Track Reconstruction at the First Level Trigger of the Belle II Experiment

Sara Pohl

München 2017

Track Reconstruction at the First Level Trigger of the Belle II Experiment

Sara Pohl

Dissertation
der Fakultät für Physik
der Ludwig-Maximilians-Universität
München

vorgelegt von
Sara Pohl (geb. Neuhaus)
aus Starnberg

München, den 18. Dezember 2017

Erstgutachter:	Prof. Dr. Christian Kiesling
Zweitgutachter:	Prof. Dr. Thomas Kuhr
Tag der mündlichen Prüfung:	11. April 2018

Zusammenfassung

Das Belle II Experiment ist ein Upgrade des Belle Experiments, das wesentlich zur Bestätigung der Kobayashi-Maskawa Theorie über den Ursprung der $\mathcal{CP}$-Verletzung beigetragen hat. Der neue e^+e^- Beschleuniger SuperKEKB soll eine instantane Luminosität von $8 \times 10^{35}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ erreichen und damit den Weltrekord seines Vorgängers KEKB um einen Faktor 40 übertreffen. Belle II wird sowohl Präzisionsmessungen an B-Mesonen durchführen als auch Suchen nach seltenen Zerfällen, wie beispielsweise Lepton-Flavor verletzende τ-Zerfälle.

Infolge der erhöhten Luminosität wird auch eine höhere Untergrundrate erwartet, die bereits auf der ersten Triggerstufe reduziert werden muss. Eine dominante Untergrundquelle sind Teilchen, die nicht von e^+e^- Kollisionen stammen, sondern von Stößen innerhalb eines Strahls oder mit Restgasmolekülen in der Strahlröhre. Die daraus entstehenden Spuren können durch ihren Entstehungspunkt identifiziert werden, der gegenüber dem Kollisionspunkt longitudinal verschoben ist.

Im Fokus der vorliegenden Arbeit steht die 3D Rekonstruktion von Spuren in der zentralen Driftkammer des Belle II Detektors auf der ersten Triggerstufe. Die Driftkammer enthält Drähte mit unterschiedlicher Ausrichtung: *Axial*-Drähte sind parallel zur z-Achse gespannt, wohingegen *Stereo*-Drähte gegen die z-Achse verdreht sind. Spuren, die von der z-Achse, aber nicht notwendigerweise vom Kollisionspunkt stammen, werden durch vier Parameter beschrieben: Den Azimuth- und Polarwinkel, die Spurkrümmung und die longitudinale Koordinate des Entstehungspunktes („z-Vertex“). Die Spurkrümmung entsteht durch ein konstantes Magnetfeld von 1.5 T entlang der z-Achse und ist umgekehrt proportional zum Transversalimpuls des Teilchens.

Die Spurrekonstruktion im Trigger besteht aus zwei Stufen: ein Spurfindungsalgorithmus in der transversalen x-y-Ebene, gefolgt von einer 3D Rekonstruktion einzelner Spuren. Der Spurfinder bestimmt die Anzahl an Spuren sowie ihren jeweiligen Azimuthwinkel und ihre Krümmung mittels einer Houghtransformation. Hits von Axial-Drähten entsprechen Punkten in der transversalen Ebene und werden zu Kurven in einer Hough-Ebene transformiert. Spuren werden als Kreuzungspunkte mehrerer Kurven gefunden. Die Spurparametrisierung für diese Transformation beschreibt Halbkreise in der transversalen Ebene, die vom Ursprung nach außen laufen. Mit dieser Definition entsprechen die Koordinaten des Kreuzungspunktes in der Hough-Ebene dem Azimuthwinkel und einer vorzeichenbehafteten Krümmung, über die man den Transversalimpuls und die Ladung der Spur erhält.

Die 3D Rekonstruktion bestimmt den Polarwinkel und den z-Vertex jeder Spur. Da

die Spuren bereits durch den Spurfinder voneinander getrennt wurden, kann jede Spur separat rekonstruiert werden. Durch die Kombination von Hits der Stereo-Drähte mit der transversalen Spur können longitudinale Hit-Koordinaten bestimmt werden. Um die Ortsauflösung zu erhöhen, werden auch die Driftzeiten der Hits berücksichtigt, die proportional zum Abstand zwischen der Spur und dem Draht sind. Anstelle einer analytischen Rekonstruktion werden mehrlagige Perzeptrons (MLPs) trainiert, ein bestimmter Typ von neuronalen Netzen. MLPs können nichtlineare Funktionen aus Beispieldaten lernen und lassen sich parallelisiert mit deterministischer Laufzeit ausführen. Die Netze erhalten die Drahtkoordinaten relativ zur transversalen Spur sowie die Driftzeiten der Hits als Eingabewerte und schätzen den Polarwinkel und den z-Vertex. Für die Spuren einzelner Myonen wird eine durchschnittliche z-Vertex-Auflösung von $(2.910 \pm 0.008)\,\mathrm{cm}$ erreicht, wodurch der Großteil der Untergrundspuren unterdrückt werden kann.

Im Anschluss an die Optimierung der Houghtransformation und der neuronalen Netze wird in dieser Arbeit die Triggereffizienz ausgewählter B und τ Ereignisse untersucht. Unterschiedliche Bedingungen des Spurtriggers werden verglichen, darunter ein z-Vertex-Veto und ein Veto gegen Bhabha-Streuprozesse, die im Belle II Experiment einen zweiten dominanten Untergrund darstellen. Durch die Kombination dieser Vetos können selbst Ereignisse mit nur ein oder zwei Spuren getriggert werden. Für den Lepton-Flavor verletzenden Zerfall $\tau \to \mu\gamma$ erreicht ein reiner Spurtrigger eine Triggereffizienz von 75 % bis 77 % (sechs mal höher als ohne die 3D Spurrekonstruktion).

Abstract

The Belle II experiment is an upgrade of the Belle experiment, which was instrumental in confirming the Kobayashi-Maskawa theory for the origin of $\mathcal{CP}$ violation. The upgraded e^+e^- collider SuperKEKB is designed to achieve an instantaneous luminosity of $8 \times 10^{35}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$, which is 40 times higher than the world record set by its predecessor KEKB. Belle II will perform precision measurements in the B meson system as well as searches for rare decays, such as lepton flavor violating τ decays.

As a consequence of the increased luminosity, the first level trigger has to cope with an increased background rate. A dominant background source are particles that do not originate from e^+e^- collisions, but from intra-beam interactions or scattering on residual gas in the beampipe. The corresponding tracks are characterized by their production vertex, which is longitudinally displaced with respect to the interaction point.

The focus of this thesis is the 3D reconstruction of tracks in the central drift chamber of the Belle II detector at the first trigger level. The drift chamber contains wires of two different orientations: *axial* wires are oriented along the z-axis, while *stereo* wires are skewed with respect to the z-axis. Assuming that tracks come from the z-axis, but not necessarily the interaction point, a track is parametrized by four parameters: the azimuth and polar angles at the track vertex, the track curvature and the longitudinal coordinate of the production vertex ("z-vertex"). The track curvature is caused by a constant magnetic field of 1.5 T that is oriented along the z-axis, and is inverse proportional to the transverse momentum of the particle.

The track reconstruction in the trigger consists of two steps: a track finding step in the transverse x-y-plane and a 3D reconstruction step on individual tracks. The purpose of the track finding is to determine the number of tracks and the azimuth angle and curvature of each, using a Hough transformation. Hits on axial wires correspond to points in the transverse plane and are transformed into curves in a Hough plane, where tracks are found as the crossing points of several curves. The track parametrization for the transformation is chosen such that it represents half circles in the transverse plane, going outward from the origin. With this definition, the crossing point coordinates in the Hough plane give the azimuth angle and the signed curvature, which is related to the transverse momentum and the charge of the track.

The purpose of the 3D reconstruction is to determine the polar angle and the z-vertex of each track. As the tracks have already been separated in the track finding step, each track is reconstructed independently. By combining the hits from stereo wires with the transverse track, longitudinal hit coordinates can be obtained. To increase

the spatial resolution, the drift times of hits are included, which are proportional to the distance between the track and the wire. Instead of an analytical reconstruction, neural networks of the Multi Layer Perceptron (MLP) type are trained. MLPs are capable of learning nonlinear functions from data samples as well as of parallel execution with a deterministic runtime. The networks receive the wire coordinates relative to the transverse track and the drift times of hits as input and estimate the polar angle and the z-vertex. An average z-vertex resolution of $(2.910 \pm 0.008)\,\mathrm{cm}$ is achieved for single muon tracks, which allows to suppress most of the displaced background tracks.

After the optimization of both the Hough transformation and the neural networks, this thesis studies the trigger efficiency for selected B and τ events. Different track trigger conditions are compared, including a z-vertex veto and a veto on Bhabha scattering events, which form the second dominant background in the Belle II experiment. By combining these vetos, events with only one or two tracks can be triggered. For the lepton flavor violating decay channel $\tau \to \mu\gamma$, a trigger efficiency of 75 % to 77 % can be obtained with a pure track trigger, which is six times higher than without a 3D track reconstruction.

Contents

1. Introduction

> In the beginning there was nothing, which exploded.
>
> *(Terry Pratchett)*

Most models of the Big Bang assume that the universe started with an equal amount of matter and antimatter. The universe we observe today consists almost exclusively of matter, so the question is how the imbalance between matter and antimatter arose. The Sakharov conditions for baryogenesis (the creation of an excess of baryons over antibaryons) in the early universe require the violation of baryon number conservation, interactions out of thermal equilibrium and the violation of charge conjugation ($\mathcal{C}$) symmetry and also of the combined charge parity ($\mathcal{CP}$) symmetry [1]. The last of these conditions ($\mathcal{C}$ violation and $\mathcal{CP}$ violation) is necessary because otherwise each process which produces an excess of baryons would be balanced by a $\mathcal{C}/\mathcal{CP}$ symmetric process which produces an equal excess of antibaryons.

$\mathcal{CP}$ violation was first observed in the neutral K meson system by Cronin and Fitch in 1964 [2]. To explain the phenomenon, Kobayashi and Maskawa proposed a third family of quarks in 1973 [3], when even the charm quark (the last particle of the second family) was not yet discovered. Precise measurements became possible with the experiments Belle and BaBar at the so-called B factories KEKB and PEP-II, which produced pairs of B mesons and their antiparticles $\overline{\mathrm{B}}$ in large amounts to study $\mathcal{CP}$ violation in the B meson system. The results of the B factories confirmed the theory for the origin of $\mathcal{CP}$ violation by Kobayashi and Maskawa, who received the Nobel Prize for their theory in 2008.

However, the observed level of $\mathcal{CP}$ violation is not strong enough to explain the excess of matter in the universe, so now the focus shifts from testing the standard model theory to a search for new phenomena that hint at physics beyond the standard model. Therefore, the Belle experiment is being upgraded to continue the studies in the B meson system. The new experiment Belle II will increase the amount of collected $\mathrm{B}\overline{\mathrm{B}}$ samples by a factor of 50. It aims to search for small inconsistencies in the branching ratios of various decays and in the Kobayashi-Maskawa theory in general. In addition, the Belle II experiment will offer unique possibilities in the search for rare processes that are forbidden or strongly suppressed in the standard model.

The Belle experiment was operated by the Japanese High Energy Accelerator Research Organisation (Japanese: *Kō Enerugī Kasokuki kenkyū kikou*, short KEK)

and powered by the asymmetric electron-positron collider KEKB, which reached the world record in instantaneous luminosity of $2.11 \times 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$. After Belle stopped taking data in 2010, an upgrade was started both for the collider and the detector. The new collider SuperKEKB is designed to achieve a peak luminosity 40 times higher than its predecessor. The Belle II detector is equipped with newly built particle identification and tracking systems, including a new pixel vertex detector to provide better resolution for vertex reconstruction.

To cope with the increasing collision rate, the data acquisition and trigger systems are also being redesigned. The dominant source of background in Belle was beam induced, that is particles were originating from interactions with residual gas in the beampipe and from intra-beam interactions. This type of background is characterized by tracks that do not come from the interaction point, but from somewhere along the beamline. In Belle II, the background rate will increase due to the higher luminosity. To suppress background tracks, a three dimensional track reconstruction which includes an estimate for the vertex position is required.

The basic trigger scheme in Belle II follows the one from Belle. The trigger signal is provided by a pipelined first level trigger running on programmable hardware, followed by a high level trigger running on the Belle II computing farm to filter the events that are actually written to disk. The focus of this thesis is the track trigger, which is part of the first level trigger. The goal is to enable an efficient low multiplicity trigger, which can detect events with only one or two tracks and still reject enough background to keep the maximal trigger rate of 30 kHz.

To achieve this goal, two components of the track trigger are upgraded. The first is the track finding in a two dimensional projection of the detector, where single hits are combined to tracks. In the track trigger of Belle, track finding was achieved by a coincidence logic within several segments of the detector. In Belle II, this logic is replaced by a Hough transformation, which provides a good estimate not only for the number of tracks, but also for the direction and the curvature.

Following the track finding, a full three dimensional track reconstruction is performed on each single track, in order to reject background tracks which do not originate at the collision point. For this purpose, a method based on neural networks was proposed in [4], which achieved the required precision within selected sectors of the detector. In contrast, the Belle trigger had only a rudimentary logic for vertex estimation, based on the coincidence of hits in a given detector region [5].

Within this thesis, both the Hough transformation and the neural network method are refined and then evaluated on simulated events within the context of the full first level trigger. The thesis is structured into three parts. The first part gives an introduction to the theory of $\mathcal{CP}$ violation and the physics goals of the Belle II experiment (chapter 2), the SuperKEKB collider and the different components of the Belle II detector (chapter 3). Following the introductory chapters, the trigger system is described, starting with a general overview (chapter 4) and then focusing on the developed algorithms, including optimization studies and separate evaluations

of each component: chapter 5 describes the track finding, chapter 6 gives a general introduction to 3D track reconstruction in the drift chamber and chapter 7 focuses on the neural network based method. Finally, in chapter 8 the trigger system is evaluated on selected physics channels, starting with a pure track trigger and then combining the track trigger results with clusters from the calorimeter trigger. Different trigger conditions are compared to estimate the efficiency of the first level track trigger system.

2. Theory of CP Violation

Although $\mathcal{CP}$ violation is not the only phenomenon that can be studied at the Belle II experiment, it is the main motivation and dictates the design of both the collider and the detector. The following chapter gives an introduction to the mechanism that allows the breaking of $\mathcal{CP}$ symmetry and to the different types of $\mathcal{CP}$ violation that can be observed in the B meson system. Since the theory by Kobayashi and Maskawa predicted the existence of three families of quarks, a historical overview over the standard model is given first, with focus on the quark sector.

2.1. The standard model of particle physics

Figure 2.1 shows the elementary particles of the standard model as it is established today, with three families of quarks and leptons, four gauge bosons and the Higgs boson. Atoms consist of fermions from the first family: protons and neutrons, which contain up and down quarks, and electrons. Neutrinos were suggested by Pauli in 1930 to explain the energy spectrum of electrons in β decays, but they were not observed until 1956 [6]. The particles from the second and third family and the heavy gauge bosons were gradually discovered in experiments with ever increasing energy, from cosmic ray experiments in the 1930s up to the discovery of the Higgs boson at the LHC in 2012. The theoretical model evolved in parallel, sometimes explaining the experimental observations, sometimes motivating experiments with predictions. A detailed overview over the history of the standard model is given in [7].

2.1.1. Light quarks

Cosmic ray experiments in the first half of the 20th century revealed muons and various hadrons (formed, as we now know, from up, down and strange quarks). Measurements of their respective mass and charge as well as their decay channels suggested that these particles could be grouped into “strong isospin” singlets, doublets, triplets and quadruplets. The term isospin is short for isotopic spin and was introduced to explain the symmetries between protons and neutrons, which are nearly identical in all aspects except for their charge. The charged and neutral pions, which also have almost identical mass, were identified as a strong isospin triplet.

In addition to strong isospin, the quantum number “strangeness” was introduced to describe particles whose weak decay rates were much smaller than expected on the

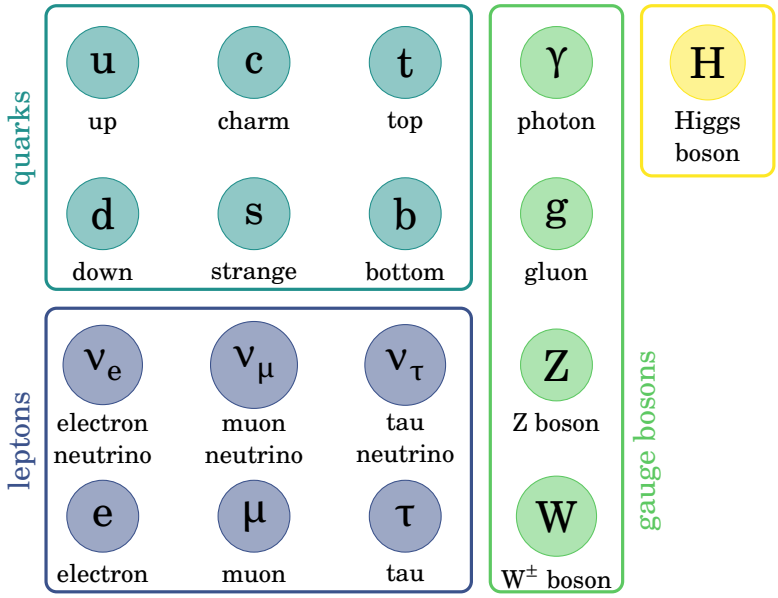

Figure 2.1.: The elementary particles in the standard model encompass three families of quarks and leptons, the gauge bosons and the Higgs boson.

basis of their masses. Combining the quantum numbers of charge, strong isospin and strangeness, the various hadrons could be grouped into octets and decuplets. The concept that hadrons are built from three types of quarks was introduced independently by Gell-Mann and Zweig in 1964 to formalize the underlying $SU(3)$ symmetry group. In the quark model, hadrons are composite states of $|qqq\rangle$ (baryons) or $|q\overline{q}\rangle$ (mesons). Strong isospin $+1/2$ $(-1/2)$ is associated with up (down) quarks and strangeness -1 is associated with strange quarks. At that time, quarks were interpreted as a mathematical concept rather than real particles, until deep inelastic scattering experiments of electrons on protons in 1968 reveiled that there are indeed point-like constituents inside of the proton [8].

The reason for the long lifetime of (unexcited) strange particles is that they can only decay via the weak interaction, which does not conserve quark flavor. More precisely, the weak interaction acts on "weak isospin" doublets, where weak isospin $+1/2$ $(-1/2)$ is associated with the upper (lower) quark row in fig. 2.1. Transitions between families are suppressed, as will become clear in the following section.

2.1.2. Cabibbo angle

The first theory of the weak interaction was formulated by Fermi for the β decay process $n \rightarrow pe^-\overline{\nu}_e$ [9]. He assumed a point interaction of four fermions with a coupling constant G_F, which gives a good description of β decays, but leads to inconsistent predictions for the cross section at high energies. Therefore, a massive vector boson (W)

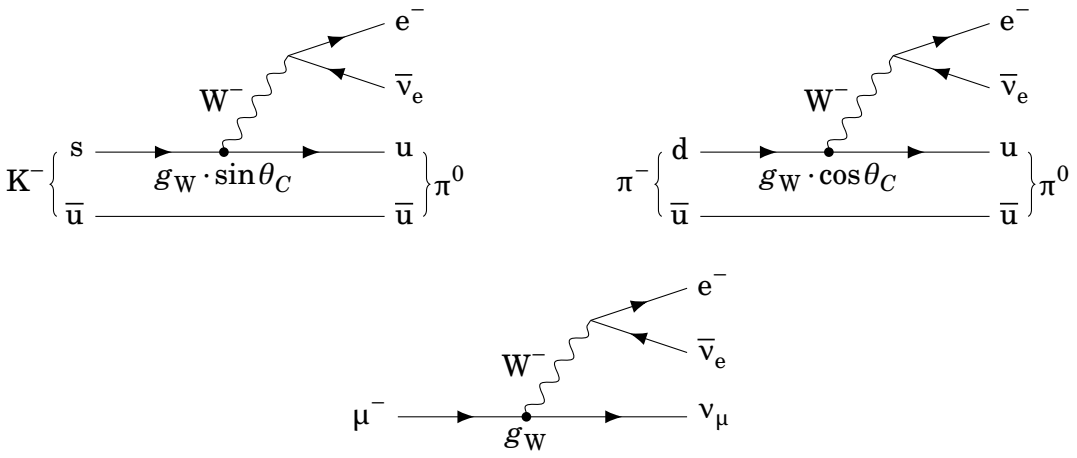

Figure 2.2.: Different weak interaction decay processes with an electron in the final state. The coupling for hadronic decays contains the Cabibbo angle θ_C.

was introduced later, which mediates the weak interaction and couples to quarks with a coupling constant g_W. For energies $\ll m_W$ (the mass of the W boson), the interaction can be approximated as a four-particle interaction with an effective coupling constant $g_W^2/m_W^2 \equiv G_F$, which corresponds to Fermi's original theory. Initially, a universal coupling constant was assumed for all weak interaction processes. However, it was found that the decay rates of strange particles (for example K^+, quark content $u\bar{s}$) to leptonic final states were much smaller than the rates for similar decays of non-strange particles (π^+, quark content $u\bar{d}$). Figure 2.2 shows the Feynman diagrams for two such processes and also for a muon decay with a similar final state.

As an explanation, Cabibbo proposed that for hadronic particles the weak current J_μ should be written as a linear combination of a strangeness conserving part $J_\mu^{(0)}$ and a strangeness changing part $J_\mu^{(1)}$ [10]. To preserve the total weak current, he introduced a mixing angle θ, which was later named Cabibbo angle θ_C, and wrote the weak current as

$$J_\mu = \cos\theta \cdot J_\mu^{(0)} + \sin\theta \cdot J_\mu^{(1)} .$$

The coupling constant g_W is then replaced by $g_W \cdot \sin\theta$ for decays involving strange particles and $g_W \cdot \cos\theta$ for decays involving non-strange hadrons. Cabibbo compared existing measurements for different decay channels and found consistent values for the mixing angle θ. In addition, his hypothesis explained the observed discrepancy between the decay rates of nuclear β decays compared to the muon.

In the quark model, the Cabibbo angle is interpreted as a mixing between quark flavors, that is the weak interaction couples to the linear combination

$$|d'\rangle = \cos\theta_C \cdot |d\rangle + \sin\theta_C \cdot |s\rangle .$$

2.1.3. Charm quark

The fourth quark (charm) was proposed theoretically to explain the suppression of flavor changing neutral currents, such as the decay $K^0 \to \mu^+\mu^-$. The branching ratio for this decay is 6.84×10^{-9} [11], so clearly there is no first order contribution to this process. Even without first order flavor changing neutral currents, the decay $K^0 \to \mu^+\mu^-$ should still be possible through a second order loop process, as shown on the left in fig. 2.3. However, the expected branching ratio for this diagram is several orders of magnitude larger than the actual branching ratio.

Glashow, Iliopoulos and Maiani observed in 1970 that such processes could be suppressed if there was a fourth quark with the same charge as the u quark [12]. They extended Cabibbo's mixing angle to a rotation matrix, which mixes down and strange quarks:

$$\begin{pmatrix} d' \\ s' \end{pmatrix} = \begin{pmatrix} \cos\theta_C & \sin\theta_C \\ -\sin\theta_C & \cos\theta_C \end{pmatrix} \begin{pmatrix} d \\ s \end{pmatrix}. \tag{2.1}$$

The weak interaction then couples the u quark to d' and the new charm quark to s'. In processes with apparent flavor changing neutral currents, a second loop diagram has to be considered, which has a charm quark in the loop instead of an up quark (right diagram in fig. 2.3). Due to the minus sign in eq. (2.1), the two contributions cancel almost exactly, except for a small term due to the mass difference between the up and the charm quark. In addition, the unitarity of eq. (2.1) explains the absence of first order flavor changing neutral currents, since the rotated states d' and s' are orthogonal to each other.

The first meson containing charm quarks to be discovered was the J/ψ, which is a bound $c\bar{c}$ state. In 1974 it was observed as a resonance at $3.1\,\mathrm{GeV}/c^2$ in two independent experiments: in the invariant mass spectrum of e^+e^- pairs produced in collisions of protons on a beryllium target at BNL [13], and in the interaction cross section of e^+e^- beams at SLAC [14]. Charmed mesons consisting of a c quark and a light quark followed shortly afterwards.

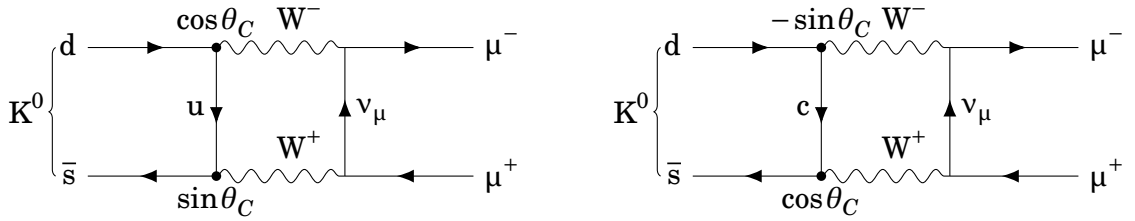

Figure 2.3.: The process $K^0 \to \mu^+\mu^-$ is suppressed by the cancellation between two diagrams: one with an up quark in the loop (left), the other with a charm quark (right).

2.1.4. Third family

The third lepton (τ) was discovered at SLAC shortly after the observation of the J/ψ. Perl and collaborators reported 64 events with e^+e^- in the initial state and an electron, a muon and missing energy in the final state [15]. Such a signature is explained by a pair of new leptons produced in the e^+e^- collision, with one lepton decaying to an electron and neutrinos, the other decaying to a muon and neutrinos.

For the quark sector, Kobayashi and Maskawa had suggested a third quark family already in 1973, before the charm quark and the τ were discovered, to explain the origin of $\mathcal{CP}$ violation [3]. With three families, the quark mixing matrix in eq. (2.1) is extended to a 3×3 matrix, which contains an irreducible complex phase. This phase is required to explain $\mathcal{CP}$ asymmetric processes, as will be shown in more detail in sections 2.3 and 2.4.

The $\Upsilon(1S)$, the first bound $b\bar{b}$ state analogous to the J/ψ, was found in 1977 at Fermilab as a resonance at $9.5\,\mathrm{GeV}/c^2$ [16]. The last quark (top) turned out to be significantly heavier, with a rest mass of about $175\,\mathrm{GeV}/c^2$. In consequence, it was not discovered until 1995 [17, 18], when the Tevatron at Fermilab provided $p\bar{p}$ collisions with enough energy to produce top quark pairs.

2.2. Discrete symmetries

Symmetries play a fundamental role in physics. On the one hand, symmetries can often be exploited to simplify calculations and models, which might not be solvable at all otherwise. On the other hand, the observation of symmetries (or their absence) in nature provides insight into the underlying physical laws. In the following, the discrete symmetries $\mathcal{P}$, $\mathcal{C}$ and $\mathcal{T}$ are formally defined and the experimental history of $\mathcal{CP}$ violation is given.

2.2.1. Parity inversion

The parity inversion $\mathcal{P}$ is defined as an inversion of all spatial coordinates. This is equivalent to a mirror reflection followed by a 180° rotation. After a parity inversion, the direction of motion of a particle (a vector) is reversed. However, the direction of the angular momentum and the spin (axial vectors) are unchanged, as illustrated in fig. 2.4.

The angular momentum part of a particle wave function $\psi(\vec{x})$ is an eigenstate of $\mathcal{P}$ with eigenvalue $+1$ (-1) if it is symmetric (antisymmetric). In addition, an intrinsic parity of ± 1 can be assigned to particles. For a multiparticle state, the parity is then given by the product of all intrinsic parities times $(-1)^L$, where L is the angular momentum of the system. For instance, the intrinsic parity of a pion is -1, so the parity of a two pion system is $(-1)^L$, while the parity of a three pion system is $(-1)^{L+1}$.

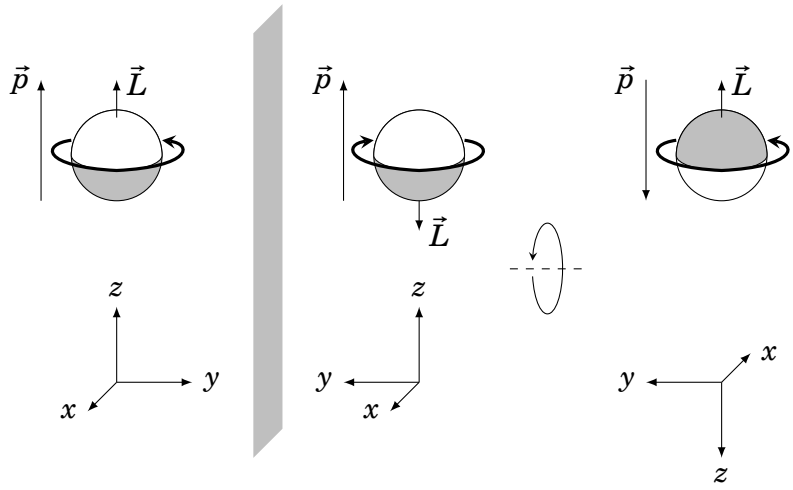

Figure 2.4.: A parity inversion is equivalent to a mirror reflection followed by a 180° rotation around the mirror axis. All spatial coordinates and the direction of motion $\vec{p}$ are reversed. The angular momentum $\vec{L}$ is unchanged.

While the electromagnetic and the strong interaction conserve parity, the weak interaction violates it maximally. Historically, the first hint of parity violation was given by the observation of two particles, then named θ^+ and τ^+ (not to be confused with the τ lepton), which had the same mass and lifetime, but decayed to states of different parity. In 1956 Lee and Yang suggested that the θ^+ and τ^+ might be the same particle, which is called K^+ today. They noted that parity conservation in the weak interaction had never been tested and proposed several experiments to do so [19].

In 1957 Wu and collaborators carried out one of the proposed experiments, which was based on the β decay of polarized nuclei [20]. Angular momentum conservation restricts the spins of the final state electron and neutrino to the direction of the spin of the decaying nucleus. A parity inversion would flip the momentum $\vec{p}$ of the decay products, but not their spin $\vec{\sigma}$, in other words the helicity $\frac{\vec{p}\cdot\vec{\sigma}}{|\vec{p}|\cdot|\vec{\sigma}|}$ is changed. Therefore, an asymmetry in the angular distribution of the electrons with respect to the polarization axis implies a violation of parity symmetry. Indeed a clear asymmetry was observed and could be correlated to the polarization of the decaying nuclei, showing a maximal violation of parity symmetry. The result was confirmed shortly afterwards by Garwin, Lederman and Weinrich, who measured the polarization of muons produced in pion decays [21]. They found that the emitted muons were strongly polarized along the direction of motion, indicating again that parity symmetry is maximally violated.

2.2.2. Charge conjugation

The charge conjugation $\mathcal{C}$ replaces all particles with their respective antiparticles and vice-versa. All charge-like quantum numbers are reversed, while the momentum,

angular momentum and spin are unchanged. A single particle state is an eigenstate of $\mathcal{C}$ if it is its own antiparticle. For example, the neutral pion is an eigenstate of $\mathcal{C}$ ($\mathcal{C}|\pi^0\rangle = |\pi^0\rangle$), while the charged pions are transformed into each other by a charge conjugation ($\mathcal{C}|\pi^\pm\rangle = |\pi^\mp\rangle$). A bound state of a fermion and the corresponding antifermion $|f\overline{f}\rangle$ is an eigenstate of $\mathcal{C}$ with eigenvalue $(-1)^{L+S}$, where L is the angular momentum of the system and S is the total spin, since the charge conjugation effectively exchanges the two particles. Other eigenstates of $\mathcal{C}$ can be formed from the superposition of particle and antiparticle states.

A symmetry of charge conjugation would imply that particles and antiparticles behave identically. In other words, the charge conjugated equivalent of a process should happen with the same rate. As mentioned in the introduction, this assumption is in contradiction to the observed imbalance between matter and antimatter in the universe. Unless the asymmetry was present already at the Big Bang, $\mathcal{C}$ symmetry has to be violated at some point in the history of the universe.

In fact, $\mathcal{C}$ symmetry is also violated maximally in the weak interaction. This conclusion could be obtained from the same experiments that showed $\mathcal{P}$ symmetry violation, based on the fact that the combined symmetry of $\mathcal{CPT}$ has to be conserved as a consequence of Lorentz invariance [22]. For example, the helicity of antimuons produced in pion decays is reversed compared to muons. Note, however, that $\mathcal{C}$ violation alone does not explain the matter-antimatter asymmetry. Further necessary conditions include the violation of baryon number conservation, interactions out of thermal equilibrium and the violation of the combined $\mathcal{CP}$ symmetry [1].

2.2.3. Time reversal

The time reversal operation $\mathcal{T}$ transforms a process into one that is happening backward in time, that is it replaces $t \to -t$. Position and charge of particles are unchanged, but the direction of motion and the angular momentum are reversed.

Macroscopically, the world is clearly not $\mathcal{T}$ symmetric, since the second law of thermodynamics states that entropy increases with time. However, microscopic processes are generally $\mathcal{T}$ symmetric. As an example for a $\mathcal{T}$ symmetric process in classical mechanics, consider the motion of a billiard ball bouncing off a wall. A backward recording of the motion follows the same equations of motion, so it would not be possible to decide if the recording is played forward or backward. If the ball hits another ball at rest, or if more than two balls are involved, we could distinguish the forward and backward motion with reasonable confidence. However, it is important to note that the backward motion of two colliding billiard balls still does not violate any laws of motion. It is simply very difficult to prepare the exact initial conditions that result in one ball being at rest after the collision. If more than two balls are involved, the difficulty to prepare the initial conditions increases, although the backward process is still physically allowed.

So in a stricter sense, $\mathcal{T}$ symmetry implies that if a system with a given initial

state evolves to a certain final state, another system prepared in the final state of the original system with reversed direction of motion will evolve to the original initial state. An example for a $\mathcal{T}$ symmetric process in particle physics are the reactions $K^- p \to \overline{K}^0 n$ and its reverse $\overline{K}^0 n \to K^- p$, which happen with the same rate.

2.2.4. Combined charge parity symmetry

$\mathcal{C}$ and $\mathcal{P}$ violation were included into the theory of weak interaction by writing the weak interaction current as a sum of vector and axial vector terms with equal coefficients. In consequence, the weak interaction couples only to left-handed fermions and right-handed antifermions, where left-(right-)handed means that the spin of a massless particle is antiparallel (parallel) to the direction of motion (for a massive particle the definition is more abstract, since the direction of motion depends on the reference frame). So $\mathcal{C}$ and $\mathcal{P}$ are maximally violated separately, but the theory is symmetric with respect to the combined transformation of parity inversion and charge conjugation $\mathcal{CP}$.

