Karlsruhe Institute of Technology

ETP-BACHELOR-KA/2024-14

Track reconstruction with Hough transformation for low momentum particles at Belle II

Florian Ohlheiser

Bachelor's Thesis

27th September 2024

Institute of Experimental Particle Physics (ETP)

Reviewer:	Prof. Dr. Torben Ferber
Second Reviewer:	Dr. Giacomo De Pietro
Advisor:	Lea Reuter

Editing time: 14th May 2024 – 27th September 2024

Karlsruhe Institute of Technology

ETP-KA/2024-14

Spurrekonstruktion mit Hough Transformation für Teilchen mit geringem Impuls bei Belle II

Florian Ohlheiser

Bachelorarbeit

27. September 2024

Institut für Experimentelle Teilchenphysik (ETP)

Referent:	Prof. Dr. Torben Ferber
Korreferent:	Dr. Giacomo De Pietro
Betreuerin:	Lea Reuter

Bearbeitungszeit: 14. Mai 2024 – 27. September 2024

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, 27. September 2024

. . . .

(**Florian Ohlheiser**)

Disclaimer

Data analyses in particle physics such as in this thesis are a collaborative effort. The thesis and the ideas in this thesis build upon the work of Lea Reuter [1], who provided the Python code for the CAT Finder and Python scripts to simulate and evaluate data. The software framework basf2 [2–4] used in this thesis to simulate and evaluate data, is developed and maintained by the Belle II collaboration. My contributions include the decision at which angles and energies the data is simulated and the simulation of said data. Additionally, writing the Python code for the Hough transformation, the 2D Binary Search and the Distance Check. The metrics from the Belle II tracking group are used to evaluate the tracking performance, where Lea Reuter provided the scripts to access the necessary information. The plots with the results, starting from Chapter 4, and the tables are created by me.

This thesis incorporates the use of Artificial Intelligence (AI) tools to help with grammatical or stylistic improvement of text creation.

Grammarly [1] is utilized throughout the thesis for spell and grammar checks, as well as for paraphrasing individual, selected sentences to improve clarity and precision in academic writing. I have approved all suggested changes.

[1] Grammarly: AI writing assistant. https://www.grammarly.com/ (Accessed: 23.09.2024)

Contents

1. Introduction

To search for new physics in the flavor sector and address open questions in particle physics, such as decays in the dark sector, it is essential to reconstruct the measured data. One important part is the reconstruction of particle trajectories. The higher luminosity results in new challenges such as significantly increased backgrounds and deteriorating detector signals. For this reason, a new track finding algorithm based on graph neural networks is being developed, the CAT Finder [1]. This thesis focuses on improving the performance of this track finding algorithm in the low energy range. Specifically on the track reconstruction of particle trajectories that curl in the Central Drift Chamber. The hit efficiency of the CAT Finder is higher than the previously used baseline algorithm, a track finding algorithm that is based on the Legendre Transformation. This higher hit efficiency leads to the assignment of the complete curled track to one particle, which is useful for the analysis level. However, the following fitting algorithm only expects a single loop and has trouble fitting the whole curled track. To solve this problem the idea is to introduce a filter to the track reconstruction algorithm that filters out every loop after the first. The filter consists of the Hough transformation and will be applied after the track finding and before track fitting.

The thesis is divided into multiple parts. First, in Chapter 2, the structure of the Belle II experiment will be explained briefly. Chapter 3 clarifies the different software aspects. Afterward, the Hough transformation and how it is applied is explained in Chapter 4. In Chapter 5 the impact of the Hough transformation is tested, and different tracking metrics proposed by the Belle II tracking performance group are calculated and compared for each approach. Finally, in Chapter 6 the results are summarized, and an outlook, on what still needs improvement, is given.

2. The Belle II Experiment

2.1. Structure

2.1.1. SuperKEKB

SuperKEKB is a particle accelerator at KEK (High Energy Accelerator Research Organization) in Tsukuba, Japan, with a center of mass energy of 10.58 GeV. More precisely, an asymmetric electron-positron collider, with a designed luminosity of $8 \times 10^{35}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ [5]. SuperKEKB is an upgrade to the KEKB, an electron-positron collider operated from 1998 to 2010 with a maximum luminosity of $2.11 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$. The almost 40 times higher luminosity of SuperKEKB compared to KEKB is intended to enable the search for new physics beyond the Standard Model and do precise measurements of Standard Model quantities in the flavour sector [5]. A schematic of the SuperKEKB can be seen in Fig. 2.1.

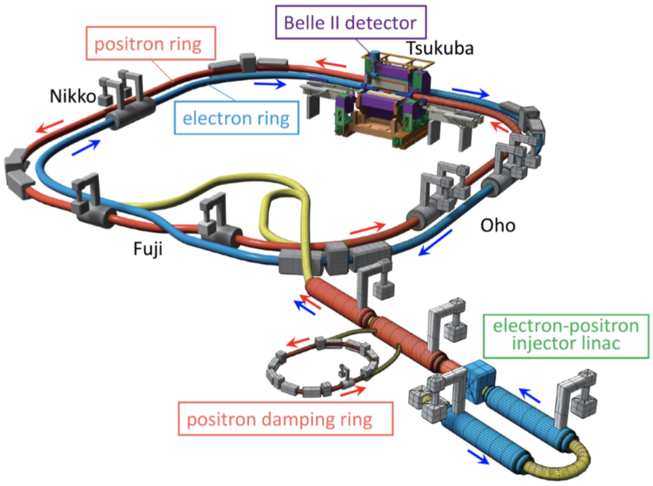


Figure 2.1.: A schematic view of the SuperKEKB Collider. Image taken from [5].

2.1.2. Belle II Detector

The Belle II detector consists of multiple different sub-detectors. The innermost part consists of two layers of pixelated silicon sensors, the Pixel Detectors (PXD), followed by four layers of double-sided silicon strip sensors, the Silicon Vertex Detectors (SVD). The PXD and SVD together are called the Vertex Detector (VXD) and measure decay vertex positions of particles, such as B mesons [6, 7]. The next sub-detector is called the Central Drift Chamber (CDC) and has three important tasks. Firstly measuring momenta and reconstructing tracks. Secondly measuring the energy loss in the CDC and thus the identification of particles with low momentum that do not reach the particle identification system. Thirdly providing reliable trigger signals for charged particles [8]. Following the CDC is the particle identification system (PID). The PID consists of two parts: the Time-Of-Propagation (TOP) counter and the Aerogel Ring-Imaging Cherenkov detector (ARICH). Their purpose is to reliably distinguish between charged kaons and pions [8]. The next sub-detector is the Electromagnetic Calorimeter (ECL), which serves the purpose of detecting photons and distinguishing electrons from hadrons. The outermost layer detects K_L^0 and muons and is called K_L^0 and muon detector (KLM) [9]. A schematic of the Belle II detector can be seen in Fig. 2.2. Additionally a top view of the Belle II detector can be seen in Fig. 2.3. More details to SuperKEKB and the components of the Belle II detector can be found at [5, 8]

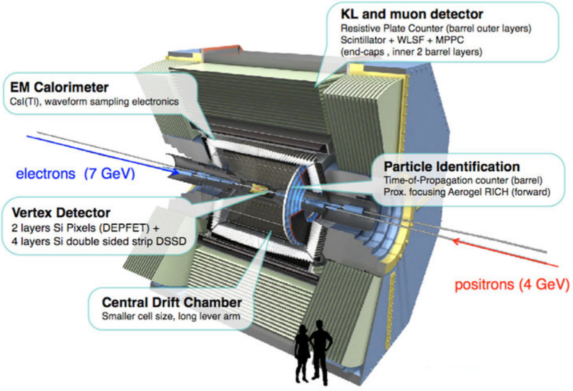

KL and muon detector
Resistive Plate Counter (barrel outer layers)
Scintillator + WLSF + MPPC
(end-caps , inner 2 barrel layers)
EM Calorimeter
CsI(Tl), waveform sampling electronics
electrons (7 GeV)
Particle Identification
Time-of-Propagation counter (barrel)
Prox. focusing Aerogel RICH (forward)
Vertex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

Figure 2.2.: A schematic view of the Belle II detector. Image taken from [10].

2.2. The Central Drift Chamber

The CDC is 2.3 m long and consists of a large chamber, that covers a polar angle between 17° and 150° and has an inner radius of 160 mm and 1130 mm [8]. The CDC contains 14 336 sense wires and 42 240 field wires arranged in 56 layers and is filled with a 50:50 mixture of He-C_2H_6. Additionally, the entire CDC is in a magnetic field of 1.5 T along the z-axis. This leads to a helix trajectory for charged particles. The layers are further categorized into nine superlayers, see Fig. 2.4 for a visualization of the layout.

The first and innermost superlayer consists of eight layers, followed by eight superlayers with six layers each. An important point is that superlayers alternate between axial layers and stereo layers, with the first superlayer consisting of axial layers. Completely parallel wires, to the magnetic field, would only lead to a 2D track on the x-y plane. For this reason, stereo layers consist of wires angled to the wires in axial layers, which allows the measurement of

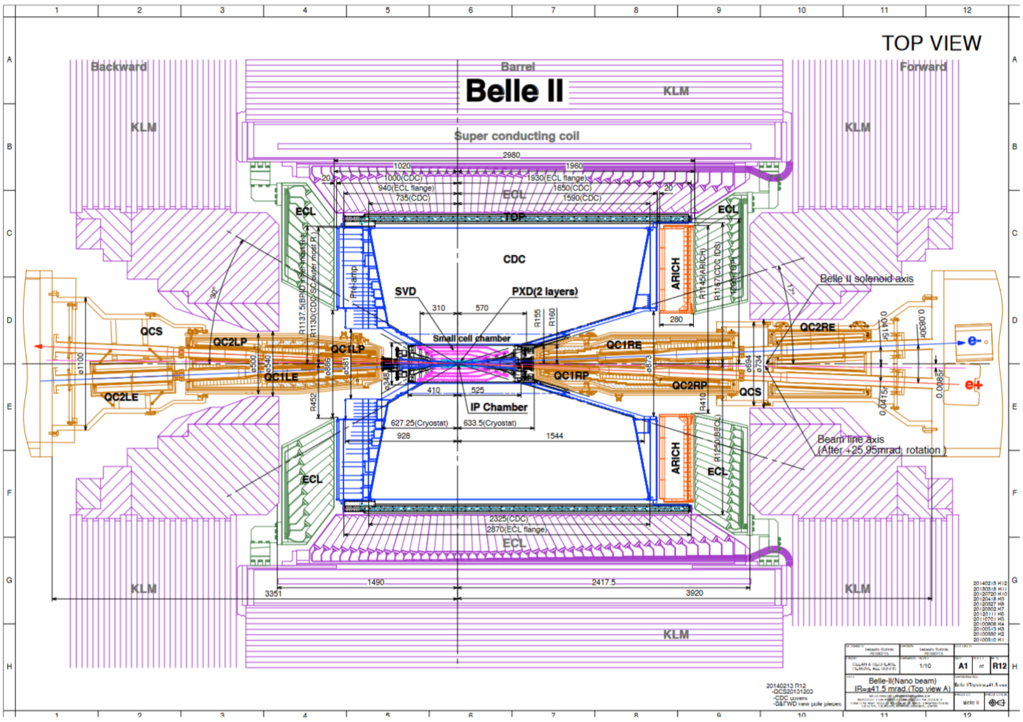


Figure 2.3.: Top view of the Belle II detector. Image taken from [9].

the particle's trajectory along the z-axis as well. This enables the measuring of a complete 3D helix track [8, 9].

Charged particles ionize the gas mixture during their path through the CDC and leave a trail of electrons. To detect these electrons, eight field wires create an electric field, which leads to the electrons drifting toward the center of these eight field wires. In the center lays one sense wire that detects the electron charge deposition. This setup of eight field wires with one sense wire in the middle is called a drift cell. Each sense wire is surrounded by eight field wires. Due to the selected field strength and gas mixture, the drift velocity is nearly constant. Thus the drift time, the time it takes one electron to reach a sense wire, is proportional to the distance. The simple distance, speed, and time relation can be used.

