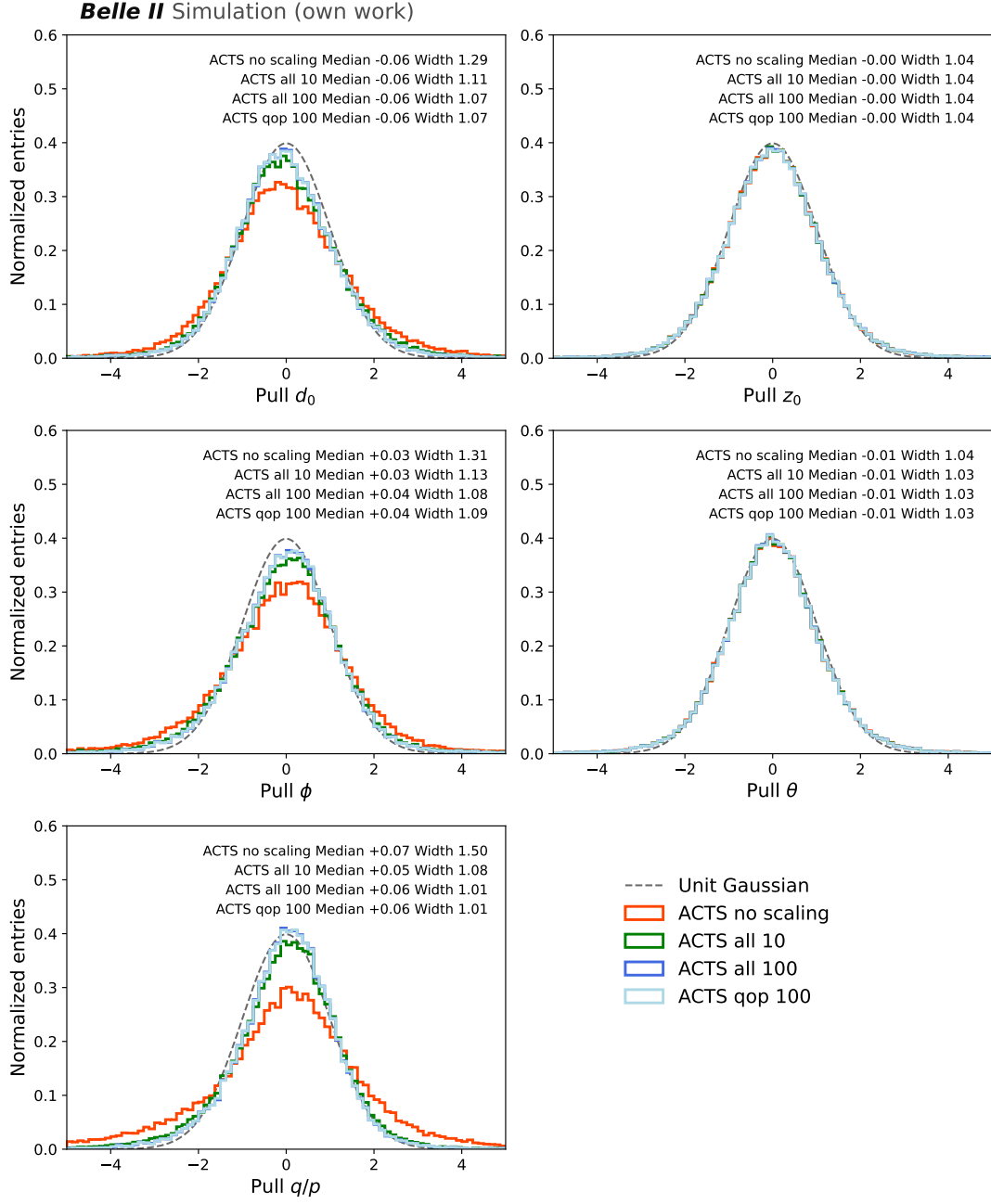


Figure B.2.: Track parameter pulls of ACTS in the SVD-only configuration with different inflation of the initial covariance. Figure 7.3 shows the same distributions for the configuration used in this work, in which the covariance of q/p is scaled by 100.