The violation of $\mathcal{CP}$ symmetry was first observed in the neutral K^0 meson system. The K^0 ($d\bar{s}$) can be distinguished from its antiparticle $\overline{K}^0$ ($\bar{d}s$) by their respective strangeness. However, a K^0 can make a transition to a $\overline{K}^0$ and vice-versa by weak interaction processes, as shown in fig. 2.5. This allows the two states to mix coherently, that is the eigenstates of the weak interaction K_1^0 and K_2^0 (states of definite mass and lifetime, also called *mass eigenstates*) are a superposition of the flavor eigenstates K^0 and $\overline{K}^0$. Assuming that the weak interaction conserves $\mathcal{CP}$, the mass eigenstates would also be eigenstates of the $\mathcal{CP}$ operator:

$$|K_1^0\rangle = \frac{1}{\sqrt{2}}\left(|K^0\rangle + |\overline{K}^0\rangle\right) \qquad \mathcal{CP}|K_1^0\rangle = +|K_1^0\rangle\,,$$
$$|K_2^0\rangle = \frac{1}{\sqrt{2}}\left(|K^0\rangle - |\overline{K}^0\rangle\right) \qquad \mathcal{CP}|K_2^0\rangle = -|K_2^0\rangle\,.$$

The K_1^0 decays dominantly to two pions in s-wave (with total angular momentum 0), which form a $\mathcal{CP}$ even state. Still assuming that $\mathcal{CP}$ is conserved, this decay

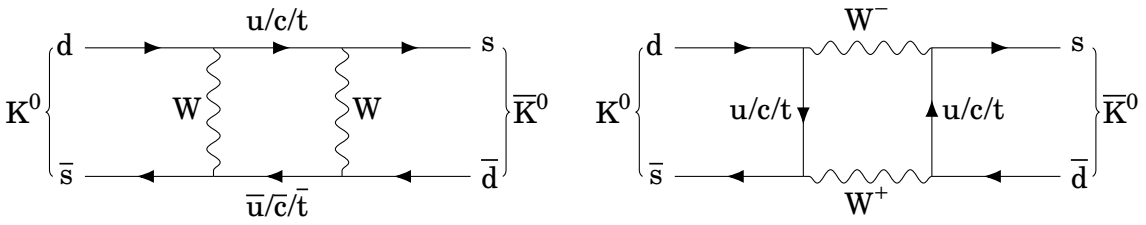

Figure 2.5.: The K^0 can transform into a $\overline{K}^0$ and vice-versa by the exchange of two W bosons and virtual up, charm or top quarks.

is forbidden for the K_2^0, which decays to a $\mathcal{CP}$ odd state with three pions instead. Since the mass of the kaon is only slightly higher than the mass of three pions, the latter process is strongly suppressed kinematically. Therefore, the lifetime of the K_2^0 of 5.116×10^{-8} s is three orders of magnitude larger than the lifetime of the K_1^0 of 8.954×10^{-11} s [11].

In 1964 Christenson, Cronin, Fitch and Turlay made use of this lifetime difference to find evidence for the decay of the K_2^0 into two pions [2], proving that $\mathcal{CP}$ symmetry is violated. They prepared a kaon beam which initially contains a mixture of K_1^0 and K_2^0 and let it travel until all K_1^0 had decayed. Then they searched for a decay of the remaining K_2^0 into two pions with an invariant mass that matches the mass of the kaon. Indeed they found two pion events from K_2^0 decays [2].

In contrast to $\mathcal{P}$ and $\mathcal{C}$, the combined symmetry $\mathcal{CP}$ is only slightly violated in the weak interaction. Therefore, the true eigenstates of the weak interaction are almost identical to the K_1^0 and K_2^0, but with a small contribution from the opposite $\mathcal{CP}$ eigenstate. They are called K_S ("short" lived) and K_L ("long" lived) and are given by

$$|K_S\rangle = |K_1^0\rangle + \varepsilon|K_2^0\rangle = \frac{1}{\sqrt{2}}\left((1+\varepsilon)\cdot|K^0\rangle + (1-\varepsilon)\cdot|\overline{K}^0\rangle\right),$$
$$|K_L\rangle = \varepsilon|K_1^0\rangle + |K_2^0\rangle = \frac{1}{\sqrt{2}}\left((1+\varepsilon)\cdot|K^0\rangle - (1-\varepsilon)\cdot|\overline{K}^0\rangle\right).$$

Once the existence of $\mathcal{CP}$ violation is accepted, it is necessary to quantify the contribution of possible sources of $\mathcal{CP}$ violation. For example, the parameter ε is related to an asymmetry between the transition probability from K^0 to $\overline{K}^0$ and the reverse process. It can be measured by comparing the decay rate of the K_L to flavor specific final states (states that identify the decaying kaon as either K^0 or $\overline{K}^0$), such as the semileptonic decays $K^0 \to \pi^- e^+ \nu_e$ / $\overline{K}^0 \to \pi^+ e^- \overline{\nu}_e$. In addition, asymmetries in the decay process itself can occur. Both effects are a consequence of a complex phase in the coupling parameter of the weak interaction, as will be shown in more detail in section 2.4.

2.3. CKM matrix

The coupling of the W bosons to quarks is described by the charged current Lagrangian [23]

$$L_{cc} \propto \overline{u}_i \gamma_\mu (1-\gamma_5) \boldsymbol{V}_{ij} d_j W^{+\mu} + \overline{d}_i \gamma_\mu (1-\gamma_5) \boldsymbol{V}_{ij}^\dagger u_j W^{-\mu}$$
$$\xrightarrow{\mathcal{CP}} L_{cc}^{\mathcal{CP}} \propto \overline{d}_j \gamma_\mu (1-\gamma_5) \boldsymbol{V}_{ij} u_i W^{-\mu} + \overline{u}_j \gamma_\mu (1-\gamma_5) \boldsymbol{V}_{ij}^\dagger d_i W^{+\mu},$$

where the up-type quarks u are coupled to the down-type quarks d through the Dirac matrices γ_μ. The chirality operator $(1-\gamma_5)$ introduces parity asymmetry by projecting to the left-handed component of the fermions. The unitary matrix $\boldsymbol{V}$ mixes down-type quarks from different families, in other words it is the generalized version of

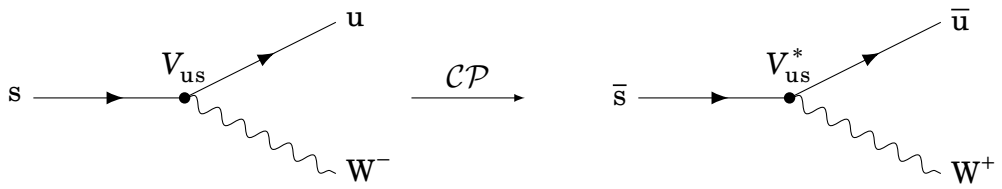

Figure 2.6.: The coupling for the decay of a strange quark to an up quark is proportional to the mixing matrix element V_{us}. In the charge conjugated decay $\overline{s} \to \overline{u}$ the complex conjugated element V^*_{us} appears instead.

the two family mixing matrix in eq. (2.1). The three family version of the mixing matrix was proposed by Kobayashi and Maskawa in 1973 [3] and is now called the *Cabibbo-Kobayashi-Maskawa* (CKM) matrix $\boldsymbol{V}_{\mathrm{CKM}}$, with corresponding quark tuples $u = (\mathrm{u}, \mathrm{c}, \mathrm{t})$ and $d = (\mathrm{d}, \mathrm{s}, \mathrm{b})$.

L_{cc} is invariant under a $\mathcal{CP}$ transformation if $\boldsymbol{V}^\dagger_{ij} = \boldsymbol{V}_{ji}$, that is if the mixing matrix is real. Conversely, this means that a $\mathcal{CP}$ violating theory requires at least one non-trivial complex phase in the mixing matrix. Figure 2.6 illustrates the effect of a $\mathcal{CP}$ transformation on a weak interaction vertex: the mixing matrix element that appears in the coupling is replaced by its complex conjugate.

2.3.1. Parameters in a unitary matrix

A general unitary $n \times n$ matrix $\boldsymbol{U}$ contains n^2 free parameters. This can be seen for example[1] by noting that $\boldsymbol{U}$ can be written as $\mathrm{e}^{\mathrm{i}\boldsymbol{H}}$, where $\boldsymbol{H}$ is hermitian. The total number of free parameters of $\boldsymbol{U}$ and $\boldsymbol{H}$ is therefore the same. The parameters in a hermitian matrix are easy to count: n real values on the diagonal plus $\frac{1}{2}n(n-1)$ complex values above the diagonal give a total of n^2 free parameters.

Unitarity requires that all column vectors of $\boldsymbol{U}$ are normalized and orthogonal to each other. This allows $(n-1)$ degrees of freedom for the magnitude of the entries in the first column, $(n-2)$ for the second column and so on, giving $\frac{1}{2}n(n-1)$ real parameters in total, which are usually written as rotation angles. The remaining $\frac{1}{2}n(n+1)$ free parameters of $\boldsymbol{U}$ are complex phases.

However, not all of these phases are physically observable. The Lagrangian L_{cc} is invariant under simultaneous transformations of

$$u_i \to \mathrm{e}^{\mathrm{i}\phi_i} u_i\,, \qquad d_j \to \mathrm{e}^{\mathrm{i}\phi_j} d_j\,, \qquad \boldsymbol{V}_{ij} \to \mathrm{e}^{\mathrm{i}(\phi_i - \phi_j)} \boldsymbol{V}_{ij}\,.$$

The relative phases between such redefinitions can absorb $(2n-1)$ phases in the mixing matrix [24]. An overall phase has no physical meaning and cannot cancel another phase in the matrix. This leaves $\frac{1}{2}(n-1)(n-2)$ physically observable phases with the

[1] Other arguments are based on the constraints imposed by the unitarity conditions.

potential to cause $\mathcal{CP}$ violation. For two families there are no such phases, which prompted Kobayashi and Maskawa to suggest a third family. In the CKM matrix ($n = 3$) there is exactly one non-trivial phase.

2.3.2. Parametrization of the CKM matrix

There are various ways to define the CKM matrix, depending on the order of rotations and on the definition of the phase. A common definition is given by [24]

$$\boldsymbol{V}_{\mathrm{CKM}} = \begin{pmatrix} V_{\mathrm{ud}} & V_{\mathrm{us}} & V_{\mathrm{ub}} \\ V_{\mathrm{cd}} & V_{\mathrm{cs}} & V_{\mathrm{cb}} \\ V_{\mathrm{td}} & V_{\mathrm{ts}} & V_{\mathrm{tb}} \end{pmatrix} = \begin{pmatrix} \mathrm{c}_{12}\mathrm{c}_{13} & \mathrm{s}_{12}\mathrm{c}_{13} & \mathrm{s}_{13}\mathrm{e}^{-\mathrm{i}\delta} \\ -\mathrm{s}_{12}\mathrm{c}_{23} - \mathrm{c}_{12}\mathrm{s}_{23}\mathrm{s}_{13}\mathrm{e}^{\mathrm{i}\delta} & \mathrm{c}_{12}\mathrm{c}_{23} - \mathrm{s}_{12}\mathrm{s}_{23}\mathrm{s}_{13}\mathrm{e}^{\mathrm{i}\delta} & \mathrm{s}_{23}\mathrm{c}_{13} \\ \mathrm{s}_{12}\mathrm{s}_{23} - \mathrm{c}_{12}\mathrm{c}_{23}\mathrm{s}_{13}\mathrm{e}^{\mathrm{i}\delta} & -\mathrm{c}_{12}\mathrm{s}_{23} - \mathrm{s}_{12}\mathrm{c}_{23}\mathrm{s}_{13}\mathrm{e}^{\mathrm{i}\delta} & \mathrm{c}_{23}\mathrm{c}_{13} \end{pmatrix},$$

where $\mathrm{s}_{ij} \equiv \sin\theta_{ij}$ and $\mathrm{c}_{ij} \equiv \cos\theta_{ij}$ are short for the sine and cosine of the rotation angles and δ is the single phase. A detailed derivation is given in [25], together with a list of alternative representations.

Another useful representation is the Wolfenstein parametrization [26], which expands the elements of the CKM matrix in terms of the parameter $\lambda = \sin\theta_{12} \approx 0.22$. The angle θ_{12} is the largest mixing angle in the CKM matrix and corresponds to the Cabibbo angle θ_C in the two family model (eq. (2.1)). Noting the empirical relation $|V_{\mathrm{cb}}| \approx \mathcal{O}(|V_{\mathrm{us}}|^2)$, Wolfenstein defined four parameters (λ, A, ρ, η) such that

$$\lambda = \sin\theta_{12}\,, \qquad A\lambda^2 = \sin\theta_{23}\,, \qquad A\lambda^3(\rho - \mathrm{i}\eta) = \sin\theta_{13}\mathrm{e}^{-\mathrm{i}\delta}\,.$$

Up to $\mathcal{O}(\lambda^3)$ the CKM matrix is then given by

$$\boldsymbol{V}_{\mathrm{CKM}} = \begin{pmatrix} 1 - \frac{1}{2}\lambda^2 & \lambda & A\lambda^3(\rho - \mathrm{i}\eta) \\ -\lambda & 1 - \frac{1}{2}\lambda^2 & A\lambda^2 \\ A\lambda^3(1 - \rho - \mathrm{i}\eta) & -A\lambda^2 & 1 \end{pmatrix} + \mathcal{O}(\lambda^4). \tag{2.2}$$

The advantage of this representation is that it directly reflects the hierarchy between the different elements: the largest coupling is found between quarks of the same family ($\mathcal{O}(1)$), the second largest between the first and the second family ($\mathcal{O}(\lambda)$), the third largest between the second and the third family ($\mathcal{O}(\lambda^2)$) and the smallest coupling is found between the first and the third family ($\mathcal{O}(\lambda^3)$). The complex phase appears only in the smallest matrix elements and in higher order terms, reflecting the fact that $\mathcal{CP}$ symmetry is only weakly violated in the standard model.

2.3.3. The unitarity triangle

An important test of the Kobayashi Maskawa model is to measure the individual elements of the CKM matrix and check if the resulting matrix is indeed unitary. If the CKM matrix turns out not to be unitary, the model is incomplete. In the Wolfenstein

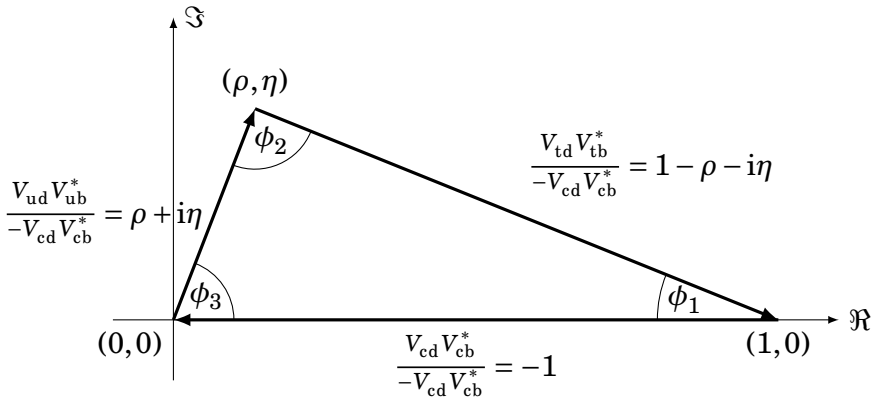

Figure 2.7.: Geometrically the unitarity condition eq. (2.7) can be represented as a triangle in the complex plane. The sides are normalized by $-V_{cd}V_{cb}^*$, so by definition two corners of the triangle are fixed at $(0,0)$ and $(1,0)$.

parametrization (eq. (2.2)), the unitarity condition $\boldsymbol{V}_{\text{CKM}}^\dagger \boldsymbol{V}_{\text{CKM}} = \mathbb{1}$ can be written up to $\mathcal{O}(\lambda^3)$ as

$$\left|V_{ud}\right|^2 + \left|V_{cd}\right|^2 + \left|V_{td}\right|^2 = 1 = (1-\lambda^2) + \lambda^2 + \mathcal{O}(\lambda^4), \tag{2.3}$$

$$\left|V_{us}\right|^2 + \left|V_{cs}\right|^2 + \left|V_{ts}\right|^2 = 1 = \lambda^2 + (1-\lambda^2) + \mathcal{O}(\lambda^4), \tag{2.4}$$

$$\left|V_{ub}\right|^2 + \left|V_{cb}\right|^2 + \left|V_{tb}\right|^2 = 1 = 1 + \mathcal{O}(\lambda^4), \tag{2.5}$$

$$V_{ud}V_{us}^* + V_{cd}V_{cs}^* + V_{td}V_{ts}^* = 0 = (\lambda - \frac{1}{2}\lambda^3) + (-\lambda + \frac{1}{2}\lambda^3) + \mathcal{O}(\lambda^5), \tag{2.6}$$

$$V_{ud}V_{ub}^* + V_{cd}V_{cb}^* + V_{td}V_{tb}^* = 0 = A\lambda^3(\rho+i\eta) - A\lambda^3 + A\lambda^3(1-\rho-i\eta) + \mathcal{O}(\lambda^5), \tag{2.7}$$

$$V_{us}V_{ub}^* + V_{cs}V_{cb}^* + V_{ts}V_{tb}^* = 0 = A\lambda^2 - A\lambda^2 + \mathcal{O}(\lambda^4). \tag{2.8}$$

Six similar equations emerge from the condition $\boldsymbol{V}_{\text{CKM}} \boldsymbol{V}_{\text{CKM}}^\dagger = \mathbb{1}$. Equations (2.3) to (2.5) are normalization conditions, while eqs. (2.6) to (2.8) are orthogonality conditions and constrain also the complex phase. Each of these conditions can be represented geometrically as a triangle in the complex plane. Of special interest is eq. (2.7), which contains three complex terms of order $\mathcal{O}(\lambda^3)$ that have to cancel perfectly, as shown in fig. 2.7. By convention the triangle is normalized such that one side is purely real and has length 1. This fixes two corners of the triangle. The third corner[2] can be determined by measuring the three angles and the length of the remaining two sides. Since the normalized triangle is fully defined by two parameters, this system is overconstrained and provides an excellent test of the Kobayashi Maskawa theory: the CKM Matrix is unitary only if the triangle closes.

[2] In the expansion up to $\mathcal{O}(\lambda^3)$ the third corner corresponds to the Wolfenstein parameters (ρ,η). For higher order expansions a correction factor appears.

The triangle in fig. 2.7 is referred to as "the unitarity triangle" of the B meson system, since it is defined by the columns V_{id} and V_{ib} of the CKM matrix and thus related to processes that involve B mesons. In principle, similar triangles can be defined for eqs. (2.6) and (2.8) if the Wolfenstein parametrization is extended to order $\mathcal{O}(\lambda^5)$. However, these triangles are almost degenerate, with one side being two respectively four orders of magnitude smaller than the other sides. The degenerate triangles are related to K mesons (eq. (2.6)) and B_s mesons (eq. (2.8)).

2.4. CP violation in the B meson system

$\mathcal{CP}$ violation can manifest itself in three different ways, which are illustrated in fig. 2.8:

1. *$\mathcal{CP}$ violation in decay*, also called direct $\mathcal{CP}$ violation, refers to decay processes that happen with a different rate than the charge conjugated process:

$$\left|\langle f|\mathrm{B}^0\rangle\right|^2 \neq \left|\langle \overline{f}|\overline{\mathrm{B}}^0\rangle\right|^2,$$

   where $|f\rangle$ is an arbitrary (semi)hadronic final state and $|\overline{f}\rangle = \mathcal{CP}|f\rangle$ is the corresponding $\mathcal{CP}$ conjugated state. $\mathcal{CP}$ violation in decay can occur both for charged and for neutral mesons.

2. *$\mathcal{CP}$ violation in mixing*, also called indirect $\mathcal{CP}$ violation, occurs only for neutral mesons which can transform into each other, as discussed for the K^0 and $\overline{\mathrm{K}}^0$ in section 2.2.4. $\mathcal{CP}$ violation in mixing refers to the case that the transition rate of a meson to the corresponding antimeson differs from the reciprocal process:

$$\left|\langle \overline{\mathrm{B}}^0|\mathrm{B}^0\rangle\right|^2 \neq \left|\langle \mathrm{B}^0|\overline{\mathrm{B}}^0\rangle\right|^2.$$

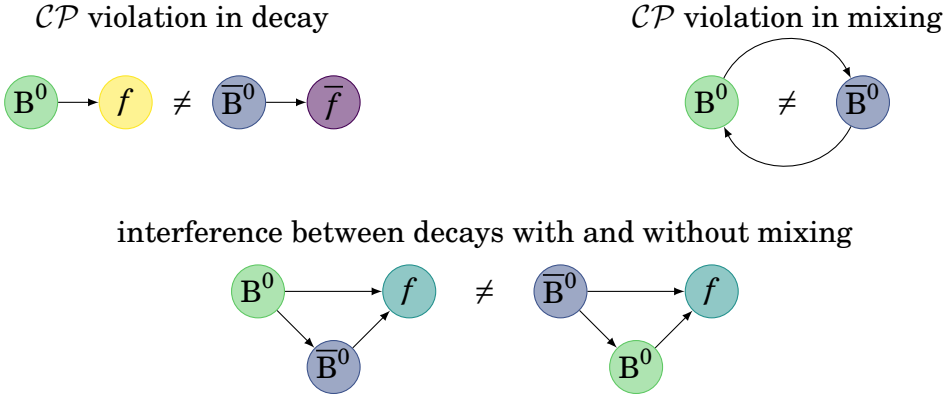

Figure 2.8.: Illustration of the three different types of $\mathcal{CP}$ violation (see text).

If this is the case, the mass eigenstates of the neutral meson system are not identical with $\mathcal{CP}$ eigenstates.

3. *$\mathcal{CP}$ violation in the interference between decays with and without mixing*, also called mixing-induced $\mathcal{CP}$ violation, can also be observed in neutral meson systems where the meson and antimeson mix coherently. If both flavor states can decay into the same final state, the decay amplitude is a coherent sum of the two flavor states. This type of process is therefore sensitive to a relative phase between the decay of a meson and the corresponding antimeson. $\mathcal{CP}$ violation is observed if

$$\left|\langle f|\mathrm{B}^0\rangle\right|^2 \neq \left|\langle f|\overline{\mathrm{B}}^0\rangle\right|^2,$$

where $|f\rangle$ is an arbitrary (semi)hadronic final state that is accessible from both B^0 and $\overline{\mathrm{B}}^0$ (typically a $\mathcal{CP}$ eigenstate).

In the following, all of these processes are discussed for the B^0 ($\mathrm{b\bar{d}}$) and its antiparticle $\overline{\mathrm{B}}^0$ ($\mathrm{b\bar{d}}$). For other meson systems, like the K^0 ($\mathrm{\bar{s}d}$), D^0 ($\mathrm{c\bar{u}}$) and B_s^0 ($\mathrm{\bar{b}s}$), the same principle phenomena occur, but the rates and timescales are very different depending on the mass and lifetime differences of the respective weak interaction eigenstates.

2.4.1. CP violation in decay

Consider the decay amplitude A_f of a B^0 to a final state f. In general, different intermediate states contribute to the decay, so the decay amplitude can be written as a sum [23]

$$A_f = \sum_i \left|a_i\right| \mathrm{e}^{\mathrm{i}(\delta_i+\phi_i)}.$$

Two types of phases appear here: the so-called weak phases ϕ_i, which change sign under a $\mathcal{CP}$ transformation, and the so-called strong phases δ_i, which are invariant under a $\mathcal{CP}$ transformation. With this definition, the decay amplitude $\overline{A}_{\overline{f}}$ of a $\overline{\mathrm{B}}^0$ to the charge conjugated final state $\overline{f}$ is given by

$$\overline{A}_{\overline{f}} = \sum_i \left|a_i\right| \mathrm{e}^{\mathrm{i}(\delta_i-\phi_i)}.$$

In the standard model, the weak phases are related to elements of the CKM matrix which appear in the diagram of a given decay. In the charge conjugated decay the complex conjugated matrix element appears, so the sign of the weak phase changes. The strong phases are due to hadronic particles in the final state, exhibiting strong interactions which conserve $\mathcal{CP}$ symmetry, so the phase does not change.

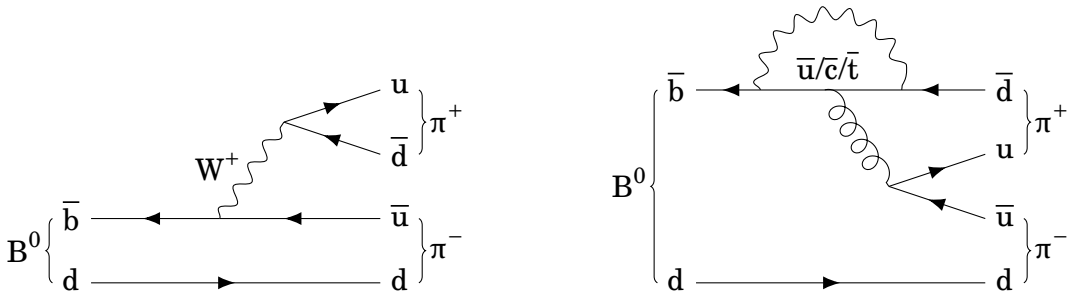

Figure 2.9.: Tree diagram (left) and penguin diagram (right) for the decay $B^0 \to \pi^+\pi^-$.

$\mathcal{CP}$ symmetry is violated if $|A_f| \neq \left|\overline{A}_{\overline{f}}\right|$. This is equivalent to the condition

$$|A_f|^2 - \left|\overline{A}_{\overline{f}}\right|^2 = -2\sum_{i,j} |a_i|\,|a_j| \sin(\delta_i - \delta_j)\sin(\phi_i - \phi_j) \neq 0\,.$$

So $\mathcal{CP}$ violation in decay is only possible if at least two intermediate states with different weak and strong phases contribute to a decay. In other words, it is a consequence of the interference between states with different complex phases. In B decays, the relevant intermediate states are usually a "tree" diagram and a "penguin" diagram, as shown in fig. 2.9 for the decay $B^0 \to \pi^+\pi^-$. In this example, the CKM matrix elements for both contributing diagrams are of order λ^3, so the amplitudes are of similar size. Therefore, a relatively large direct $\mathcal{CP}$ violation is found [27]. In contrast, for decays that are dominated by a single diagram, direct $\mathcal{CP}$ violation is very small.

The decay amplitudes $|A_f|^2$ and $\left|\overline{A}_{\overline{f}}\right|^2$ are proportional to the decay rates, so they are directly accessible experimentally. However, the individual decay amplitudes $|a_i|$ and the strong phases δ_i are very complicated to calculate theoretically and have large uncertainties. Therefore, quantities that depend only on the weak phases provide much cleaner tests of the standard model.

2.4.2. CP violation in mixing

The neutral B^0 and $\overline{B}^0$ mesons can transform into each other by diagrams analogous to fig. 2.5. The general state of such a system is given by a linear combination

$$|\psi\rangle = a|B^0\rangle + b|\overline{B}^0\rangle\,,$$

which evolves according to the time-dependent Schrödinger equation [23]

$$\mathrm{i}\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} a \\ b \end{pmatrix} = \boldsymbol{H}\begin{pmatrix} a \\ b \end{pmatrix} = (\boldsymbol{M} - \frac{\mathrm{i}}{2}\boldsymbol{\Gamma})\begin{pmatrix} a \\ b \end{pmatrix},$$

where $\boldsymbol{M}$ and $\boldsymbol{\Gamma}$ are hermitian matrices. Note that the total Hamiltonian $\boldsymbol{H}$ is not hermitian, that is probability is not conserved. This reflects the fact that the total number of B^0 and $\overline{\mathrm{B}}^0$ is not constant, since the B mesons decay after some time. The diagonal elements of $\boldsymbol{M}$ and $\boldsymbol{\Gamma}$ are related to the mass and lifetime of the pure flavor eigenstates, respectively. As a consequence of $\mathcal{CPT}$ invariance the B^0 and the $\overline{\mathrm{B}}^0$ have the same mass and lifetime, which implies $H_{11} = H_{22}$. The Hamiltonian can therefore be written as

$$\boldsymbol{H} = \begin{pmatrix} M - \frac{\mathrm{i}}{2}\Gamma & M_{12} - \frac{\mathrm{i}}{2}\Gamma_{12} \\ M_{12}^* - \frac{\mathrm{i}}{2}\Gamma_{12}^* & M - \frac{\mathrm{i}}{2}\Gamma \end{pmatrix}.$$

The off-diagonal terms are responsible for the mixing and are thus related to box diagrams like the one in fig. 2.5 and to the CKM matrix. To find the mass eigenstates, $\boldsymbol{H}$ is diagonalized, which gives the eigenvalues

$$\begin{aligned} m_{\mathrm{H/L}} - \frac{\mathrm{i}}{2}\Gamma_{\mathrm{H/L}} &= M - \frac{\mathrm{i}}{2}\Gamma \pm \sqrt{\left(M_{12} - \frac{\mathrm{i}}{2}\Gamma_{12}\right)\left(M_{12}^* - \frac{\mathrm{i}}{2}\Gamma_{12}^*\right)} \\ &\equiv \left(M \pm \frac{1}{2}\Delta m\right) - \frac{\mathrm{i}}{2}\left(\Gamma \mp \frac{1}{2}\Delta\Gamma\right), \end{aligned}$$

where $m_{\mathrm{H/L}}$ and $\Gamma_{\mathrm{H/L}}$ are the mass and lifetime of the heavy (H) and light (L) mass eigenstates and $\Delta m = m_{\mathrm{H}} - m_{\mathrm{L}}$ and $\Delta\Gamma = \Gamma_{\mathrm{L}} - \Gamma_{\mathrm{H}}$ are the differences between them. By definition Δm is positive, while $\Delta\Gamma$ can in principle be either positive or negative, depending on the mixing matrix (for the K^0 system discussed in section 2.2.4, the $\mathrm{K_S}$ corresponds to the lighter mass eigenstate). The corresponding mass eigenstates are found to be

$$\begin{aligned} |\mathrm{B_L}\rangle &= p|\mathrm{B}^0\rangle + q|\overline{\mathrm{B}}^0\rangle, \\ |\mathrm{B_H}\rangle &= p|\mathrm{B}^0\rangle - q|\overline{\mathrm{B}}^0\rangle, \end{aligned}$$

with the coefficients obeying

$$\frac{q}{p} = -\sqrt{\frac{M_{12}^* - \frac{\mathrm{i}}{2}\Gamma_{12}^*}{M_{12} - \frac{\mathrm{i}}{2}\Gamma_{12}}}, \qquad |p|^2 + |q|^2 = 1.$$

$\mathcal{CP}$ violation occurs if $\left|\frac{q}{p}\right| \neq 1$, that is if the mass eigenstates are not also $\mathcal{CP}$ eigenstates, as in the case of the $\mathrm{K_L}$ and $\mathrm{K_S}$. This is only the case if the off-diagonal terms M_{12} and Γ_{12} are of similar magnitude and at least one of them has a non-trivial complex phase.

Another way to see that $\mathcal{CP}$ symmetry is violated is to look at the time evolution of an arbitrary initial state $|\psi\rangle$

$$\begin{aligned} |\psi(t=0)\rangle &= \alpha|\mathrm{B_L}\rangle + \beta|\mathrm{B_H}\rangle, \\ |\psi(t)\rangle &= \alpha \cdot \mathrm{e}^{-\mathrm{i}m_{\mathrm{L}}t}\mathrm{e}^{-\frac{1}{2}\Gamma_{\mathrm{L}}t}|\mathrm{B_L}\rangle + \beta \cdot \mathrm{e}^{-\mathrm{i}m_{\mathrm{H}}t}\mathrm{e}^{-\frac{1}{2}\Gamma_{\mathrm{H}}t}|\mathrm{B_H}\rangle. \end{aligned}$$

An initially pure flavor state $|\mathrm{B}^0\rangle$ corresponds to $\alpha = \beta = \frac{1}{2p}$; an initially pure $|\overline{\mathrm{B}}^0\rangle$ state corresponds to $\alpha = -\beta = \frac{1}{2q}$. The time evolution is then given by

$$\begin{aligned}
|\mathrm{B}^0(t)\rangle &= g_+(t)|\mathrm{B}^0\rangle + \frac{q}{p} g_-(t)|\overline{\mathrm{B}}^0\rangle\,, \\
|\overline{\mathrm{B}}^0(t)\rangle &= \frac{p}{q} g_-(t)|\mathrm{B}^0\rangle + g_+(t)|\overline{\mathrm{B}}^0\rangle\,, \\
g_\pm(t) &= \frac{1}{2}\mathrm{e}^{-\mathrm{i}Mt}\mathrm{e}^{-\frac{1}{2}\Gamma t}\left(\mathrm{e}^{\frac{\mathrm{i}}{2}\Delta m t}\mathrm{e}^{-\frac{1}{4}\Delta\Gamma t} \pm \mathrm{e}^{-\frac{\mathrm{i}}{2}\Delta m t}\mathrm{e}^{\frac{1}{4}\Delta\Gamma t}\right).
\end{aligned} \tag{2.9}$$

So if $\left|\frac{q}{p}\right| \neq 1$, the probability to find an initial B^0 as a $\overline{\mathrm{B}}^0$ at time t is not equal to the probability to find an initial $\overline{\mathrm{B}}^0$ as a B^0 after the same time. The time integrated probabilities then also differ. Experimentally these probabilities can be related to the decay rates of a B meson with known initial flavor state to a flavor specific final state.

In contrast to the K meson system, the lifetime difference $\Delta\Gamma$ in the B meson system is much smaller than the mass difference Δm [23], since due to the large mass of the B meson there are many more accessible final states than for the kaons. Similarly, for the off-diagonal matrix elements one finds $\Gamma_{12} \ll M_{12}$ [28]. Therefore, to $\mathcal{O}(10^{-2})$ accuracy the following approximation holds:

$$\frac{q}{p} = -\frac{|M_{12}|}{M_{12}} \quad \rightarrow \quad \left|\frac{q}{p}\right| \approx 1\,.$$

So $\mathcal{CP}$ violation in mixing is very small in the B meson system. In this approximation, the mixing coefficients in eq. (2.9) can be simplified to

$$g_\pm(t) = \frac{1}{2}\mathrm{e}^{-\mathrm{i}Mt}\mathrm{e}^{-\frac{1}{2}\Gamma t}\left(\mathrm{e}^{\frac{\mathrm{i}}{2}\Delta m t} \pm \mathrm{e}^{-\frac{\mathrm{i}}{2}\Delta m t}\right).$$

2.4.3. Interference between decays with and without mixing

When the final state of a B decay is accessible from both B^0 and $\overline{\mathrm{B}}^0$, both terms in eq. (2.9) contribute to the decay amplitude. Of special interest is the case where the final state is a $\mathcal{CP}$ eigenstate, that is

$$\mathcal{CP}|f_{\mathcal{CP}}\rangle = \pm|f_{\mathcal{CP}}\rangle\,.$$

Denoting the decay amplitudes of a pure B^0 ($\overline{\mathrm{B}}^0$) to $f_{\mathcal{CP}}$ as $A_{\mathcal{CP}}$ ($\overline{A}_{\mathcal{CP}}$), the decay amplitudes after a time t can be derived from eq. (2.9) as

$$\begin{aligned}
A_{\mathcal{CP}}(t) &= \mathrm{e}^{-\mathrm{i}Mt}\mathrm{e}^{-\frac{1}{2}\Gamma t}A_{\mathcal{CP}}\left(\cos\frac{\Delta m t}{2} + \mathrm{i}\cdot\frac{q}{p}\frac{\overline{A}_{\mathcal{CP}}}{A_{\mathcal{CP}}}\sin\frac{\Delta m t}{2}\right), \\
\overline{A}_{\mathcal{CP}}(t) &= \mathrm{e}^{-\mathrm{i}Mt}\mathrm{e}^{-\frac{1}{2}\Gamma t}A_{\mathcal{CP}}\frac{p}{q}\left(\mathrm{i}\sin\frac{\Delta m t}{2} + \frac{q}{p}\frac{\overline{A}_{\mathcal{CP}}}{A_{\mathcal{CP}}}\cos\frac{\Delta m t}{2}\right).
\end{aligned}$$

The $\mathcal{CP}$ dependent parameters can be summarized in a single complex number $\lambda_{\mathcal{CP}}$, which is defined as

$$\lambda_{\mathcal{CP}} = \frac{q}{p}\frac{\overline{A}_{\mathcal{CP}}}{A_{\mathcal{CP}}}.$$

The time dependent decay rates are then given by

$$\left|A_{\mathcal{CP}}(t)\right|^2 = \mathrm{e}^{-\Gamma t}\left|A_{\mathcal{CP}}\right|^2\left(\frac{1+\left|\lambda_{\mathcal{CP}}\right|^2}{2} + \frac{1-\left|\lambda_{\mathcal{CP}}\right|^2}{2}\cos(\Delta m t) - \Im(\lambda_{\mathcal{CP}})\sin(\Delta m t)\right),$$

$$\left|\overline{A}_{\mathcal{CP}}(t)\right|^2 = \mathrm{e}^{-\Gamma t}\left|A_{\mathcal{CP}}\right|^2\left|\frac{p}{q}\right|^2\left(\frac{1+\left|\lambda_{\mathcal{CP}}\right|^2}{2} - \frac{1-\left|\lambda_{\mathcal{CP}}\right|^2}{2}\cos(\Delta m t) + \Im(\lambda_{\mathcal{CP}})\sin(\Delta m t)\right).$$

A difference between the rates can be observed for $t \neq 0$ if $\lambda_{\mathcal{CP}}$ has a non-trivial complex phase, even if $\left|\lambda_{\mathcal{CP}}\right| = 1$, in other words if there is neither $\mathcal{CP}$ violation in decay nor $\mathcal{CP}$ violation in mixing. Furthermore, since the rates depend on $\Im(\lambda_{\mathcal{CP}})$, the complex phase can be measured explicitly. Note that it is essential to measure the time dependent rates. If the rates are integrated over time, the term $\sin(\Delta m t)$ becomes 0 and all information about the phase of $\lambda_{\mathcal{CP}}$ is lost.