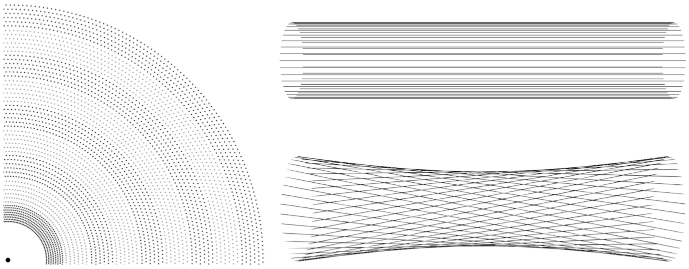


Figure 2.4.: Left: A quadrant of a slice of the r-ϕ projection of the drift chamber. Right: A visualization of stereo wires (bottom) relative to axial wires (top). The skew is exaggerated. Image taken from [11].

The result is a drift circle, with a radius equal to the drift velocity multiplied by the drift time:

$$r = v_{\text{drift}} \cdot t_{\text{drift}}. \tag{2.1}$$

The particles' size is significantly smaller than that of the drift cell, making it unlikely for a particle to pass precisely through the cell's center. Consequently, measuring the x and y coordinates at the center of the drift cell introduces an offset compared to the particle's true track. Combining all drift circles of an event yields a more accurate particle track [8, 9, 12].

However, at the start of my thesis, the setup was not designed for measuring the coordinates with drift circles. Therefore in the following everything is evaluated with the measurement of the coordinates at the center of the drift cells.

3. Software Background

3.1. Track Reconstruction at Belle II

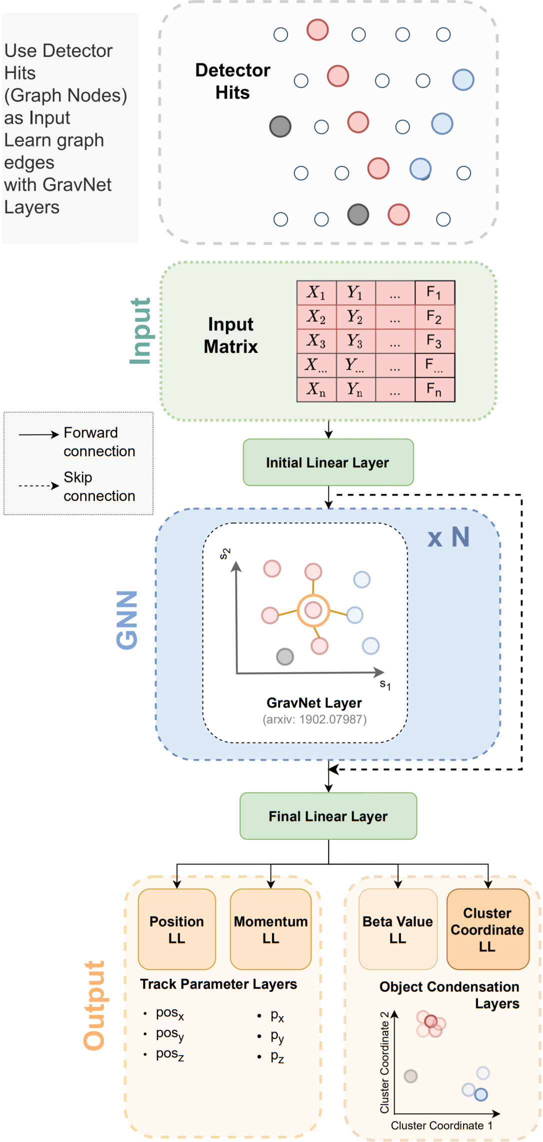
The baseline algorithm that is used for track finding in the CDC is based on the Legendre transformation. Before the Legendre transformation can be executed on the data, a Conformal transformation is applied to the x and y coordinates. This transformation reduces the circular trajectory to a straight line. Afterwards the Legendre transformation, according to [7], defined as:

$$\rho_{\pm}(\theta) = \frac{2}{x^2 + y^2 - r^2}(x\cos(\theta) + y\sin(\theta) \pm r) \qquad \theta \in [0, \pi], \tag{3.1}$$

is applied. Here, r is the measured radius of a drift circle. Similarly to the Hough transformation, which will be explained in detail in Chapter 4, the transformed hits result in sinusoidal curves in the ρ-θ space. A 2D Binary Search is applied to these sinusoidal curves. A detailed explanation of how a 2D Binary search works is given in Section 4.2. The only difference is that the baseline filter has a given resolution parameter, whereas we test which amount of iterations leads to the best result. When this resolution parameter is reached the 2D Binary Search is stopped and all hits, that correspond to the lines, that lie in the final quadrant are chosen as the track of the particle [7, 11, 12].

The baseline algorithm has trouble finding tracks that do not originate at the interaction point since the applied Legendre transformation assumes that all trajectories go through $x = 0$ and $y = 0$. To improve upon the baseline, the CDC AI Track Finder (CAT Finder) was designed [1]. The CAT Finder aims to do this through object condensation and graph neural networks (GNN). An overview of the structure of the GNN of the CAT Finder can be seen in Fig. 3.1. The CAT Finder processes the CDC hits of each event and returns a list with the CDC hits that correspond to the found track, as well as the predicted starting position, the predicted momentum and the predicted charge of the track [1]. The CDC hits are ordered before the results are passed to GENFIT [13], an open-source, experiment-independent framework for track fitting. The hit ordering is extremely relevant to ensure that the fit converges, especially for low momentum particles and curlers. First, the point at which the particle enters the CDC is extrapolated and selected as the starting position. Afterward, the CDC hit that is the closest to the previous one is chosen as the next hit, this is repeated until all CDC hits are ordered.

After either method is used, the resulting tracks are passed to GENFIT, which tries to fit them.



X_1	Y_1	...	F$_1$
X_2	Y_2	...	F$_2$
X_3	Y_3	...	F$_3$
$X_{...}$	$Y_{...}$	...	F$_{...}$
X_n	Y_n	...	F$_n$

Figure 3.1.: Structure of the GNN of the CAT Finder. Image taken from [1].

3.2. Simulating Data

Particles with sufficient transverse momentum (p_T), the amount of a particle's momentum which is perpendicular to the beam direction, pass through the CDC and leave a singular track in the CDC. The tracks of those particles are comparatively easy to find and fit, especially for low background levels. However, particles with low p_T may exit the CDC and then re-enter it, which leads to a second loop. The issue with this is that the particle loses more energy outside of the CDC, compared to in the CDC especially , leading to a track that curls in on itself, making it difficult to find and fit. As there are two possibilities for a low p_T particle to leave the CDC, we need to distinguish between the two cases. In the first case, the particle has so little p_T, that it leaves the CDC in the direction of the VXD but then re-enters the CDC. I will refer to this as an inner curler. In the second case, the particle has enough p_T to leave the CDC in the direction of the PID detectors but then re-enters the CDC. We will refer to this as an outer curler below. To properly simulate and analyze inner curlers and outer curlers, we check which p_T and which polar angle θ lead to the respective curler types.

In every following chapter the Belle II Analysis Software Framework basf2 [2–4], release-08-01-05, is used to simulate and evaluate the data. The data is simulated with the basf2 module "ParticleGun". The Python code, that is used for simulation and evaluation, can be found at [14].

3.2.1. Inner Curler

An example of an inner curler is seen in Fig. 3.2. The particle's trajectory consists of three circular shapes. The hits in the outermost circle stem from the first passage of the particle through the CDC. The second circle is the trajectory of the particle after leaving and then re-entering the CDC. The smaller radius of the second circle highlights the energy loss outside of the CDC. The third and innermost circle depicts the particle's trajectory after further energy loss outside of the CDC, resulting in a smaller circle radius.

To determine which p_T and θ values lead to an inner curler, we simulate 1000 single-muon events with low p_T, $p_\mathrm{T} = [0\,\mathrm{GeV}, 0.4\,\mathrm{GeV}]$, and varying θ angles, $\theta = [0\,^\circ, 180\,^\circ]$. Additionally, for a track to be recognized as a curler it needs to have a distance of 20 cm between two subsequent hits of the track and save these events. After that we plot a 2D histogram that consists of events with said events, see Fig. 3.3. Since an inner curler can form as soon as the particle has enough energy to re-enter the CDC after leaving it at the innermost part, we can conclude that inner curlers start at the lowest p_T values in the 2D histogram, 0.076 GeV. Furthermore, we know that inner curlers do not have enough energy to leave the CDC in the direction of the PID detectors. Thus there needs to be an upper boundary for the p_T. Tests lead to the result that tracks with a p_T of approximately 0.25 GeV and above can leave the CDC. The upper boundary of the corresponding p_T bin in Fig. 3.3 is 0.253 GeV, making this our upper boundary for p_T values. As we can see in Fig. 3.3 for p_T values between 0.076 GeV and 0.253 GeV, 80.0 % of the events have a θ value between 79 ° and 96 °, marking this interval as our θ boundaries for inner curlers.

All inner curlers in the following chapters will be simulated with the following parameters: $p_\mathrm{T} = [0.076\,\mathrm{GeV}, 0.253\,\mathrm{GeV}]$, $\theta = [79\,^\circ, 96\,^\circ]$, particle pdg $= [13, -13]$(muon and antimuon) and one particle per event.

3.2.2. Outer Curler

Fig. 3.4 shows an example of an outer curler. The trajectory of the particle is indicated by the color of the hits, starting at the brightest hit and moving towards the darkest.

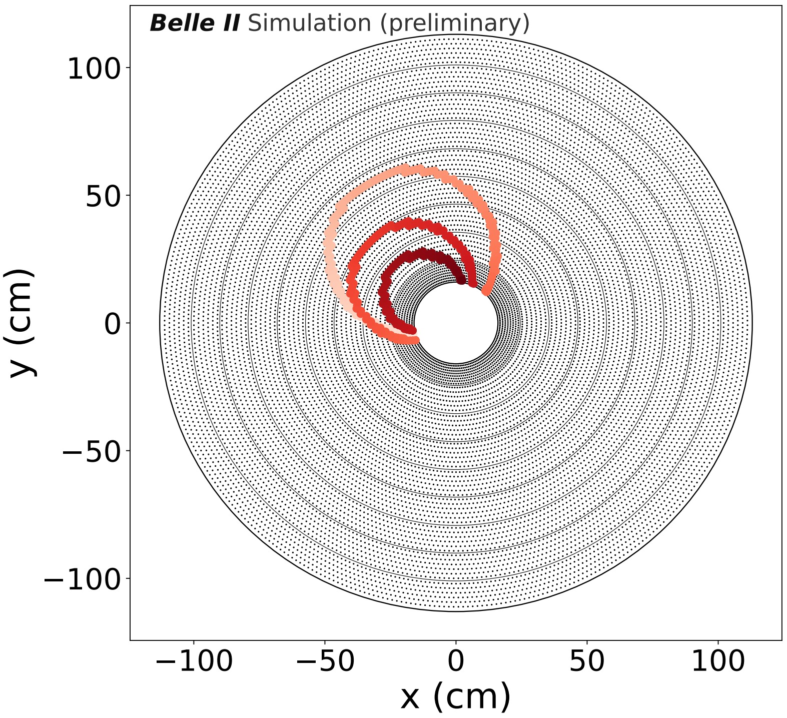


Figure 3.2.: Example of a simulated Inner Curler. The trajectory of the particle is indicated by the color of the hits, starting at the brightest hit and moving towards the darkest. Plot adapted from Lea Reuter.

We observe that the trajectory starts at the middle of the CDC and moves outwards. Subsequently, the particle exits the CDC and then re-enters it as it lacks the p_{T} to leave completely. Additionally, the circular path, after re-entering the CDC, has a smaller radius compared to the original track. Similarly to the inner curler in Section 3.2.1 the particle loses energy outside of the CDC. As the particle completes the second circular path, it exits and re-enters the CDC again, resulting in further energy loss and a consequently even smaller circular track. Compared to an inner curler, an outer curler crosses more material outside of the CDC, leading to a larger energy loss.

As mentioned in Section 3.2.1, a particle needs an p_{T} of approximately 0.25 GeV to leave the CDC. In addition, if a particle has sufficient energy to completely leave the CDC, it will not curl. For this reason, an upper limit for the p_{T} is needed. To determine the exact p_{T} and θ values leading to outer curlers, we re-examine Fig. 3.3. The lower boundary of the p_{T} for outer curlers therefore corresponds to 0.253 GeV, the lower boundary of the bin in the 2D histogram. As we can see almost all of the outer curler events take place between a p_{T} of 0.253 GeV and 0.312 GeV, thereby defining these p_{T} values as our p_{T} boundaries. The vast majority of outer curler events, 93.10 %, are in the range from 53 ° to 104 °, thus we choose this as our θ range.