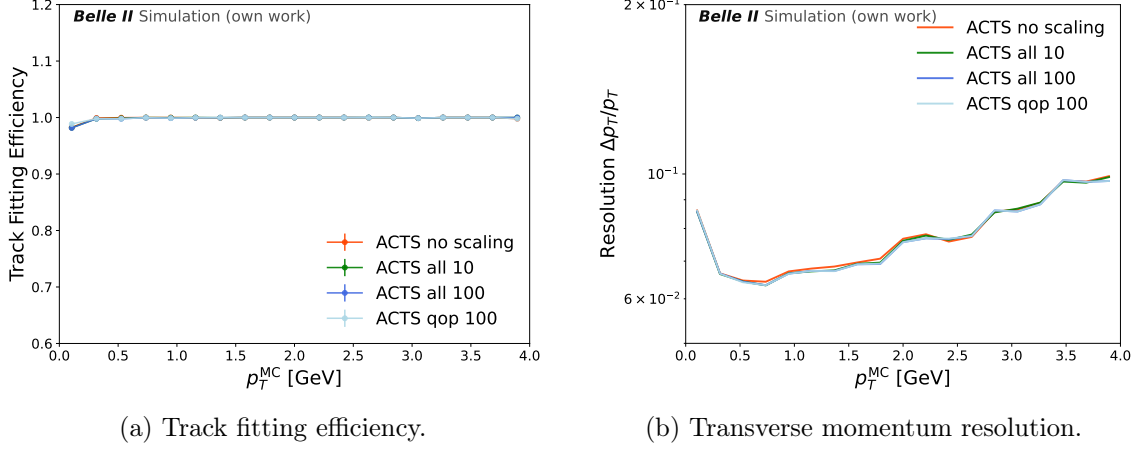


Figure B.3.: Track fitting efficiency and transverse momentum resolution in the SVD-only setup as a function of the true transverse momentum, for different scalings of the covariance matrix of the initial track parameters.

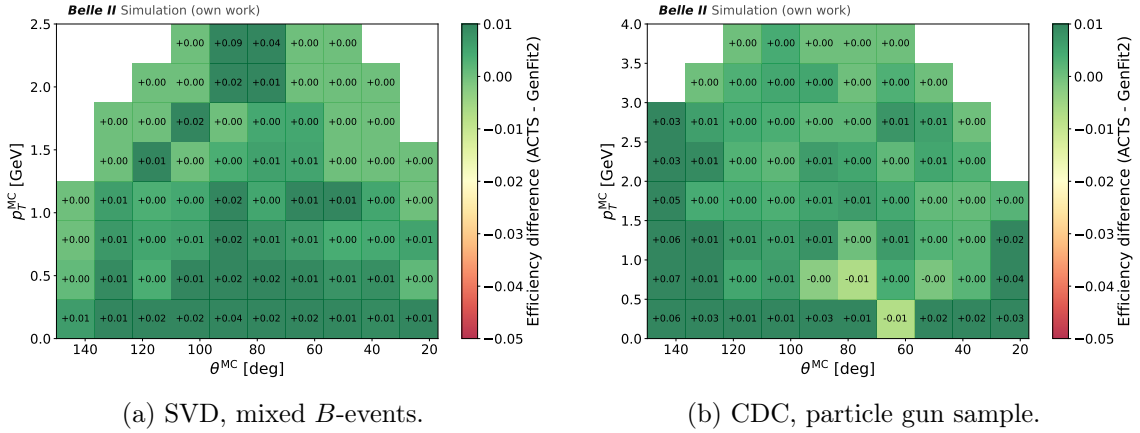


Figure B.4.: Difference of the track fitting efficiency between ACTS and GenFit2 in bins of the transverse momentum and the polar angle, for runs in which tracks are counted as fitted even when the extrapolation to the perigee surface fails after a converged fit. A positive value indicates that ACTS performs better.