With the approximation $\left|\frac{p}{q}\right|^2 \approx 1$, the time dependent $\mathcal{CP}$ asymmetry is given by

$$a_{\mathcal{CP}}(t) \equiv \frac{\left|A_{\mathcal{CP}}(t)\right|^2 - \left|\overline{A}_{\mathcal{CP}}(t)\right|^2}{\left|A_{\mathcal{CP}}(t)\right|^2 + \left|\overline{A}_{\mathcal{CP}}(t)\right|^2} = \frac{1-\left|\lambda_{\mathcal{CP}}\right|^2}{1+\left|\lambda_{\mathcal{CP}}\right|^2}\cos(\Delta m t) - \frac{2\Im(\lambda_{\mathcal{CP}})}{1+\left|\lambda_{\mathcal{CP}}\right|^2}\sin(\Delta m t). \quad (2.10)$$

From this observable the parameter $\lambda_{\mathcal{CP}}$ can be directly extracted. So the task is to measure the two decay rates $\left|A_{\mathcal{CP}}(t)\right|^2$ and $|\overline{A}_{\mathcal{CP}}(t)|^2$ as precisely as possible as functions of time. Unfortunately, the B meson has many different decay channels, due to its high mass of $5.28\,\mathrm{GeV}/c^2$. Therefore, the branching fractions to specific $\mathcal{CP}$ eigenstates are generally small, so a very large number of events is required. This motivates the construction of a dedicated B factory, which is designed to produce a large number of B and $\overline{\mathrm{B}}$ mesons.

More precisely, B factories produce $\mathrm{B}\overline{\mathrm{B}}$ mesons in pairs, which originate from the decay of a $\Upsilon(4S)$ meson produced in an $\mathrm{e}^+\mathrm{e}^-$ annihilation. Approximately 50 % of these pairs consist of neutral $\mathrm{B}^0\overline{\mathrm{B}}^0$ mesons, which are produced in the maximally entangled state

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|\mathrm{B}^0\rangle|\overline{\mathrm{B}}^0\rangle - |\overline{\mathrm{B}}^0\rangle|\mathrm{B}^0\rangle).$$

Although the flavor states of each particle can oscillate via mixing, they evolve in phase, so until one particle decays the flavor states of the two particles are always

orthogonal to each other. The time evolution of the entangled system is given by

$$\begin{aligned}|\psi(t_1,t_2)\rangle &= \frac{1}{\sqrt{2}}(|\mathrm{B}^0(t_1)\rangle|\overline{\mathrm{B}}^0(t_2)\rangle - |\overline{\mathrm{B}}^0(t_1)\rangle|\mathrm{B}^0(t_2)\rangle)\\ &= \frac{1}{\sqrt{2}}\mathrm{e}^{-\mathrm{i}M(t_1+t_2)}\mathrm{e}^{-\frac{1}{2}\Gamma(t_1+t_2)}\cdot\left(\cos\left(\frac{\Delta m(t_1-t_2)}{2}\right)\left(|\mathrm{B}^0\overline{\mathrm{B}}^0\rangle - |\overline{\mathrm{B}}^0\mathrm{B}^0\rangle\right)\right.\\ &\qquad \left.-\mathrm{i}\sin\left(\frac{\Delta m(t_1-t_2)}{2}\right)\left(\frac{p}{q}|\mathrm{B}^0\mathrm{B}^0\rangle - \frac{q}{p}|\overline{\mathrm{B}}^0\overline{\mathrm{B}}^0\rangle\right)\right).\end{aligned}$$

Until one particle decays $t_1 = t_2$, so the term with two B mesons of the same flavor vanishes. After the decay, t_1 and t_2 correspond to the respective decay times. From the time evolved state $|\psi(t_1,t_2)\rangle$ one can calculate the decay rate for one particle decaying to the final state f_1 after a time t_1 and the other decaying to the final state f_2 after a time t_2.

To measure the time dependent $\mathcal{CP}$ asymmetry $a_{\mathcal{CP}}$, one looks for decays where one particle decays to a $\mathcal{CP}$ eigenstate $f_{\mathcal{CP}}$ at the time $t_{\mathcal{CP}}$, while the other decays to a flavor specific final state f_{tag} at the time t_{tag}. The latter is called the *tag side*, as it "tags" the flavor of both particles at the decay time t_{tag}. If the tag side meson decays first, the $\mathcal{CP}$ side meson is projected to the orthogonal flavor state at the moment t_{tag}. If the $\mathcal{CP}$ side meson decays first, at $t_{\mathcal{CP}}$ the $\mathcal{CP}$ side meson was in the state that would have evolved to the flavor state orthogonal to f_{tag}. Thus, the decay of two entangled B mesons to the final states f_{tag} and $f_{\mathcal{CP}}$ is equivalent to the decay of a single B meson with known flavor to the final state $f_{\mathcal{CP}}$. Going through the full calculation of the two particle decay rate, one can rederive eq. (2.10), except that the time t is replaced by $t_{\mathcal{CP}} - t_{\text{tag}}$ [23]. Therefore, the most important observable for the measurement of time

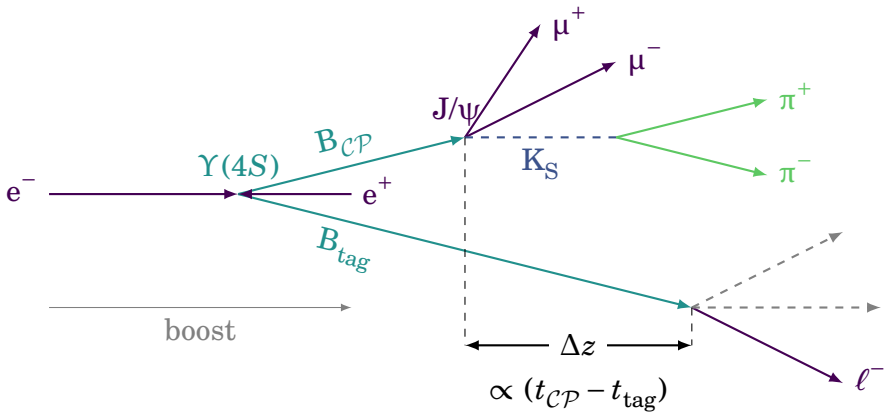

Figure 2.10.: Typical $\mathrm{B}^0\overline{\mathrm{B}}^0$ decay for the measurement of time dependent $\mathcal{CP}$ violation. One B decays to a $\mathcal{CP}$ eigenstate (here $\mathrm{J}/\psi\mathrm{K}_\mathrm{S}$), the other decays to a flavor specific final state (here a semileptonic decay, where the charge of the lepton is related to the flavor of the B_{tag}).

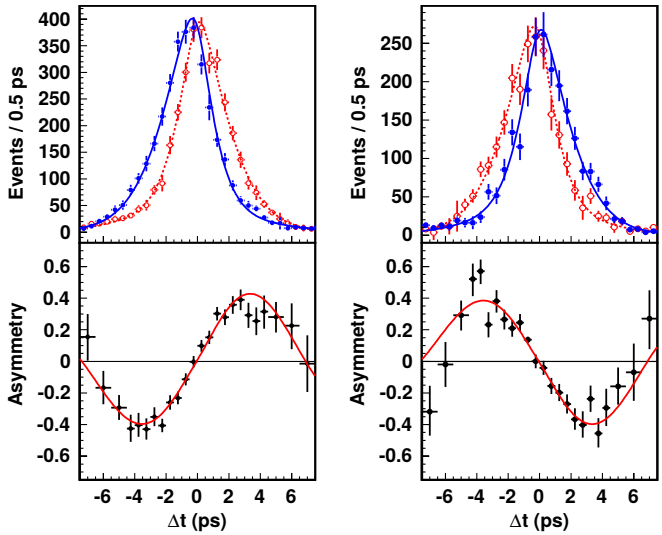

Figure 2.11.: Measured $\mathcal{CP}$ asymmetry in $B^0 \to (c\bar{c})K^0$ decays in the Belle experiment [29]. Shown are the background subtracted Δt distributions for $B_{tag} = B^0$ (red) and $B_{tag} = \overline{B}^0$ (blue) and the resulting asymmetry $a_{\mathcal{CP}}(\Delta t)$. Left: combined $\mathcal{CP}$-odd decay modes, right: $\mathcal{CP}$-even decay mode.

dependent $\mathcal{CP}$ violation is the difference between the decay times of the two B mesons.

To measure the oscillation in eq. (2.10), a timing resolution on the order of pico seconds is required. Since such small differences in the decay time cannot be measured, the $\Upsilon(4S)$ is produced with a boost, that is the colliding electrons and positrons have different energies. Thus, the difference in the decay time is translated to a distance between the two decay vertices, which can be reconstructed more easily by precise silicon vertex detectors. The principle of the measurement is illustrated in fig. 2.10. The details about the boost of the $\Upsilon(4S)$ system are given in the next chapter. Figure 2.11 shows the results of a time dependent $\mathcal{CP}$ violation measurement in the Belle experiment for the $\mathcal{CP}$-odd decay $B^0 \to J/\psi K_S$ and related $\mathcal{CP}$-odd and $\mathcal{CP}$-even decay channels with the same quark content. For this decay, the asymmetry at $\Delta t = 0$ is negligible, so there is no direct $\mathcal{CP}$ violation, but in the time-dependent distribution there is a clear asymmetry, which is directly related to the angle ϕ_1 of the unitarity triangle shown in fig. 2.7 [29].

2.5. Open questions of the standard model

The standard model, with the particles and interactions shown in fig. 2.1, gives a remarkably good description of the phenomena observed in high energy physics experiments, including $\mathcal{CP}$ violation processes. Figure 2.12 shows the present constraints on

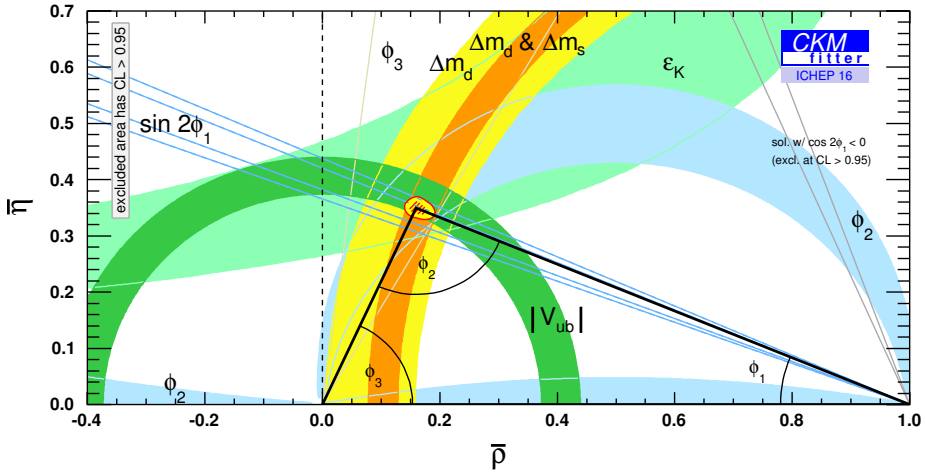

Figure 2.12.: Constraints on the unitarity triangle from measurements up to 2016 [30].

the angles and sides of the unitarity triangle, which are all compatible with each other within the present precision. However, there are many questions not answered by the standard model. One of them is how the asymmetry between matter and antimatter observed in the universe could arise. The measured $\mathcal{CP}$ violation is too weak to explain the asymmetry quantitatively. In addition, no baryon number violating processes are known in the standard model. Another problem is that the standard model has no explanation for dark matter and dark energy, which are necessary in cosmological models to explain the dynamics of galaxies and galaxy clusters and the expansion of the universe. Finally, the standard model does not include any description of gravity.

Many theories have been proposed that extend the standard model, mostly by including new particles, interactions or symmetries. To discriminate between the different models, experiments need to observe some effect that is not in agreement with the standard model, which can give a hint at the underlying physics. There are two complementary experimental approaches. Experiments at the *energy frontier*, such as the Large Hadron Collider (LHC) at CERN, aim to create collisions at ever higher energies to find new heavy particles. In contrast, Belle II is an experiment at the *intensity frontier*, which searches for small deviations from the standard model predictions, using high statistics event samples.

One possibility to observe tensions in the standard model is that the unitarity triangle might not close. Although the present constraints are compatible with each other, as shown in fig. 2.12, it is still possible that tensions will appear when the uncertainties are reduced. New particles can be observed indirectly if they contribute to known decays through loop processes. Another approach is the search for rare decays, which are forbidden or strongly suppressed in the standard model. The observation of such a decay would be an immediate proof of physics beyond the

standard model. All of these measurements require very high statistics, motivating the luminosity upgrade of the KEKB collider to SuperKEKB, which will increase the statistics of collected events by a factor of 50 compared to its predecessor.

3. The Belle II Experiment

The following chapter gives an overview over the Belle II experiment. It starts with an introduction to the SuperKEKB collider, based on the requirements for the measurement of time dependent $\mathcal{CP}$ violation that were discussed in the previous chapter, followed by an introduction to the Belle II detector. Then some of the physics processes that can be studied in Belle II are discussed, including but not restricted to B physics. The focus is set on those event signatures that are most difficult to trigger, namely events with low track multiplicity. These event types will be the subject of the trigger efficiency studies in chapter 8. Finally, different sources of background are discussed that need to be suppressed by the trigger.

3.1. SuperKEKB

SuperKEKB is an asymmetric electron-positron collider, where "asymmetric" refers to the energy difference between the colliding electrons and positrons. SuperKEKB consists of a linear accelerator and a double storage ring of 3 km circumference [31], which is sketched in fig. 3.1. In addition, there is a *damping ring* to cool down the transverse momentum of the positrons, which are produced with a too large emittance for the nano beam scheme of SuperKEKB (described in section 3.1.2). Electrons and positrons are then accelerated to their respective target energies by the same linear accelerator and injected in bunches into independent storage rings: the *high energy ring* (HER) for the electrons and the *low energy ring* (LER) for the positrons. Then the electrons and positrons are brought to collision at the interaction point (IP), which is surrounded by the Belle II detector. In order to separate the beams before and after the collision, the beams cross at an angle of 83 mrad.

3.1.1. Beam energies

The most important running mode of SuperKEKB is at the $\Upsilon(4S)$ resonance, where the center of mass energy of the e^+e^- collision matches the mass of the $\Upsilon(4S)$ meson of $10.58\,\mathrm{GeV}/c^2$. The $\Upsilon(4S)$ is the fourth excited bottomonium ($b\bar{b}$) state and has a mass just above the production threshold of two B mesons, which have a mass of $5.28\,\mathrm{GeV}/c^2$. Therefore, the $\Upsilon(4S)$ decays with more than 96 % probability to B meson pairs, which are produced nearly at rest in the center of mass system. The momentum in the laboratory system is thus given solely by the boost due to the asymmetric

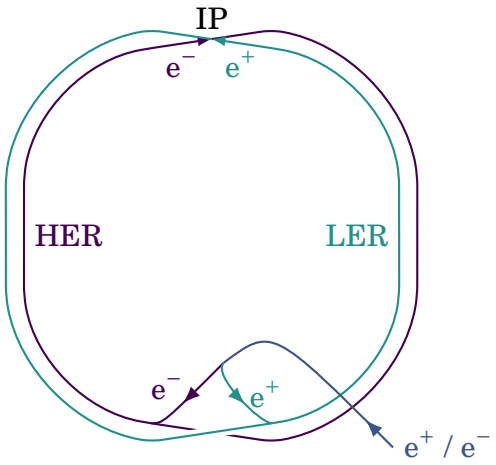

Figure 3.1.: Schematic of the SuperKEKB storage ring structure, with the low energy ring (LER), the high energy ring (HER) and the injection tunnel leading to the linear accelerator. The electrons and positrons collide at the interaction point (IP).

energies of the electron and positron beams. This allows to directly relate the distance Δz between the decay vertices to the decay time difference Δt, which is required for the measurement of time dependent $\mathcal{CP}$ violation:

$$\Delta t = \frac{\Delta z}{c\beta\gamma}.$$

The boost depends on the energies of the electrons and positrons and on the crossing angle between the beams. For a head-on collision, the boost would be given by

$$\beta = \frac{E_{\mathrm{HER}} - E_{\mathrm{LER}}}{E_{\mathrm{HER}} + E_{\mathrm{LER}}}.$$

Since the electrons and positrons collide at a finite crossing angle, the actual boost is slightly larger. Table 3.1 shows the beam energies and the resulting boost for SuperKEKB and its predecessor KEKB. Compared to KEKB, the boost is reduced by a

Table 3.1.: Comparison of the beam energies and the resulting boost at KEKB [32] and SuperKEKB [33].

parameter	KEKB	SuperKEKB
e^+/e^- energy	3.5 GeV / 8.0 GeV	4.0 GeV / 7.0 GeV
crossing angle	22 mrad	83 mrad
boost $\beta\gamma$	0.425	0.287

factor of $\approx 2/3$. This is necessary to achieve the design luminosity, but it reduces the average spatial distance between the B decay vertices. Nevertheless, by improving the spatial resolution of the vertex detector, the Δt resolution at Belle II can even by improved with respect to the resolution achieved at Belle [34].

3.1.2. Luminosity upgrade

The instantaneous luminosity $\mathcal{L}$ is a measure for the collision rate. For a process with cross section σ, the number of events per second is given by

$$\frac{\mathrm{d}N}{\mathrm{d}t} = \mathcal{L} \cdot \sigma \,.$$

The cross section for $\Upsilon(4S)$ production at the default beam energy of SuperKEKB is 1.05 nb [34]. KEKB achieved an integrated luminosity of $1040\,\mathrm{fb}^{-1}$ and collected 770×10^6 $\mathrm{B\overline{B}}$ events in total [35]. SuperKEKB aims for an increase of the instantaneous luminosity by a factor of 40 compared to peak luminosity at KEKB, as well as a 50 times higher integrated luminosity.

For two beams with a Gaussian profile of horizontal size σ_x and vertical size σ_y, the luminosity is given by

$$\mathcal{L} = \frac{N_+ N_- f_c}{4\pi\sigma_x\sigma_y} \cdot R_L \,, \tag{3.1}$$

where $N_{+/-}$ is the number of particles in a positron/electron bunch and f_c is the bunch crossing frequency. The factor R_L is a reduction factor that includes geometrical effects from the finite crossing angle and the bunch length [31]. From eq. (3.1) it can be seen that the luminosity can be increased both by increasing the beam currents $I_\pm \propto N_\pm f_c$ and by reducing the beam size.

The beam size depends in part on the so-called *betatron oscillations*. Particles that deviate from the ideal orbit are bent back by focusing magnets. In consequence, as particles travel along the ring, they oscillate around the ideal orbit. The beam size $\sigma_{x/y}$ corresponds to the envelope of these oscillations. At any given point s along the ring it can be written as

$$\sigma_{x/y}(s) = \sqrt{\epsilon_{x/y}\beta_{x/y}(s)} \,.$$

Here, two new parameters are introduced: the *emittance* ϵ and the *beta function* β. The beta function changes along the ring and describes the effect of focusing and defocusing magnets. The emittance is related to the initial beam divergence at injection into the main ring, and is unaffected by the focusing system. A detailed description of the beam dynamics is given in [36]. The change of the beam energies compared to KEKB, which is given in table 3.1, was chosen to reduce the emittance of the positron beam, since a low emittance is easier to achieve with a higher beam energy [33]. The damping

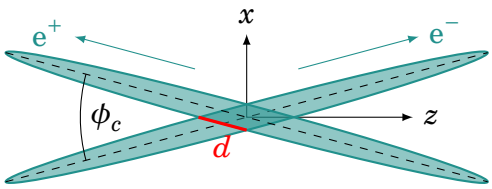

Figure 3.2.: Schematic of an electron-positron bunch collision in the nano beam scheme.

ring for the positrons was also newly introduced in SuperKEKB to further reduce their emittance. Note that due to synchrotron radiation, the horizontal emittance ϵ_x is larger than the vertical emittance ϵ_y by two orders of magnitude.

The main new feature in the luminosity upgrade of SuperKEKB is the so-called *nano beam scheme* [33], where the vertical beta function β_y at the interaction point is squeezed down to 0.3 mm, which is 20 times smaller than at KEKB, leading to a vertical beam size of only $\sigma_y \approx 60\,\text{nm}$ [37]. An inherent property of the beam focusing is that the minimal beta function can only be maintained over a short distance, which is of the same order of magnitude as β_y itself. Therefore, if the bunch length is longer than β_y, the actual luminosity is smaller than the nominal value. This is known as the *hourglass effect* [38]. The bunch length is on the order of mm, considerably larger than β_y. However, since the beams collide at an angle ϕ_c, the effective bunch length for the collision is given by the length of the overlap region, as shown in fig. 3.2:

$$d = \frac{\sigma_x}{\sin\phi_c}.$$

With a horizontal beam size of $\sigma_x \approx 10\,\mu\text{m}$ and the crossing angle of $\phi_c = 83\,\text{mrad}$, the hourglass requirement of $d \leq \beta_y$ is fulfilled. Another reason for the rather large crossing angle is that is allows to separate the final focusing magnets of the two beams [37].

In addition to the decrease of the vertical beta function β_y, the beam currents are increased by a factor of 2 compared to KEKB. Writing the luminosity in terms of the beta function and the beam current, the following scaling behaviour is obtained [33]:

$$\mathcal{L} \propto \frac{I_\pm \xi_{y\pm}}{\beta_y}, \qquad \xi_{y\pm} \propto \frac{N_\mp \beta_y}{\sigma_y \sigma_x}.$$

The index +/− refers to the positron/electron beam. The parameter ξ_y is called the vertical *beam-beam parameter*. It describes the focusing force excerted on a bunch by the electromagnetic field of the opposite bunch [39]. For low beam currents, the beam-beam parameter increases with the number of particles in a bunch. However, for high beam currents, the emittance increases as the two beams disrupt each other, and the increasing particle number is compensated by the increasing beam size. At that

limit, the beam-beam parameter saturates and the luminosity depends only linearly on the beam currents. This leads to a total luminosity increase by a factor of 40 compared to KEKB, assuming that the same beam-beam parameter can be achieved. The first e^+e^- collisions at SuperKEKB are expected in March 2018.

3.2. The Belle II detector

The Belle II detector surrounds the interaction point and detects particles produced in the e^+e^- collision. These final particles are mainly electrons, positron, photons, muons, pions and kaons. The detector consists of several subdetectors, which are arranged in layers one within the other and measure the momentum, energy, charge and production vertex of the particles and determine the particle type.

Figure 3.3 shows an overview of the detector, as well as the Belle II coordinate system: the z-axis points roughly in the direction of the electron beam, the y-axis points upward and the x-axis points radially away from the center of the accelerator ring. The origin is at the interaction point. The z-axis forms the symmetry axis of the detector, so due to the crossing angle it is not perfectly aligned with either beam, but with the angle bisector of the two beams (compare fig. 3.2). The x-y-plane is referred to as the *transverse* plane. In spherical coordinates, the polar angle θ is measured from the z-axis, while the azimuth angle φ is measured in the transverse plane from the x-axis.

To account for the boost, the detector acceptance is asymmetric, with a coverage of about 17° to 150° in the polar angle θ (depending on the subdetector). In the center of mass system, this corresponds to a symmetric acceptance of about 23° to 157°. In the following, the individual components of the Belle II detector are described. The focus is set on the drift chamber, which provides the input for the track trigger.

3.2.1. Magnetic field

Except for the outermost subdetector, all detector components are within a homogenous magnetic field of 1.5 T, which is produced by a superconducting solenoid with an inner radius of 1.7 m and a length of 4.4 m. The magnetic flux is returned through the iron structure of the Belle II detector, which also serves as absorber for the K_L and μ detector (KLM, described in more detail later).

Charged particles experience a Lorentz force in the magnetic field and follow a curved trajectory. The magnetic field is aligned with the z-axis, so the Lorentz force acts on the transverse motion of the particle. More precisely, the particles follow a helix trajectory, whose radius is proportional to the transverse momentum $p_\perp$. The total momentum of the particle can be obtained from the trajectory by combining $p_\perp$ and the polar angle.

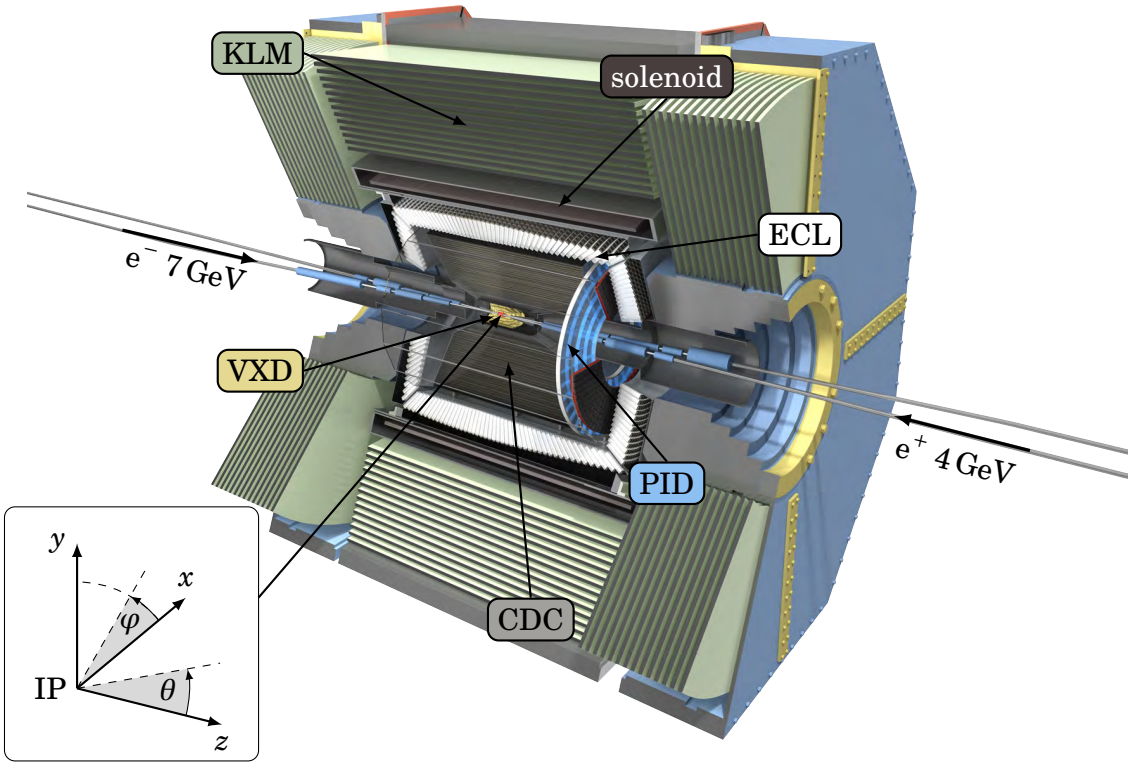

Figure 3.3.: Overview of the Belle II detector [40], with the vertex detector (VXD), the central drift chamber (CDC), the particle identification system (PID), the electromagnetic calorimeter (ECL), the solenoid, the K_L and μ detector (KLM) and a schematic of the Belle II coordinate system.

3.2.2. The vertex detector

The vertex detector (VXD) is the innermost subdetector, directly surrounding the interaction point. It is part of the tracking system and measures the trajectory of charged particles close to the interaction point with maximal precision. By extrapolation and matching of the tracks, the decay vertices of short-lived particles can be determined with an accuracy of about 50 μm [34]. As explained in section 2.4.3, this is essential to measure time dependent $\mathcal{CP}$ violation. For comparison, the average distance between the two B meson decay vertices is about 130 μm [34].

The vertex detector is designed completely new for Belle II. It consists of two layers of pixel detectors (PXD), followed by four layers of silicon strip detectors (SVD). The PXD layers are located at radii of 14 mm and 22 mm, while the four SVD layers are located between 38 mm and 140 mm from the z-axis. In contrast, the vertex detector in Belle consisted only of a three layer silicon strip detector, with the innermost layer at 30 mm [41]. The addition of the new pixel detector is possible because of the nano

beam scheme, which allows a much smaller beampipe radius. Measuring the trajectory closer to the interaction point reduces errors in the extrapolation of tracks to the decay vertex. Details of the new PXD technology can be found in [42].

Both the PXD and the SVD are based on silicon sensors. Charged particles crossing the sensor create electron-hole pairs, which drift towards different sides of the sensor. The charges are collected and read out to determine the position of the incident particle. The difference between the two detectors is the shape of the sensors. The SVD consists of double sided silicon detectors, with orthogonal strips on the top and bottom side of each detector layer. By combining the coordinates from both sides, a two dimensional hit is obtained. In contrast, the PXD consists of an array of pixels and can directly provide two dimensional coordinates.

The pixel detector is chosen for the region directly around the beampipe because of the high beam background, which increases with the inverse square of the radius [37]. If strip detectors were used for the innermost layers, the occupancy (≡ fraction of channels hit in an event) would be too high to reconstruct any vertices. The pixel sensors have a much larger number of channels and thus a much smaller occupancy. On the other hand, the PXD produces a much higher data rate. In fact, to reduce the amount of data for each event, only PXD hits from certain "regions of interest" are kept for the reconstruction. The regions of interest are determined by extrapolating tracks from the SVD to the PXD [43].

3.2.3. The Central Drift Chamber

The Central Drift Chamber (CDC) is the main tracking detector of Belle II and occupies a cylindrical volume with an inner radius of 16 cm and an outer radius of 113 cm. The main purpose of the CDC is to precisely measure the momenta of charged particles. In addition, it can provide particle identification by measuring the energy loss of particles, which depends on the particle velocity. This is of special importance for low momentum particles that do not reach the outer particle identification detectors. Finally, the CDC provides the sole input for the track trigger, since the vertex detector readout is too slow for the first trigger level.

The CDC is a wire chamber filled with a gas mixture of 50 % helium and 50 % ethane. It is larger than the drift chamber at Belle, but follows the same principal design. It contains an array of 14336 sense wires and 42240 field wires, with a high voltage applied between sense and field wires. The wires are arranged in radial layers of rectangular cells, with a sense wire in the center surrounded by eights field wires, as illustrated in fig. 3.4. Charged particles ionize the gas as they pass through the chamber. The free electrons drift towards the sense wires, where they ionize further gas atoms in the high electrical field close to the wire. When this electron avalanche reaches the sense wire, a measurable current is produced, which is registered as a hit.

The drift electrons are accelerated by the electrical field of the sense and field wires, but slowed by the resistance of the gas. The gas mixture and the wire configuration are

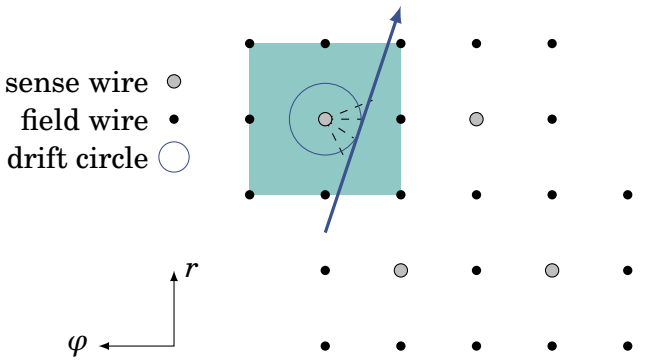

Figure 3.4.: A drift cell of the CDC is formed of one sense wire and eight field wires. The cells of adjacent layers are shifted with respect to each other. Charged particles ionize the gas and the free electrons drift towards the sense wire. The drift time restricts the hit position to a circle around the sense wire.

carefully chosen such that the drift velocity of the electrons is almost constant, with a nominal value of $40\,\mu\mathrm{m\,ns}^{-1}$. Therefore, the timing of the hit signal is approximately proportional to the distance between the particle trajectory and the sense wire. The readout of the timing is digitized to 1 ns for the data acquisition and 2 ns for the track trigger. Including the drift times, the spatial resolution of the CDC is 100 µm [37], whereas the distance between wires is about 2 cm.

In total, there are 56 layers of sense wires, which are arranged in 9 so-called superlayers. Within one superlayer, all layers have the same number of wires and the same wire orientation, which can be either *axial* or *stereo*. Axial wires are oriented parallel to the z-axis, so a hit on an axial wire provides coordinates in the transverse plane, while the longitudinal coordinate is completely unknown. Stereo wires are skewed with respect to the z-axis by shifting the azimuthal wire position at the forward endplate with respect to the position at the backward endplate. The different wire orientation is illustrated in fig. 3.5. By combining axial and stereo hits, a three dimensional trajectory can be reconstructed. Figure 3.6 shows the superlayer configuration with the corresponding wire numbers and stereo angles. Axial and stereo superlayers alternate, as well as the sign of the stereo angle. All superlayers contain 6 layers, except for the innermost superlayer, which contains 8 layers of smaller drift cells to cope with the higher background level. To reduce the occupancy, the cell size in the innermost superlayer is only 1 cm.

There are different counting schemes for the sense wires: by superlayer, layer within superlayer and wire within layer; by layer and wire within layer; or by a continuous ID. In any case, the counting within a layer starts at $\varphi = 0°$ at the backward endplate, where the readout electronics is located.