Subsequently, all outer curler events discussed in the following chapters will be simulated with the following parameters: $p_{\mathrm{T}} = [0.253\,\mathrm{GeV}, 0.312\,\mathrm{GeV}]$, $\theta = [53\,^\circ, 104\,^\circ]$, particle pdg $= [13, -13]$(muon and antimuon) and one particle per event.

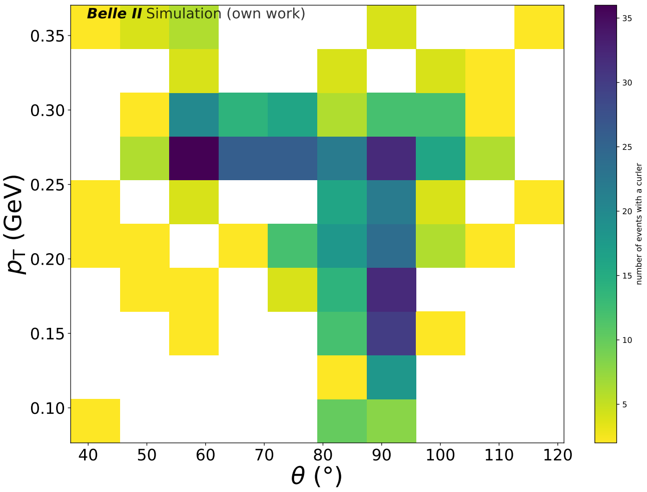


Figure 3.3.: 2D histogram to determine which p_T and θ values lead to Inner and Outer Curler. Out of 1000 simulated muon events 544 are curling events, only these 544 events are plotted in the 2D histogram.

Compared to inner curlers, outer curler events display a smaller interval for the p_T. However, we notice a larger θ interval for outer curlers than for inner curlers. As discussed in Section 2.2, the geometry of the CDC changes according to the coordinates. See Fig. 2.3 for a top view on the Belle II detector. In this schematic the horizontal axis corresponds to the z-axis. We observe that for small vertical distances from the SVD, the CDC is restricted in size in the z direction due to the size of the electron and positron rings. This limitation in z direction means that if an inner curler with a polar angle, θ, deviating significantly from $90\,^\circ$, leaves the CDC in z direction. If we look at the x-y projection for inner curlers and outer curlers, we see a circular shaped track. However inner curlers and outer curlers with $\theta \neq 90\,^\circ$ move in z direction therefore the true trajectory has the shape of a helix. To be recognized as a Curler the particle has to re-enter the CDC with a distance of 20 cm between two hits. Since an inner curler needs to complete a full helix loop to re-enter and to be recognized as a Curler, the particle travels further along the z-axis compared to an outer curler that completes less than half of the first loop. Due to these circumstances, the particle does not curl multiple times in the CDC and can not be identified as an inner curler, if it's θ angle deviates too much from $90\,^\circ$.

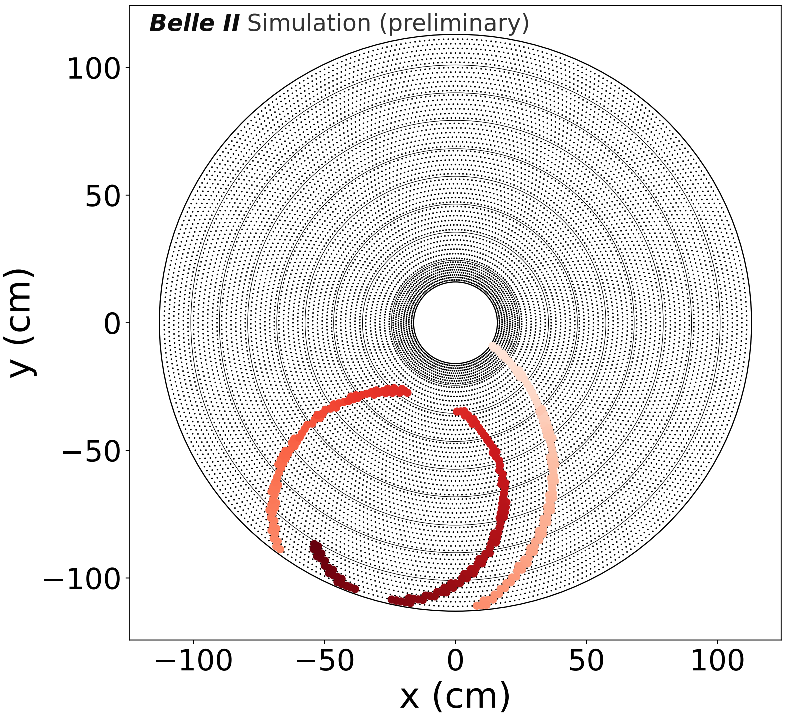


Figure 3.4.: Example of a simulated Outer Curler. The trajectory of the particle is indicated by the color of the hits, starting at the brightest hit and moving towards the darkest. Plot adapted from Lea Reuter.

4. Hough-Based Track Filtering

Since tracks with low p_{T} can curl multiple times in the CDC, GENFIT has difficulties fitting these tracks. A filtering of the tracks can improve the performance which is tested in this thesis. Due to the circular shape of tracks that curl several times, it is useful to apply a Hough transformation (HT). The HT filters the multiple circle-like tracks in the CDC to a singular circle-like track

The aim with this is that we remove hits of a track, that leaves the CDC and loses energy outside the CDC, for example in the VXD, but re-enters the CDC. The goal is to keep all the hits of the first loop, since removing hits results in a deteriorating resolution. Additionally, if too many hits are removed GENFIT can not fit the track, because at least five hits are needed. Finally, as mentioned in Section 2.2, the stereo layers are angled, so that the hits in the stereo layers contain information about the z position, the removal of which would therefore be undesirable. Fig. 4.1 is selected as an example on which a Hough transformation is performed, in Section 4.1.

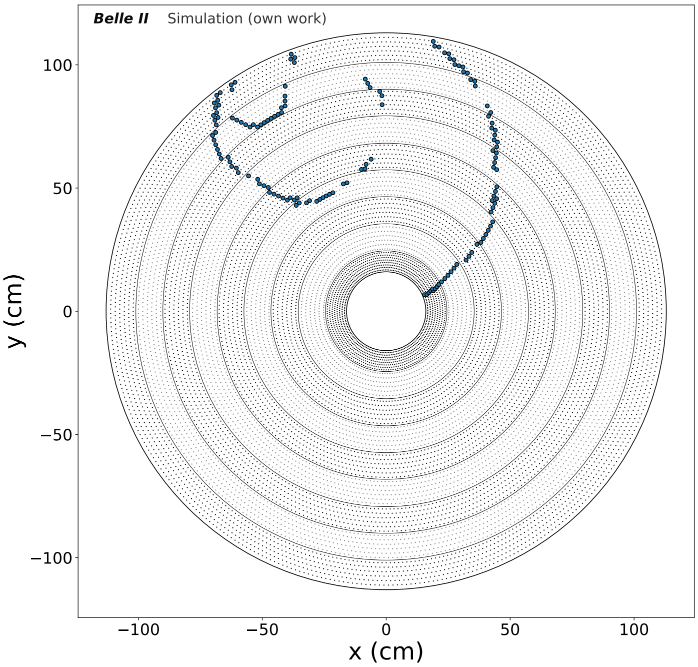


Figure 4.1.: Example of a simulated outer curler the Outer Curler, that is used as an example for the Hough transformation.

4.1. Hough transformation

The HT transforms lines into sinusoidal curves, however since inner curlers and outer curlers are circular shaped it is necessary to first perform a Conformal Transformation. According to [12] the Conformal Transformation is defined as:

$$u = \frac{2x}{x^2 + y^2}, \quad v = \frac{2y}{x^2 + y^2}. \tag{4.1}$$

Conformal means we preserve the angles between curves [15]. By performing a Conformal Transformation we can transform a circle to a straight line. In Fig. 4.2 we can see the result of a Conformal Transformation executed on the outer curler in Fig. 4.1.

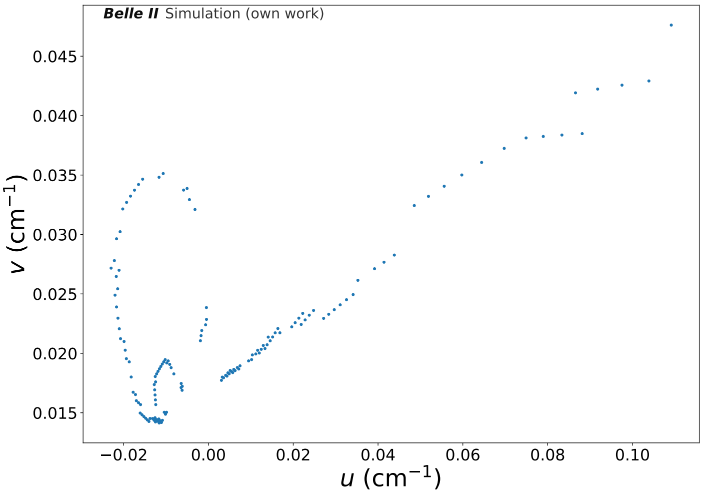


Figure 4.2.: Conformal transformation of the Outer Curler seen in Fig. 4.1.

As we can see, the outer curler is not transformed into just one line. There are three main reasons for this. First, the Conformal Transformation we apply assumes that the circle to transform passes through $x = 0$ and $y = 0$, which is no longer the case after the particle lost energy after passing through material as described in Chapter 3. Second, as mentioned in Section 2.2, the particle does not necessarily pass through the center of the drift cell. Therefore, there is a shift in the x and y coordinates, compared to the true path of the particle. Lastly, as mentioned in Section 2.2, stereo layers are angled. Thus the x and y coordinates are depending on z, resulting in an offset of the x and y coordinates for stereo wires, if the particle moves in z direction. This means that the hits are not on a perfect circle, but can deviate within a few centimeters due to the assumptions we make. Therefore the hits are not on a perfect line after transformation.

Finally we can transform these hits into the $\rho - \theta$ space, also called Hough space. According to [15] the HT is defined as:

$$\rho = u\,cos\,(\theta) + v\,sin(\theta) \qquad \theta \in [0, \pi]. \tag{4.2}$$

This transforms one hit at a time. Thus we get the same amount of sinusoidal curves as hits in the CDC. We use a 2D Binary Search to find the ρ and θ intervals where the most lines are. A CDC hit corresponding to a line in this interval is kept. A CDC hit corresponding to a line not in this interval is removed. Finally, the remaining CDC hits are approximately a circle given by the HT. See Fig. 4.3 for an example of a HT. Here the sinusoidal curves are the transformed wire hits.

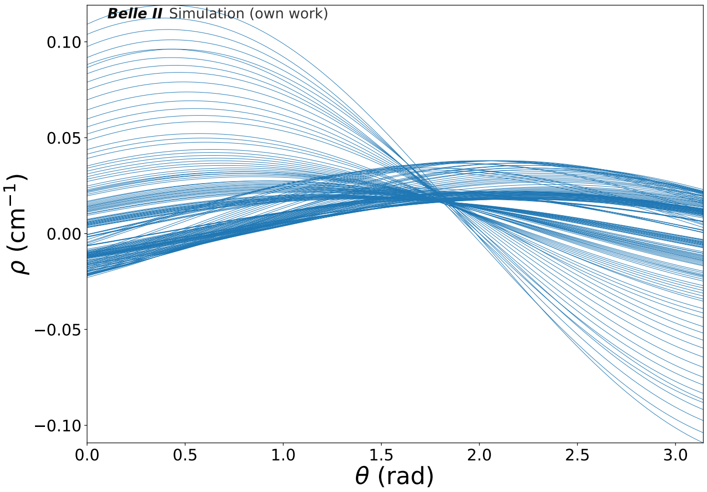


Figure 4.3.: CDC hits after the HT on Fig. 4.1, every sinusoidal curve corresponds to a CDC hit in Fig. 4.1.

4.2. 2D Binary Search

To find the final interval, in which the most lines lie, we divide the Hough space into four equal-sized bins. Next, we search for the quadrant with the most lines and repeat the process until a certain amount of iterations are reached. We determine the needed amount of iterations through testing which amount leads to the highest track charge efficiency ($\varepsilon_{\text{track,charge}}$), this can be seen in Chapter 5. An example of a 2D Binary Search can be seen in Fig. 4.4. The region of interest is zoomed in to highlight which quadrant has the most lines. As we can see in the zoomed-in image, the red rectangle has the most lines. Thus the lines in the red rectangle represent the wire hits accepted by the HT. The lines that are not in the red rectangle represent the hits that are filtered.