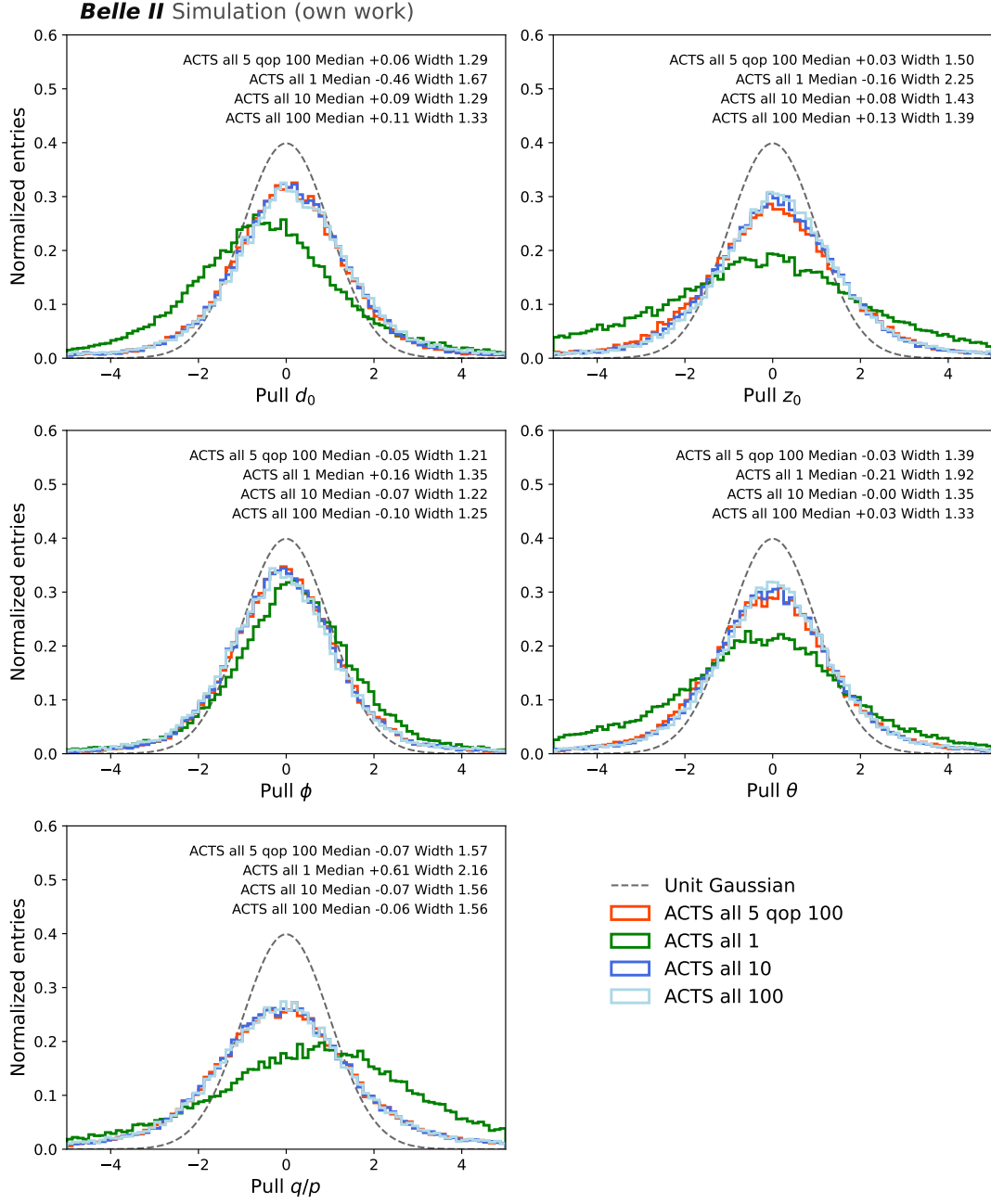
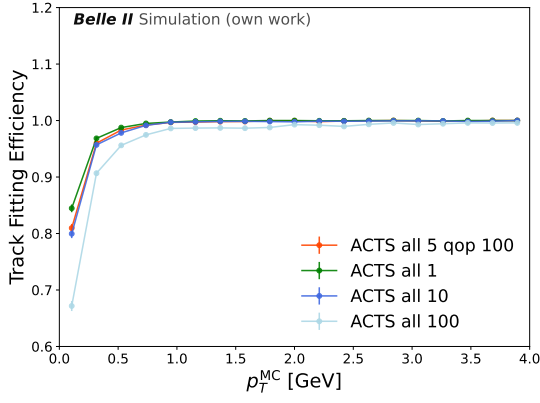
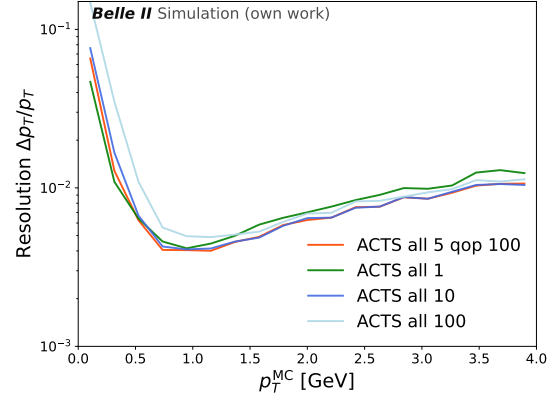