The CDC covers a polar angle range of 17° to 150°. However, in fig. 3.5 it can be seen that for very flat polar angles, particles do not pass all layers. A particle reaches

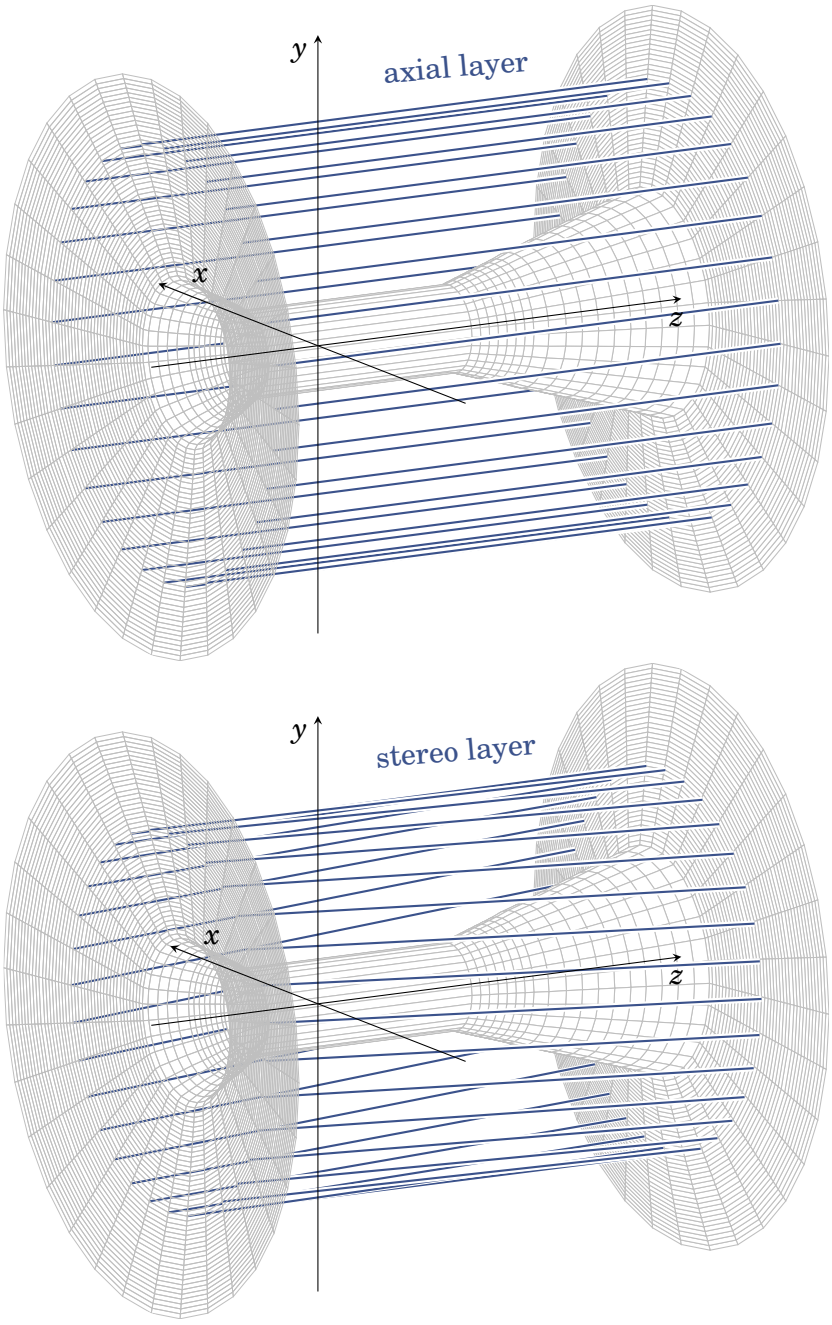

Figure 3.5.: Inner boundary of the CDC and example layers. Top: axial layer with wires parallel to the z-axis (every 10th wire shown). Bottom: stereo layer with skewed wires (every 9th wire shown).

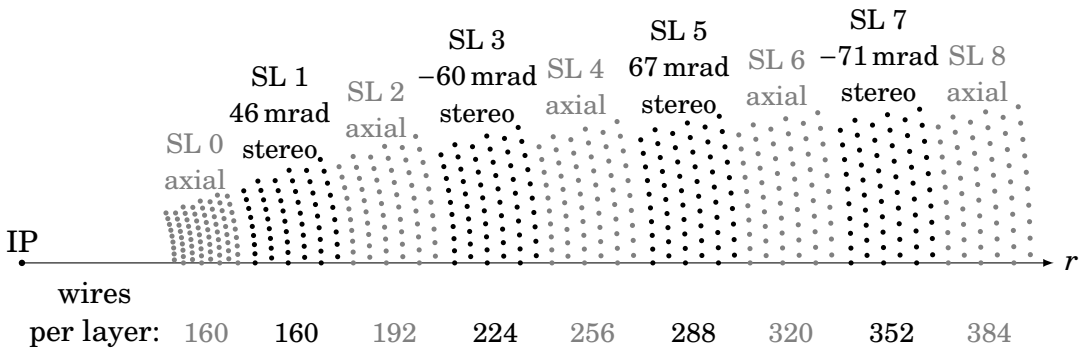

Figure 3.6.: Layer configuration of the CDC with 9 superlayers. The stereo angles within a superlayer vary by a few mrad; the given numbers are average values.

the 19th layer with $\theta = 17°$ and the 13th layer with $\theta = 150°$. The outermost layer covers a polar angle range of 35° to 123°.

3.2.4. Cherenkov detectors for particle identification

If a charged particle passes through a dielectric medium at a speed greater than the speed of light in that medium, it emits a cone of Cherenkov photons at an angle that depends on the particle velocity. Thus, the particle velocity can be measured independently from the momentum to obtain the mass of the particle. Belle II contains two independent Cherenkov detectors, whose main task is to discriminate charged kaons and pions. Both systems are newly developed for Belle II.

In the forward endcap region, an aerogel ring-imaging Cherenkov (ARICH) detector is installed, which covers the polar angle range of 17° to 35°. It consists of an aerogel radiator of 2 cm thickness, where Cherenkov photons are produced, and an array of position sensitive photon detectors at 20 cm distance from the aerogel radiator. The Cherenkov photons form a ring on the photon detectors, whose radius can be related to the velocity of the incident particle.

In the barrel region, the outer wall of the CDC is surrounded by 16 time-of-propagation (TOP) counters, which consist of rectangular quartz radiator bars. The TOP counters cover a polar angle range of 32° to 120°. Cherenkov photons emitted in the radiator are transported to the backward end of the bar by total internal reflection, where they are measured by photon detectors with a timing resolution of 40 ps. The path length of the photons from the crossing point of the incident particle to the photon detectors depends on the Cherenkov angle. Therefore, the particle velocity can be related to the arrival time of the Cherenkov photons. For kaons and pions with momenta of 2 GeV/c, a propagation time difference of $\mathcal{O}(100\,\text{ps})$ is obtained [44]. With a thickness of only 2 cm, the TOP provides a very compact measurement of the

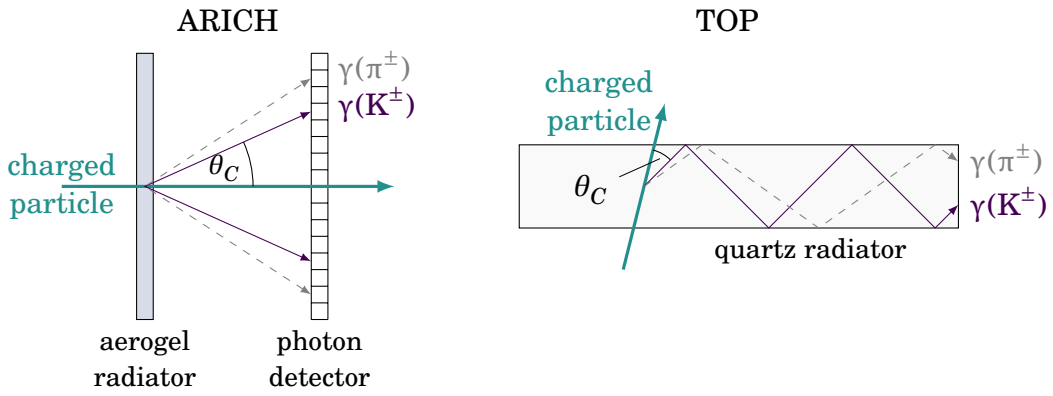

Figure 3.7.: Working principle of the two Cherenkov detectors. In the ARICH, the Cherenkov angle θ_C is determined from the radius of the rings on the photon detector. In the TOP, the Cherenkov angle θ_C is related to the time of propagation of the Cherenkov photon.

particle velocity.

The working principle of the TOP and ARICH is illustrated in fig. 3.7.

3.2.5. The electromagnetic calorimeter

The main task of the electromagnetic calorimeter (ECL) is to detect photons, which are not visible in the tracking detectors, and to measure their energy and position. Photons are produced for example as the decay products of neutral particles such as π^0, which constitute one third of the decay products from a typical B decay [37]. In addition, the ECL is used to identify electrons by comparing the momentum and energy deposition of particles, to measure the luminosity and to detect K_L together with the KLM. Finally, the ECL trigger is an important part of the first level trigger.

The ECL consists of 8736 Tl-doped CsI scintillation crystals with a cross section of about $6 \times 6\,\mathrm{cm}^2$ and a length of 30 cm. In the crystals, photons are converted to electron-positron pairs, which in turn emit bremsstrahlung photons. This process is repeated, causing a particle shower, until the energy of the shower particles falls below a certain threshold. Low energetic electrons and positrons in the shower produce scintillation light, which is measured at the end of each crystal and is proportional to the deposited energy. Electrons and positrons can be identified by matching tracks with calorimeter showers. Showers that cannot be matched to any track correspond to neutral particles. Photons can be distinguished from hadronic neutral showers by the size of the shower.

The ECL is divided into a barrel section and two endcap sections. In total, it covers a polar angle region of 12° to 155°, with two gaps of 1° between the barrel and the endcaps. The calorimeter from Belle is mostly reused, although the readout electronics

was upgraded [45].

3.2.6. Kaon and muon detector

The purpose of the K_L and μ detector (KLM) is to identify muons and long-lived kaons. It is positioned outside of the solenoid coil and consists of alternating layers of 4.7 cm thick iron plates and active detector elements. Kaons produce hadronic showers both in the ECL and in the iron plates of the KLM. Muons do not create showers, but are visible as tracks in the active detector elements. As mentioned earlier, the iron plates also serve to return the magnetic flux of the solenoid.

Muons are characterized by a track in the CDC, which can be associated with matching hits in the KLM. To discriminate muons from charged hadrons, the predicted and actual range are compared: a muon should show up in all layers that the extrapolated track can reach.

A K_L is identified by a cluster of hits in the KLM that cannot be matched with a charged track. If the KLM cluster can be matched with a shower in the ECL, the two are combined and the cluster direction is given by the more precise measurement from the ECL.

3.3. Physics processes of interest

The most important goal of Belle II is the precise measurement of $\mathcal{CP}$ violation in the B meson system, as explained in chapter 2. Depending on the studied decay channel, the observed $\mathcal{CP}$ asymmetry can be related to different parameters of the CKM matrix. In addition to B decays, the Belle II physics program includes for example searches for rare τ decays, charm physics and, most importantly, searches for "exotic" channels beyond the standard model, such as dark matter candidates and reactions involving supersymmetric mediators [46].

Two types of processes have been selected for the studies in this thesis. The first selected process are lepton flavor violating τ decays, which produce typically only two charged tracks and are thus an extreme example for a low multiplicity signal. The second are "invisible" B^0 decays, where the signal side B^0 meson produces neither charged particles nor photons and is thus invisible for the Belle II detector. Both processes belong to the searches for rare decays, which are strongly suppressed in the standard model, but may be enhanced in certain theories beyond the standard model.

3.3.1. Lepton flavor violation

In the standard model, separate additive lepton numbers L_ℓ ($\ell \in \{e, \mu, \tau\}$) are associated with each generation of leptons, which are conserved in all interactions. In a standard model τ decay, such as the one shown in fig. 3.8a, the lepton number L_τ is

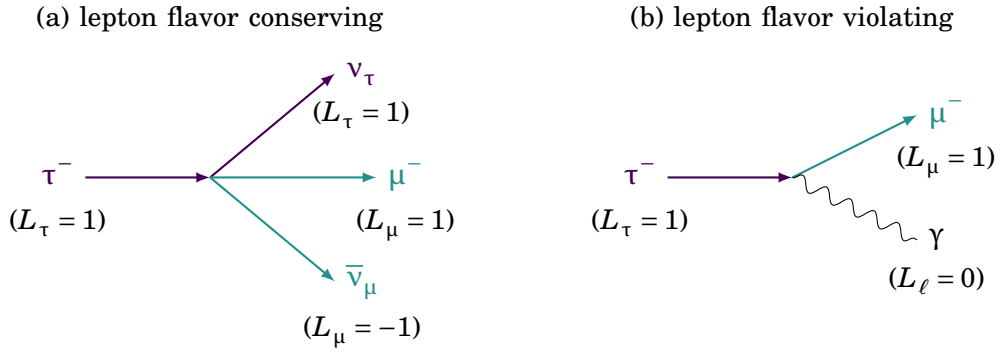

Figure 3.8.: Examples for a lepton flavor conserving and a lepton flavor violating process.

transfered to a ν_τ neutrino, while an electron or muon is produced together with the corresponding antineutrino, such that the lepton numbers add up to $L_{\mu/e} = 0$.

Lepton flavor violation has already been observed in the neutrino sector in form of neutrino oscillations [47, 48]. In consequence, the standard model has to be modified by introducing neutrino masses. Then, the neutrino mass eigenstates can be related to the neutrino flavor eigenstates by a mixing matrix, which is analogous to the CKM matrix and known as *Pontecorvo-Maki-Nakagawa-Sakata* (PMNS) matrix [49, 50]. However, neutrino oscillations do not lead to measurable lepton flavor violating phenomena in the charged lepton sector, due to the tiny neutrino masses. Processes such as the one shown in fig. 3.8b are suppressed by the fourth power of (m_ν/m_W), where m_ν is the neutrino mass and m_W is the mass of the W boson [46]. Therefore, searches for lepton flavor violation in charged lepton decays constitute a direct search for physics beyond the standard model.

Naturally, searches for rare decays require very high statistics. The cross section for $\tau^+\tau^-$ production at the $\Upsilon(4S)$ resonance is 0.9 nb [34], which is of the same order of magnitude as the $B\overline{B}$ cross section, so τ pairs are also produced in very large numbers. In addition, the unambiguous identification of a lepton flavor violating decay is only possible with an e^+e^- collider, where the initial state is precisely known. Therefore, Belle II is ideally suited to carry out such searches.

There are several extensions of the standard model that predict charged lepton flavor violating processes, such as supersymmetry, little Higgs models and extra dimensions (see for example [46] for a list of references). Lepton flavor violation can be introduced for example by the mixing of heavy leptons, or by new scalar particles with lepton flavor violating couplings [23]. A list of different models is given in tab. 5.19 of [46]. Depending on the specific model, different branching fractions are predicted. Thus, if lepton flavor violating decays are indeed found, the ratios between the branching fractions of different decay modes can point towards the underlying physics and discriminate some of the proposed theories. On the other hand, if no such decays are

found, Belle II will improve the experimental limits on the branching ratios, which helps to confine the parameter space for any theories beyond the standard model.

The current experimental limit on the branching fraction of the lepton flavor violating muon decay $\mu \to e\gamma$ is 4.2×10^{-13} [11]. For tau decays, the experimental limits are several orders of magnitudes higher, with 4.4×10^{-8} for $\tau \to \mu\gamma$ [11] and 3.3×10^{-8} for $\tau \to e\gamma$ [11]. The experimental limits for other τ decay channels are of similar magnitude. Also in most theoretical models, the τ branching ratios are expected to be higher than those of the corresponding muon decays, due to the higher mass of the τ. Therefore, the efficiency studies in this thesis focus on τ events, with the decay channel $\tau \to \mu\gamma$ (shown in fig. 3.8b) as an explicit example.

3.3.2. Invisible B decay channels

Invisible final states consist solely of neutral particles that do not interact with the electromagnetic calorimeter and the KLM, that is they leave no visible signal in the detector. Such invisible particles can be either neutrinos or some unknown new neutral particles. The invisible standard model decay $B^0 \to \nu\bar{\nu}$ is strongly helicity suppressed: since the B^0 has spin 0, the two neutrinos need to have the same helicity, so there is either a right-handed neutrino or a left-handed antineutrino in the final state. This decay would be forbidden for massless neutrinos. Assuming small but finite neutrino masses, the decay is suppressed by a factor of the order $(m_\nu/m_B)^2$ [51], where m_ν is the neutrino mass and m_B is the mass of the B^0. Thus, an observation of an invisible B^0 decay is a clear indication for new physics. The current experimental limit on the branching fraction of B^0 decays to invisible final states was set by the BaBar experiment to 2.4×10^{-5} [11, 51]. Belle II will be able to improve this limit by two orders of magnitude.

The search for invisible final states is based on a full reconstruction of the other B meson. The trigger signal also has to be derived solely from the tag side B meson. Thus, the trigger efficiency for invisible B decays gives also a lower limit for other signal channels which are hard to detect at the first level trigger. For example, final states with few charged particles and one or more neutrinos, such as $B \to \tau\nu$, may produce only low momentum tracks, which are below the detection threshold of the trigger.

3.4. Background processes

Compared to KEKB, the background level at SuperKEKB is expected to increase dramatically. This is an unavoidable consequence of the increased luminosity. Two types of background can be distinguished: machine background, which occurs due to particle loss in the vicinity of the detector, and background from the e^+e^- collisions, which scales directly with the luminosity.

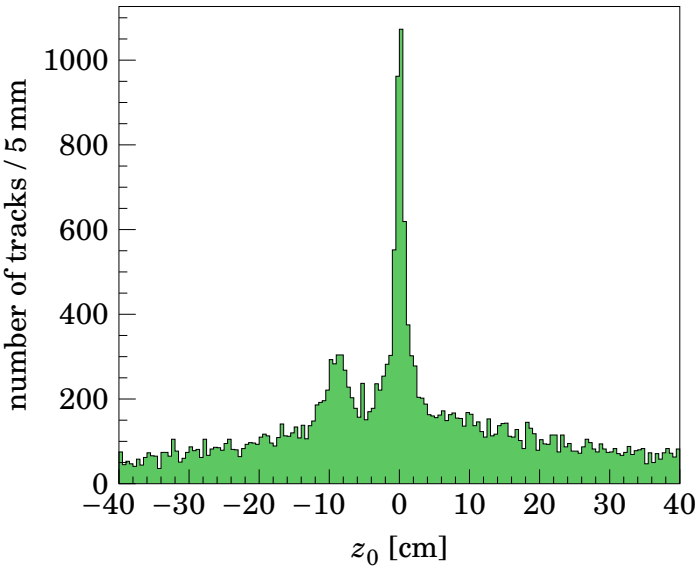

Figure 3.9.: Distribution of z-vertices in random triggered events in Belle [37]. The peak at $z_0 = 0\,\text{cm}$ corresponds to tracks from e^+e^- collisions. The broad distribution beneath is caused by machine background.

3.4.1. Machine background

The main sources of machine background are *Touschek scattering* [52] and *beam-gas scattering*.

Touschek background is caused by Coulomb scattering of particles within the same bunch. When the scattering process transfers transverse momentum to longitudinal momentum, the momentum change is amplified by the Lorentz factor. Then one scattered particle has too much energy with respect to the design beam energy, while the other has too little. The Touschek scattering rate is proportional to the inverse beam size and to E^{-3}, where E is the beam energy [53]. The dependence on the beam size means that the Touschek scattering around the interaction point will strongly increase compared to KEKB, due to the nano beam scheme. The energy dependence means that the Touschek background is dominated by the LER. In fact, reducing the Touschek scattering is another reason for increasing the positron beam energy.

Beam-gas background is caused by scattering of electrons or positrons on the nuclei of residual gas molecules in the beam pipe. The scattering can be either elastic or inelastic, when bremsstrahlung is emitted during the scattering process. In either case, the particles are deflected away from the design orbit. The beam-gas scattering rate depends on the beam current and on the vacuum pressure in the beampipe. The vacuum level around the interaction point is expected to be 100 to 1000 times higher than at KEKB [37]. This is a consequence of the much smaller beampipe radius of only 1 cm.

Both Touschek scattering and beam-gas scattering lead to changes of the energy and momentum of the scattered particles, which are then no longer on a stable trajectory along the storage ring. At some point, the scattered particles hit the beampipe and are lost. On the one hand, particle loss limits the lifetime of the beams, which are refilled continuously to maintain the design beam currents. On the other hand, if particles are lost within a few meters around the interaction point, they can scatter into the detector and cause background tracks. These tracks can be identified by their longitudinal vertex position (z-vertex), which is typically not at the interaction point, but somewhere along the beam line. Figure 3.9 shows the distribution of the z-vertices of reconstructed tracks in random triggered events at Belle. Beneath the peak from the e^+e^- collision, there is a broad distribution of tracks caused by the machine background. Since this background contribution is expected to increase significantly in Belle II, the goal is to reject these tracks at the first trigger level.

3.4.2. Luminosity background

The e^+e^- cross section is dominated by $e^+e^- \to e^+e^-$ scattering, which is known as *Bhabha scattering*. Two diagrams contribute to this process in leading order, which are shown in fig. 3.10: the s-channel, which appears also in annihilation processes, and the t-channel, which appears in scattering processes. While the s-channel term is equivalent to lepton pair production of $\mu^+\mu^-$ or $\tau^+\tau^-$, the t-channel term strongly increases for flat polar angles and dominates the cross section. Since Bhabha scattering is a very well understood QED process and easy to identify, the Bhabha scattering rate is used to measure the luminosity. However, due to the large cross section, Bhabha events need to be prescaled by a factor of ≈ 100 [37].

In addition to identifiable Bhabha events, Bhabha scattering can cause secondary background tracks in the detector. For very flat scattering angles, the electron and positron do not directly cross the acceptance region of the detector, but they can hit the

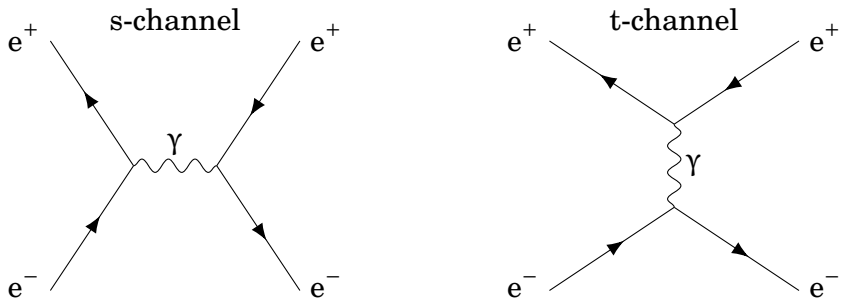

Figure 3.10.: Leading order diagrams for Bhabha scattering. With an additional photon in the final state, the process is refered to as radiative Bhabha scattering.

detector structure and cause secondary particles, which back-scatter into the detector. These secondary tracks are similar to the machine background in that they do not originate at the interaction point, and should be likewise rejected. Similarly, photons from radiative Bhabha scattering can produce neutrons when they interact with the material of the magnet or the beampipe. Although neutrons do not produce visible tracks, they can damage the detector and produce background hits in the ECL and KLM.

Another luminosity dependent QED background is the two photon mediated process $e^+e^- \to e^+e^-e^+e^-$. Within the polar acceptance of the detector, the cross section for this process is even higher than for Bhabha scattering, but it produces mostly very low momentum e^+e^- pairs. Therefore, it is only relevant for the vertex detector, where it is the dominant background [54].

3.4.3. Background mixing

Generally, two types of background have to be distinguished, independent of their source: background tracks and background hits. Background tracks are caused by particles from background processes, which go through the detector and look similar to tracks from interesting physics processes. The purpose of the present thesis is to identify and reject such background tracks.

In contrast, background hits are isolated hits, which are not part of a full track. They can be caused by particles that pass only a part of the detector, for example very low momentum secondary particles produced within the CDC. Isolated background hits are much more frequent than full background tracks, so they are typically overlaid with physics or background events as random noise and complicate the reconstruction. Note that with this definition hits which are caused by a full, reconstructable background track are not referred to as “background hits”.

Background hits can be simulated by adding random hits to an event, according to the expected background rate and distribution. For the present studies, these distributions are taken from simulations of the machine background [55]. For consistency, the same background samples are used throughout this thesis, although new background campaigns are generated regularly. The simulated background hits are added before the digitizer, which simulates the response of the CDC readout electronics. This concept is referred to as *background mixing*. When the experiment is running, the background hits from random triggered events can be overlaid with simulated physics events.

4. The Trigger System

The Belle II trigger consists of two levels. The first level trigger performs a partial online event reconstruction in order to provide a trigger signal for the data acquisition system when the trigger criteria are satisfied. All algorithms are pipelined and implemented on configurable hardware (FPGAs) to allow a dead time free system with a total latency of 5 µs and a maximum average trigger rate of 30 kHz [37]. The high level trigger is implemented in software and performs an event selection on physics level to further reduce the event rate that is written to disk.

The following chapter gives an introduction to the main components of the first level trigger. The trigger system is adopted from Belle and consists of several subtrigger systems, which process signals from different detector components. The summary of each subtrigger is passed to the *Global Decision Logic* (GDL), which combines all components and makes the final trigger decision. The two main subtriggers are the track trigger, which provides charged track information based on signals from the CDC, and the ECL trigger, which gives information about the energy deposition in the calorimeter. In addition, the TOP trigger provides precise timing information and the KLM trigger gives muon track information. The vertex detector readout is too slow to provide input for the first level trigger.

4.1. The track trigger

The logical sequence of the track trigger is shown in fig. 4.1. It is the most complicated subtrigger system and has the longest latency, barely meeting the requirement of a total trigger latency of 5 µs.

The first step in the track trigger is the so-called *Track Segment Finder* (TSF), which combines hits within the same superlayer to track segments in order to reduce the input for the following modules. Then, the track finder combines hits from the axial superlayers to tracks in the transverse plane. In parallel, the event time is determined, which allows to calculate drift times and thus gain precise spatial information from the hits. This precision is necessary for the following 3D reconstruction of the tracks. Two independent approaches are applied in parallel. The standard approach pursued by the trigger group is based on a linear least squares fit. The method presented in this thesis uses neural networks, which can learn to approximate also nonlinear models.

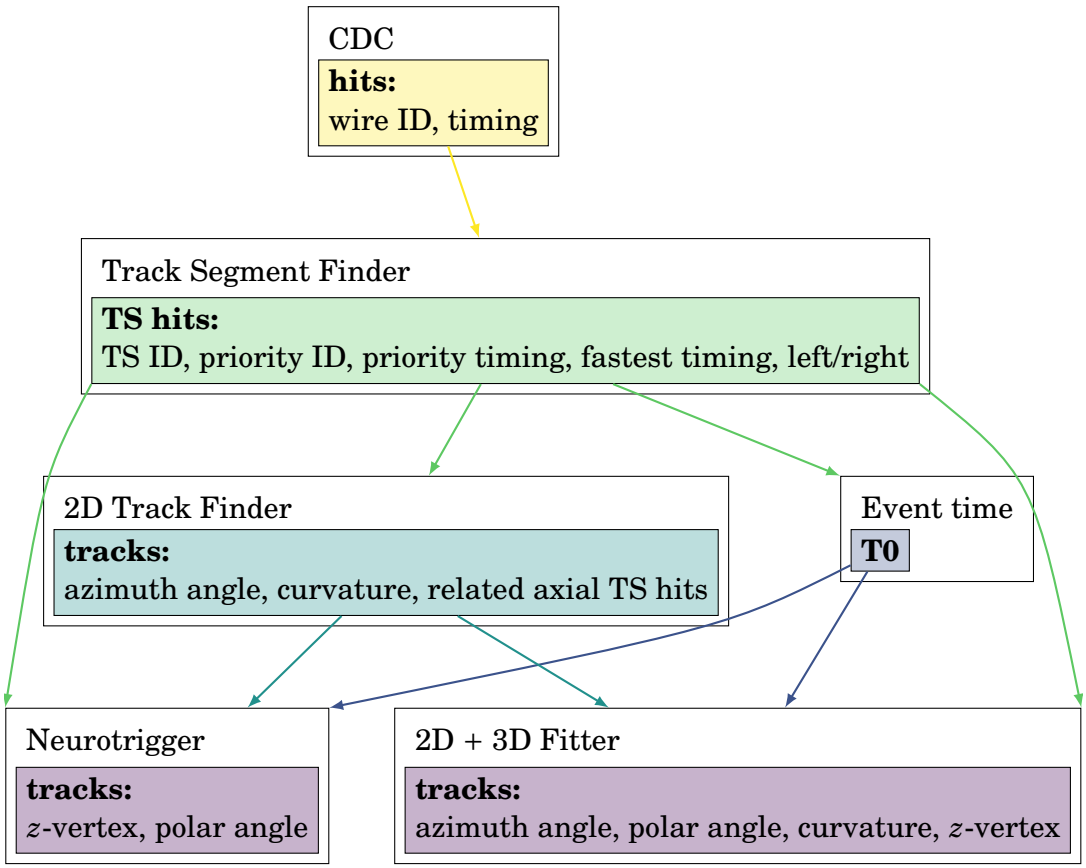

Figure 4.1.: Dataflow in the track trigger. The sequence of modules is shown, as well as the input and output objects of each module.

4.1.1. The Track Segment Finder

Track segments are defined as a particular arrangement of wires from 5 layers within a superlayer. The track segment shape is shown in fig. 4.2. It includes the definition of three priority wires, called *first priority* and *left / right second priority* wires, which are used as reference points in the track segment. The track segment ID is derived from the wire number of the first priority wire and the superlayer number.

When there are hits in at least four different layers of the track segment during a certain time window, a hit is produced. For the following modules, two types of summary information are created. The event time module receives only the *fastest timing* (≡ shortest drift time) of all hits in the track segment. The tracking modules (track finder and 3D reconstruction modules) receive instead the *priority timing*, which is the timing of one of the priority wires. The first priority wire is preferred if it is hit (hence the name), otherwise one of the second priority wires is chosen. If both second priority wires are hit and the first priority wire is not, the hit with faster timing is selected. To denote which wire the priority timing belongs to, a 2 bit *priority ID* is sent

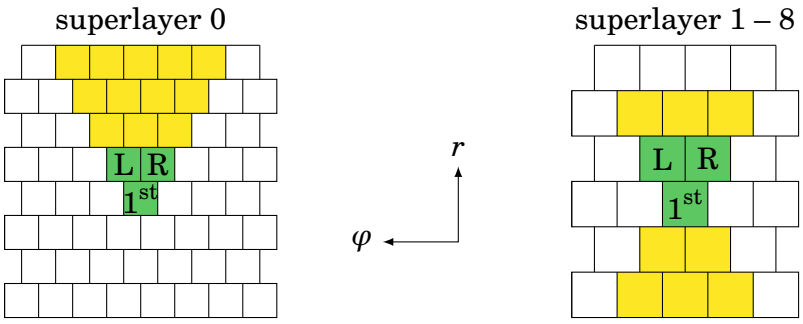

Figure 4.2.: Left: a track segment in the innermost superlayer consists of 15 wires in a triangle configuration. The three innermost layers are not used because of the high background level. Right: a track segment in the outer superlayers consists of 11 wires in an hourglass shape configuration. The outermost layer within each superlayer is not used. For each track segment three priority wires are defined (1^{st} priority, second priority Left and Right).

together with the priority timing. Note that the requirement for a track segment hit guarantees that at least one of the priority wires is hit.

In addition to the timing, the track reconstruction modules receive a 2 bit *left / right state* which denotes on which side of the wire the track passed. The left/right state is determined from the hit pattern in the track segment and can be either left, right or undecided. Undecided patterns include both realistic but ambiguous hit patterns, where typically the track is very close to the wire, and unlikely hit patterns that can only occur from the crossing of several tracks or background noise. Figure 4.3 shows some examples of hit patterns and their respective left/right state.

The mapping from hit pattern to left/right state is determined from simulated tracks. For each track segment hit, the hit pattern and the true left/right state are determined. Then, for each pattern the number of hits with true left passage n_L and the number

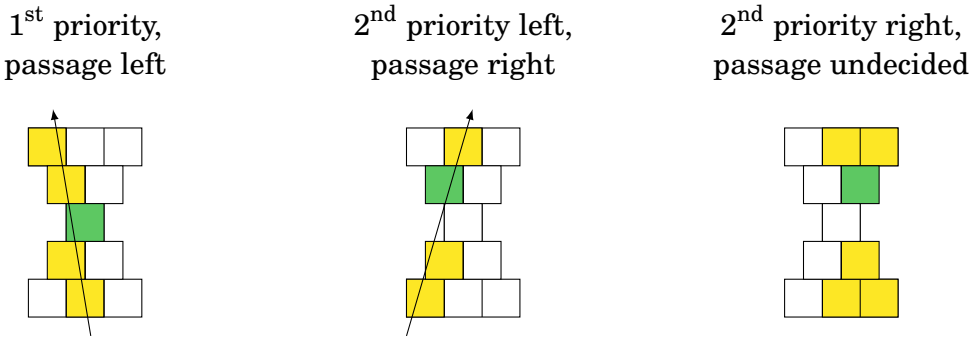

Figure 4.3.: Examples of track segment hits with different priority ID and left/right state. The priority cell is shown in green.

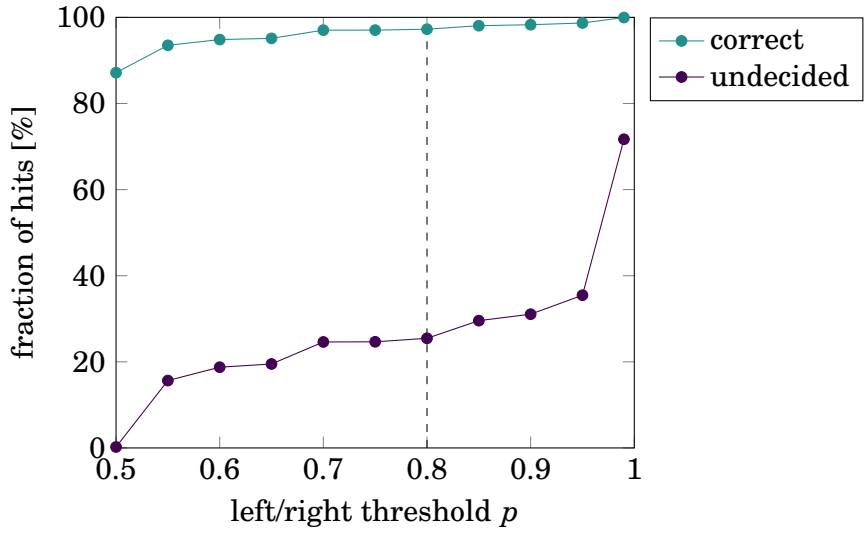

Figure 4.4.: Fraction of hits with correct left/right state (normalized to the number of hits with decided left/right state) and fraction of hits with undecided left/right state (normalized to the total number of hits), depending on the threshold p from eq. (4.1), which is used to define the left/right state for a pattern. The dashed line shows the default threshold.

of hits with true right passage n_R are counted. To decide the left/right state for the pattern, the following condition is checked:

$$\text{left/right state} = \begin{cases} \text{left} & \text{if } n_L > p \cdot (n_L + n_R) + 3\sigma \\ \text{right} & \text{if } n_R > p \cdot (n_L + n_R) + 3\sigma \\ \text{undecided} & \text{otherwise}, \end{cases} \tag{4.1}$$

where $\sigma = \sqrt{(n_L + n_R) \cdot p \cdot (1-p)}$ is the width of a binomial distribution with probability p. The threshold p is set somewhat arbitrarily to 0.8 in the default lookup table provided by the trigger group. The effect of the 3σ term is to set a threshold for the minimal number of hits that are required in the training to determine the left/right state for a pattern. In other words, the pattern is undecided either if the ratio of true left/right for a given pattern is below the decision threshold, that is if the pattern is ambiguous, or if the total number of hits with this pattern is too low to make a statistically significant decision. Figure 4.4 shows the fractions of hits with correct left/right state and undecided left/right state for different threshold values p. The selected value of $p = 0.8$ gives a reasonable trade-off between the correct and undecided fractions, although $p \approx 0.7$ might also be a good choice. Note that since the default lookup table was generated without considering background, the fractions shown in fig. 4.4 were also obtained from a simulation without background. More detailed studies including background will be done in section 7.5.2.