After applying the HT and the 2D Binary Search we finally get the filtered wire hits, see Fig. 4.5. Additionally in Fig. A.1 we can see the filtered conformal transformation.

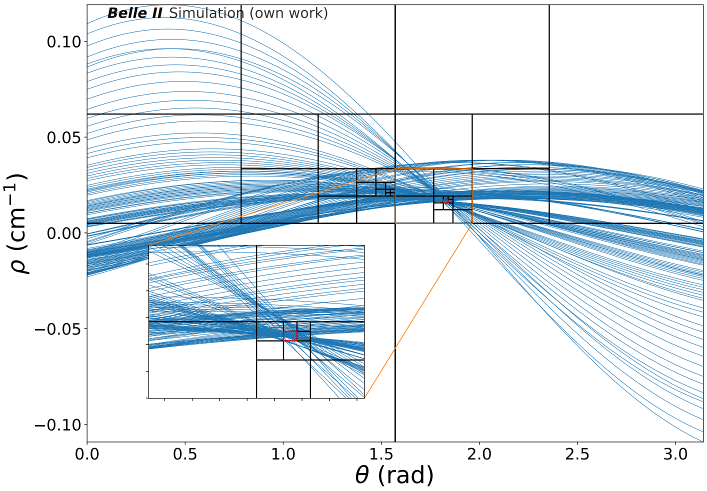


Figure 4.4.: 2D Binary Search after applying the HT on Fig. 4.1.

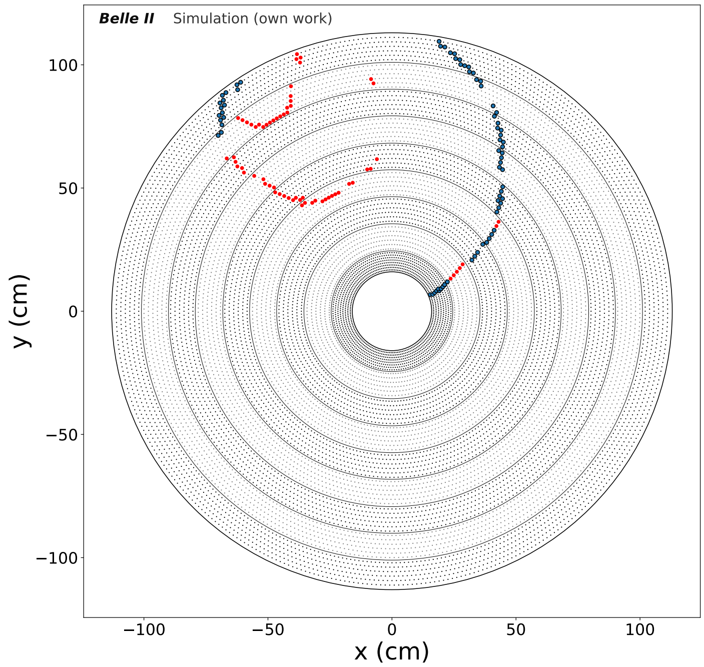


Figure 4.5.: Filtered Track after the Hough transformation and the 2D Binary Search. The hits that were filtered out are marked red.

5. Results

To improve the number of fitted tracks, we try different approaches. For each approach, we apply the HT and compare the results. To determine the best result we look at different tracking metrics proposed by the Belle II tracking performance group. As established in Section 4.2 removing hits of a track might lead to a deteriorated resolution. Additionally the CAT Finder rejects tracks with less than seven hits. To prevent this we define and calculate the track finding efficiency. The track finding efficiency, $\varepsilon_{\text{found}}$, is the ratio of tracks, that are found by the CAT Finder and satisfy the requirements after post-processing (n_{found}(matched)), to the number of all simulated particles that are matched to at least one hit($n_{\text{simulated}}(\geq 1\,\text{matched hit})$):

$$\varepsilon_{\text{found}} = \frac{n_{\text{found}}(\text{matched})}{n_{\text{simulated}}(\geq 1\,\text{matched hit})}. \tag{5.1}$$

Next, for an overview of how well our approach fits the simulated tracks, we look at the track efficiency as the ratio of fitted tracks to the number of simulated particles:

$$\varepsilon_{\text{track}} = \frac{n_{\text{tracks}}(\text{fitted})}{n_{\text{simulated}}(\geq 1\,\text{matched hit})}. \tag{5.2}$$

Finally, we look at the track charge efficiency, which is the ratio of matched tracks with the correct charge to the number of simulated particles:

$$\varepsilon_{\text{track,charge}} = \frac{n_{\text{tracks}}(\text{matched to particle and charge})}{n_{\text{simulated}}(\geq 1\,\text{matched hit})}. \tag{5.3}$$

For each of the efficiencies, a value of 1.0 or 100% would be a perfect result. For example $\varepsilon_{\text{track,charge}} = 100.0\%$ would mean, we matched every track to a simulated particle with the correct charge. The most important of the three efficiencies is the track charge efficiency since the baseline algorithm has trouble matching the correct charge to the particles, especially for inner curlers, and our goal is to improve upon the baseline algorithm. Additionally, to reconstruct the whole event, both the correct charge and particle are important. 1000 events each are chosen for inner curlers and outer curlers to analyze the different approaches.

5.1. Iterations

The first step is to analyze which amount of iterations during the 2D Binary Search leads to the best track charge efficiency. During the 2D Binary Search, different numbers of iterations lead to varying efficiencies. Thus we check, for inner curlers and outer curlers separately, which number of iterations leads to the best track charge efficiency.

5.1.1. inner curler

As we see in Table 5.1 the efficiencies for inner curlers improve with each added iteration of the 2D Binary Search, up until the seventh iteration. The reason for this is, that we reduce the number of hits that do not lie on the circle indicated by the HT. For every additional iteration after the seventh, the efficiencies worsen. As mentioned in Section 4.2, removing too many hits of a track may lead to it not getting fitted by GENFIT. Furthermore, we can see a drop in the track finding efficiency for each increasing iteration past the fourth.

Table 5.1.: Efficiencies in percentage and absolute values for various iterations for inner curlers.

Iterations	$\varepsilon_{\text{found}}(\%)$	$\varepsilon_{\text{track}}(\%)$	$\varepsilon_{\text{track,charge}}(\%)$	n_{found}	n_{tracks}	$n_{\text{correct charge}}$
1	98.8	56.0	49.5	988	560	495
2	98.8	56.0	49.5	988	560	495
3	98.9	56.0	49.5	989	560	495
4	98.9	56.1	49.6	989	561	496
5	98.8	57.6	51.1	988	576	511
6	98.8	61.2	52.5	988	612	525
7	98.2	64.3	55.4	982	643	554
8	97.3	60.4	51.7	973	604	517
9	86.6	36.2	28.2	866	362	282

5.1.2. outer curler

For outer curlers, we observe, that the efficiencies improve up to six iterations. Afterward, the performance drops with each additional iteration. Similarly to Section 5.1.1 the deterioration of the efficiencies for a higher number of iterations is due to the removal of too many CDC hits, thus making it difficult for the GENFIT to fit them. However, if we compare the results for both outer curlers and inner curlers, we can see that the track charge efficiency for six and seven iterations are close together. Going forward it makes sense to test different numbers of iterations for each approach. We select an interval from four iterations to eight iterations for the 2D Binary Search to make sure we do not overlook an increase or decrease in efficiency.

Table 5.2.: Efficiencies in percentage and absolute values for outer curlers.

Iterations	$\varepsilon_{\text{found}}(\%)$	$\varepsilon_{\text{track}}(\%)$	$\varepsilon_{\text{track,charge}}(\%)$	n_{found}	n_{tracks}	$n_{\text{correct charge}}$
1	99.5	53.2	52.3	995	532	523
2	99.5	53.3	52.5	995	533	525
3	99.5	53.5	52.7	995	535	527
4	99.2	56.0	54.5	992	560	545
5	99.0	59.8	58.4	990	598	584
6	98.4	64.6	63.0	984	646	630
7	97.5	64.4	61.4	975	644	614
8	96.5	56.8	51.7	965	568	517
9	92.4	42.5	37.7	924	425	377

5.2. x and y coordinates at $z = 0$

As mentioned in Section 4.1, the x and y coordinates for stereo wires are depending on z. Thus it is important to test if measuring the coordinates at $z = 0$, compared to the middle of the wire in Section 5.1, has an impact on the efficiencies. We expect particles with $\theta = 90^\circ$ to have aligned stereo and axial layers, thus improving the HT accuracy for those particles.

5.2.1. Inner Curler at $z = 0$

We can see, in Table 5.3, that measuring the coordinates at $z = 0$ improves the calculated efficiencies for every number of iterations. Yet seven iterations still performs the best with a track finding efficiency of 98.4 %, a track efficiency of 68.0 % and a track charge efficiency of 59.1 %.

Table 5.3.: Efficiencies in percentage and absolute values at $z = 0$ for inner curlers.

Iterations	$\varepsilon_{\text{found}}(\%)$	$\varepsilon_{\text{track}}(\%)$	$\varepsilon_{\text{track,charge}}(\%)$	n_{found}	n_{tracks}	$n_{\text{correct charge}}$
4	98.9	56.5	50.0	989	565	500
5	98.8	58.6	51.7	988	586	517
6	98.8	63.0	55.1	988	630	551
7	98.4	68.0	59.1	984	680	591
8	97.5	65.9	57.7	975	659	577

The reason for this improvement is that, as mentioned in Section 4.1, the angled stereo layers lead to an offset in the x and y coordinates, if the particle moves in z direction. However if the particle does not move in z direction and the coordinates are measured at the middle of the wire it will lead to a shift in the x and y coordinates as well. The HT followed by the 2D Binary Search filters these shifted hits, since they do not lie on the circle given by the HT, leading to deteriorating efficiencies. Thus using the wire coordinates at $z = 0$ for the HT, as inner curlers occur at $\theta \approx [80^\circ, 95^\circ]$, improves the efficiencies.

5.2.2. Outer Curlers at $z = 0$

Table 5.4 shows the efficiencies for outer curlers measured at $z = 0$ for varying number of iterations. The best result is at seven iterations for the 2D Binary Search. If we compare the previously best result, six iterations x and y measured at the middle of the wire, we observe a decrease of 0.6 percentage points for the track finding efficiency. However the track efficiency increases by 4.2 percentage points to 68.8 % and the track charge efficiency increases by 3.4 percentage points to 66.4 %. This is again because we expect outer curlers at $\theta \approx [50\,^{\circ}, 105\,^{\circ}]$

Table 5.4.: Efficiencies in percentage and absolute values at $z = 0$ for outer curlers.

Iterations	$\varepsilon_{\text{found}}(\%)$	$\varepsilon_{\text{track}}(\%)$	$\varepsilon_{\text{track,charge}}(\%)$	n_{found}	n_{tracks}	$n_{\text{correct charge}}$
4	99.2	57.6	55.7	992	576	557
5	99.3	61.2	59.7	993	612	597
6	98.3	65.8	64.6	983	658	646
7	97.8	68.8	66.4	978	688	664
8	97.4	58.4	54.6	974	584	546

Going forward we will take the x and y coordinates at $z = 0$, since for both inner curlers and outer curlers the efficiencies improve.

5.3. Distance Check

As the angular distribution for outer curlers is much broader than for inner curlers, we try another approach. If a hit has a distance equal to or greater than d from the last hit, we remove this hit and every following hit. We will call this method Distance Check in the following. The distance d is defined as:

$$d = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}. \tag{5.4}$$

We apply this distance cut, in the outermost superlayer, in the x-y plane. The mostly correct ordering of the hits for outer curlers, as mentioned in Section 3.1, enables this straightforward method for outer curlers. For inner curlers the hit ordering is not good enough and thus the Distance Check does not improve the results for inner curlers. We choose seven iterations for the 2D Binary Search, but the impact of different iterations for the Distance Check will be studied later in this section. We test different distances for the Distance Check and calculate the efficiencies, see Table 5.5.