Figure B.5.: Track parameter pulls of ACTS in the CDC-only configuration with different inflation of the initial covariance. Figure 7.7 shows the same distributions for the configuration used in this work, in which the covariance of all parameter is scaled by 5 and the covariance of q/p is scaled by 100.



(a) Track fitting efficiency.



(b) Transverse momentum resolution.

Figure B.6.: Track fitting efficiency and transverse momentum resolution in the CDC-only setup as a function of the true transverse momentum, for different scalings of the covariance matrix of the initial track parameters. All x means that every element of the matrix is scaled by x .

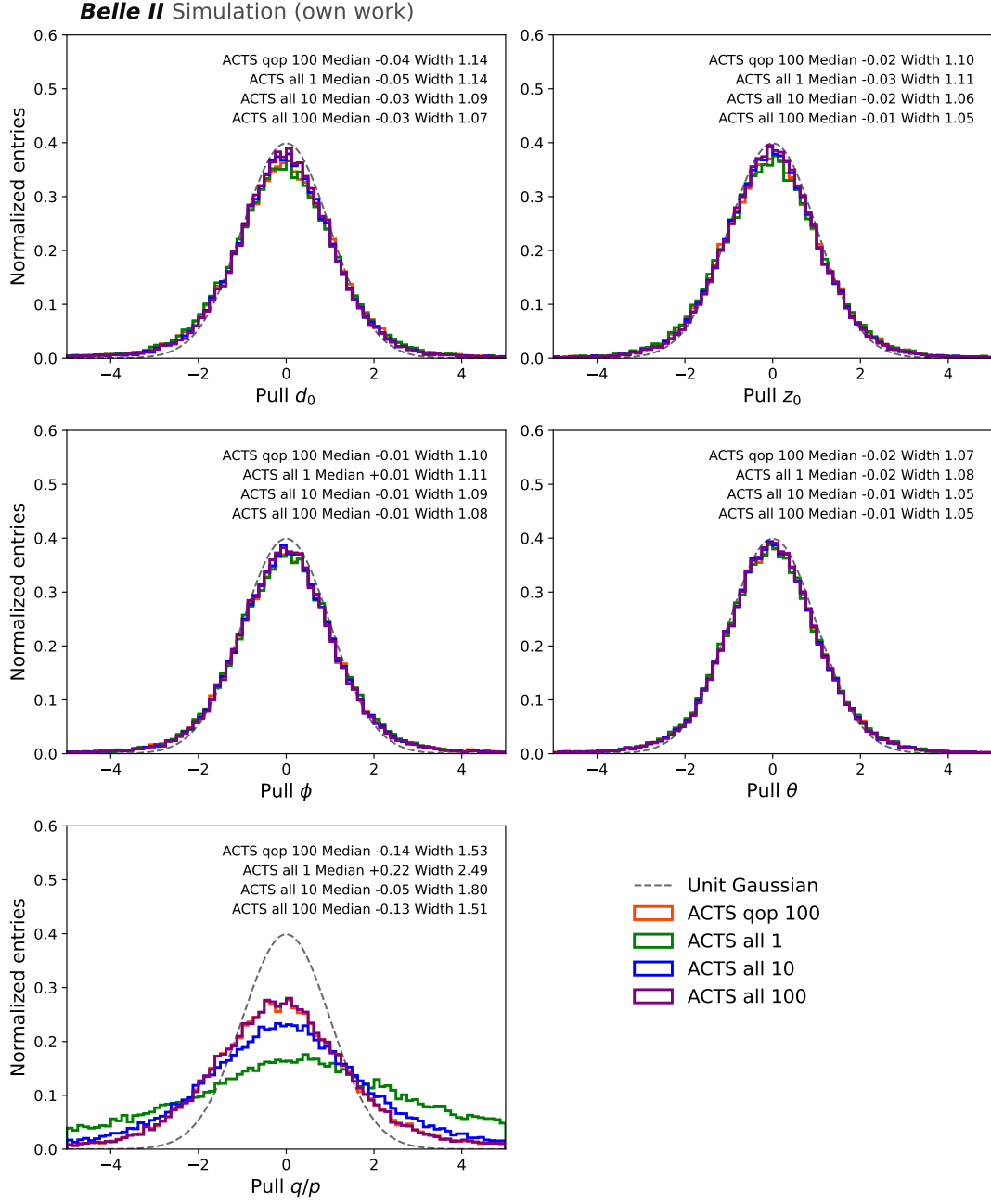
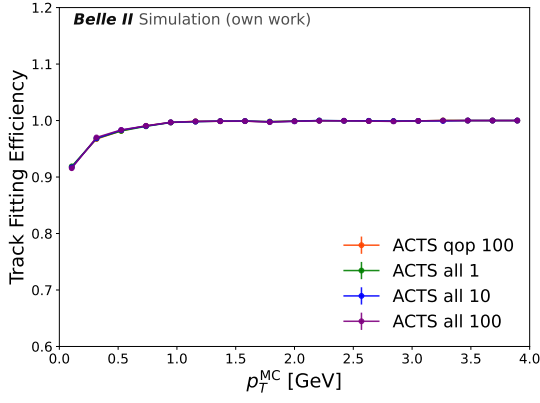
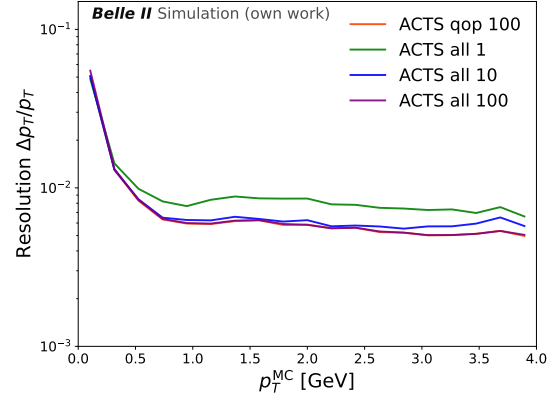


Figure B.7.: Track parameter pulls of ACTS in the SVD+CDC configuration with different inflation of the initial covariance. Figure 7.13 shows the same distributions for the configuration used in this work, in which the covariance of q/p is scaled by 100.