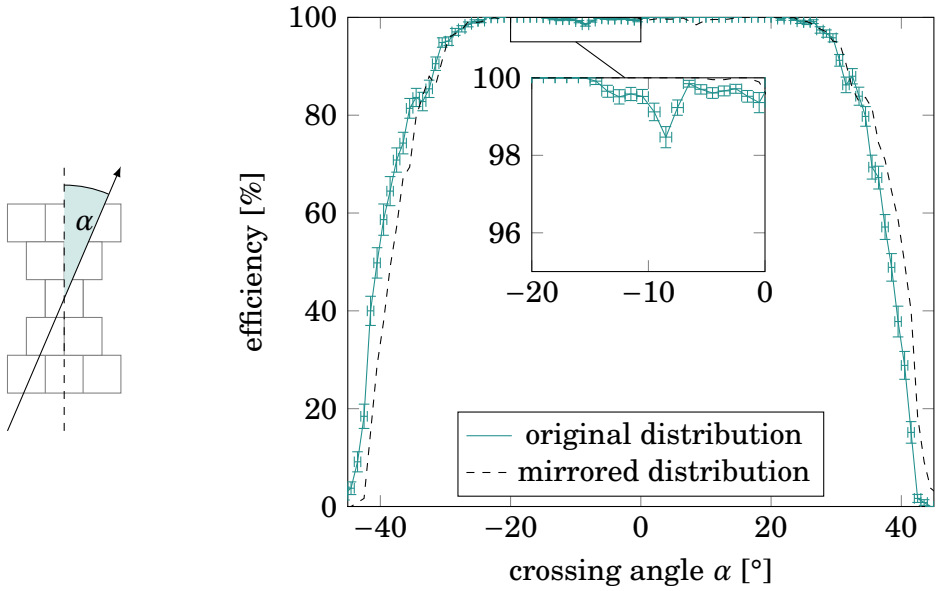

Figure 4.5.: Track segment finder efficiency depending on the crossing angle α of the track, which is defined in the sketch on the left. There is a slight asymmetry, which can be seen by comparing the distribution to a mirrored version, and a dip around a crossing angle of $-10°$.

Using track segment hits instead of single wire hits has several ramifications for the subsequent tracking. First, the information contained in the hits is compressed, so the tracking has to handle less hits while still having access to local information such as the left/right state. On the other hand, the low number of hits limits the accuracy that can be achieved in the track reconstruction. With the layer criterion, the track segment finder also reduces noise, which is essential for track reconstruction with only a few hits. However, it also limits the parameter space for particles that can be tracked. The special configuration of the track segments is only sensitive to tracks with a crossing angle no larger than $\approx 45°$. This reduces the acceptance for low momentum tracks.

Figure 4.5 shows the actual acceptance, which is even smaller than the theoretical limit, with an efficiency of $\approx 90\,\%$ reached at a crossing angle of $\pm 30°$. This is a consequence of the time window imposed in the track segment finder, and of nonlinearities in the drift velocity, which cause hits with long drift times to drop out of the time window. For small crossing angles, tracks pass all five layers of the track segment, so a hit is found even if one layer is missing. For larger crossing angles, only four layers within the track segment are hit geometrically, so the track segment hit is lost if one hit is out of the time window. In addition, the acceptance is slightly asymmetric, with a small dip in the efficiency around a crossing angle of $\approx -10°$. These effects are also caused by the nonlinearities in the drift velocity, which are related to the path of the

ionized gas molecules in the electrical field from the wires and the magnetic field from the solenoid, and depend themselves on the crossing angle of the track.

To summarize, the compression of hit information in the track segment finder is a necessary step in the first level trigger, which could not handle the full hit information in the assigned latency. However, the low number of hits is also a challenge for the subsequent track reconstruction, especially in extreme regions of the detector.

4.1.2. Event time estimation

The event time is approximated as the arrival time of the first hit in an event. This is a reasonable estimate if all hits are caused by the event, with only a very small delay by the shortest drift time. However, the drift time distribution of hits caused by the event is overlayed with background hits, which can occur at any time before or after the event. A real event is distinguished from random background noise by the sudden appearance of many hits within a short time window.

Therefore, the event time is determined from a histogram of the fastest timings of all track segment hits. The first bin that exceeds a certain threshold is identified as the event time. The event time is harder to determine for low multiplicity events, which cause comparatively few track segment hits. If the threshold on the histogram bins is set too high, the event time may not be found at all for such events. On the other hand, if the threshold is too low, the probability to get a fake event time from background hits increases.

Another option is to consider only those hits that can be related to a track. Since the track finder does not use the event time, such a selection is possible for the axial track segment hits after the track finding. Using only hits related to a track increases the signal to noise ratio, but also reduces the total number of hits. Moreover, it is not possible for the stereo hits.

Studies about the optimal event time algorithm are still ongoing. Therefore, in most of the studies in this thesis, the true event time is used instead. To estimate the influence of errors in the event time, section 7.5.3 contains a study where a random event time jitter of varying magnitude is applied.

4.1.3. Track reconstruction

In general, the track of a charged particle in a homogeneous magnetic field has the shape of a helix, which is aligned with the magnetic field axis:

$$\begin{pmatrix} x(\mu) \\ y(\mu) \\ z(\mu) \end{pmatrix} = \begin{pmatrix} r \cdot \left(\sin\left(\frac{\mu}{r} - \varphi_0\right) + \sin\varphi_0\right) + x_0 \\ r \cdot \left(\cos\left(\frac{\mu}{r} - \varphi_0\right) - \cos\varphi_0\right) + y_0 \\ \cot\theta_0 \cdot \mu + z_0 \end{pmatrix}, \tag{4.2}$$

where μ is the so-called helix parameter, (x_0, y_0, z_0) is a reference point on the helix, φ_0 and θ_0 are the azimuth and polar angle of the momentum at the reference point and r is the radius of the helix. As will be shown later, the helix parameter corresponds to the arclength of the transverse track projection from the reference point to a general point on the helix and is typically in the interval $[0, \pi r]$. The radius is related to the transverse momentum $p_\perp$ and the magnetic field strength B:

$$r[\mathrm{m}] = \frac{p_\perp[\mathrm{GeV}/c]}{0.3 \cdot B[\mathrm{T}]}.$$

In the first level trigger, it is assumed that tracks originate from the beamline, such that $x_0 = y_0 = 0$, which is a valid assumption within the precision of the track trigger. Note also that the assumption of constant $p_\perp$ is not exactly true if particles loose energy in the CDC, but again it is valid within the precision of the trigger.

The track reconstruction is described in detail in the following chapters. Chapter 5 introduces the track finding algorithm, which uses a Hough transformation to combine the axial track segment hits to circles in the transverse plane. From the circles, the helix parameters φ_0 and r are obtained. Chapter 6 explains the basics of a three dimensional track reconstruction, which estimates θ_0 and z_0. The formulas derived there are the basis both for the least squares fit and for the input of the neural networks, which are introduced in chapter 7. Since the 3D resolution depends strongly on the 2D track parameters, the least squares fit is split into two steps. First, the 2D parameter estimates from the track finding are improved by a circle fit that takes the drift times into account. Then, the 3D fit is performed. For the neural network method, these steps are combined, so no additional 2D fit is required.

The final results of the track reconstruction for a given event are the number of tracks and an estimate for the helix parameters $(r, \varphi_0, \theta_0, z_0)$ for each track. Different ways to combine the estimates from the various track trigger modules are discussed in chapter 8.

4.2. The calorimeter trigger

The track information is complemented by the energy deposition in the ECL, which can detect neutral particles and identify electrons. Since the acceptance of the ECL is larger than the acceptance of the CDC, the ECL trigger can also find particles outside of the acceptance of the track trigger.

The ECL trigger identifies isolated energy clusters and determines their position and energy. The final trigger decision can take into account both the number of clusters above a given energy threshold and the total energy deposited in the calorimeter. The opening angle between two clusters can be determined from the cluster positions.

In addition, the ECL clusters can be matched with tracks from the track trigger. This allows to identify electrons by comparing the track momentum and the cluster

energy. Clusters that cannot be matched are identified as neutral clusters. In the following, both the cluster finding and the matching algorithm are described.

4.2.1. Clusters

The ECL trigger operates on so-called *trigger cells*, which consist of 4×4 crystals. Similar to the track segment concept in the track trigger, this reduction suppresses noise and reduces the latency. The signals of the sixteen crystals are merged by a fast shaper circuit to an analog signal, which is then digitized. For each trigger cell, the pulse height and the timing at the peak are determined [37]. In total, there are 576 trigger cells in 17 rings (3 in the forward region, 12 in the barrel region and 2 in the backward region), with the number of cells in each ring depending on the region.

Adjacent trigger cells are combined to clusters, starting from a seed that satisfies the ICN (*Isolated Cluster Number*) logic depicted in fig. 4.6. The logic is equivalent to searching for an isolated top left corner in the grid of trigger cells and was used also in Belle for counting the number of clusters, then without further analyzing the cluster. Within the 3×3 trigger cells surrounding the seed, the cell with the highest energy is selected. The cluster is defined as the 3×3 rectangle around this highest energy cell [57]. The cluster energy is given by the sum over all trigger cells in the cluster. The timing and position of the cluster are determined as an energy weighted average over the individual trigger cells.

4.2.2. Matching between tracks and clusters

Tracks and clusters can be matched based on their transverse and longitudinal distance from each other. This is done by the *General Reconstruction Logic* (GRL), which takes the results of both the track trigger and the ECL trigger and combines them. Then, the matched objects are sent to the GDL for the final decision. To match tracks and clusters, each track is extrapolated to the radius of the ECL. Then, the cluster with the shortest transverse distance Δr to the extrapolated point is searched. The

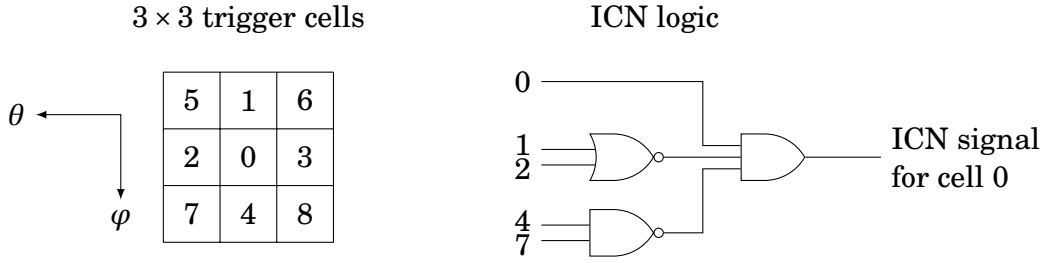

Figure 4.6.: ECL clusters are formed around seeds (cell 0) that satisfy the ICN logic, which combines signals from trigger cells within a 3×3 rectangle (diagram redrawn after [56]).

cluster is matched to the track, if both Δr and the longitudinal distance Δz are below a given threshold.

The extrapolation will be calculated explicitly in section 8.1.2, after the details of the track reconstruction have been described.

4.3. Global decision logic

The GDL has two important tasks. The first is to decide whether a trigger signal should be sent. This decision can be based for example on the number of tracks, the number of clusters or the total energy in the calorimeter. Several independent trigger logics will be defined and combined by an *or*-logic, that is at least one of them must be fulfilled to send a trigger signal. In addition to the trigger logics, different veto logics can be defined to reject background events. For example, a trigger with low track multiplicity should be combined with a z-vertex veto to reject machine background. Finally, individual trigger logics can be *prescaled* by a prescale factor N, which means that for every N events that fulfill this particular trigger logic, only one is actually triggered. An example for prescaled triggers are Bhabha events, which are generally vetoed, but necessary for calibration purposes and luminosity determination.

The second important task of the trigger is to determine the event timing. The best precision is expected from the TOP trigger. Outside of the TOP acceptance, the event timing is provided by the ECL trigger. The trigger signal is sent after a fixed delay upon the arrival of the event timing signal from the TOP trigger or the ECL trigger [37]. If no event timing is determined, the trigger signal is sent after a fixed delay from the time when the trigger was generated, leading to a much larger trigger jitter.

The final setup of the GDL is not yet defined. Plans in the Belle II technical design report [37] include a three-track trigger, a total energy trigger, a four-cluster trigger and a two-track trigger. More recent studies focus on low multiplicity triggers in combination with Bhabha vetos and on the number of clusters above different energy thresholds [58]. Different track triggers, z-vertex vetos and Bhabha vetos will be studied in chapter 8.

4.4. Trigger simulation

As mentioned before, the first level trigger is implemented on dedicated hardware to obtain a trigger decision in the required latency. However, the trigger is also simulated in the Belle II analysis software framework (*basf2*) [59], to allow studies of the trigger performance. The studies in this thesis are based on the so-called *fast simulation*, which gives a bit-precise simulation of the core logic of the first level trigger. In contrast, the *firmware simulation* aims to simulate also the datastream between the

different trigger boards, as well as the pipelined clock-by-clock simulation, and is still being developed.

The fast simulation of the track trigger was finalized during this thesis. For the track segment finder, the event time finder and the 2D and 3D fitter, basf2 modules were developed that run the simulation provided by the trigger group. In particular, the new modules make the results available outside of the trigger simulation to allow evaluation studies. For the track finding and the neural network based trigger, completely new modules were developed. The track finding module is based loosely on a module developed in [60] for the data reduction system of the PXD, which uses a similar algorithm to find tracks in the SVD. Two independent modules for the training and execution of the neural network trigger have been newly developed within this thesis.

5. 2D Track Finding

In the track finding step, track segments from the axial superlayers are combined to circles in the transverse plane. The results of the algorithm are the number of tracks, an estimate for the 2D track parameters φ_0 and $p_\perp$ of each track and a list of the track segments that form each track. The track finder uses only the wire position of hits, but no drift times, and can therefore run in parallel to the event time finder. The subsequent 3D track reconstruction adds the drift time information to individual tracks and includes the stereo wires, as will be explained in the next chapter.

The following chapter introduces the Hough transformation as track finding algorithm and describes the track parametrization that was newly introduced in this thesis. To compare the resulting tracks to simulated particles, a matching algorithm is defined, based on the procedure used for the offline track reconstruction. Then the free parameters of the algorithm are selected with regard to optimal performance, with some adjustments to make a hardware implementation in FPGA feasible. The last step of the track finding is the identification of related track segments, allowing the subsequent modules to handle tracks independently from each other.

5.1. Hough transformation

The purpose of the Hough transformation is the detection of patterns within a set of discrete points. In the simplest case the patterns are straight lines, in other words the task is to find groups of collinear points, but the concept can be extended to more complicated patterns [61]. Originally the Hough transformation was invented to detect straight line segments in photographs of bubble chambers [62]. In the trigger, it is used to find the transverse projection of tracks (circles) from the priority wire coordinates of axial track segment hits (points).

5.1.1. Principle

The key idea is to transform each point in geometrical space into a line in a parameter space, the *Hough curve*. The coordinates in parameter space correspond to track parameters in geometrical space. A Hough curve for a given point is formed by the parameters of all possible tracks that contain this point (see fig. 5.1). The shape of the curve depends on the chosen track parametrization. A dual relationship exists between geometrical space and parameter space [63]:

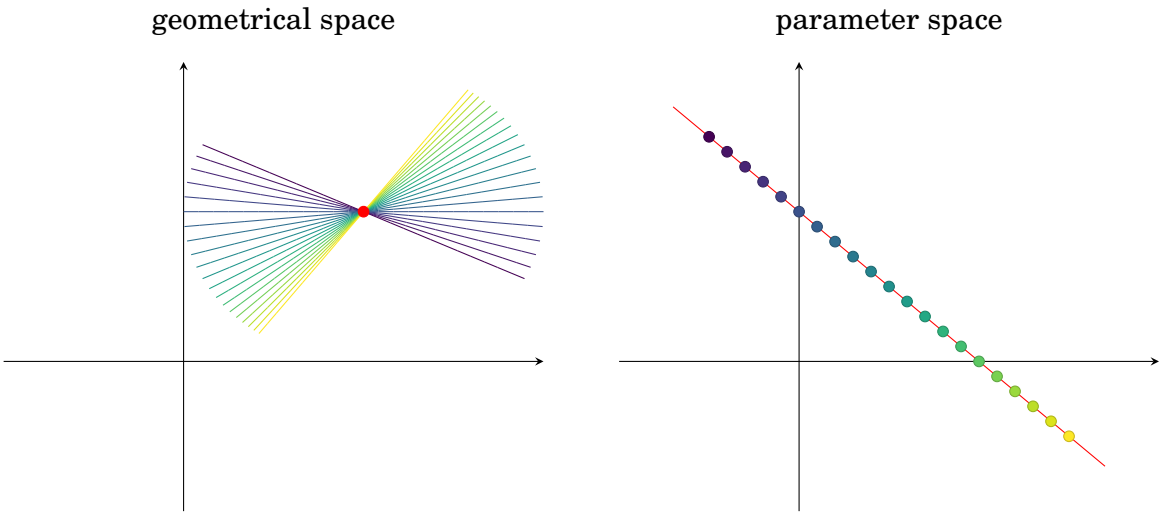

Figure 5.1.: Principle of the Hough transformation. A point in geometrical space is transformed to a curve in parameter space. Each point on the curve corresponds to a track through the point in geometrical space. The full curve represents the family of all possible tracks through the point.

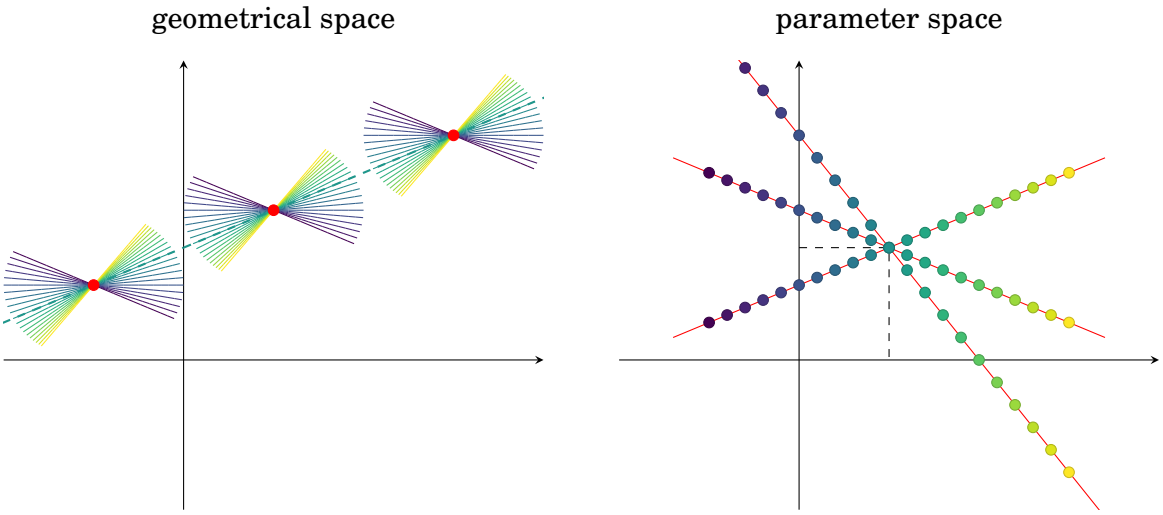

Figure 5.2.: Track finding with a Hough transformation. Points on the same track correspond to Hough curves through a common point in parameter space. The track parameters are given by the coordinates of the crossing point.

1. A point in geometrical space corresponds to a curve in parameter space.
2. A point in parameter space corresponds to a track in geometrical space.
3. Points lying on the same track in geometrical space correspond to curves through a common point in parameter space.
4. Points lying on the same curve in parameter space correspond to tracks through the same point in geometrical space.

It follows that tracks in the drift chamber can be found by transforming the wire coordinates of all hits to Hough curves and looking for crossing points in parameter space. This concept is illustrated in fig. 5.2. The track parameters are given by the coordinates of the crossing points. The hits that form a given track are those whose Hough curves pass through the corresponding crossing point.

5.1.2. Hough space parametrization

To find an explicit formula for the Hough transformation, a track parametrization must be chosen, defining the axes of the parameter space. The conceptual drawings in figs. 5.1 and 5.2 transform straight lines to straight lines. Such a transformation is obtained when the parameter space is defined by the slope and the intercept of the track. However, this definition runs into trouble for vertical tracks, where both slope and intercept become infinite. A bounded two-dimensional parameter space can be achieved by using the normal parametrization to define a track [63]

$$\rho = x \cdot \cos\theta + y \cdot \sin\theta\,, \tag{5.1}$$

where ρ is the distance of the track to the origin and θ is the angle of the normal (see fig. 5.3). The Hough curves in the $\theta - \rho$ plane are then sinusoidal and the parameters

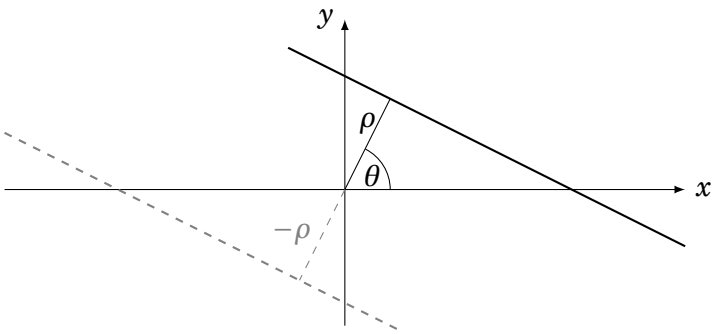

Figure 5.3.: Normal parametrization of a straight line. The line is defined by the distance ρ to the origin and the angle θ of the normal. The dashed gray line has the same normal, but a negative distance and is equivalent to a line with positive distance and rotated normal.

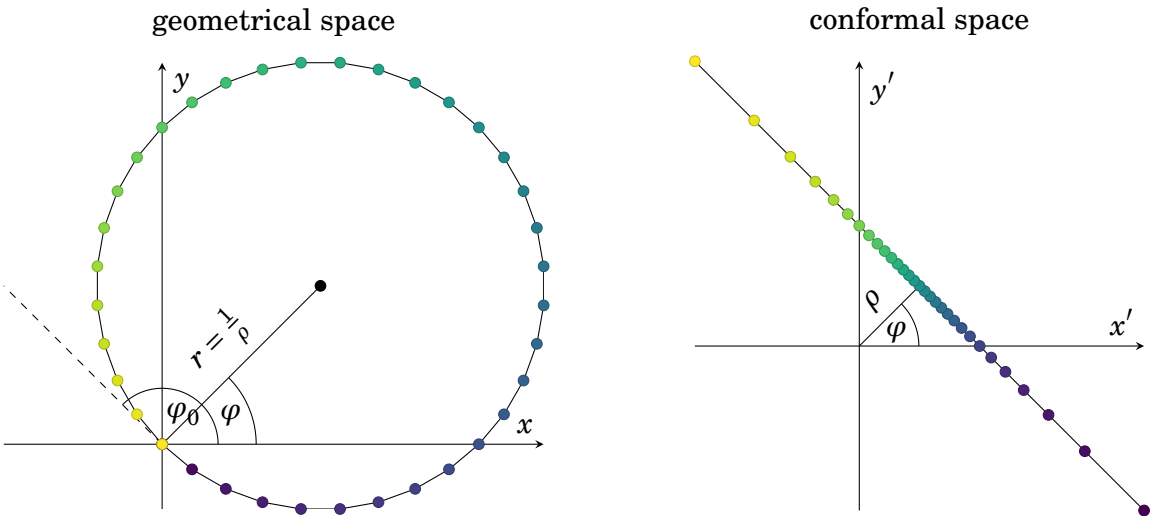

Figure 5.4.: Circles through the origin can be transformed to straight lines by a conformal mapping. The circle radius r is the inverse of the distance ρ of the straight line to the origin. The angle to the circle center φ is the same as the angle to the normal of the straight line. The angle φ_0 is the tangent to a clockwise circle.

for a given point are bounded by $\theta \in [0, 2\pi]$ and $\rho \in [-\sqrt{x^2+y^2}, \sqrt{x^2+y^2}]$. Note that there is an ambiguity in this definition: (θ, ρ) and $(\theta + \pi, -\rho)$ describe the same track. The usual way to solve this ambiguity is to limit either θ to $[0, \pi]$ or ρ to positive values.

In the presence of a magnetic field, the tracks in the transverse plane are not straight lines but circles. In general, a circle is defined by three parameters, for example the radius and the coordinates of the center. If tracks are assumed to originate from the beamline, the circle is constrained to pass through the origin, so two parameters are enough to fully parametrize the tracks. A convenient choice is the circle radius r and the angle to the circle center φ. To find the Hough curve, the circle equation is solved for the radius:

$$\begin{aligned} r^2 &= (x - r \cdot \cos\varphi)^2 + (y - r \cdot \sin\varphi)^2 \\ 0 &= x^2 + y^2 - 2r \cdot (x \cdot \cos\varphi + y \cdot \sin\varphi) \\ \frac{1}{r} &= \frac{2x}{x^2+y^2} \cdot \cos\varphi + \frac{2y}{x^2+y^2} \cdot \sin\varphi \,. \end{aligned} \tag{5.2}$$

This equation is equivalent to eq. (5.1) by introducing the conformal mapping (see fig. 5.4)

$$x' = \frac{2x}{x^2+y^2}, \qquad y' = \frac{2y}{x^2+y^2}.$$

The distance ρ in the conformal plane is identical to the inverse of the circle radius or the curvature. The angle φ to the circle center is the same as the angle θ to the normal in the conformal plane. The parameter space for the Hough transformation is thus spanned by φ and $\frac{1}{r}$.

For the track finding, it is convenient to slightly change the definition of the variables. First, the track segment coordinates are easier to obtain in polar coordinates $(r_{\mathrm{TS}}, \varphi_{\mathrm{TS}})$. Second, the angle in the track parametrization is replaced by the angle $\varphi_0 = \varphi + 90°$ of a tangent to the circle, or equivalently the direction of the transverse momentum at the origin for a track curling clockwise. With this definition eq. (5.2) becomes

$$\begin{aligned}\frac{1}{r} &= \frac{2\cos\varphi_{\mathrm{TS}}}{r_{\mathrm{TS}}} \cdot \sin\varphi_0 - \frac{2\sin\varphi_{\mathrm{TS}}}{r_{\mathrm{TS}}} \cdot \cos\varphi_0 \\ &= \frac{2}{r_{\mathrm{TS}}} \cdot \sin(\varphi_0 - \varphi_{\mathrm{TS}}).\end{aligned} \tag{5.3}$$

5.1.3. Crossing point ambiguity and charge

Figure 5.5 shows the Hough curves for a sample circle. As mentioned before, there are two crossing points in the parameter plane, one at $(\varphi_0, \frac{1}{r})$, the other at $(\varphi_0 + \pi, -\frac{1}{r})$. While a negative radius is somewhat counterintuitive, the correct interpretation of $\frac{1}{r}$ is the track curvature, which is in general signed. The sign of the curvature gives the curling direction or the charge of the particle. The reason for the ambiguity is that the track of a positively charged particle is identical to a negatively charged particle starting in the opposite direction.

As stated before, a common approach is to remove half of the parameter space. However, this creates an edge in the parameter space where determining the crossing points becomes difficult. For instance, if the lower half of the parameter space is removed, straight tracks with curvature 0 give two “half” crossing points. A similar problem arises for a specific angle if the parameter space is restricted in φ_0.

Instead the ambiguity is resolved by removing half of each curve, namely the half that corresponds to tracks curling back through the point in geometrical space. This approach assumes that all tracks leave the drift chamber towards the outer detector components. For the Belle II geometry this is true for transverse momenta above 256 MeV/c. If there is a low momentum track with a full circle in the drift chamber, it will be found as two tracks. However, the efficiency for this momentum range is very low already at the track segment finder level, which is limited to crossing angles of at most ±45° (compare fig. 4.5).

Limiting the Hough curves instead of the parameter space has two advantages. The first is a parameter space without cutoffs, since the angle is periodic and the curvature is limited by the efficiency of the track segment finder. The second is that the crossing point gives not only the direction and transverse momentum of the track but also the charge of the particle, which is the same as the sign of the curvature.

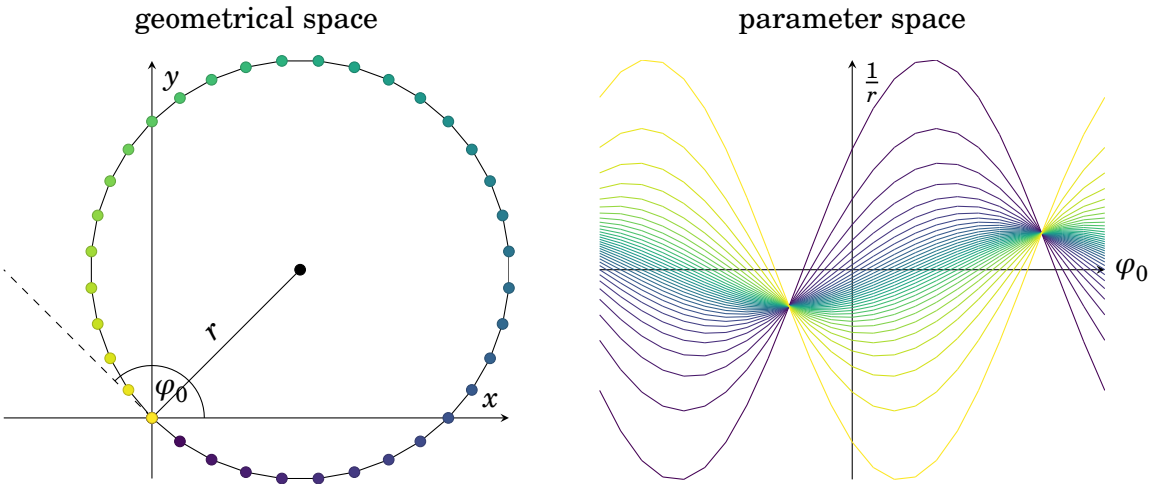

Figure 5.5.: Hough transformation of a circle. There are two crossing points, one for positive and one for negative curvature. The positive curvature result corresponds to a track going clockwise around the circle, the negative curvature corresponds to a track going counterclockwise with opposite starting direction φ_0.

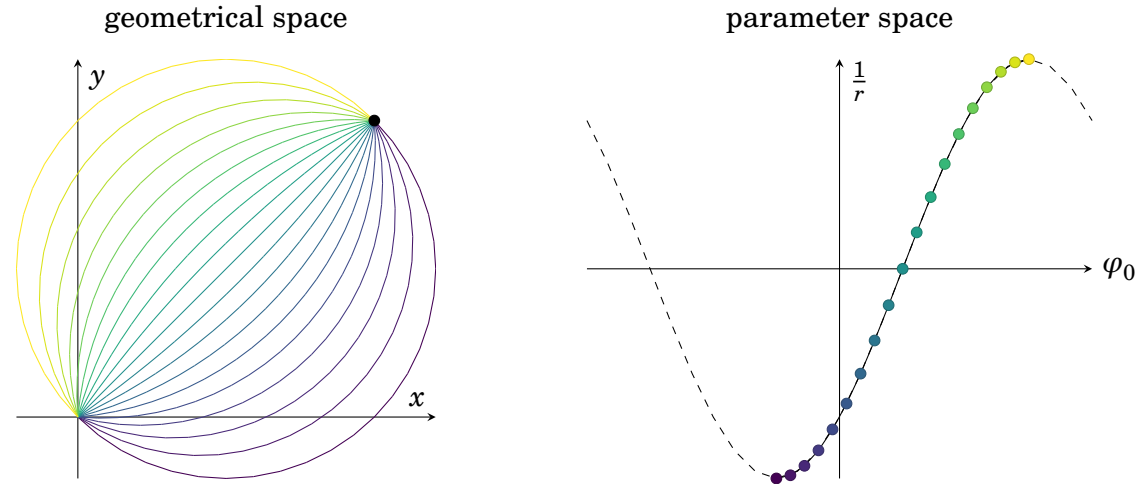

Figure 5.6.: Outgoing tracks through a given point correspond to the part of the Hough curve with rising slope.

Outgoing tracks through a given point have to fulfill $\varphi_0 \in [\varphi_{\mathrm{TS}} - 90°, \varphi_{\mathrm{TS}} + 90°]$, with the limits corresponding to tracks with maximal curvature. In the Hough curve this corresponds to the half with rising slope, as can be seen by taking the derivative of eq. (5.3):

$$\frac{\mathrm{d}}{\mathrm{d}\varphi_0}\left(\frac{1}{r}\right) = \frac{2}{r_{\mathrm{TS}}} \cdot \cos(\varphi_0 - \varphi_{\mathrm{TS}}). \tag{5.4}$$

The cosine is positive for $(\varphi_0 - \varphi_{\mathrm{TS}}) \in [-90°, 90°]$, which is equivalent to the range for outgoing tracks. The relation is illustrated in fig. 5.6.

5.1.4. Finding the crossing points

The most difficult part from the algorithmic point of view is the correct detection of crossing points in the parameter space. In general the crossing point is not exact, since the hit position has an uncertainty on the order of the distance between wires. The event time is not known at the track finding level, so it is not possible to use the drift times to get more precise hit coordinates.

The usual approach is to divide the parameter space into a grid of rectangles and count the number of lines crossing each grid cell [63]. The parameter space then becomes a matrix of line numbers and the crossing points are given by local maxima in the matrix, as shown in fig. 5.7. The size of the cells (or equivalently the number of cells in the grid) is treated as a free parameter that is optimized by comparing the track finding performance on simulated tracks. The optimal cell size should be similar to the uncertainty of the Hough curve. If the cell size is too small, the curves no longer cross the same cell, if it is too large, several crossing points cannot be separated and the crossing point position cannot be accurately determined.

When counting the lines through each grid cell, hits from the same superlayer are only counted once to avoid fake tracks from several neighboring hits. Such a cluster of track segment hits can be caused for example by a low energetic secondary particle. Particles produce one or two neighboring hits in the same superlayer with approximately equal probability, depending on the crossing angle, but the information from double hits is mostly redundant. A track is characterized by a set of hits on a circle-like path in several successive superlayers, regardless of the number of hits in each layer.