Table 5.5.: Efficiencies in percentage and absolute values for the Distance Check with seven iterations for the 2D Binary Search on outer curlers.

d(cm)	$\varepsilon_{\text{found}}$(%)	$\varepsilon_{\text{track}}$(%)	$\varepsilon_{\text{track,charge}}$(%)	n_{found}	n_{tracks}	$n_{\text{correct charge}}$
1	93.1	76.6	74.2	931	766	742
2	94.7	77.9	75.1	947	779	751
3	95.6	77.5	74.6	956	775	746
4	96.8	79.6	76.8	968	796	768
5	97.0	79.2	76.6	970	792	766
6	97.1	79.0	76.2	971	790	762
7	97.2	78.9	76.2	972	789	762
10	97.5	77.8	75.2	975	778	752
15	97.5	77.5	74.7	975	775	747
20	97.5	76.5	73.9	975	765	739

We observe an increase in the track finding efficiency for increasing distance. However, the track efficiency and track charge efficiency grow only up to the best value at 4 cm, afterward decreasing with more distance. Compared to our previous result for $z = 0$, see Section 5.2.2, we recognize a drop in the track finding efficiency. Nevertheless, the track efficiency and track charge efficiency improve significantly. Comparing the best results of both approaches, seven iterations for $z = 0$ and a maximum of 4 cm for the distance check, we can see that the Distance Check performs 10.8 percentage points better than the HT without the Distance Check for the track efficiency. The track charge efficiency of the Distance Check surpasses the normal HT at $z = 0$ by 10.4 percentage points.

Since different numbers of iterations for the 2D Binary Search yielded different results for the previous approaches, we test the impact of the number of iterations on the efficiencies for the Distance Check. For the Distance Check with different iterations, see Appendix A and the following tables. The general trend is that combining the Distance Check with a lower number of iterations for the 2D Binary Search leads to increasing efficiencies. Yet using one iteration or not filtering the tracks at all, leads to a slight deterioration of the efficiencies. The cause of this is that the Distance Check removes everything after the first half loop. But if the second loop is close, the additional HT can remove hits from there too.

The best result for outer curlers is the Distance Check with a maximum distance between two hits of 2 cm and two iterations for the 2D Binary search, see Table 5.6 for the efficiencies.

Table 5.6.: Efficiencies in percentage and absolute values for the Distance Check with two iterations for the 2D Binary Search on outer curlers.

d(cm)	$\varepsilon_{\text{found}}$(%)	$\varepsilon_{\text{track}}$(%)	$\varepsilon_{\text{track,charge}}$(%)	n_{found}	n_{tracks}	$n_{\text{correct charge}}$
1	99.0	91.9	91.5	990	919	915
2	99.1	92.2	92.0	991	922	920
3	99.2	91.9	91.7	992	919	917
4	99.2	91.9	91.7	992	919	917
5	99.3	91.0	90.5	993	910	905
6	99.3	90.3	89.7	993	903	897

The track efficiency for this result improves by 23.4 percentage points compared to not using the Distance Check. Additionally, the track charge efficiency improves by 25.6 percentage points and even the track finding efficiency improves by 1.3. In the following section, we will compare this final result with $d = 2\,\mathrm{cm}$ and 2 iterations for the 2D Binary Search to the Legendre filter of the baseline algorithm.

5.4. Comparison with Baseline

In addition to comparing the efficiencies of the baseline algorithm with the HT filter we look at three additional tracking metrics proposed by the Belle II tracking performance group. Firstly, the fake rate, $\mathfrak{r}_{\text{fake}}$, which is defined as the ratio of fake tracks to the number of all fitted tracks:

$$\mathfrak{r}_{\text{fake}} = \frac{n_{\text{fake}}}{n_{\text{tracks}}} \; . \tag{5.5}$$

Secondly, the ratio of number of clone tracks to the number of all tracks that are related to a particle is called the clone rate:

$$\mathfrak{r}_{\text{clone}} = \frac{n_{\text{clone}}}{n_{\text{tracks}}(\text{related to particle})} \; . \tag{5.6}$$

Thirdly, the wrong charge rate, $\mathfrak{r}_{\text{wrong charge}}$, is defined as the ratio of matched tracks with the wrong charge to the number tracks that are matched to a particle:

$$\mathfrak{r}_{\text{wrong charge}} = \frac{n_{\text{tracks}}(\text{matched to particle and wrong charge})}{n_{\text{tracks}}(\text{matched to particle})} . \tag{5.7}$$

The difference compared to the efficiencies is that for the clone rate, clone rate and wrong charge rate a perfect result would be 0 %. Finally, a brief examination of the relative p_{T} resolution, the absolute p_{z} resolution and the absolute θ resolution is done.

After [11], the resolution is defined as:

$$R_x = P_{68\%}(|x - P_{50\%}(x)|), \tag{5.8}$$

P_q calculates the qth percentile of a distribution.

For the comparison between baseline and Hough filter 10000 events each are chosen for inner curlers and outer curlers.

5.4.1. Inner Curler

Table 5.7 shows the efficiencies for the CAT Finder with HT filter and the baseline algorithm. Additionally the CAT Finder without filter is shown for reference, how much the HT improves the results. The baseline algorithm performs the best for the track efficiency, however it performs the worst for the track charge efficiency with 37.02 %. The CAT Finder with HT filter shows the best performance for the track charge efficiency with 60.19 %, an increase of 23.17 percentage points compared to the baseline algorithm and an increase of 9.37 percentage points compared to the CAT Finder without filter.

Table 5.7.: Comparing the efficiencies for the different methods for inner curlers (10000 events).

Method	$\varepsilon_{\text{found}}(\%)$	$\varepsilon_{\text{track}}(\%)$	$\varepsilon_{\text{track,charge}}(\%)$	n_{found}	n_{tracks}	$n_{\text{correct charge}}$
CAT Finder	99.05	56.72	50.82	9905	5672	5082
Hough filter	98.46	67.81	60.19	9846	6781	6019
Baseline	89.26	74.08	37.02	8926	7408	3702

In Table 5.8 the fake, clone and wrong charge rates are shown. We observe a slight increase in fake rate, for the HT filter, compared to the baseline algorithm. However both the clone rate and the wrong charge rate decrease drastically for the HT filter compared to the baseline algorithm. For the Hough filter, the clone rate is 7.48 % compared to 23.74 % for the baseline algorithm. The wrong charge rate is 11.24 % for the Hough filter, a decrease of 38.79 percentage points, compared to 50.03 % for the baseline algorithm.

Table 5.8.: Comparing the fake, clone and wrong charge rate for the different methods for inner curlers (10000 events).

Method	$\mathfrak{r}_{\text{fake}}(\%)$	$\mathfrak{r}_{\text{clone}}(\%)$	$\mathfrak{r}_{\text{wrong charge}}(\%)$	n_{fake}	n_{clone}	$n_{\text{wrong charge}}$
Hough filter	0.64	7.48	11.24	110	549	762
Baseline	0.33	23.74	50.03	191	2308	3706

Finally, in Fig. 5.1 the relative p_{T} resolution for events that have the correct charge and are fitted by both the HT filter and the baseline algorithm are plotted. In Fig. 5.2 the absolute p_{z} resolution is seen and Fig. 5.3 shows the absolute θ resolution. We can see that for events with the correct charge and fitted by both methods the baseline algorithm performs slightly better. Additionally, since the baseline algorithm fits significantly less events with the correct charge than the HT filter, we look at events that are fitted by both methods (relative p_{T} resolution in Fig. A.2, absolute p_{z} resolution in Fig. A.3, absolute θ resolution in Fig. A.4), as well as every event that was fitted by the individual method, regardless of whether it was also fitted by the other method (relative p_{T} resolution in Fig. A.5, absolute p_{z} resolution in Fig. A.6, absolute θ resolution in Fig. A.7). The resolutions for these events show an improvement compared to the baseline algorithm.

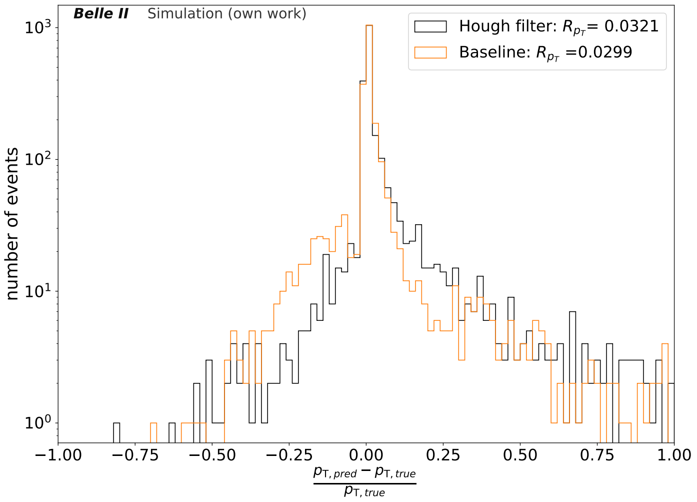


Figure 5.1.: Comparison of the relative p_T resolution between the Hough filter and the baseline algorithm for inner curlers. Only events, that are fitted by both methods and have the correct charge for both methods are used in the plot.

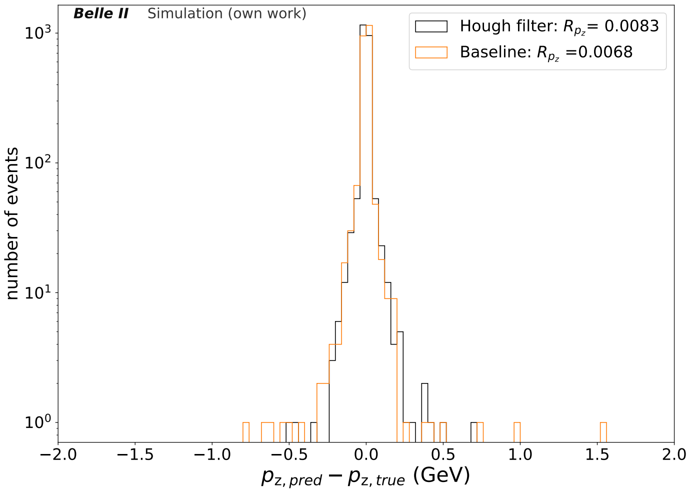


Figure 5.2.: Comparison of the absolute p_z resolution between the Hough filter and the baseline algorithm for inner curlers. Only events, that are fitted by both methods, and have the correct charge for both methods are used in the plot. Histograms are clipped at ± 2 GeV.

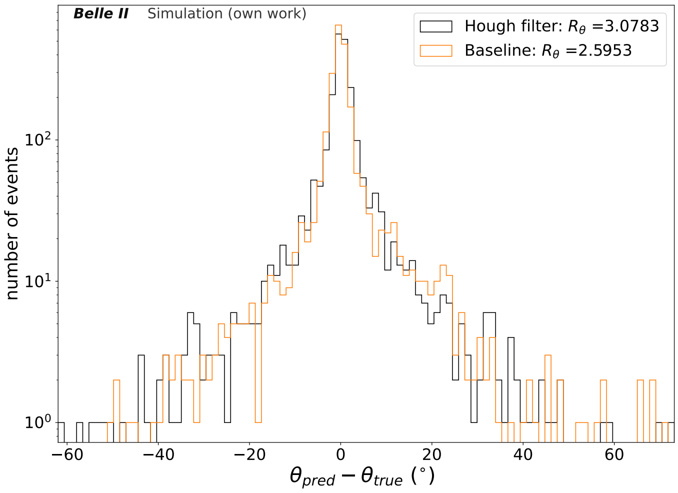


Figure 5.3.: Comparison of the absolute θ resolution between the Hough filter and the baseline algorithm for inner curlers. Only events, that are fitted by both methods, and have the correct charge for both methods are used in the plot.

5.4.2. Outer Curler

In this section we discuss the evaluation of efficiencies, rates and resolutions for outer curlers. For the efficiencies we look at Table 5.9. Compared to the standard CAT Finder we can see a huge increase for the Hough filter in the track efficiency of 35.78 percentage points to 91.98 % and the track charge efficiency increases drastically as well, from 55.10 % for the standard CAT Finder to 91.35 % for the Hough filter. However this time the baseline algorithm already has a high track charge efficiency. The track charge efficiency of the CAT Finder with HT filter only increases by 1.18 percentage points to 91.35 %. The only significant improvement is the track finding efficiency with an increase of 7.32 percentage points, compared to the baseline algorithm, to 99.62 %.

Table 5.9.: Comparing the efficiencies for the different methods for outer curlers (10000 events).