(a) Track fitting efficiency.



(b) Transverse momentum resolution.

Figure B.8.: Track fitting efficiency and transverse momentum resolution in the SVD+CDC setup as a function of the true transverse momentum, for different scalings of the covariance matrix of the initial track parameters. All x means that every element of the matrix is scaled by x .

Bibliography

- [1] G. Aad *et al.*, “Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC,” *Physics Letters B* **716** no. 1, (2012) 1–29. <http://arxiv.org/abs/1207.7214>. ATLAS Collaboration. arXiv:1207.7214 [hep-ex].
- [2] “Belle II.” <https://www.belle2.org/>.
- [3] X. Ai, X. Huang, and Y. Liu, “Implementation of ACTS for STCF track reconstruction,” *Journal of Instrumentation* **18** no. 07, (2023) P07026. <http://arxiv.org/abs/2301.04306>. arXiv:2301.04306 [hep-ex].
- [4] E. Kou, P. Urquijo, W. Altmannshofer, F. Beaujean, G. Bell, M. Beneke, I. I. Bigi, F. B. M. Blanke, C. Bobeth, M. Bona, *et al.*, “The Belle II Physics Book,” *Progress of Theoretical and Experimental Physics* **2019** no. 12, (2019) 123C01. <http://arxiv.org/abs/1808.10567>. arXiv:1808.10567 [hep-ex].
- [5] K. Akai, K. Furukawa, and H. Koiso, “SuperKEKB Collider,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **907** (2018) 188–199. <http://arxiv.org/abs/1809.01958>. arXiv:1809.01958 [physics.acc-ph].
- [6] F. Takahashi and others, “Review of Particle Physics,” *Int. J. Mod. Phys. A* **41** (2026) 2630011. Particle Data Group.
- [7] A. Natochii, T. E. Browder, L. Cao, K. Kojima, D. Liventsev, F. Meier, K. R. Nakamura, H. Nakayama, C. Niebuhr, A. Novosel, *et al.*, “Beam background expectations for Belle II at SuperKEKB,” *Nuclear Instruments and Methods in Physics Research Section A* **1055** (2023) 168550. <http://arxiv.org/abs/2203.05731>. Belle II Collaboration. arXiv:2203.05731 [hep-ex].
- [8] T. Abe, I. Adachi, K. Adamczyk, S. Ahn, H. Aihara, K. Akai, M. Aloï, L. Andricek, K. Aoki, Y. Arai, *et al.*, “Belle II Technical Design Report,” belle ii technical design report, KEK, 2010. <http://arxiv.org/abs/1011.0352>. Belle II Collaboration. arXiv:1011.0352 [physics.ins-det].
- [9] D. Matvienko, “The Belle II experiment: status and physics program,” *EPJ Web of Conferences* **191** (2018) 02010. https://www.epj-conferences.org/articles/epjconf/abs/2018/26/epjconf_quarks2018_02010/epjconf_quarks2018_02010.html.