The consequence of this definition is that the maximum line number in the Hough matrix is 5 (the total number of axial superlayers). In most cases, the peak is not a single grid cell but a plateau of several connected cells. So instead of looking for a local maximum, a *peak candidate* is defined as a cell with an entry above a certain threshold, where the threshold is again a free parameter to be optimized. Connected peak candidates are combined to a cluster, where “connected” means that the cells share an edge or a top right/bottom left corner (see fig. 5.8). Connections over the top left/bottom right corner are not considered, since by definition the Hough curves have

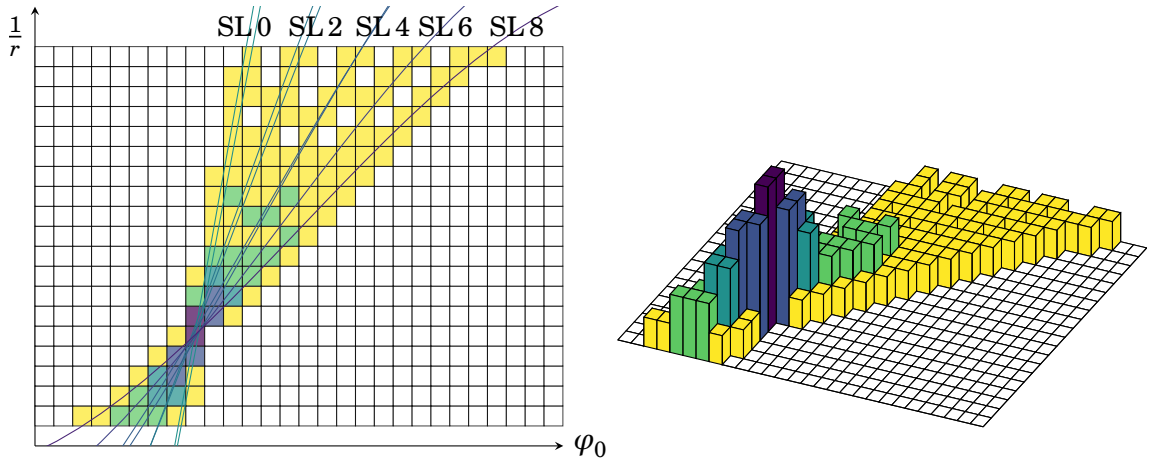

Figure 5.7.: Left: Construction of the Hough matrix. The parameter space is covered with a grid and the number of lines in each cell are counted. Lines from the same superlayer are only counted once. Right: The Hough matrix as a histogram. Cells above a given threshold are peak candidates.

a rising slope. The center of the cluster gives an approximation of the crossing point and can be found by averaging the center coordinates of all cells in the cluster.

Note that the definition of a peak candidate is independent of the surrounding cells. For example, for a peak threshold of four, a cell with lines from four different superlayers is a peak candidate even if it is connected to a cell with five lines. This allows to evaluate all cells in parallel to find peak candidates. To get only local maxima it would be necessary to check not only direct connections, but also indirect connections, like neighbors of neighbors. It turns out that the performance does not improve enough to justify the additional complexity.

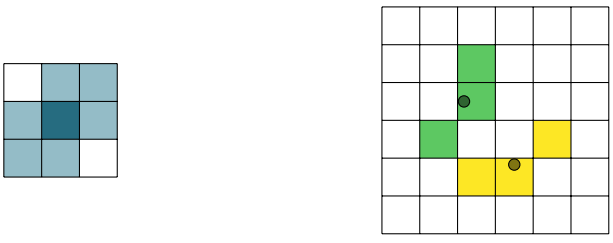

Figure 5.8.: Peak candidates are combined to a cluster if they are connected over a cell edge or over the top right to bottom left corner. Left: The center cell is connected to the six shaded cells. Right: Two clusters of peak candidates, not connected to each other. The center of each cluster is marked with a dot.

5.1.5. Alternative methods

An early implementation of the 2D track finding used the parameters φ and $\log(r)$ to define the parameter space [64]. Instead of eq. (5.2) the Hough curve was defined as

$$\log(r) = \log\left(\frac{x^2 + y^2}{2x \cdot \cos\varphi + 2y \cdot \sin\varphi}\right).$$

The principle of the Hough transformation is independent of the track parametrization, but the shape of the crossing point area changes with the definition of the Hough curve. For nearly straight tracks or a curvature of ≈ 0 the crossing point would be at infinity, that is the Hough curves don't cross, but converge to the same asymptote. For high but finite transverse momentum the crossing point is already deformed, leading to a bad track parameter resolution and difficulties in the clustering. In addition, since the logarithm is defined only for positive numbers, the logarithmic parametrization is equivalent to removing the negative half of the Hough plane, leading to the problem of ambiguous crossing points for tracks with high transverse momentum and to frequent "double counting", where two tracks with opposite charge are found instead of one. Both of these issues were solved within this thesis by defining the parameter space and the Hough curves according to eq. (5.3).

A possible future upgrade of the track finding is described in [65]. The polar angle of tracks is added to the parameter space as a third dimension and track segment hits from both stereo and axial layers are transformed into planes in this parameter space. For hits on the same helix, the planes cross in one single point. Instead of calculating the Hough planes analytically, they are learned from simulated data. A weight is introduced to reflect the uncertainty of the hit position. Together with the increased hit number this leads to a higher track finding efficiency and a better track parameter resolution as well as an initial estimate for the polar angle.

The following evaluation focuses on the pure 2D track finder with the parameter space defined by φ_0 and $\frac{1}{r}$, as newly introduced in this thesis. The logarithmic parametrization is obsolete; the 3D track finding is treated in detail in [66].

5.2. Evaluation of 2D track finding

To determine the free parameters of the track finding, some measures of quality are needed. For most parameters, there is a trade-off between different objectives. The first part of the following section describes the possible types of errors that can occur in the track finding and the quality measures derived from them. The second part introduces the algorithm for track matching that is required to compare the output of the track finder to simulated particles.

In the following sections, *track* refers to the output of the track finder which may or may not be correctly reconstructed, while *particle* refers to a true charged particle in

the detector. The track finder is tested on simulated data, where the true particles are known. To evaluate the trigger with real data, the results of the offline tracking could be used as "truth", since they are much more precise than the estimate at the first trigger level.

5.2.1. Types of errors

Missing tracks A track is said to be missing if no crossing point is found close to the parameters of a particle. The frequency of missing tracks is related to the *track finding efficiency*. The most likely reason for a missing track is that there are not enough track segment hits to exceed the threshold for a peak candidate. This is unavoidable and depends on the track segment finder efficiency and on the track parameters. The acceptance region depends on the selected peak threshold.

In principle, the grid cells can be so small that the Hough curves don't cross in the same cell. However, this effect does not occur in the tested parameter range. Another cause for missing tracks are the grid limits. To avoid this, the grid should cover the full acceptance region of the finder.

Merged tracks Two particles that are too close to each other cannot be separated. This effect happens mostly for high multiplicity events which give a clear trigger signature anyway. Merged tracks cause very large clusters, so they provide a border case for the clustering. The figure of merit for merged tracks is the minimal distance between tracks that can still be separated.

Ghost tracks When two tracks are close to each other, but not close enough to be merged, hits can sometimes be combined in different ways (see fig. 5.9). Such ambiguities are called *ghost tracks*. Background hits can also be combined with part of a track to form a ghost. The probability for ghost tracks decreases with higher peak candidate threshold and with better resolution. Another possibility to reduce ghosts would be a several step algorithm, where all curves that cross the most dominant cluster are removed before the second cluster is searched and so on. However, for the trigger such a procedure is neither practicable nor necessary, since ghost tracks are also only an issue for jet-like high multiplicity events, which should be triggered in any case.

Clone tracks Another reason for additional tracks are so-called *clones*. They are defined as two or more tracks that are found for only one particle, in contrast to ghosts that originate from combining two tracks. Clones happen if peak candidates are not correctly combined to a cluster, that is if the cluster has a gap. There are two approaches to suppress clones. The first is to change the clustering definition and merge tracks across gaps, increasing at the same time the possibility of merging two

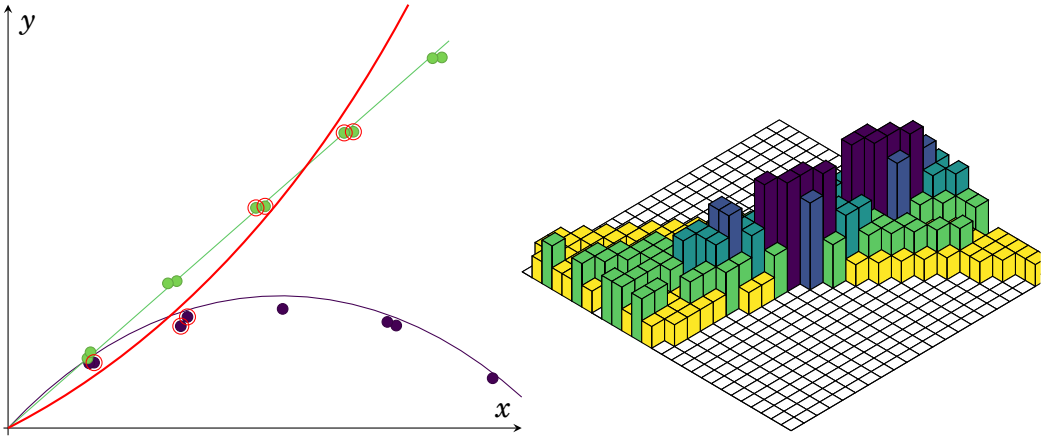

Figure 5.9.: Illustration of a ghost track as an example for errors in the track finding. Left: Two correctly reconstructed tracks and one ghost track (red). The ghost track combines hits of the other two tracks. Right: The Hough matrix for the same event. The ghost track appears as a third peak where the patterns for the other two tracks overlap.

separate tracks. The second is to require a minimum number of cells in each cluster, reducing also the efficiency. The latter approach is chosen in the trigger, since it does not increase the clustering complexity and a very good trade-off between efficiency and clone rate can be found.

Wrong track parameters Finally, a cluster can be distorted by additional background hits close to the track. This affects the resolution of the track parameters, but otherwise does not change the track finding performance. As for merged tracks, distorted tracks can pose a challenge to the clustering.

5.2.2. Track matching

To classify tracks according to the categories defined above, they must be compared to the simulated particles. The algorithm that is used for track matching is an adapted version of the offline track matching that is used to evaluate the Belle II tracking [67]. The matching works on the basis of hit relations. In the simulation, each particle keeps a relation to the hits it produced. Relations between tracks and hits can be found by inverting the Hough transformation. All hits whose Hough curves cross the cluster are related to the track. A hit can be related to more than one track if the Hough curve crosses more than one cluster, as in the case of ghost tracks.

To relate tracks and particles, a matching matrix is built with the rows corresponding to particles and the columns corresponding to tracks. The elements are given by the number of hits related to both the row particle and the column track. An additional

background row counts the hits not related to any particles. For normalization, the total number of hits related to each particle is also kept. The total number of hits related to a track can be inferred from the sum over a column, since each hit is related to at most one particle.

For the example in fig. 5.9 the matching matrix looks like this:

	track 1	track 2	track 3	total hits
particle 1	10	2	4	10
particle 2	2	8	3	8
background	0	0	0	0

The hits in the innermost superlayer are shared by both tracks. There is no background in this example, so the last row is empty. Relations are built separately for particles and tracks from the maxima of each column and row. In the example, the maxima in each row give relations from particle 1 to track 1 and from particle 2 to track 2:

	track 1	track 2	track 3	total
particle 1	10	2	4	10
particle 2	2	8	3	8
background	0	0	0	0

The *relation efficiency* is defined as the fraction of hits related to both the particle and the track over the total number of hits for the particle. In the example, the relation efficiency for both particles is 100 %. The relation is only created if the relation efficiency exceeds a given threshold.

The maxima in each column give relations from track 1 to particle 1, from track 2 to particle 2 and from track 3 to particle 1:

	track 1	track 2	track 3	total
particle 1	10	2	4	10
particle 2	2	8	3	8
background	0	0	0	0

The *relation purity* is defined as the fraction of hits related to both the particle and the track over the total number of hits for the track. In the example, the relation purity is 83 % for track 1, 80 % for track 2 and 57 % for track 3. Again the relation purity has to exceed a certain threshold to create a relation.

Depending on the relations from particles to tracks and vice-versa, the following matching categories are defined (see fig. 5.10 for illustration):

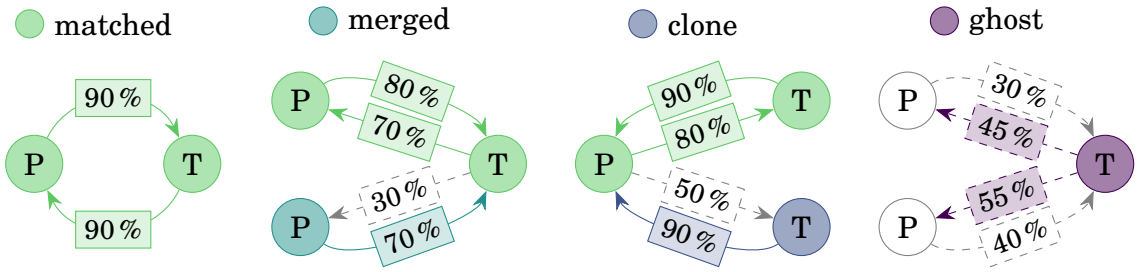

Figure 5.10.: Track (T) and particle (P) classification in the track matching [67]. The percentages on each relation are arbitrary examples for the relation efficiency and purity (see text). Dashed lines represent connections that do not result in a relation. For the sake of illustrating also ghost tracks, a purity threshold around 60 % is assumed here (the actual selected threshold is 10 %, so ghost tracks are usually classified as clones).

Matched: If the relation between particle and track is mutual, both are classified as *matched*. A relation weight is defined as the relation efficiency for particles and the relation purity for tracks.

Missing particle: If a particle has no relation to any track, it is classified as *missing*. To avoid losing particles in the matching, only a relation efficiency of 0 is classified as missing.

Merged particle: If a particle is related to a track that is related to another particle, the particle is classified as *merged*. The relation weight is set to the negative relation efficiency to distinguish merged and matched particles.

Background track: If a track has the highest relation to background hits, no relation is created and the track is classified as *background*.

Clone track: If a track is related to a particle that is related to another track, the track is classified as *clone*. The relation weight is set to the negative relation purity.

Ghost track: If the relation purity is below the threshold for creating a relation, the track is classified as *ghost*. After the matching, ghost tracks are indistiguishable from background tracks, since both have no relation to particles. The purity threshold determines whether an additional track is classified as clone or ghost.

For the track finder evaluation in the following sections, the thresholds for the relation efficiency and the relation purity are both set very low (1 % for the efficiency,

10 % for the purity). This means that particles are only missing if there is no nearby track and that there is no distinction between ghosts and clones.

5.3. Optimization of free parameters

In the following section, the free parameters of the track finding are scanned to find the optimal settings. For this purpose, different quality measures are defined using the matching categories defined in fig. 5.10. The studies are carried out for transverse momenta $p_\perp \geq 0.3\,\mathrm{GeV}/c$, meaning that the limits for the Hough plane are

$$\left|\frac{1}{r}\right| \leq \frac{0.3 \cdot B[\mathrm{T}]}{p_\perp[\mathrm{GeV}/c]} = 0.015\,\mathrm{cm}^{-1}. \tag{5.5}$$

The parameters to be optimized are summarized in table 5.1, together with reasonable ranges that are covered in the tests.

The peak threshold N_{hits} must be at least 3 to overdetermine a track and cannot be above 5. For the optimal number of cells the uncertainty of the Hough curve as defined in eq. (5.3) is estimated. In φ_0 it can be derived from the number of wires in a layer:

$$\Delta\varphi_0 \approx 2 \cdot \Delta\varphi_{\mathrm{TS}} \approx \frac{2\pi}{N_{\mathrm{wires}}}, \qquad N_\varphi = \frac{2\pi}{\Delta\varphi_0} \approx N_{\mathrm{wires}}.$$

The uncertainty is dominated by the innermost superlayer with 160 wires, so this value is used as baseline for N_φ. To find a reasonable starting point for optimizing N_r, the shape of the crossing point region is examined. The width is given by the cell size in φ_0. The height depends on the slope of the curve from the outermost superlayer as given in eq. (5.4). So to get an approximately quadratic crossing point region for the maximal slope, the cell size can be estimated as follows (using the limits from eq. (5.5)):

$$\Delta\frac{1}{r} \approx \Delta\varphi_0 \cdot \frac{2}{r_{\mathrm{TS}}} \approx \frac{4\pi}{N_\varphi \cdot r_{\mathrm{TS}}}, \qquad N_r = \frac{2 \cdot 0.015\,\mathrm{cm}^{-1}}{\Delta\frac{1}{r}} \approx 0.25 \cdot N_\varphi \approx 40, \tag{5.6}$$

with $r_{\mathrm{TS}} = 105.68\,\mathrm{cm}$ for the priority layer of the outermost superlayer. These estimates should give very rough guidelines about the Hough grid. In the optimization, the cell number is varied from 0.5 to 1.5 times the estimated optimum.

Table 5.1.: Free parameters of the track finding and tested parameter ranges.

parameter	symbol	tested values
threshold for peak candidates	N_{hits}	3, 4, 5
number of grid cells in φ_0	N_φ	{80, 90, …, 240}
number of grid cells in $\frac{1}{r}$	N_r	{20, 22, …, 60}
minimum cells in cluster	N_{cells}	1, 2, 3, 4, 5

5.3.1. Peak threshold: efficiency

The track finding efficiency is defined as

$$\text{efficiency} = \frac{\text{number of matched particles}}{\text{total number of particles}}.$$

It is the most important figure of merit for the track finding and dominated by the hit efficiency and the peak threshold. Figure 5.11 shows the efficiency for different

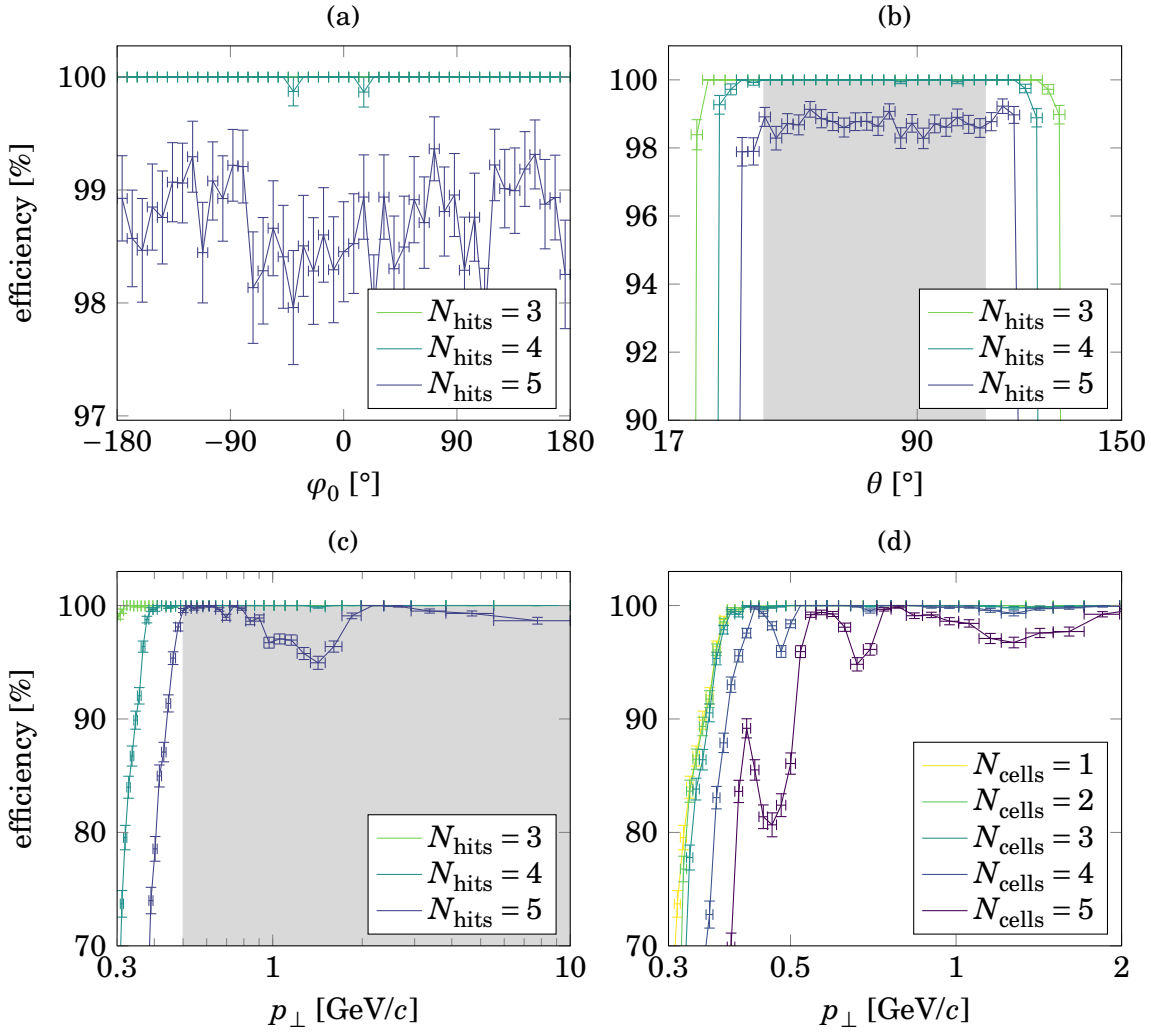

Figure 5.11.: Single track efficiency depending on the track parameters. To disentangle the dependence on the parameters, $p_\perp$ and θ are limited to the shaded areas for the other two parameters, respectively. (a) to (c): Efficiency for different peak threshold and fixed cluster size $N_{\text{cells}} = 1$. (d): Efficiency for different cluster size and fixed peak threshold $N_{\text{hits}} = 4$.

peak thresholds, measured on simulated single muon tracks with background mixing. There is no cut on the cluster size ($N_{\text{cells}} = 1$) and the Hough grid is the estimated optimum of 160×40 cells.

In the azimuth φ_0 the efficiency is flat, in the transverse momentum $p_\perp$ and the polar angle θ there is an acceptance region that depends on the peak threshold. For the polar angle, the acceptance is limited by the requirement to reach the third, fourth or fifth axial superlayer before leaving the CDC. For the transverse momentum, the acceptance is limited by the crossing angle dependence of the track segment finder (compare fig. 4.5).

For a peak threshold of 5, the efficiency has a dip around transverse momenta of 1 GeV/c to 2 GeV/c. This is related to a crossing angle of $-10°$ in one of the layers, which means that the track is not lost due to a mistake in the track finding but because there are simply not enough hits. With a peak threshold of 4, the efficiency is above 99 % within an acceptance region of $p_\perp > 0.38\,\text{GeV}/c$ and $\theta \in [31°, 126°]$. Although a threshold of 3 leads to a larger acceptance, the background rate and clone rate also increase (by a factor of ≈ 3 for the background rate and one order of magnitude for the clone rate), as well as the average cluster size.

Therefore, the peak threshold is fixed at $N_{\text{hits}} = 4$. For tracks with flat polar angles, an alternative track finding setup with $N_{\text{hits}} = 3$ is studied in appendix C, as well as the consequences for the 3D track reconstruction. To optimize the clone rate, the minimum cluster size N_{cells} is varied rather than the peak threshold. The dependence of the efficiency on the cluster size is shown in fig. 5.11d. For the given grid, a cluster size of 2 or 3 can be required without significant efficiency decrease. A minimal cluster size of 4 still gives a better efficiency than a peak threshold of 5.

Note that the typical cluster size depends on the Hough grid, so after the optimization of N_φ and N_r the efficiency needs to be checked again.

5.3.2. Hough grid: clone rate and resolution

For the optimization of the Hough grid two quality measures are considered. The first is the clone rate, which is defined as

$$\text{clone rate} = \frac{\text{number of clone tracks}}{\text{total number of tracks}}.$$

The second is the resolution of the track parameters as given by the cluster center. It is defined only for matched tracks as the standard deviation of the difference between the track parameters (φ_0 and $p_\perp^{-1}$) and the corresponding parameters of the matched particle.

The resolution is again measured on single muon tracks. The clone rate is additionally estimated on Bhabha scattering events, since Bhabha scattering is a dominant background and can only be suppressed if all tracks are cleanly reconstructed. Therefore, a very low clone rate is required. All simulations include background mixing.

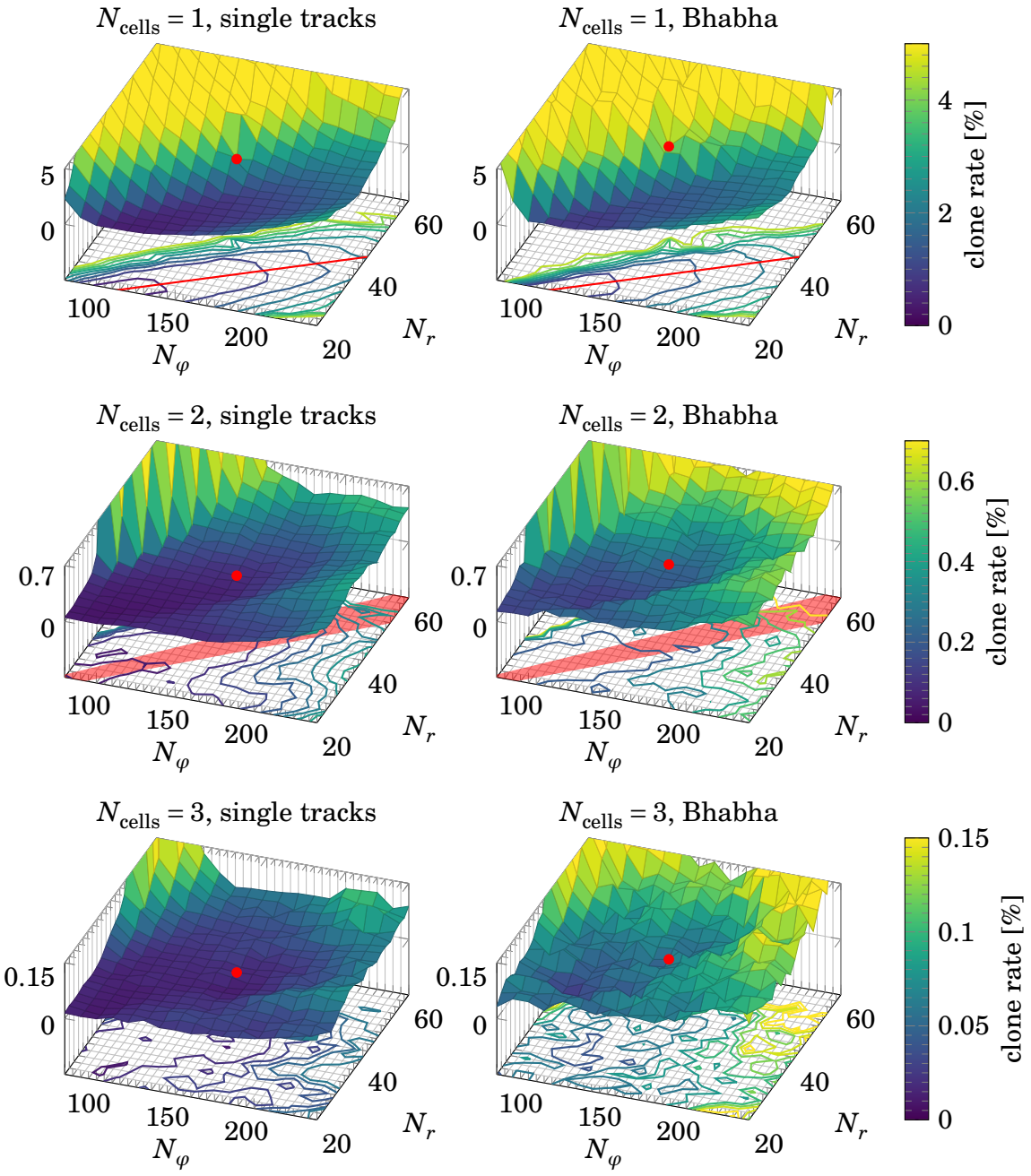

Figure 5.12.: Clone rate for single muon tracks and Bhabha events for different cell numbers in the Hough grid and different cluster size requirement. The red trendline shows the optimal ratio between N_φ and N_r. The red dot shows the a priori estimation for the optimum of (160×40) cells.

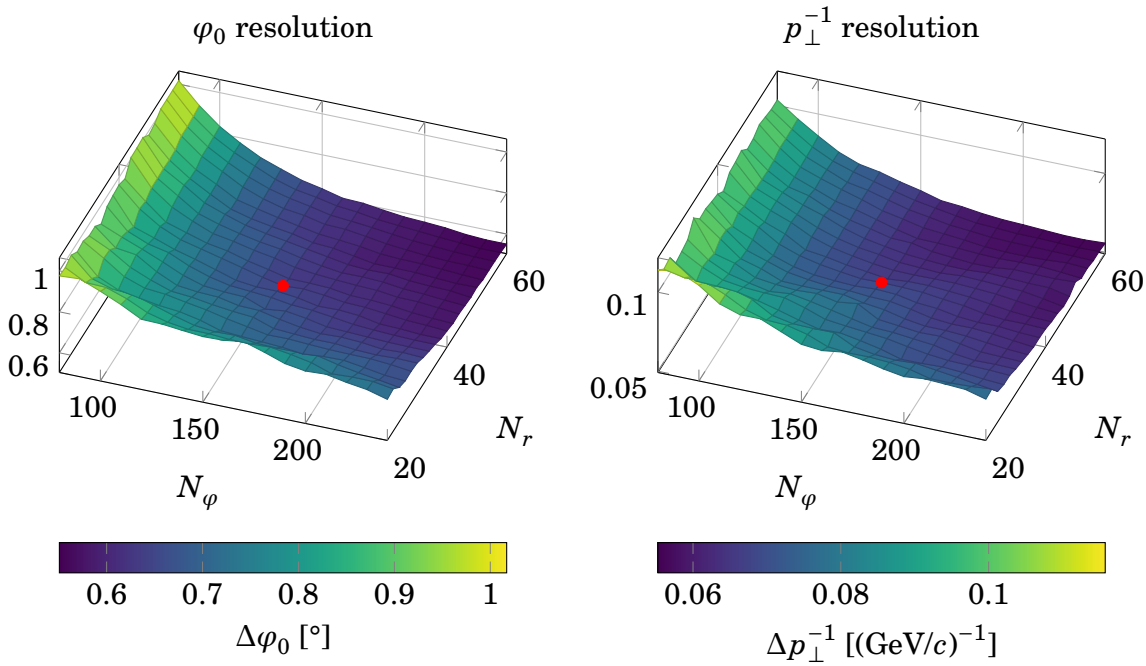

Figure 5.13.: Track parameter resolution for different cell numbers in the Hough grid and cluster size requirement $N_{\text{cells}} = 2$. The red dot shows the a priori estimation for the optimum of (160×40) cells.

Figure 5.12 shows the clone rate for different cluster size requirements. For $N_{\text{cells}} = 1$ it depends mostly on the ratio between N_φ and N_r, with an optimum at $\frac{N_r}{N_\varphi} \approx 0.175$, smaller than the estimated optimum in eq. (5.6), where only the Hough curve from the outermost superlayer was taken into account. Combining the curves from all superlayers gives a more elongated cluster shape than assumed in the simple estimate, so the real optimum is found for lower N_r. Within the optimal valley the clone rate increases with the cell number. For $N_{\text{cells}} = 2$ the clone rate decreases by almost one order of magnitude and the valley is less pronounced, with the optimal ratio shifted to $\frac{N_r}{N_\varphi} \approx 0.22$ to 0.26, that is a slightly elongated cluster shape is preferred. For $N_{\text{cells}} = 3$ the clone rate decreases again and is dominated by statistical fluctuations, but the general tendency is unchanged.

The track parameter resolution is shown in fig. 5.13. The cluster size requirement affects only the number of found tracks, not the parameter estimation of individual tracks, so it is not varied for the resolution study. In contrast to the clone rate the resolution improves for higher cell number and is independent of the ratio $\frac{N_r}{N_\varphi}$. The resolution of both φ_0 and $p_\perp^{-1}$ depends equally on N_φ and N_r.

Figure 5.14 illustrates the trade-off between the clone rate for Bhabha events and the φ_0 resolution. Since the resolution of both track parameters shows the same

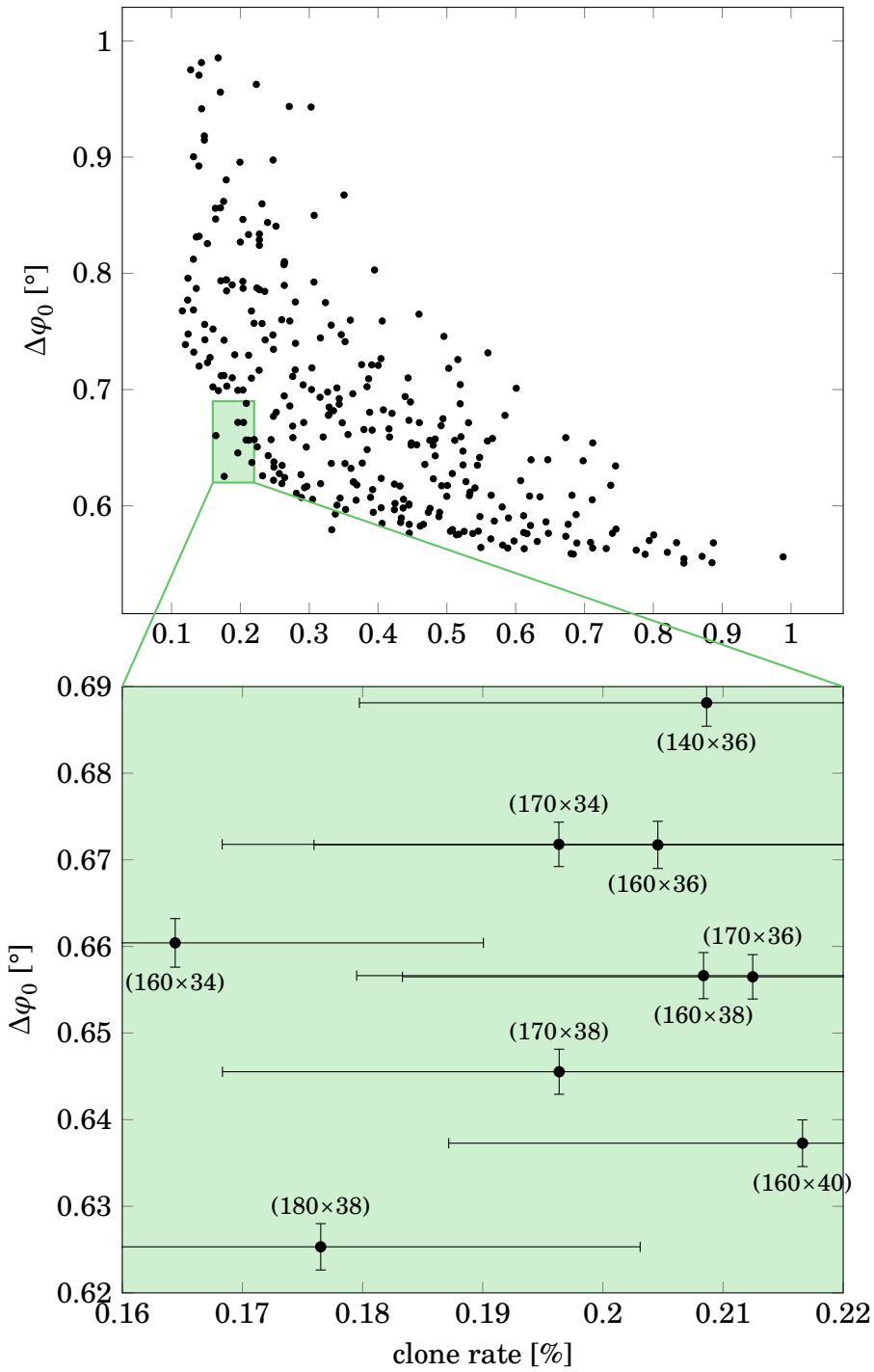

Figure 5.14.: Top: Clone rate for Bhabha events vs. φ_0 resolution for cluster size requirement $N_{\text{cells}} = 2$ and different cell numbers in the Hough grid. Bottom: Close up of the optimal region (green rectangle), with the cell numbers ($N_\varphi \times N_r$) as labels.

dependence on the cell number, $\Delta p_\perp^{-1}$ is not compared to the clone rate separately. The best compromise is found in the region of $N_\varphi \in [160, 180]$, $N_r \in [34, 40]$. Within this region the difference between individual settings is below the uncertainty. The original estimate of ($N_\varphi = 160 \times N_r = 40$) is within the optimal region, with a clone rate for Bhabha events of $(0.22 \pm 0.03)\,\%$ for $N_{\text{cells}} = 2$ or $(0.053 \pm 0.015)\,\%$ for $N_{\text{cells}} = 3$ and a resolution of $\Delta\varphi_0 = 0.64°$ and $\Delta p_\perp^{-1} = 0.069\,(\text{GeV}/c)^{-1}$.