Method	$\varepsilon_{\text{found}}(\%)$	$\varepsilon_{\text{track}}(\%)$	$\varepsilon_{\text{track,charge}}(\%)$	n_{found}	n_{tracks}	$n_{\text{correct charge}}$
CAT Finder	99.70	56.20	55.10	9970	5620	5510
Hough filter	99.62	91.98	91.35	9962	9198	9135
Baseline	92.30	91.32	90.17	9230	9132	9017

In Table 5.10 the fake rate, clone rate and wrong charge rate is shown. We observe a slight improvement for the fake and wrong charge rate. The more significant difference is the clone rate, where we get an improvement of 2.68 percentage points to 1.68 %.

Table 5.10.: Comparing the fake, clone and wrong charge rate for the different methods for outer curlers (10000 events).

Method	$\mathfrak{r}_{\text{fake}}(\%)$	$\mathfrak{r}_{\text{clone}}(\%)$	$\mathfrak{r}_{\text{wrong charge}}(\%)$	n_{fake}	n_{clone}	$n_{\text{wrong charge}}$
Hough filter	0.36	1.68	0.68	101	157	63
Baseline	0.78	4.36	1.26	217	422	120

Looking at the resolution plots, for events that have the correct charge and are fitted by both the HT filter and the baseline algorithm (relative p_{T} resolution in Fig. 5.4, absolute p_{z} resolution in Fig. 5.5, absolute θ resolution in Fig. 5.6), shows once more that the baseline algorithm has a better resolution compared to the HT filter. The same additional resolution plots as for inner curlers are created for outer curlers as well. Events that are fitted by both methods (relative p_{T} resolution in Fig. A.8, absolute p_{z} resolution in Fig. A.9, absolute θ resolution in Fig. A.10) and every fitted event (relative p_{T} resolution in Fig. A.11, absolute p_{z} resolution in Fig. A.12, absolute θ resolution in Fig. A.13). However, unlike for inner curlers, the baseline algorithm also performs better if we compare events that both methods fitted or if we look at all fitted events.

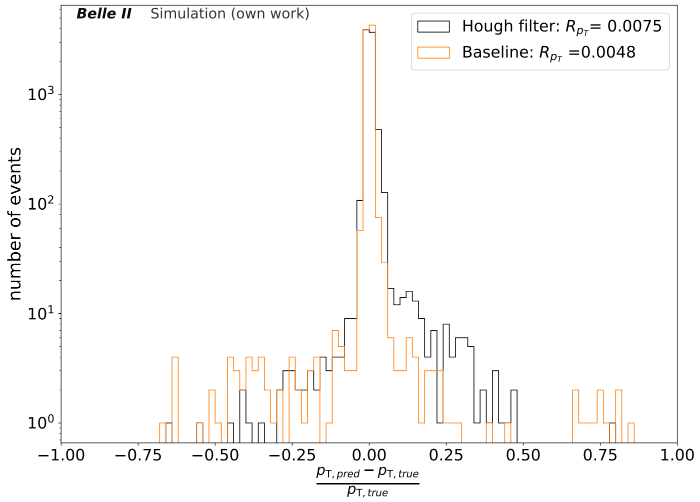


Figure 5.4.: Comparison of the relative p_{T} resolution between the Hough filter and the baseline algorithm for outer curlers. Only events, that are fitted by both methods are used in the plot.

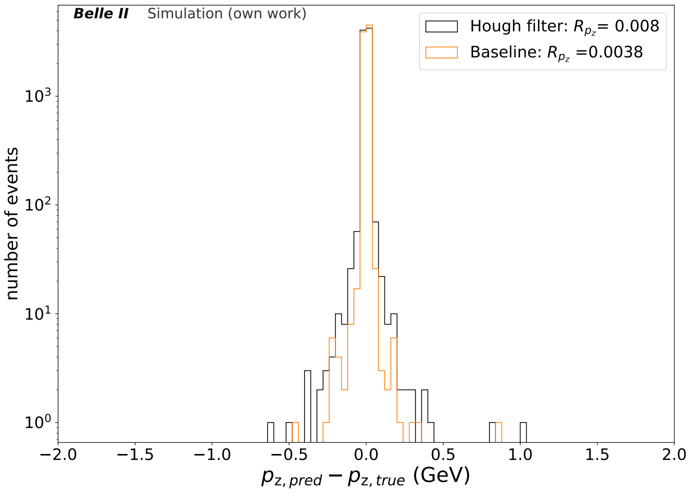


Figure 5.5.: Comparison of the absolute p_z resolution between the Hough filter and the baseline algorithm for outer curlers. Only events, that are fitted by both methods, and have the correct charge for both methods are used in the plot. Histograms are clipped at ± 2 GeV.

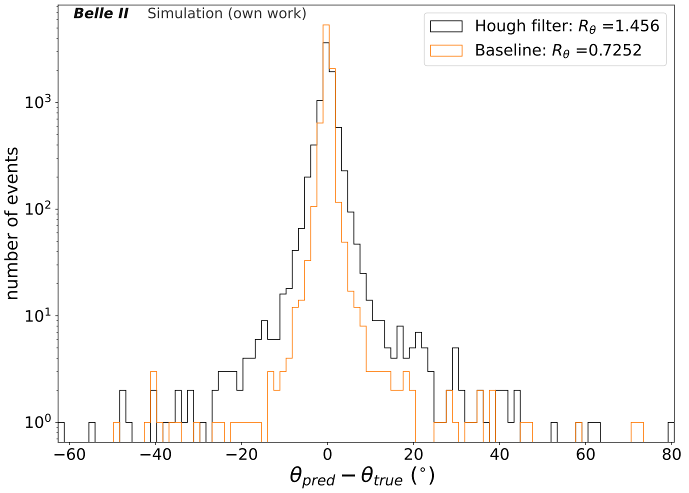


Figure 5.6.: Comparison of the absolute θ resolution between the Hough filter and the baseline algorithm for outer curler. Only events, that are fitted by both methods, and have the correct charge for both methods are used in the plot.

6. Conclusion and Outlook

As stated in Chapter 1, the aim of this thesis is the improvement of the performance of the new GNN based CAT Finder for low p_{T} events, specifically the improvement of the track charge efficiency. We identified two different problematic trajectories, firstly inner curlers and secondly outer curlers. The CAT Finder has a high hit efficiency, therefore it matches the complete curled track to one particle. Since the track fitting algorithm has trouble fitting trajectories with more than one loop it has trouble fitting these tracks. To solve this problem we developed a method each for inner curlers and outer curlers, to filter out hits that do not belong to the first loop.

For inner curlers, we choose the Hough transformation with seven iterations for the 2D Binary Search. When comparing the efficiencies with the standard CAT Finder, we noticed an improvement in the track charge efficiency of 9.37 percentage points to 60.19 % and a significant improvement of 23.17 percentage points compared to the baseline algorithm. In addition to the efficiencies we also compared the fake rate, clone rate, and wrong charge rate of our filter with the baseline algorithm. The wrong charge rate of the Hough filter, 11.24 %, decreases by 38.79 percentage points compared to the baseline algorithm. The clone rate is 7.48 % compared to 23.74 % for the baseline algorithm. This means, that both metrics improve for our Hough filter. The fake rate of both methods performs similarly. Additionally, a brief examination of the resolutions leads to the result, that for events, that have the correct charge and are fitted by both methods, the baseline algorithm performs slightly better. However, for all fitted events, independent of the correct charge, and events fitted by both methods, independent of the correct charge, the Hough filter performs better.

For outer curlers, we select the Distance Check with 2 cm in addition to the Hough transformation with two iterations for the 2D Binary Search. Comparing the efficiencies of the selected Hough filter with the standard CAT Finder, we can see a sharp increase in the track charge efficiency of 36.25 percentage points, to 91.35 %. Compared to the baseline algorithm, we observe an increase of 1.18 percentage points in the track charge efficiency. The fake rate and the wrong charge rate perform similarly, however, the clone rate of the Hough filter, 1.68 %, performs 2.68 percentage points better than the clone rate of the baseline algorithm. Lastly, we briefly examined the resolutions for the Hough filter and the baseline algorithm. For events, that have the correct charge and are fitted by both methods, the Hough filter has a relative p_{T} resolution of 0.0075, an absolute p_{z} resolution of 0.008 GeV and an absolute θ resolution 1.456 °. For these events the baseline algorithm has a has a relative p_{T} resolution of 0.0048, an absolute p_{z} resolution of 0.0038 GeV and an absolute θ resolution $0.7252\,^\circ$. In addition, for all fitted events, independent of the correct charge, and events fitted by both methods, independent of the correct charge, the baseline performs slightly better as well.

The objective of this thesis, improving the efficiencies of the CAT Finder through the application of a Hough filter for low p_T events, was reached for both inner curlers and outer curlers. Additionally, the other tracking metrics showed good performance as well.

The next immediate step is determining the exact p_T and θ range at which our filters are applied. However, these working points can change depending on the particle. Since other charged particles, besides muons and antimuons, can pass through the CDC, it is important to evaluate the performance of the applied Hough filters for each type of particle at different p_T and θ ranges to identify the optimal working points. Yet, due to time constraints, this is beyond the scope of this bachelor thesis. Another possible step is to include the measurement of the coordinates, for axial layers, with drift circles and replace the Hough transformation with the Legendre transformation. This change may lead to further performance increases.

A. Appendix

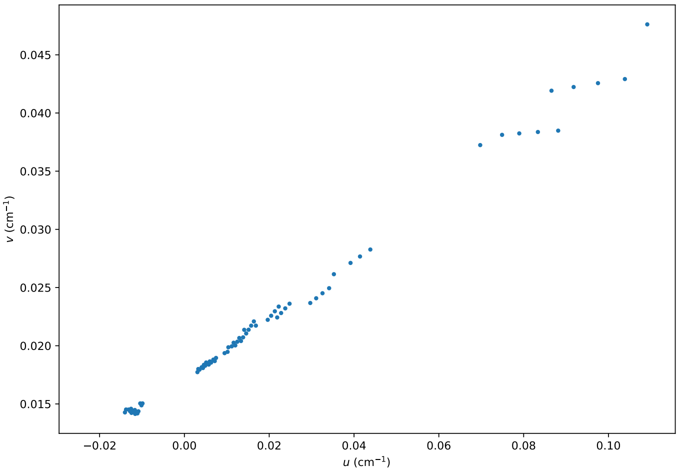


Figure A.1.: Filtered Conformal Transformation after the Hough Transformation and the 2D Binary Search.

Table A.1.: Efficiencies in percentage and absolute values for the Distance Check, without using both the HT and the 2D Binary Search, on outer curlers.

d(cm)	$\varepsilon_{\text{found}}(\%)$	$\varepsilon_{\text{track}}(\%)$	$\varepsilon_{\text{track,charge}}(\%)$	n_{found}	n_{tracks}	$n_{\text{correct charge}}$
1	99.0	91.3	90.8	990	913	908
2	99.0	91.3	90.8	990	913	908
3	99.1	91.6	91.2	991	916	912
4	99.3	91.2	90.7	993	912	907
5	99.3	90.6	90.2	993	906	902
6	99.3	90.0	89.6	993	900	896

Table A.2.: Efficiencies in percentage and absolute values for the Distance Check with one iteration for the 2D Binary Search on outer curlers.

d(cm)	$\varepsilon_{\text{found}}$(%)	$\varepsilon_{\text{track}}$(%)	$\varepsilon_{\text{track,charge}}$(%)	n_{found}	n_{tracks}	$n_{\text{correct charge}}$
1	99.0	91.7	91.1	990	917	911
2	99.1	91.9	91.5	991	919	915
3	99.1	91.7	91.4	991	917	914
4	99.3	91.5	91.1	993	915	911
5	99.3	90.9	90.6	993	909	906
6	99.3	90.1	89.8	993	901	898

Table A.3.: Efficiencies in percentage and absolute values for the Distance Check with three iterations for the 2D Binary Search on outer curlers.

d(cm)	$\varepsilon_{\text{found}}$(%)	$\varepsilon_{\text{track}}$(%)	$\varepsilon_{\text{track,charge}}$(%)	n_{found}	n_{tracks}	$n_{\text{correct charge}}$
1	99.0	90.8	90.1	990	908	901
2	99.1	92.1	91.9	991	921	919
3	99.1	91.3	90.7	991	913	907
4	99.3	90.3	89.8	993	903	898
5	99.3	89.4	88.9	993	894	889
6	99.3	88.9	88.3	993	889	883