- [10] H. Ye, F. Abudinen, K. Ackermann, P. Ahlburg, M. Albalawi, O. Alonso, L. Andricek, R. Ayad, V. Babu, T. Bilka, *et al.*, “Commissioning and performance of the Belle II pixel detector,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **987** (2021) 164875. <https://www.sciencedirect.com/science/article/pii/S0168900220312729>. Belle II PXD Collaboration.
- [11] K. Adamczyk *et al.*, “The Design, Construction, Operation and Performance of the Belle II Silicon Vertex Detector,” *Journal of Instrumentation* **17** no. 11, (2022) P11042. <http://arxiv.org/abs/2201.09824>. arXiv:2201.09824 [physics.ins-det].
- [12] T. V. Dong, S. Uno, M. Uchida, H. Ozaki, N. Taniguchi, E. Nakano, and K. Trabelsi, “Calibration and alignment of the Belle II central drift chamber,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **930** (2019) 132–141. <https://www.sciencedirect.com/science/article/pii/S0168900219304024>.
- [13] L. Reuter, *Track Finding with Graph Neural Networks in the Belle II Drift Chamber*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2026. <https://publikationen.bibliothek.kit.edu/1000189859>.
- [14] X. Ai, C. Allaire, N. Calace, A. Czirkos, I. Ene, M. Elsing, R. Farkas, L.-G. Gagnon, R. Garg, P. Gessinger, *et al.*, “A Common Tracking Software Project,” *Computing and Software for Big Science* **6** no. 1, (2022) 8. <http://arxiv.org/abs/2106.13593>. arXiv:2106.13593 [physics.ins-det].
- [15] P. Gessinger-Befurt, *Development and improvement of track reconstruction software and search for disappearing tracks with the ATLAS experiment*. PhD thesis, Johannes Gutenberg-Universität Mainz, 2021. <https://openscience.ub.uni-mainz.de/handle/20.500.12030/5910>.
- [16] J. Myrheim and L. Bugge, “A fast Runge-Kutta method for fitting tracks in a magnetic field,” *Nuclear Instruments and Methods* **160** no. 1, (1979) 43–48. <https://www.sciencedirect.com/science/article/pii/0029554X79901630>.
- [17] E. Lund, L. Bugge, I. Gavrilenko, and A. Strandlie, “Track parameter propagation through the application of a new adaptive Runge-Kutta-Nyström method in the ATLAS experiment,” *Journal of Instrumentation* **4** no. 04, (2009) P04001. <https://doi.org/10.1088/1748-0221/4/04/P04001>.
- [18] E. Lund, L. Bugge, I. Gavrilenko, and A. Strandlie, “Transport of covariance matrices in the inhomogeneous magnetic field of the ATLAS experiment by the application of a semi-analytical method,” *Journal of Instrumentation* **4** no. 04, (2009) P04016–P04016. <https://iopscience.iop.org/article/10.1088/1748-0221/4/04/P04016>.
- [19] S. Navas, C. Amsler, T. Gutsche, C. Hanhart, J. J. Hernández-Rey, C. Lourenço, M. Masoni, M. Mikhasenko, R. E. Mitchell, C. Patrignani, *et al.*, “Review of Particle Physics,” *Phys. Rev. D* **110** no. 3, (2024) 030001. <https://link.aps.org/doi/10.1103/PhysRevD.110.030001>. Particle Data Group.