Without cluster size cutoff the best compromise is around ($N_\varphi = 180 \times N_r = 32$), with a clone rate of $(1.39 \pm 0.07)\,\%$, which is worse than the clone rate for $N_{\text{cells}} = 2$ by a factor of 6. On the other hand, a cluster size requirement of $N_{\text{cells}} = 3$ gives a very low clone rate even for very high cell number. However, the resolution improves by only $\approx 10\,\%$. Taking into account also the slightly reduced acceptance resulting from an increased cluster size cutoff (see fig. 5.11d), the minimal cluster size is fixed at $N_{\text{cells}} = 2$.

To decide on the size of the Hough grid, one more issue is taken into account, namely the computational complexity of the clustering procedure. It increases with the typical cluster size, so a smaller grid is preferred, but it depends also on the cluster shape.

5.4. Fast clustering

In the studies above, clusters were found by recursively searching through the neighbors of each peak candidate until no more connected candidates were found. This procedure is repeated until all peak candidates are associated to a cluster. The advantage of this method is that it makes no assumption about the cluster size and is thus independent of the cell number, but it is not practicable for the hardware implementation, where a parallel clustering algorithm with deterministic runtime is required. To enable a parallel search of clusters, a unique *cluster seed* needs to be defined from which the clustering procedure starts. If there is exactly one seed in each cluster, the seed condition can be checked in parallel for all peak candidates. The number of tracks is then identical to the number of cluster seeds. To get the center of each cluster in a finite number of steps, the maximum *cluster area* around the seed needs to be limited. Any peak candidates beyond this area are ignored, distorting the estimate of the cluster center.

Table 5.2 shows the likelihood to find different cluster areas, defined as the minimum bounding rectangle of a cluster, with the grid sizes from the optimal region in fig. 5.14 for single tracks without background hits. From these estimates a cluster area of at least (4×5) cells needs to be searched. With background hits and merged tracks the cluster area can be significantly larger. For this size it is convenient to introduce a two step clustering procedure, where the Hough grid is first divided into 2×2 squares of cells. This approach was developed for the first version of the track finder described in [64] and adjusted to the altered cluster shape. The substructure of each square is encoded in a binary pattern of four bits, which signify whether the corresponding cell

Table 5.2.: Percentages of different cluster areas in single tracks without background hits in different Hough grids. For each setting the 5 most common cluster areas are listed. The remaining clusters are summarized in two categories, namely larger than the most common clusters and smaller but differently shaped.

		cluster area						
N_φ	N_r	(2×3)	(3×4)	(3×5)	(4×5)	(4×6)	$>(4\times6)$	other
140	36	15.5 %	18.7 %	27.7 %	11.7 %	11.6 %	0.4 %	14.4 %
160	38	15.5 %	27.1 %	15.3 %	20.1 %	8.8 %	1.1 %	12.1 %
160	40	15.7 %	20.2 %	24.1 %	14.9 %	13.6 %	1.3 %	10.1 %
		(2×3)	(3×4)	(3×5)	(4×4)	(4×5)	$>(4\times5)$	other
160	34	11.1 %	36.4 %	5.5 %	6.6 %	26.1 %	3.4 %	10.9 %
160	36	14.2 %	31.0 %	10.8 %	4.5 %	23.7 %	6.0 %	9.8 %
170	36	11.2 %	31.4 %	6.2 %	7.6 %	26.7 %	4.6 %	12.3 %
170	38	12.7 %	29.6 %	11.3 %	4.4 %	25.6 %	6.7 %	9.7 %
180	38	9.8 %	31.1 %	5.8 %	7.6 %	27.3 %	5.9 %	12.4 %
		(2×3)	(3×3)	(3×4)	(4×4)	(4×5)	$>(4\times5)$	other
170	34	9.1 %	8.6 %	29.9 %	12.6 %	25.5 %	3.8 %	10.3 %

is a peak candidate or not. The squares with at least one peak candidate are then clustered by a neighborhood definition that takes the substructure into account and is illustrated in fig. 5.15. These clustering rules are almost equivalent to the previous definition of connected peak candidates illustrated in fig. 5.8, except for the case where only the top left and bottom right cell within a 2×2 square are peak candidates. By the cell to cell connection definition these candidates should be unconnected, but the two step pattern algorithm does not allow to split a 2×2 square into separate clusters.

With this neighborhood definition the clustering algorithm works as follows: A cluster seed is defined as a square that is not connected to its left and bottom neighbors, in other words the bottom left corner of the cluster is used as seed. Equivalently, the top right corner could be used. The obvious disadvantage of this definition is that it introduces a bias for clusters that exceed the limits of the maximum cluster area. A seed close to the cluster center would avoid this bias but is difficult to define without ambiguities, which would lead to clone tracks. Starting from the cluster seed, connected squares are added to the cluster within a rectangle of fixed size extending above and to the right of the seed. The center of each cluster is estimated from the top right and bottom left corner, again taking into account the substructure of the squares. This method is faster and more simple than an average over all peak candidates and thus easier to implement in the hardware (for details of the hardware implementation

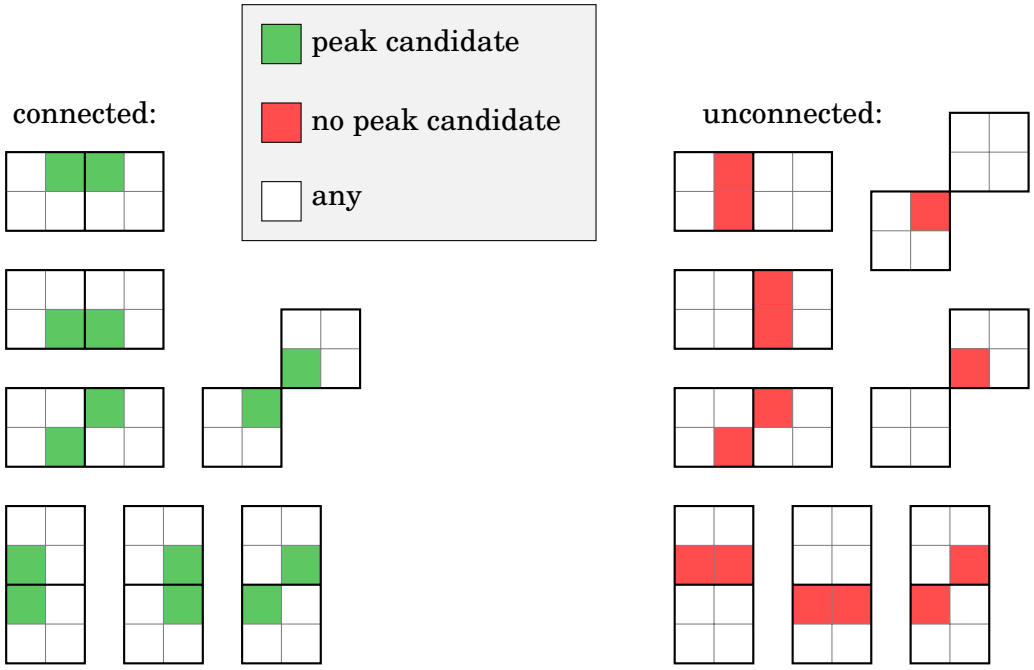

Figure 5.15.: To cluster 2×2 squares of Hough cells, the patterns within the squares are compared. Left: Connected squares. Right: Unconnected squares. The two sets of definitions for connected / unconnected are equivalent to each other, that is one set is enough to fully define the clustering rules.

see appendix B.1).

5.4.1. Size of the cluster area

According to table 5.2 the clustering rectangle needs to contain at least 3×3 squares. Note that although most clusters fit into an area of 2×3 squares, there is a random offset with respect to the grid of 2×2 squares, which is accounted for by enlarging the clustering rectangle by one square. Thus a 3×3 clustering rectangle is guaranteed to hold any (5×5) cluster, while a (5×6) cluster has a 50 % likelihood to extend above the clustering rectangle. With this consideration, the Hough grid is fixed to $(N_\varphi = 160 \times N_r = 34)$ cells, which gives very good results for both the clone rate and the resolution (see fig. 5.14) and the smallest likelihood for protruding from a 3×3 clustering rectangle (see table 5.2).

Figure 5.16 shows the percentages for different cluster areas with the selected grid in different kinds of events. The first event class is the idealized case of single tracks without background. For this case most clusters fit into a 3×3 clustering rectangle. The second event class is a test sample with two muon tracks created by a particle gun in a limited part of the detector ($\varphi_0 \in [0°, 45°]$, $\theta \in [45°, 110°]$) with additional

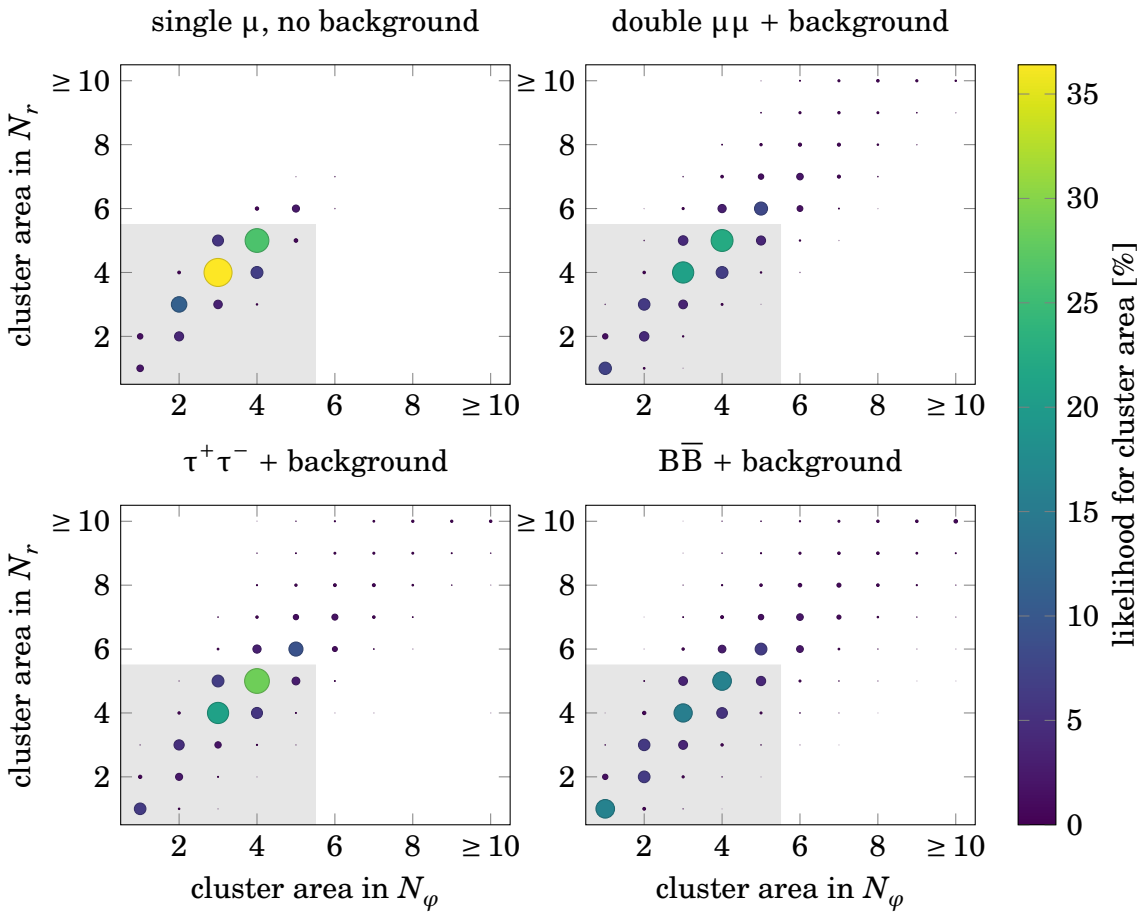

Figure 5.16.: Percentages of different cluster areas in a Hough grid of (160 × 34) cells for different event types (see text). The gray area marks the clusters that are guaranteed to be covered by a clustering area of 3 × 3 squares.

noise from background mixing. These events have an enhanced number of merged particles and are examined in order to isolate the effect of two merged particles on the clustering. In the following, they are referred to as *double muon* events. The last two event classes are realistic events, namely $\tau^+\tau^-$ pairs and $B\overline{B}$ events, both with background mixing.

As expected, the likelihood for large clusters increases in the presence of background hits and additional tracks. Merged particles produce clusters that are up to twice as large as clusters from isolated tracks. The probability for merged particles is given by the merge rate, which is defined as

$$\text{merge rate} = \frac{\text{number of merged particles}}{\text{number of matched or merged particles}}.$$

This definition does not contain the number of missing particles, so it is independent

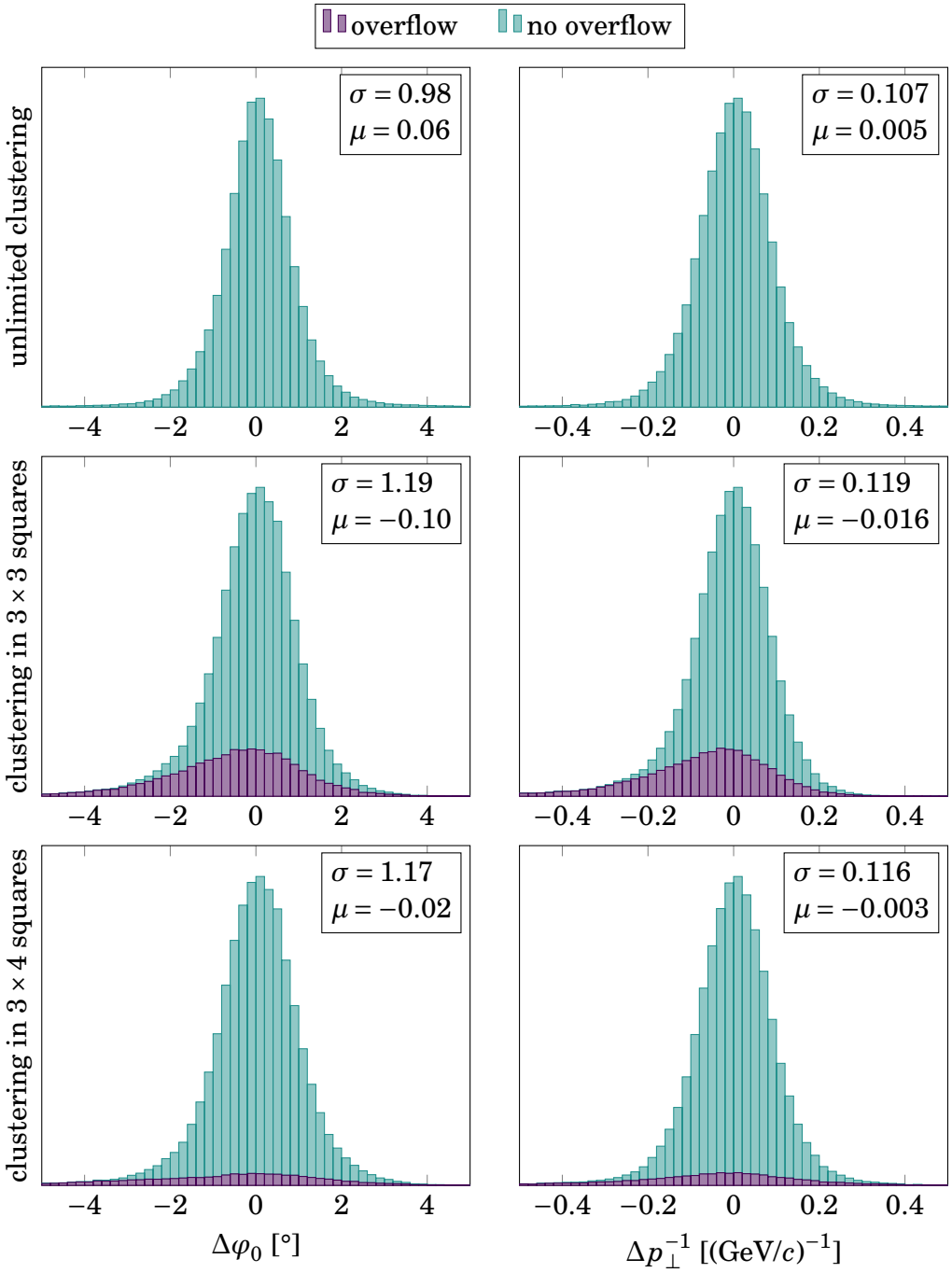

Figure 5.17.: Track parameter resolution for events with two muons and increased probability for merged particles ($\varphi_0 \in [0°, 45°]$, $\theta \in [45°, 110°]$). σ and μ are the width and the mean of the total distribution (with and without overflow from the clustering rectangle). The bias μ is at least one order of magnitude smaller than the resolution σ.

of the acceptance region. For $\tau^+\tau^-$ pair events the merge rate is $(3.37 \pm 0.04)\,\%$, for the double muon sample it is $(4.49 \pm 0.05)\,\%$ and for $B\overline{B}$ events it is $(16.61 \pm 0.05)\,\%$. In other words, for low multiplicity events the merge rate is low enough that the cluster size for single tracks gives a reasonable estimate for the cluster area. For high multiplicity events the track parameter resolution is less important, so an increased overflow probability does not affect the trigger performance.

Figure 5.17 shows the effect of a limited cluster area on the resolution for the double muon sample. Even with an unlimited clustering area, the resolution is worse than for single tracks, because the center of a cluster that contains two merged particles gives only an average over the track parameters of both particles. With a clustering rectangle of 3×3 squares, there is a clear bias for clusters that protrude from the clustering rectangle, especially in the tails of the distribution. However, such an overflow is rare enough that the bias of the overall distribution is still one order of magnitude smaller than the width of the distribution. Due to the simplified center estimation there is a resolution loss of $\approx 20\,\%$ in φ_0 and $\approx 10\,\%$ in $p_\perp^{-1}$. By increasing the clustering area to 3×4 squares, the probability for a cluster overflow can be reduced so far that the overall bias vanishes completely, but the resolution does not improve significantly.

In addition to the almost identical bias and resolution, a larger clustering rectangle requires more hardware resources and might not be feasible (see appendix B.1), so to avoid overestimating the performance of the track finding in the simulation, the clustering rectangle is set to 3×3 squares in the following studies.

5.4.2. Irregular cluster shapes

A further difficulty that arises from merged particles is the cluster shape. In the simplified clustering, the cluster center is estimated from the top right and bottom left corner of the cluster, that is the cluster is assumed to be symmetric around the

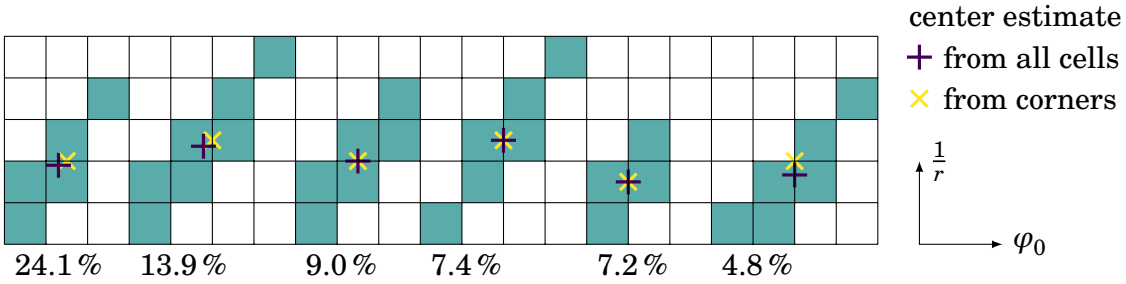

Figure 5.18.: Most common cluster shapes for single tracks without background. For shapes that are not symmetric under a 180° rotation, the percentages includes both variants, which occur with equal probability. The center estimate from the cluster corners is only precise for symmetric clusters.

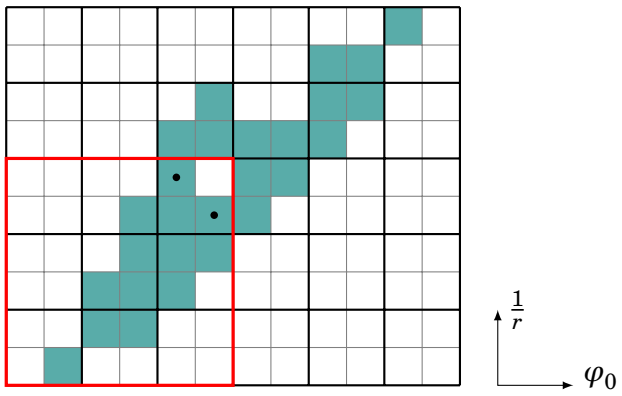

Figure 5.19.: Example for a very large and irregular cluster produced from the merging of two particles in a τ decay. Within the clustering rectangle (red) there are two possibilities to define the top right corner of the cluster (marked with dots).

center. This is in general not true, resulting in the resolution loss observed in fig. 5.17. The most common cluster shapes for single tracks without background are shown in fig. 5.18, together with the center estimate derived from an average over all cells and the simplified center estimate derived from the corners.

For merged particles, the cluster corners are not even necessarily unique. An example for a cluster where the top right corner is ambiguous is shown in fig. 5.19. For such cases it is necessary to define some general rule for the selection of the corners to get a fully deterministic algorithm. The following rules are chosen (see fig. 5.20):

1. The bottom left square is the cluster seed, so it is always unique.

2. If the top right square is not unique, *top* is preferred over *right*. This reflects the typical cluster shape for single tracks, which is vertically elongated.

3. If the bottom left cell within the bottom left square is not unique, the top left cell is selected as corner. This choice is arbitrary; the “correct” selection depends on the relative location of the merged cluster parts.

4. If the top right cell within the top right square is not unique, the bottom right is selected as corner. This choice is symmetric to the definition of the bottom left cell.

Events with ambiguous cluster corners are already included in the distributions shown in fig. 5.17, so there is no further resolution loss.

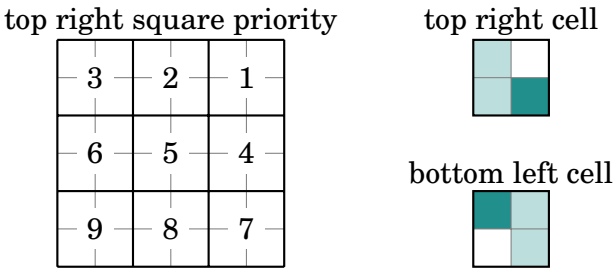

Figure 5.20.: Rules for the selection of cluster corners within a clustering rectangle of 3×3 squares. Left: Selection order for the top right square. The first square with at least one peak candidate is selected. Right: Selection of the top right and bottom left cell within a square in case of an ambiguity.

5.4.3. Related track segments

The last task of the track finding is to create a list of related track segments for each track. This can be done by inverting the Hough relation: a track segment is related to a track if the corresponding Hough curve crosses the cluster. For the fast clustering algorithm, the easiest way for finding the related hits is to check only the cluster corners, which have already been identified for the purpose of estimating the cluster center. All track segment hits whose Hough curve crosses one of the corner cells are related to the track.

Another possibility is to create relations for all hits whose Hough curve crosses the center of the cluster. However, two difficulties arise for this approach. First, the estimated center can be either in the middle of a cell, on a cell edge or on a cell corner, so up to 4 cells need to be checked. Second and more important, for clusters that are distorted by background hits or merged tracks the estimated center is not necessarily a peak candidate, so it is possible that *no* track segments are related to the center. In this case the track cannot be further processed in the following trigger modules and is effectively lost. However, the probability to get a center which is not a peak candidate is relatively low, with $\approx 0.5\,\%$ for $B\overline{B}$ events and less for low track multiplicity.

The quality of the track segment selection is given by the relation efficiency and purity of the track matching, which was defined in section 5.2.2. The relation efficiency describes the fraction of true hits that are correctly related to the track. The relation purity describes the fraction of related hits that are also related to the true particle. Both should be as high as possible. Table 5.3 shows the average relation efficiency and purity for single tracks and for $\tau^+\tau^-$ pair events (both with background). The relation efficiency is highest when relations are created for all cells in a cluster, but the center alone also gives a very high relation efficiency and a better purity, since additional hits are unlikely to cross the cluster center. The relations derived from the corners give a comparatively low efficiency and purity.

Therefore, the center is selected for creating relations despite the risk to get a few

Table 5.3.: Relation efficiency and purity for two exemplary event classes. Relations are either created from the Hough curves in all cells of a cluster, only from the center cells or only from the corners.

	all cells		center		corners	
events	efficiency	purity	efficiency	purity	efficiency	purity
single μ	99.95 %	93.50 %	98.66 %	94.60 %	91.53 %	93.38 %
$\tau^+\tau^-$ pairs	99.68 %	88.44 %	97.53 %	91.28 %	88.98 %	88.88 %

tracks without relations. If necessary, the hit relations from the corners could be used as fallback option to make sure that each track has some related hits.

Only one track segment per track and superlayer is transmitted to the following modules. Since the fitter cannot handle second priority hits due to resource limitations, a first priority hit is selected when possible. If there are several first priority hits (or no first but several second priority hits), the track segment with higher ID is selected.

5.5. Final setup

Table 5.4 summarizes once again the algorithm and the selected parameters. Two setups are defined: an *optimal 2D finder* setup, which uses the best setting found without regard for the hardware limitations and a *realistic 2D finder* setup, which uses the limited clustering algorithm developed for the hardware implementation. All following studies are based on the realistic setup. The optimal setup is shown only as reference.

The most important quality figures are also listed once again. The only significant differences between the two setups are observed in the track parameter resolution and in the merge rate, which is reduced by the cluster cutoff. The latter is an effect of the track matching and the definition of hit relations. Clusters are still merged, but the resulting tracks are matched to only one particle, while the second particle is classified as missing instead of merged.

Table 5.4.: Summary of the selected parameters for the 2D track finder and the most important quality figures. The parameters that are different in the two defined setups are highlighted.

	optimal 2D finder	realistic 2D finder
algorithm		
Hough curves	φ_0 vs. $\frac{1}{r}$, eq. (5.3)	φ_0 vs. $\frac{1}{r}$, eq. (5.3)
Hough matrix	160×34 cells	160×34 cells
$\varphi_0 \in$	$[-180°, 180°]$	$[-180°, 180°]$
$\frac{1}{r} \in$	$[-0.015, 0.015]\,\mathrm{cm}^{-1}$	$[-0.015, 0.015]\,\mathrm{cm}^{-1}$
peak candidates	≥ 4 superlayers	≥ 4 superlayers
clustering	connected regions (fig. 5.8)	2×2 squares (fig. 5.15)
cluster size	≥ 2 cells	≥ 2 cells
limited to	—	3×3 squares
center estimate	average over all cells	average corners (fig. 5.20)
hit relations	from all cells	from center
performance		
threshold for efficiency $\geq 50\,\%$		
$p_\perp$ [GeV/c]	0.310 ± 0.001	0.310 ± 0.001
θ [°]	$[29.6 \pm 0.2,\ 127.1 \pm 0.2]$	$[29.6 \pm 0.2,\ 127.1 \pm 0.2]$
threshold for efficiency $\geq 99\,\%$		
$p_\perp$ [GeV/c]	0.380 ± 0.002	0.380 ± 0.002
θ [°]	$[30.6 \pm 0.5,\ 126.0 \pm 0.5]$	$[30.6 \pm 0.5,\ 126.0 \pm 0.5]$
Bhabha clone rate	$(0.16 \pm 0.03)\,\%$	$(0.16 \pm 0.03)\,\%$
resolution		
$\Delta\varphi_0$ [°]	0.660 ± 0.003	0.818 ± 0.003
$\Delta p_\perp^{-1}$ [(GeV/c)$^{-1}$]	0.0734 ± 0.0002	0.0790 ± 0.0003
$\tau^+\tau^-$ merge rate	$(3.37 \pm 0.04)\,\%$	$(2.91 \pm 0.04)\,\%$

6. 3D Track Reconstruction

A 3D track reconstruction is possible by combining the information from axial and stereo wires. The following chapter describes how to extract three dimensional hit coordinates from the stereo wires and a 2D track estimate. These coordinates determine the full 3D helix which can be extrapolated to the beamline to obtain the z-vertex position. A rough estimate for the accuracy of such a reconstruction is derived from the uncertainty of the hit positions.

6.1. Stereo wire crossing

A 3D track reconstruction, for example with a fit, requires a set of points in space. Since a single wire hit contains no information about where along the wire a particle passed, the 3D hit point position has to be obtained from the crossing point of two non-parallel wires, in other words by combining axial and stereo hits of adjacent superlayers. The radial distance between the axial and stereo layers means that the longitudinal position at the two wires is in general not the same, so it cannot be derived precisely from only one crossing point. The combination of three hits from different superlayers allows to estimate the longitudinal position of the middle wire by interpolation between the two crossing points (see fig. 6.1). Note that this is only possible if the two outer wires are coplanar, that is for one stereo and two axial wires. For three skew wires the interpolation cannot be uniquely defined.

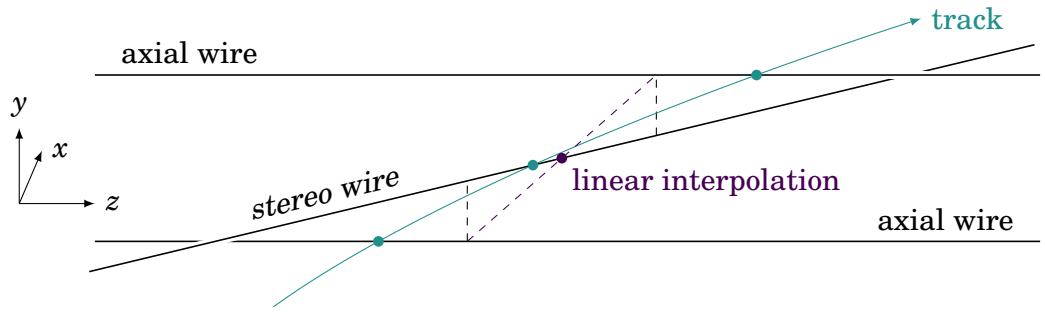

Figure 6.1.: The hit position along a wire can be estimated from the crossing of axial and stereo wires in adjacent superlayers. Due to the radial distance between the wires at least three layers must be combined to obtain a precise space point.

6.1.1. Crossing point between stereo wire and 2D track

Since a linear interpolation ignores the track curvature, a better estimate for the longitudinal position is obtained from the crossing point between the stereo wire and the 2D track estimate given by the track finder described in the previous chapter (see fig. 6.2). This approach has the additional advantage that the 2D track combines the information from all axial hits, so the track position at a certain radius is less uncertain than a single axial hit at the same radius.

To calculate the crossing point in the transverse plane, it is convenient to use a normal parametrization similar to eq. (5.1) to describe the wire:

$$\begin{aligned} \rho &= x \cdot \cos\theta + y \cdot \sin\theta \\ &= r\cos\varphi \cdot \cos\theta + r\sin\varphi \cdot \sin\theta\,, \\ r(\varphi) &= \frac{\rho}{\cos(\varphi - \theta)}\,, \end{aligned} \tag{6.1}$$

where (x, y) and (r, φ) are the cartesian and polar coordinates of a general point on the wire. The minimal distance ρ of the projected wire to the origin and the angle θ to the normal can be derived from the transverse wire coordinates at both endplates $(r_{\text{wire}}, \varphi_F)$ and $(r_{\text{wire}}, \varphi_B)$, where the index F/B denotes the forward/backward endplate:

$$\rho = r_{\text{wire}} \cdot \cos\left(\frac{\varphi_B - \varphi_F}{2}\right), \qquad \theta = \frac{\varphi_B + \varphi_F}{2}\,.$$

The track is parametrized according to eq. (5.3) by φ_0 and $\frac{1}{r} \equiv \omega$,[1] replacing the track segment coordinates $(r_{\text{TS}}, \varphi_{\text{TS}})$ by a general point on the track:

$$\begin{aligned} \omega &= \frac{2}{r} \cdot \sin(\varphi_0 - \varphi)\,, \\ r(\varphi) &= \frac{2}{\omega} \cdot \sin(\varphi_0 - \varphi)\,. \end{aligned} \tag{6.2}$$

The crossing point is found by equating eqs. (6.1) and (6.2):

$$\begin{aligned} \rho \cdot \omega &= 2 \cdot \sin(\varphi_0 - \varphi_{\text{cross}}) \cdot \cos(\varphi_{\text{cross}} - \theta) \\ &= \sin(\varphi_0 - 2\varphi_{\text{cross}} + \theta) + \sin(\varphi_0 - \theta)\,, \\ \varphi_{\text{cross}} &= \frac{1}{2} \cdot \left(\varphi_0 + \theta - \arcsin\left(\rho \cdot \omega - \sin(\varphi_0 - \theta)\right)\right). \end{aligned} \tag{6.3}$$

The radial component $r_{\text{cross}} = r(\varphi_{\text{cross}})$ can be obtained either from eq. (6.1) or eq. (6.2). Unfortunately, the expressions for $(r_{\text{cross}}, \varphi_{\text{cross}})$ contain several nested trigonometric functions that can only be evaluated with a sequence of lookup tables in the FPGA,

[1] To avoid confusion between the track radius and the radial components of the polar coordinates, the track curvature ω is used in this chapter to describe the track.

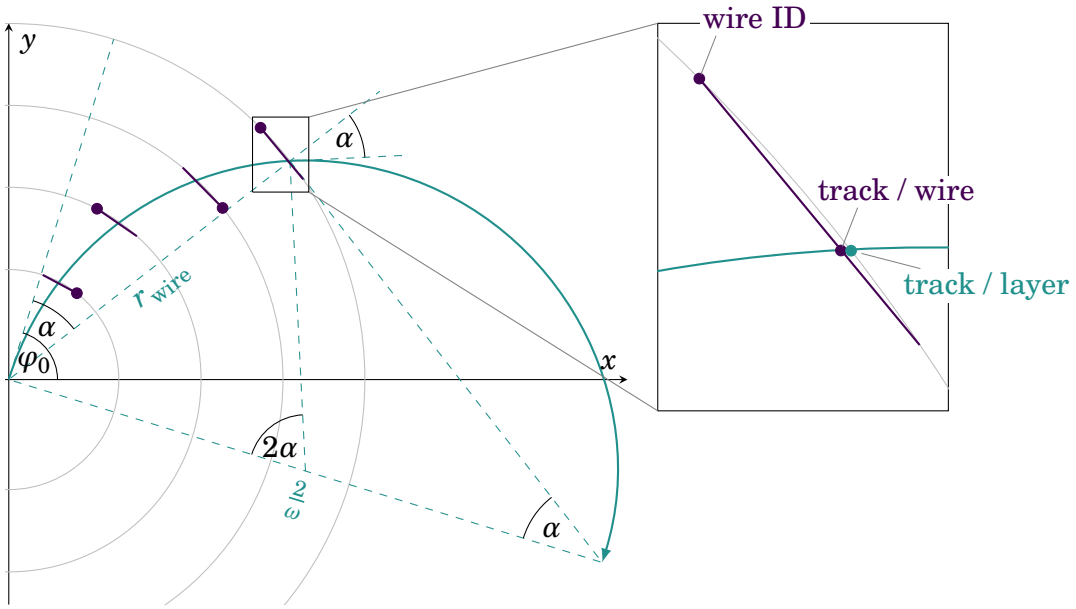

Figure 6.2.: Projection onto the transverse plane of a track and the stereo wires of related hits. The hit position along a stereo wire can be estimated from the crossing point of the wire and the track. The wire ID is defined at the backward end. The crossing point with the stereo layer gives an approximation of the crossing point with the wire.

with some uncertainty added in each step due to the inevitable rounding. An approximation of the crossing point is obtained by neglecting the variation of the wire radius and replacing eq. (6.1) by $r(\varphi) = r_{\text{wire}} = \text{const}$. This is equivalent to calculating the crossing point between the track and the stereo layer (see fig. 6.2). The crossing point is found by solving eq. (6.2) for φ:

$$\varphi'_{\text{cross}} = \varphi_0 - \arcsin\left(\frac{1}{2} \cdot r_{\text{wire}} \cdot \omega\right) \equiv \varphi_0 - \alpha\,. \tag{6.4}$$

Here the *crossing angle* α is introduced, which will appear again in the longitudinal projection of the track and when the drift time is included in the calculation of the hit position. The angle α can be calculated by a single lookup table and does not depend on the hit coordinates, so it can be precalculated for each track and superlayer.