Table A.4.: Efficiencies in percentage and absolute values for the Distance Check with four iterations for the 2D Binary Search on outer curlers.

d(cm)	$\varepsilon_{\text{found}}$(%)	$\varepsilon_{\text{track}}$(%)	$\varepsilon_{\text{track,charge}}$(%)	n_{found}	n_{tracks}	$n_{\text{correct charge}}$
1	98.7	90.2	89.7	987	902	897
2	99.1	91.6	91.2	991	916	912
3	99.0	90.9	90.2	990	909	902
4	99.1	90.7	89.6	991	907	896
5	99.1	89.7	88.6	991	897	886
6	99.1	88.9	88.1	991	889	881

Table A.5.: Efficiencies in percentage and absolute values for the Distance Check with five iterations for the 2D Binary Search on outer curlers.

d(cm)	$\varepsilon_{\text{found}}(\%)$	$\varepsilon_{\text{track}}(\%)$	$\varepsilon_{\text{track,charge}}(\%)$	n_{found}	n_{tracks}	$n_{\text{correct charge}}$
1	98.4	86.7	85.8	984	867	858
2	98.7	88.0	87.5	987	880	875
3	98.7	87.4	86.8	987	874	868
4	98.7	87.0	86.3	987	870	863
5	98.7	86.1	85.5	987	861	855
6	98.7	85.3	84.6	987	853	846

Table A.6.: Efficiencies in percentage and absolute values for the Distance Check with six iterations for the 2D Binary Search on outer curlers.

d(cm)	$\varepsilon_{\text{found}}(\%)$	$\varepsilon_{\text{track}}(\%)$	$\varepsilon_{\text{track,charge}}(\%)$	n_{found}	n_{tracks}	$n_{\text{correct charge}}$
1	98.4	86.8	85.9	984	868	859
2	98.7	88.2	87.5	987	882	875
3	97.5	84.1	83.3	975	841	833
4	97.7	84.8	83.7	977	848	837
5	97.8	85.1	83.9	978	851	839
6	97.8	84.8	83.5	978	848	835

Table A.7.: Efficiencies in percentage and absolute values for the Distance Check with eight iterations for the 2D Binary Search on outer curlers.

d(cm)	$\varepsilon_{\text{found}}(\%)$	$\varepsilon_{\text{track}}(\%)$	$\varepsilon_{\text{track,charge}}(\%)$	n_{found}	n_{tracks}	$n_{\text{correct charge}}$
1	93.7	78.0	75.5	937	780	755
2	94.9	79.2	76.3	949	792	763
3	95.8	78.9	76.0	958	789	760
4	97.0	80.8	78.1	970	808	781
5	97.1	80.3	77.8	971	803	778
6	97.2	80.1	77.4	972	801	774

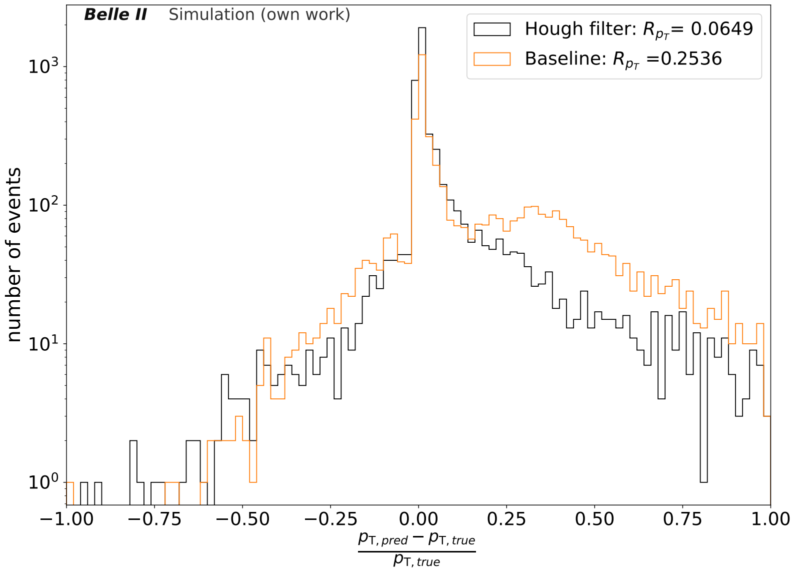


Figure A.2.: Comparison of the relative p_T resolution between the Hough filter and the Baseline algorithm for inner curlers. Only events, that are fitted by both methods are used in the plot.

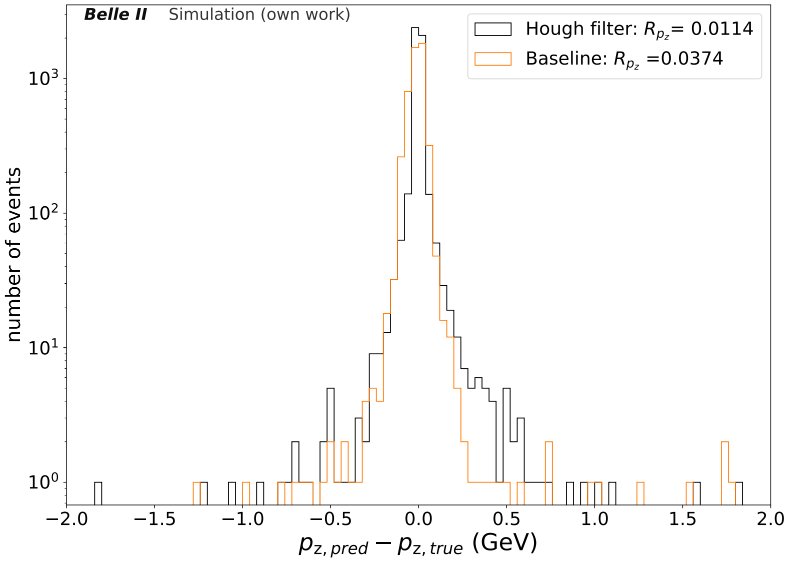


Figure A.3.: Comparison of the absolute p_z resolution between the Hough filter and the Baseline algorithm for inner curlers. Only events, that are fitted by both methods are used in the plot. Histograms are clipped at ± 2 GeV.

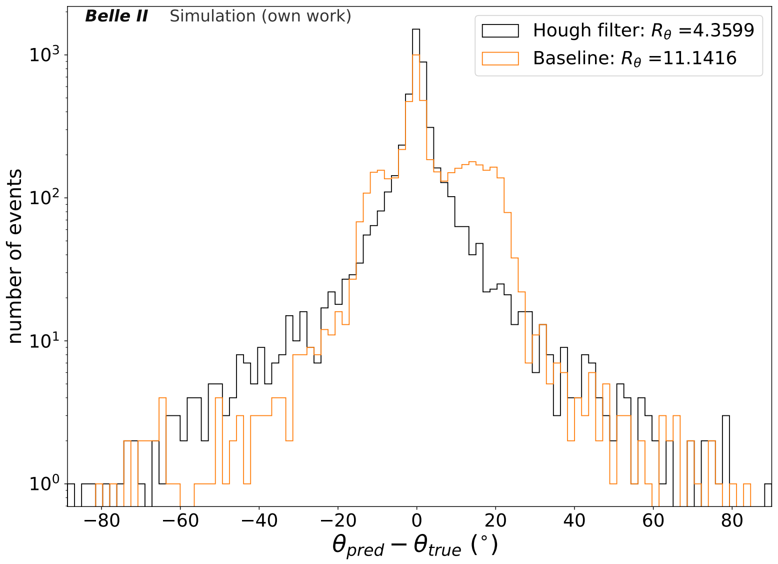


Figure A.4.: Comparison of the absolute θ resolution between the Hough filter and the Baseline algorithm for inner curlers. Only events, that are fitted by both methods are used in the plot.

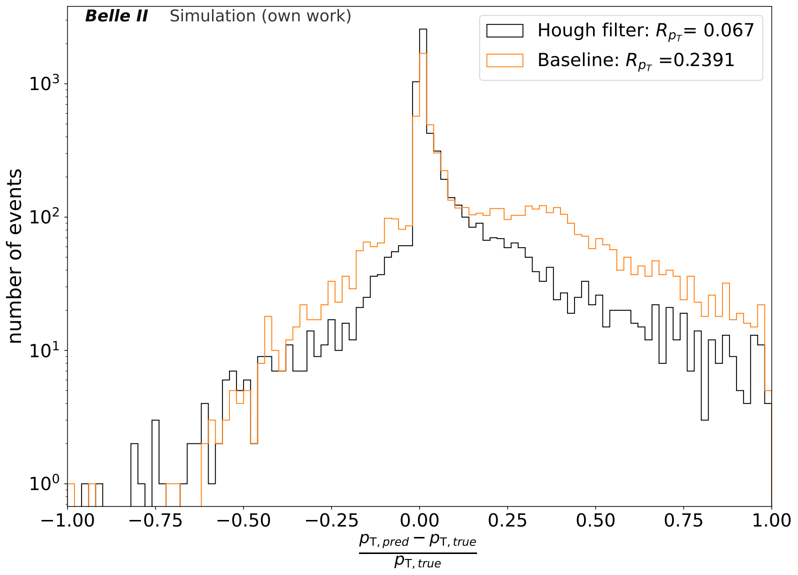


Figure A.5.: Comparison of the relative p_T resolution between the Hough filter and the Baseline algorithm for inner curlers. All fitted events for either method are used.

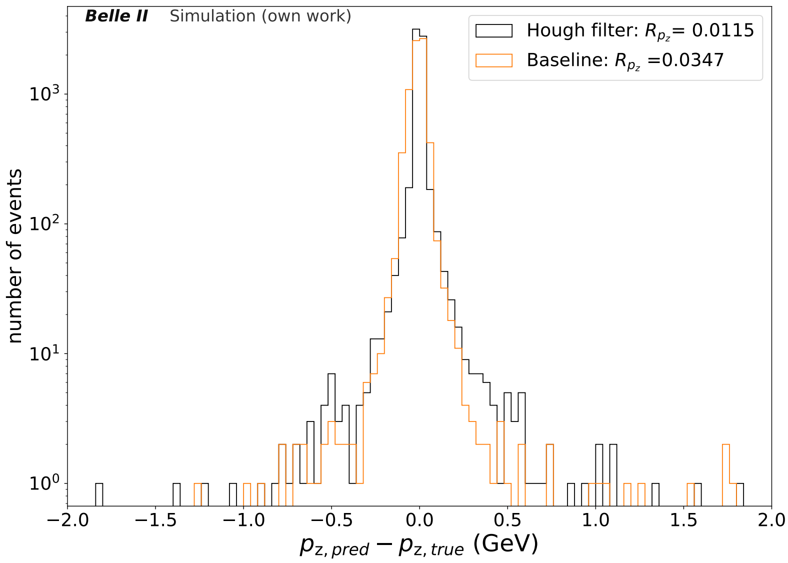


Figure A.6.: Comparison of the absolute p_z resolution between the Hough filter and the Baseline algorithm for inner curlers. All fitted events for either method are used. Histograms are clipped at ± 2 GeV.

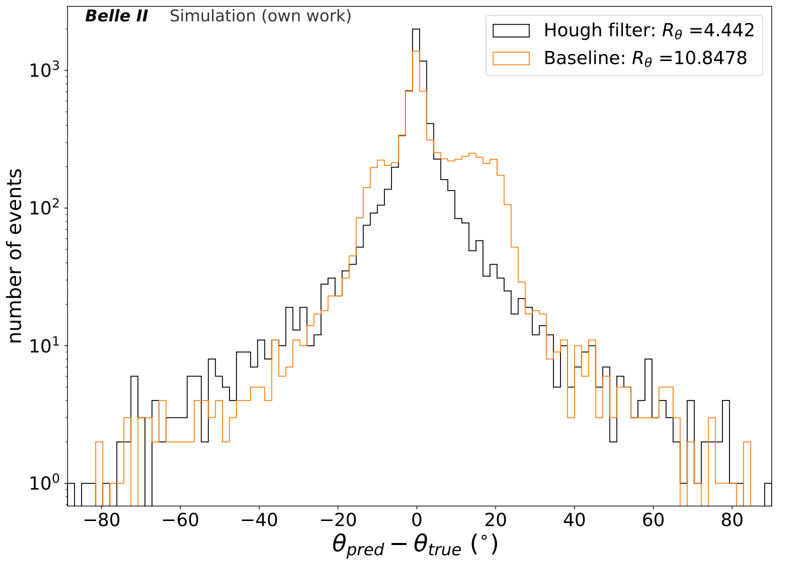


Figure A.7.: Comparison of the absolute θ resolution between the Hough filter and the Baseline algorithm for inner curlers. All fitted events for either method are used.