- [20] R. Frühwirth and A. Strandlie, “Track Fitting,” in *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*, R. Frühwirth and A. Strandlie, eds., pp. 103–127. 2021. https://doi.org/10.1007/978-3-030-65771-0_6.
- [21] R. Mankel, “Pattern Recognition and Event Reconstruction in Particle Physics Experiments,” *Reports on Progress in Physics* **67** no. 4, (2004) 553–622. <http://arxiv.org/abs/physics/0402039>. arXiv:physics/0402039.
- [22] L. Reuter, T. Brandes, G. De Pietro, and T. Ferber, “Multi-Modal Track Reconstruction using Graph Neural Networks at Belle II,” 2026. <https://arxiv.org/abs/2602.10897v1>.
- [23] V. Bertacchi, T. Bilka, N. Braun, G. Casarosa, L. Corona, S. Cunliffe, F. Dattola, G. D. Marino, M. D. Nuccio, G. D. Pietro, *et al.*, “Track Finding at Belle II,” *Computer Physics Communications* **259** (2021) 107610. <http://arxiv.org/abs/2003.12466>. Belle II Collaboration. arXiv:2003.12466 [physics.ins-det].
- [24] R. Frühwirth and A. Strandlie, “Track fitting with ambiguities and noise: A study of elastic tracking and nonlinear filters,” *Computer Physics Communications* **120** no. 2, (1999) 197–214. <https://www.sciencedirect.com/science/article/pii/S0010465599002313>.
- [25] L. Reuter, G. De Pietro, S. Stefkova, T. Ferber, V. Bertacchi, G. Casarosa, L. Corona, P. Ecker, A. Glazov, Y. Han, *et al.*, “End-to-End Multi-Track Reconstruction using Graph Neural Networks at Belle II,” 2024. <https://arxiv.org/abs/2411.13596v2>. Belle II Collaboration.
- [26] J. Kieseler, “Object condensation: one-stage grid-free multi-object reconstruction in physics detectors, graph and image data,” *The European Physical Journal C* **80** no. 9, (2020) 886. <http://arxiv.org/abs/2002.03605>. arXiv:2002.03605 [physics.data-an].
- [27] C. Höppner, S. Neubert, B. Ketzer, and S. Paul, “A Novel Generic Framework for Track Fitting in Complex Detector Systems,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **620** no. 2-3, (2010) 518–525. <http://arxiv.org/abs/0911.1008>. arXiv:0911.1008 [hep-ex].
- [28] R. Frühwirth, “Application of Kalman filtering to track and vertex fitting,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **262** no. 2-3, (1987) 444–450. <https://linkinghub.elsevier.com/retrieve/pii/0168900287908874>.
- [29] N. L. Braun, *Combinatorial Kalman Filter and High Level Trigger Reconstruction for the Belle II Experiment*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2019. <https://publikationen.bibliothek.kit.edu/1000089317>.
- [30] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering* **82** no. 1, (1960) 35–45. <https://doi.org/10.1115/1.3662552>.

- [31] T. Kuhr, C. Pulvermacher, M. Ritter, T. Hauth, and N. Braun, “The Belle II Core Software,” *Computing and Software for Big Science* **3** no. 1, (2019) 1.
<http://arxiv.org/abs/1809.04299>. arXiv:1809.04299 [physics.comp-ph].
- [32] R. Brun and F. Rademakers, “ROOT — An object oriented data analysis framework,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **389** no. 1, (1997) 81–86.
<https://www.sciencedirect.com/science/article/pii/S016890029700048X>.
- [33] S. Agostinelli, J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G. Barrand, *et al.*, “Geant4—a simulation toolkit,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **506** no. 3, (2003) 250–303.
<https://www.sciencedirect.com/science/article/pii/S0168900203013688>.
- [34] J. Rauch and T. Schlüter, “GENFIT - a Generic Track-Fitting Toolkit,” *Journal of Physics: Conference Series* **608** (2015) 012042. <http://arxiv.org/abs/1410.3698>. arXiv:1410.3698 [physics.ins-det].
- [35] X. Ai, “Tracking with A Common Tracking Software,” 2020.
<http://arxiv.org/abs/2007.01239>. arXiv:2007.01239 [physics.ins-det].
- [36] P. Gessinger and A. Salzburger, “Next generation geometry model for Tracking in ACTS,”.
- [37] B. Yeo, “Traccc: GPU Track Reconstruction Library,”.
- [38] B. Yeo, K. A. Kusiak, C. Leggett, G. Mania, J. Niermann, N. Styles, S. N. Swatman, A. Salzburger, H. GRASLAND, P. Gessinger, *et al.*, “CTD2022: traccc - GPU Track reconstruction demonstrator for HEP,”. <https://zenodo.org/records/8119504>. Meeting Name: Connecting the Dots Workshop 2022 (CTD2022).
- [39] P. Gessinger, H. M. Gray, A. Krasznahorkay, C. Leggett, J. Niermann, A. Salzburger, S. N. Swatman, and B. Yeo, “traccc: GPU track reconstruction library for HEP experiments,” *EPJ Web of Conferences* **337** (2025) 01187.
<http://arxiv.org/abs/2505.22822>. arXiv:2505.22822 [hep-ex].