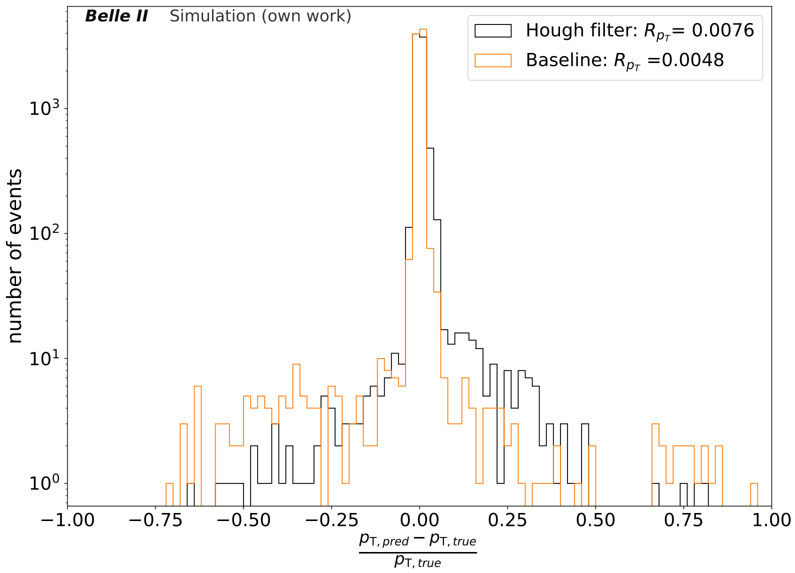


Figure A.8.: Comparison of the relative p_T resolution between the Hough filter and the Baseline algorithm for outer curlers. Only events, that are fitted by both methods are used in the plot.

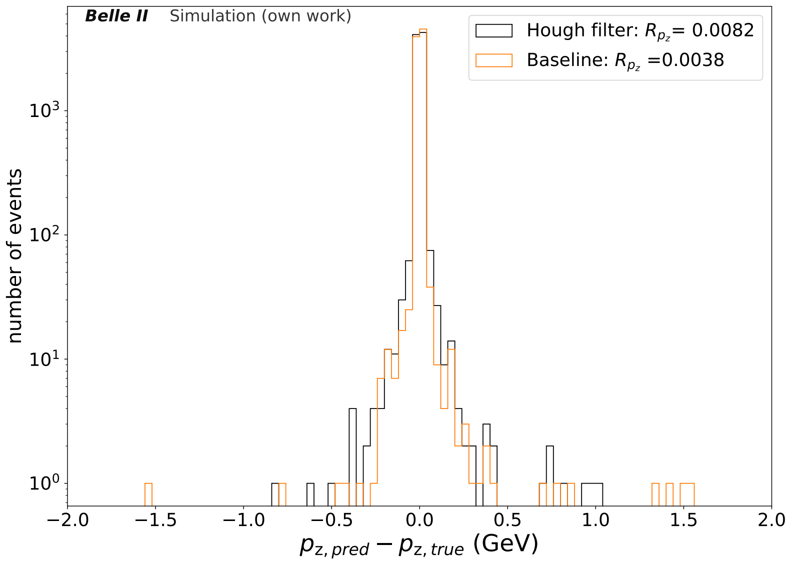


Figure A.9.: Comparison of the absolute p_z resolution between the Hough filter and the Baseline algorithm for outer curlers. Only events, that are fitted by both methods are used in the plot. Histograms are clipped at ± 2 GeV.

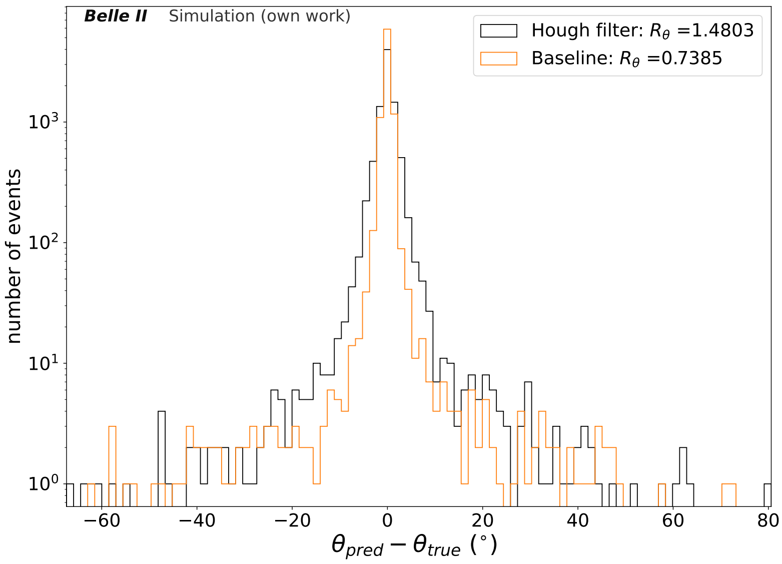


Figure A.10.: Comparison of the absolute θ resolution between the Hough filter and the Baseline algorithm for outer curlers. Only events, that are fitted by both methods are used in the plot.

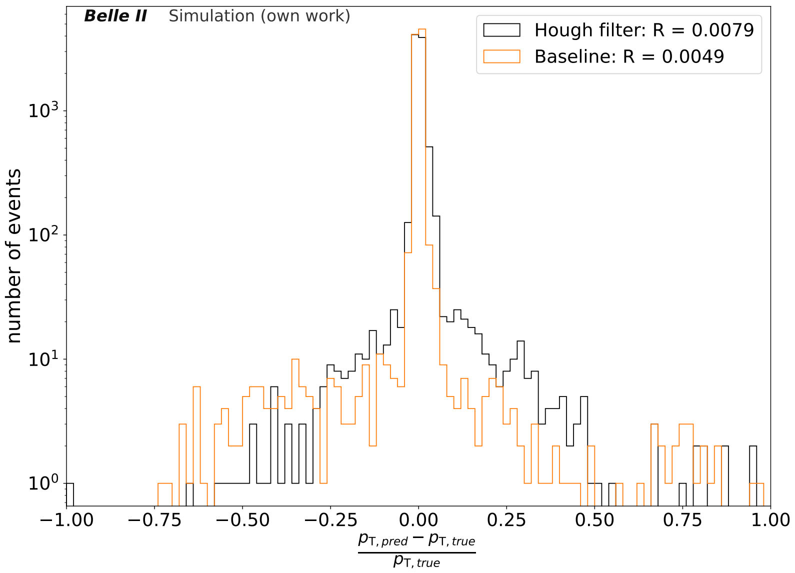


Figure A.11.: Comparison of the relative p_T resolution between the Hough filter and the Baseline algorithm for outer curlers. All fitted events for either method are used.

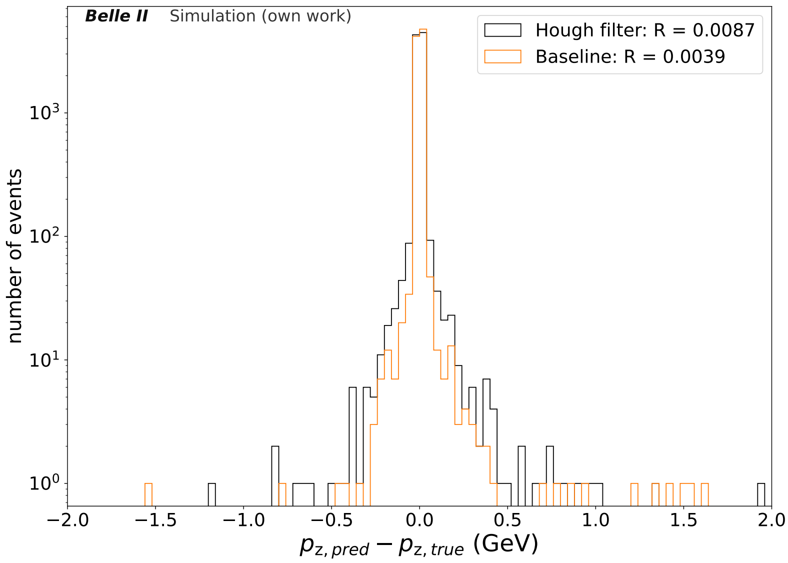


Figure A.12.: Comparison of the absolute p_z resolution between the Hough filter and the Baseline algorithm for outer curlers. All fitted events for either method are used. Histograms are clipped at ± 2 GeV.

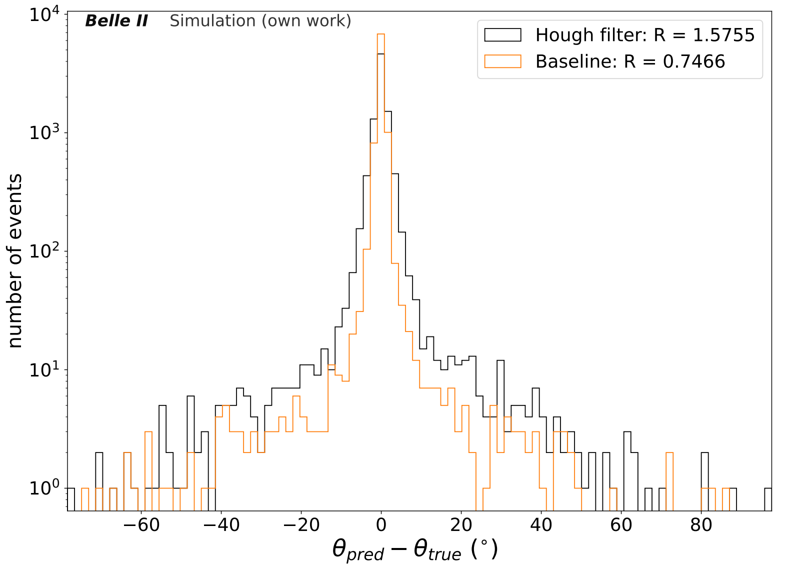


Figure A.13.: Comparison of the absolute θ resolution between the Hough filter and the Baseline algorithm for outer curlers. All fitted events for either method are used.

Bibliography

[1] L. Reuter, P. Dorwarth, S. Stefkova, and T. Ferber, "Graph Neural Network based Track Finding in the Central Drift Chamber at Belle II." https://indico.jlab.org/event/459/contributions/11761/. Last accessed: 22. September 2024.

[2] **Belle II Framework Software Group**, 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, (Nov., 2018) 1–12.

[3] Belle II collaboration, "Belle II Software Documentation." https://software.belle2.org/development/sphinx/index.html. Last accessed: 23. September 2024.

[4] **The Belle II Collaboration**, "Belle II Analysis Software Framework (basf2) ," *Zenodo* (2022) .

[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.

[6] Belle II, "Detector." https://www.belle2.org/research/detector/. Last accessed: 1. August 2024.

[7] N. L. Braun, *Combinatorial Kalman Filter and High Level Trigger Reconstruction for the Belle II Experiment*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2019.

[8] T. Abe *et al.* (Belle II Collaboration), "Belle II Technical Design Report," 2010. https://arxiv.org/abs/1011.0352.

[9] E. Kou *et al.* (Belle II Collaboration), "The Belle II Physics Book," *Progress of Theoretical and Experimental Physics* **2019** no. 12, (Dec., 2019) 123C01.

[10] D. Matvienko, "The Belle II experiment: status and physics program," *EPJ Web Conf.* **191** (2018) .

[11] V. Bertacchi *et al.* (Belle II Collaboration), "Track finding at Belle II," *Computer Physics Communications* **259** (Feb., 2021) 107610.

[12] V. Trusov, *Development of Pattern Recognition Algorithms for the Central Drift Chamber of the Belle II Detector*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2016.

[13] J. Rauch and T. Schlüter, "Genfit — a generic track-fitting toolkit," *Journal of Physics: Conference Series* **608** (May, 2015) 1–8.

[14] L. Reuter and F. Ohlheiser, "Hough Filter for CAT Finder." `https://gitlab.desy.de/kit-etp/hough-filter-for-cat-finder`.

[15] R. Frühwirth and A. Strandlie, *Track Finding*, pp. 81–102. Springer International Publishing, Cham, 2021. `https://doi.org/10.1007/978-3-030-65771-0_5`.