6.1.2. Including the drift time

So far it was assumed that the track actually crosses a stereo wire. In fact there is usually a distance between the sense wire and the track that is proportional to the drift time. To calculate the precise hit position, some directional information is needed in addition to the drift time (see fig. 6.3). With the left/right state determined by the

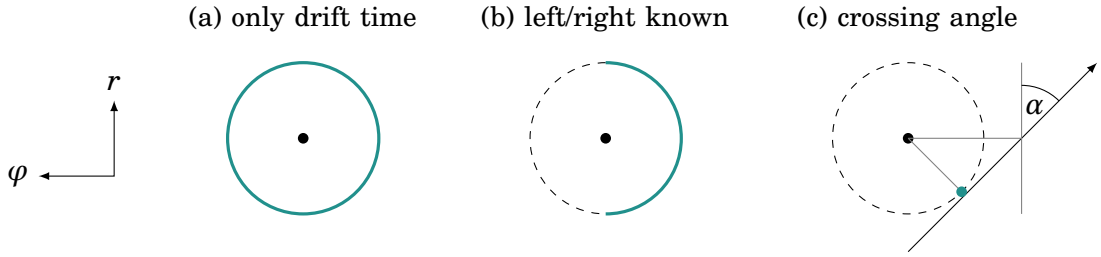

Figure 6.3.: The hit position can be calculated from the wire coordinates and the drift time. (a) Without additional information, the direction of the drift correction is unknown and the correct position is somewhere on the drift circle. (b) The track segment finder provides a binary information about the side of the hit position. (c) Using the crossing angle α provided by the 2D track estimate, a precise hit position can be obtained.

track segment finder (see section 4.1.1) the possible hit positions can be constrained to a half circle. The best approximation is to assume a crossing angle of $\alpha = 0°$ and set the hit position to

$$\begin{aligned} \varphi_{\text{hit}} &= \varphi_{\text{wire}} \pm \arcsin\left(\frac{v_{\text{drift}} \cdot t_{\text{drift}}}{r_{\text{wire}}}\right) \approx \varphi_{\text{wire}} \pm \frac{v_{\text{drift}} \cdot t_{\text{drift}}}{r_{\text{wire}}}, \\ r_{\text{hit}} &= r_{\text{wire}}. \end{aligned} \tag{6.5}$$

The sign of the correction depends on the left/right state. If it is undecided, only the original wire position can be used as hit position. If an estimate of the 2D track is given in addition to the left/right state, the hit position of an axial wire can be determined more precisely from the crossing angle:

$$\begin{aligned} \varphi_{\text{hit}} &= \varphi_{\text{wire}} \pm \arcsin\left(\frac{v_{\text{drift}} \cdot t_{\text{drift}} \cdot \cos\alpha}{r_{\text{wire}}}\right), \\ r_{\text{hit}} &= r_{\text{wire}} \pm v_{\text{drift}} \cdot t_{\text{drift}} \cdot \sin\alpha. \end{aligned} \tag{6.6}$$

The axial hit positions obtained in this way can be used to improve the precision of the 2D track parameters φ_0 and ω. For the stereo wires the correct treatment of the drift times is more complex and should take into account both the skewing angle of the wire and the polar angle of the track. Since the latter is a priori unknown, while the former is between 45 mrad and 74 mrad (depending on the superlayer), a good approximation is given by applying eq. (6.5) or eq. (6.6) to the transverse coordinates of the forward and backward end of the wire. The projection of the wire is then a straight line between the end point positions corrected by the drift time and the crossing with the track is calculated as before.

6.2. Extrapolation to z-vertex

The longitudinal hit coordinates can be derived by projecting the 2D crossing point back to the stereo wire. The following relation holds for the longitudinal (z) and azimuthal (φ) coordinates of the wire endpoints and the crossing point, where the coordinates at the forward/backward endplate are again denoted by an index F/B:

$$\frac{z_{\text{cross}} - z_B}{z_F - z_B} = \frac{\sin(\varphi_{\text{cross}} - \varphi_B)}{\sin(\varphi_F - \varphi_{\text{cross}}) + \sin(\varphi_{\text{cross}} - \varphi_B)}. \tag{6.7}$$

Similar to the approximation in eq. (6.4) this expression can be simplified by calculating the 2D distance along the circular arc instead of the straight connection between the wire endpoints:

$$\frac{z_{\text{cross}} - z_B}{z_F - z_B} = \frac{\varphi'_{\text{cross}} - \varphi_B}{\varphi_F - \varphi_B}. \tag{6.8}$$

As before, the advantage of this approximation is that the trigonometric expressions have been dropped. In addition, the denominator of the simplified calculation is constant within each layer, so the longitudinal coordinate z_{cross} is directly proportional to the angle φ_{cross}.

With the three dimensional crossing point coordinates the full helix can be reconstructed. Since the transverse helix parameters are already known, it is convenient to project the helix to the longitudinal plane spanned by r and z and do another two dimensional reconstruction. A possible parametrization of a helix with track vertex on the z-axis is given by (compare eq. (4.2))

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{1}{\omega} \cdot \left(\sin\left(\omega\mu - \varphi_0\right) + \sin\varphi_0\right) \\ \frac{1}{\omega} \cdot \left(\cos\left(\omega\mu - \varphi_0\right) - \cos\varphi_0\right) \\ z_0 + \cot\theta_0 \cdot \mu \end{pmatrix}, \tag{6.9}$$

where θ_0 is the polar angle of the momentum at the track vertex, z_0 is the longitudinal vertex displacement and φ_0 and $\omega \equiv \frac{1}{r}$ are the established circle parameters of the transverse projection (defined in eq. (5.3)). The radial component in this parametrization is given by

$$\begin{aligned} r(\mu) = \sqrt{x^2 + y^2} &= \sqrt{\frac{2}{\omega^2} \cdot \left(1 + \sin\left(\omega\mu - \varphi_0\right) \cdot \sin\varphi_0 - \cos\left(\omega\mu - \varphi_0\right) \cdot \cos\varphi_0\right)} \\ &= \sqrt{\frac{2}{\omega^2} \cdot \left(1 - \cos(\omega\mu)\right)} = \left|\frac{2}{\omega} \cdot \sin\frac{\omega\mu}{2}\right|. \end{aligned} \tag{6.10}$$

For a track that does not curl back within the volume of the drift chamber ($p_\perp > 256\,\text{MeV}/c$), the sine is always positive and the modulus can be dropped. To get a

helix representation in the r-z-plane eq. (6.10) is inverted and substituted into the z component of eq. (6.9):

$$\begin{aligned} z(r) &= z_0 + \cot\theta_0 \cdot \frac{2}{\omega} \arcsin\frac{r\omega}{2} \\ &= z_0 + \cot\theta_0 \cdot \frac{2\alpha(r)}{\omega}. \end{aligned} \tag{6.11}$$

The helix parameter μ can be identified as the arclength of the transverse track projection from the origin to a general point on the helix (compare fig. 6.2). The longitudinal track parameters z_0 and θ_0 can be obtained from a linear fit of eq. (6.11), using the points $\left(z_{\text{cross}}, \alpha\left(r_{\text{cross}}\right)\right)$ of each stereo superlayer. The z-vertex is given by the intercept of the resulting linear function, while the polar angle can be calculated from the slope.

6.3. Accuracy estimation

The following section gives an estimate for the z-vertex resolution that can be achieved with a linear fit under ideal circumstances and discusses the effect of various approximations and other errors, like the influence of a hit that was wrongly assigned to a track. The purpose is not to precisely calculate the resolution, which can be measured with much higher confidence by feeding simulated tracks to the actual reconstruction algorithm. Instead the intention is to give a rough order of magnitude for the resolution of the analytical reconstruction and its dependence on the track parameters, in particular on the curvature, and to motivate the neural network approach that will be introduced in chapter 7.

6.3.1. Ideal precision

The following derivation assumes one correctly assigned hit in each superlayer with known and correct left/right state. The wire coordinates alone give a spatial resolution of $\approx \frac{1}{12}$ times the distance between wires or $\approx 0.4\,\text{cm}$. The uncertainty of the hit position according to eq. (6.6) depends on the uncertainty of the drift length and the event time. The resolution of the drift length is $\approx 100\,\mu\text{m}$ [37]. The final precision of the event time is not yet known. A timing jitter of 2 ns would correspond to a drift length error of 80 µm. For the purpose of a very rough estimate, a spatial resolution of 200 µm is assumed for the hit position with drift time. Assuming that the uncertainty of the 2D track parameters scales linearly with the resolution of the input values, the resolution of φ_0 and ω can be extrapolated from table 5.4 to $\Delta\varphi_0 \approx 0.03°$ and $\Delta\omega \approx 2 \times 10^{-5}\,\text{cm}^{-1}$.

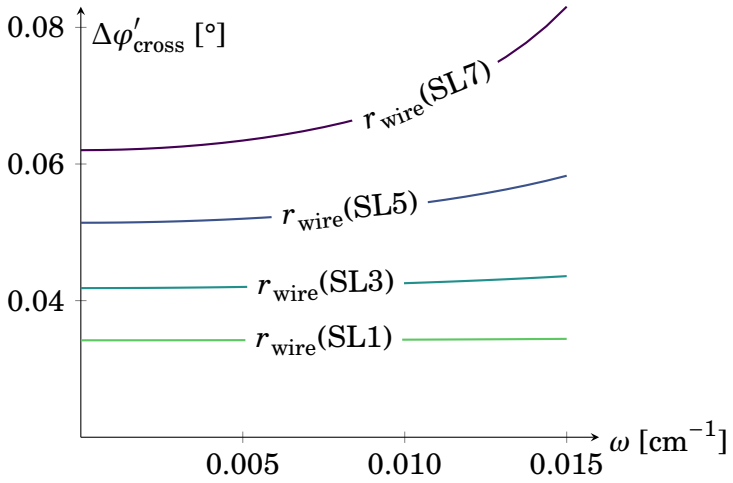

Figure 6.4.: Uncertainty of the crossing point angle φ'_{cross} according to eq. (6.12) for the different stereo superlayers, depending on the track curvature.

The uncertainty of the crossing point according to eq. (6.4) is

$$\Delta\varphi'_{\text{cross}}(r_{\text{wire}},\omega) = \sqrt{(\Delta\varphi_0)^2 + \frac{r_{\text{wire}}^2\cdot(\Delta\omega)^2}{4-(r_{\text{wire}}\omega)^2}} \approx 0.03° \text{ to } 0.08°. \tag{6.12}$$

Since the track is extrapolated from the origin due to the constraint of the track finding, the uncertainty increases with r_{wire}. The functions for all stereo superlayers are shown in fig. 6.4. The dependence on the curvature ω becomes significant for the last stereo superlayer.

The uncertainty of the longitudinal hit coordinates is estimated from eq. (6.8). The uncertainty of the wire end position after the drift time correction is $\Delta\varphi_B \approx \frac{200\,\mu\text{m}}{r_{\text{wire}}}$. The difference $\varphi_F - \varphi_B$ has no uncertainty, because the drift time correction for both wire ends is the same. The scale factor is therefore not affected by an error in the drift time. The uncertainty of z_{cross} is then given by

$$\Delta z_{\text{cross}} = \frac{z_F - z_B}{\varphi_F - \varphi_B}\cdot\sqrt{(\Delta\varphi'_{\text{cross}})^2 + (\Delta\varphi_B)^2} \approx \begin{cases} 0.4\,\text{cm to } 1.4\,\text{cm} & \text{for high } p_\perp \\ 0.4\,\text{cm to } 1.8\,\text{cm} & \text{for low } p_\perp, \end{cases}$$

where the uncertainty increases again with the radius. It is reasonable to assume that the z-vertex can be reconstructed with a resolution similar to the individual hit coordinates, with some additional uncertainty for flat polar angles arising from the extrapolation to the beamline. Indeed it has been shown that a z-vertex resolution on the order of 1 cm can be achieved under ideal conditions [4, 68]. It is also important to note that the resolution depends directly on the uncertainty of the hit coordinates. Due to the small skewing angle, the uncertainty of the longitudinal component is two

orders of magnitude larger than for the transverse components. Including the drift times in the 3D reconstruction is therefore essential for a sufficient track parameter resolution. Early studies in the Belle II technical design report show that with the wire coordinates alone a z-vertex resolution of only 9 cm can be achieved [37].

6.3.2. Errors due to approximations

Three major approximations were introduced in this chapter for the reconstruction of the 3D track parameters. All of them are similarly applied in the 3D fitter module developed by the trigger group. The first is to ignore the crossing angle α in applying the drift times and calculating the hit position according to eq. (6.5), which results in a shift of the hit coordinates towards the wire and away from the track. The magnitude of the error increases linearly with the drift length and quadratically with the crossing angle, in other words it is worst for the outer superlayers and low transverse momentum, ranging from $\approx 0.1\,\mathrm{mm}$ to $1\,\mathrm{mm}$. The sign depends on the position of the wire relative to the track, so in principle it is random. However, if the hit selection favors higher or lower wire ID, as is the case with the axial hit selection in the track finder, a bias is introduced.

The second approximation is made in calculating the crossing point with the stereo wires according to eq. (6.4) instead of eq. (6.3). As can be seen in the inlay in fig. 6.2, the error increases again with the crossing angle and is largest in the center of the wire. For low transverse momenta at the edge of the track finding acceptance ($\omega \approx 0.015\,\mathrm{cm}^{-1}$) the shift in $\varphi'_{\mathrm{cross}}$ can be up to 0.2°. The sign of the shift depends on the charge of the track and is the same for all superlayers. However, the alternating sign of the skewing angle means that the shift of the longitudinal coordinate z_{cross} alternates likewise between superlayers, so in the end the effect is an increased uncertainty in the longitudinal fit.

The last approximation happens in the projection of the azimuthal to the longitudinal crossing point coordinates in eq. (6.8). The error with respect to the precise expression in eq. (6.7) depends on the superlayer and on z_{cross} itself. A positive/negative shift is introduced for hits in the forward/backward half of a wire, in other words hits are shifted away from the wire center. The shift is largest for the innermost stereo layer, where it goes up to 3 mm.

All three approximations can have a significant effect on the reconstruction, especially for low transverse momentum, while for nearly straight tracks the errors are negligible. Accordingly, the 3D fitter is very sensitive to the track curvature [66].

6.3.3. Other limitations of an analytic reconstruction

In addition to the increased uncertainty from the approximations, some other issues have been neglected or simplified in the accuracy estimation so far. First, the drift velocity is only approximately constant, so the correct relation between the drift

length and the drift time should be modeled by a polynomial of at least second order. While this is in principle possible, a nonlinear function is more sensitive to an event time jitter. For undetermined left/right state the hit position cannot be corrected at all, while for an incorrect left/right state the correction actually increases the error. Therefore, the uncertainty of the hit position is in fact larger than the ideal estimate.

The last and most severe kind of error is the selection of a hit that does not belong to the track at all. Additional hits can originate either from background or from tracks that are close to each other, causing a mismatch of hits. Due to the low number of hits (at most one hit from each of the four stereo superlayers), a single wrong hit has a strong effect on the result of the fit. A reliable stereo hit selection is therefore essential.

All these problems motivate the investigation of an alternative reconstruction method. The next chapter introduces an algorithm based on neural networks, which are capable of learning nonlinear functions from data samples. In particular, they may be able to find a better approximation for the complicated trigonometric expressions and the drift time correction. Although a neural network approach cannot solve all of the discussed difficulties, it is more robust against many kinds of imperfection and offers a complementary solution to the analytic reconstruction.

7. z-Vertex Reconstruction with Neural Networks

The 3D track reconstruction is the task of two independent algorithms. The original approach suggested in the Belle II technical design report [37] is a linear least squares fit of eq. (6.11). At that time only the wire coordinates were used without taking the drift times into account, leading to a very poor resolution of the z-vertex. In addition, a similar algorithm developed for Belle had never come into use due to insufficient performance [69]. Therefore, an alternative approach based on neural networks with the drift times as input was investigated in [4] and gave the desired resolution of $\mathcal{O}(1\,\mathrm{cm})$ in selected sectors of the track parameter phase space. The neural network algorithm was also proven to be robust against background hits [68]. Within this thesis, the algorithm is extended from a proof of principle to a realistic setup for the full track parameter phase space.

In the meantime, the analytical algorithm was also improved, including the drift times and achieving the same resolution of $\mathcal{O}(1\,\mathrm{cm})$ for clean events. The two algorithms are intended to run in parallel, complementing each other. A comparison of the respective performance will be shown within the track trigger efficiency studies in chapter 8.

The following chapter first introduces the basics of neural networks in general and the training algorithm. Then the application in the neural z-vertex trigger is explained, including the input and output representation and the combination of several networks for specific tasks. The network structure is optimized and the influence of background hits is investigated, then the networks are adapted for fixed point calculation in the FPGA. In the last section, networks are prepared for the first real tests of the trigger with cosmic rays.

7.1. Neural networks

An artificial neural network consists of a collection of computing units called *neurons*, which mimic the behavior of biological neurons in the human brain. The terms “neural network” and “neuron” are used throughout this thesis as shorthand of the more precise terms “artificial neural network” and “artificial neuron”, which refer to a mathematical model of the brain. The biological brain cells are explicitly called “biological neurons” here for clarity. There are two main motivations for the study

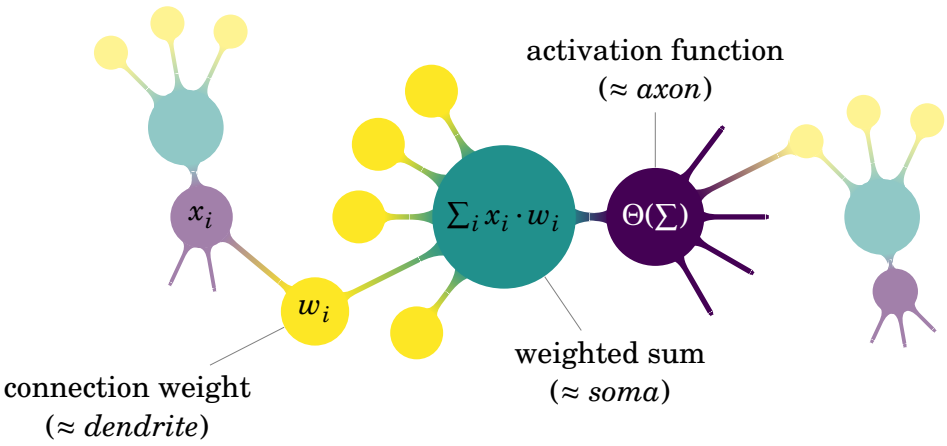

Figure 7.1.: Schematic drawing of a neuron. Inputs are received from other neurons, multiplied by a connection specific weight and summed up. The sum is fed into an activation function similar to a step function. The result is sent to other neurons.

of artificial neural networks. The first is to create algorithms for tasks where the human mind is superior to traditional computer algorithms, such as image or speech recognition. The second is to gain a deeper understanding of the human brain itself by simulating its behaviour. For the z-vertex trigger, the intent is clearly the first: to find an algorithm as efficient as possible for the problem at hand. For this purpose the artificial neurons do not need to be completely realistic and can ignore some properties that are not relevant for the problem or that do not match the timing constraints of the trigger. Accordingly, the following simplified description of biological neurons focuses on the parts that are modeled in the artificial neuron. A more detailed explanation is given for example in [70].

Biological neurons consist of three main parts: the cell body or *soma* and two types of neurites (fibers that connect the cell to other neurons), called *dendrites* and *axons*. Each neuron has multiple branching dendrites, which receive synaptic signals, and a single axon, which transmits electrical pulses to other neurons. The axon can branch into several axon terminals, which are typically connected to the dendrites of other neurons. When an electrical pulse is sent through the axon, neurotransmitters are released in the connected dendrites that either increase or decrease the electrical potential in the soma. When the potential exceeds a certain threshold, a new pulse is sent through the axon.

This behavior is modeled in an artificial neuron by mathematical functions (see fig. 7.1). A simple type is the *perceptron*, which ignores all time dependent properties of the biological neuron. The connections between neurons are represented by individual numerical weights, which combine the effect of the neurotransmitter type (exciting or inhibiting), the amount of neurotransmitters released by a single pulse and the

pulse frequency of input signals. The total potential is calculated as the sum of a given input vector multiplied by the connection weights. The output signal is the result of an activation function operating on the weighted sum of inputs. To model the all-or-none pulse of the biological axon a step function would be used, but for the training process a continuous activation function is necessary. In summary, a neuron with N input connections computes

$$y = f_{\text{act}}\left(\sum_{i=1}^{N} x_i \cdot w_i\right), \tag{7.1}$$

where x_i are the input values, w_i are the connection weights, f_{act} is the activation function and y is the output value of the neuron.

In a neural network many neurons are connected such that the inputs to a neuron are given by the outputs of other neurons. The network can be represented by a directed graph, where the neurons are represented by nodes and the connections are represented by edges. The interface is provided by N_{in} *input nodes*, which receive input values from the outside, and N_{out} *output nodes*, whose output values form the result of the network. All other neurons are called *hidden nodes*, since their input and output values are not visible outside of the network. Note that the input nodes are not actually neurons according to the previous definition. Each input node receives a single input value and sends it unaltered to several other nodes, that is the sole purpose of the input nodes is to distribute the input values. The artificial neural network forms a function $f_{\text{NN}} : \Re^{N_{\text{in}}} \to \Re^{N_{\text{out}}}$, which is defined by the structure of the network and the connection weights.

7.1.1. Multi Layer Perceptron

Neural networks are typically arranged in layers and can be divided in feed-forward networks, where nodes receive input only from the previous layers, and recurrent networks, which may contain arbitrarily complex loops. The latter do not fulfill the requirements of the trigger for a deterministic runtime and pipelined operation. Among feed-forward networks, the structure of the Multi Layer Perceptron (MLP) is particularly suited for the trigger demands. An MLP consists of at least three layers of perceptrons, with each layer fully connected to the next one and only the next one, that is there are no shortcut connections that skip layers. Missing connections are formally possible by setting specific connection weights to 0, but they are not a priori built into the structure. The graph of a general MLP is shown in fig. 7.2. The structure of the MLP is ideally suited for a pipelined implementation, with all nodes in a layer computed in parallel and each layer depending only on the directly preceding layer.

It can be proven that a three layer MLP is capable of approximating arbitrary continuous real functions to any degree of accuracy, given a sufficiently large but finite number of hidden nodes and an appropriate set of connection weights [71, 72].

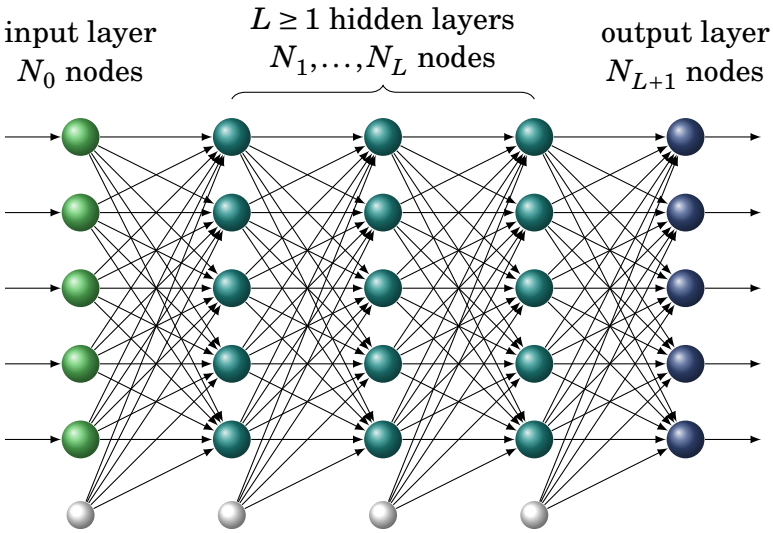

Figure 7.2.: Structure of a Multi Layer Perceptron (MLP). Neurons are arranged in fully feed-forward connected layers. An MLP has one layer of input nodes, one layer of output nodes and at least one layer of hidden nodes. The small white nodes denote bias nodes, which effectively set an individual activation threshold for each node (see eq. (7.2)).

The activation function of the hidden layer is required to be nonconstant, bounded, monotonically increasing and differentiable. Note that the proof does not imply that a three layer structure is the most efficient solution for a given problem. However, for the z-vertex trigger, the problem is comparatively simple and furthermore limited by the uncertainty of the input values (the formal proof assumes the existence of a function that describes the data perfectly). In the proof of principle studies in [4] a single hidden layer with 60 to 80 nodes was sufficient to achieve the desired resolution (more precisely, the networks contain a factor of 3 times as many hidden nodes as input nodes).

As activation function the hyperbolic tangent $\tanh\left(\frac{x}{2}\right)$ is used both for hidden and output nodes. For large $|x|$ this function behaves like a step function, except that its codomain is $[-1,1]$ instead of $[0,1]$. However, in contrast to the step function the hyperbolic tangent has a well defined and easy to calculate derivative, which is important for the training of the network. The threshold of the activation function can be effectively adjusted by adding a fixed bias b to the neuron sum, so eq. (7.1) becomes

$$y = \tanh\left(\frac{1}{2}\left(\sum_{i=1}^{N} x_i \cdot w_i + b\right)\right) \equiv \tanh\left(\sum_{i=0}^{N} \frac{x_i \cdot w_i}{2}\right). \tag{7.2}$$

In the second part of the equation, the bias is written as an additional *bias node* with fixed input $x_0 = 1$ and a weight $w_0 = b$ corresponding to the bias.

The full function $\vec{y}(\vec{x})$ of an MLP with input values $\vec{x}$ and output values $\vec{y}$ is defined by (compare fig. 7.2):

$$\begin{aligned}
\text{hidden layers: } & L & & L \geq 1 \\
\text{nodes per layer: } & N_l & & l \in [0, L+1] \\
\text{weights: } & w_{lij} & & l \in [1, L+1],\ i \in [0, N_{l-1}],\ j \in [1, N_l] \\
\text{input nodes: } & n_{0j} = x_j & & j \in [1, N_0] \\
\text{bias nodes: } & n_{l0} = 1 & & l \in [0, L] \\
\text{other nodes: } & n_{lj} = \tanh\left(\sum_{i=0}^{N_{l-1}} \frac{n_{(l-1)i} \cdot w_{lij}}{2} \right) & & l \in [1, L+1],\ j \in [1, N_l] \\
\text{outputs: } & y_j = n_{(L+1)j} & & j \in [1, N_{L+1}].
\end{aligned}$$

The number of hidden layers L and their size $N_l, l \in [1, L]$ are user defined parameters of the network. The size of the input and output layer has to match the dimension of $\vec{x}$ and $\vec{y}$. The weights w_{lij} are determined automatically by a training algorithm. The total number of weights is given by

$$N_w = \sum_{l=1}^{L+1} (N_{l-1} + 1) \cdot N_l \,. \tag{7.3}$$

7.1.2. Training

The network is initialized with random weights in the range $[-0.1, 0.1]$, which are then adjusted by a supervised training algorithm such that the output values of the network for a given set of input samples approximate a corresponding set of target values. The existence of explicit target values distinguishes supervised from unsupervised training methods, which aim to find correlations in the input samples. For the supervised training, the set of training data consists of N_{train} pairs of input and target vectors $(\vec{x}_k, \vec{t}_k), k \in [1, N_{\text{train}}]$, called *training patterns*. The quality of a network with a given weight set w is measured by an error function

$$E(w, \vec{x}, \vec{t}) = \sum_{k=1}^{N_{\text{train}}} \left(\vec{y}(\vec{x}_k, w) - \vec{t}_k \right)^2 . \tag{7.4}$$

The weights are optimized by a gradient descent. A single *training epoch* consists of evaluating the error function for all samples and then updating the weights according to the gradient of the error function:

$$w_{lij}^{\text{epoch}} = w_{lij}^{\text{epoch}-1} - \Delta \cdot \frac{\partial E}{\partial w_{lij}} (w^{\text{epoch}-1}, \vec{x}, \vec{t}), \tag{7.5}$$

where the learning rate Δ scales the step size of the weight updates. An alternative would be to update the weights after each individual training pattern.

Calculating the gradient is only possible for a differentiable activation function. The choice of the hyperbolic tangent as activation function has the additional advantage that it can be expressed by itself:

$$\frac{\mathrm{d}}{\mathrm{d}x}\tanh\left(\frac{x}{2}\right) = \frac{1}{2}\cdot\left(1-\tanh^2\left(\frac{x}{2}\right)\right).$$

The derivative $\frac{\partial E}{\partial w_{lij}}$ of the full error function can be calculated by repeated application of the chain rule:

$$\begin{aligned}
\frac{\partial E}{\partial w_{(L+1)ij}} &= \sum_{k=1}^{N_{\text{train}}} \underbrace{2\cdot(n_{(L+1)j,k}-t_{j,k})\cdot\frac{1}{2}\cdot(1-n_{(L+1)j,k}^2)}_{\equiv\delta_{(L+1)j,k}}\cdot\frac{n_{Li,k}}{2}, \\
\frac{\partial E}{\partial w_{Lij}} &= \sum_{k=1}^{N_{\text{train}}} \underbrace{\sum_{a=1}^{N_{L+1}}\delta_{(L+1)a,k}\cdot\frac{w_{(L+1)ja}}{2}\cdot\frac{1}{2}\cdot(1-n_{Lj,k}^2)}_{\equiv\delta_{Lj,k}}\cdot\frac{n_{(L-1)i,k}}{2}, \\
&\vdots \\
\frac{\partial E}{\partial w_{1ij}} &= \sum_{k=1}^{N_{\text{train}}} \underbrace{\sum_{a=1}^{N_2}\delta_{2a,k}\cdot\frac{w_{2ja}}{2}\cdot\frac{1}{2}\cdot(1-n_{1j,k}^2)}_{\equiv\delta_{1j,k}}\cdot\frac{n_{0i,k}}{2}.
\end{aligned}$$

By introducing the helper variables $\delta_{lj}, l \in [1, L+1]$, the derivative of each layer depends only on the directly neighboring layers. Because of the order of the calculation the procedure is called *backpropagation*: starting from the output layer, the error is propagated back through the hidden layers, updating the weights in the process.

While the gradient components are fairly straightforward to calculate, the topology of the N_w dimensional error surface $E(w)$ is very hard to analyze. Therefore, it is not trivial to choose an appropriate learning rate Δ for the updating rule eq. (7.5). A small learning rate on a relatively flat error surface leads to a very slow convergence, while a large learning rate on an uneven error surface can lead to erratic oscillations around the optimum (see fig. 7.3). In addition, the shape of the gradient may be very different with respect to each weight, making it even more difficult to choose a global learning rate.

To overcome this problem, various adaptive learning algorithms have been developed, which take into account not only the gradient in the current training epoch, but also the behavior since the last epoch. The training algorithm chosen for the z-vertex trigger is a variant of the Resilient backPROPagation algorithm (RPROP) [73]. The global learning rate Δ is replaced by individual, variable update values $\Delta_{lij}^{\text{epoch}}$ for each weight. At each epoch, the update value is adapted depending on the relative sign of

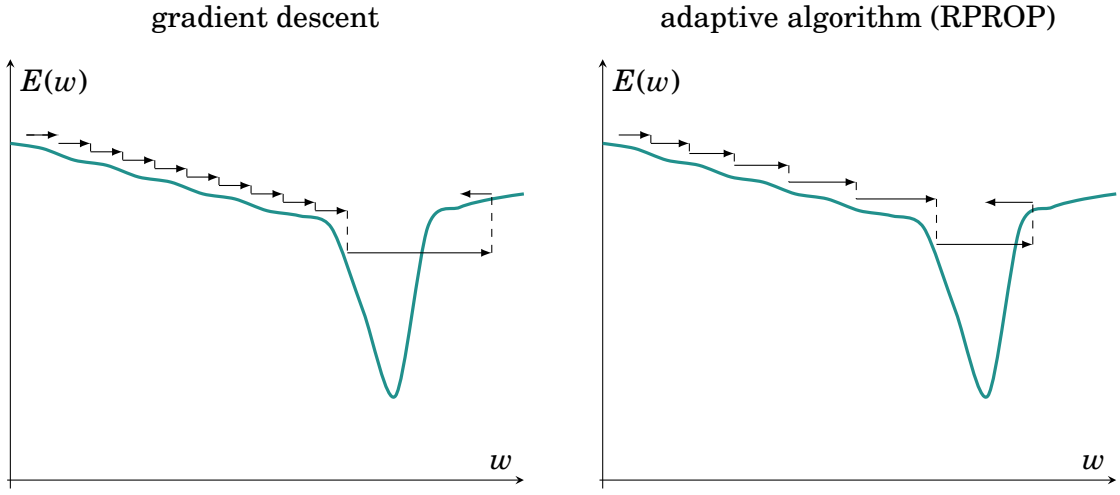

Figure 7.3.: Irregular error surface where gradient descent (left) may have problems to find the minimum (redrawn after Figure 2 in [73]). The descent is either very slow in the flat region or the minimum may be missed. An adaptive algorithm (right) increases the step size as long as the gradient sign is constant and decreases it when the sign changes.

the gradient compared to the previous epoch:

$$\Delta_{lij}^{\text{epoch}} = \Delta_{lij}^{\text{epoch}-1} \cdot \begin{cases} \eta^+ & \text{if } \frac{\partial E}{\partial w_{lij}}^{\text{epoch}} \cdot \frac{\partial E}{\partial w_{lij}}^{\text{epoch}-1} > 0 \\ \eta^- & \text{if } \frac{\partial E}{\partial w_{lij}}^{\text{epoch}} \cdot \frac{\partial E}{\partial w_{lij}}^{\text{epoch}-1} < 0 \\ 1 & \text{if } \frac{\partial E}{\partial w_{lij}}^{\text{epoch}} \cdot \frac{\partial E}{\partial w_{lij}}^{\text{epoch}-1} = 0 \end{cases}$$
$$0 < \eta^- < 1 < \eta^+ .$$

The increase and decrease factors are taken from [73] to be $\eta^+ = 1.2$ and $\eta^- = 0.5$. Further free parameters are the initial update value Δ_0 and an upper bound $\Delta_{\max}$ for the update values. According to the authors, both have only a minor influence on the training, so the default values are used here as well. The learning rule eq. (7.5) is replaced by

$$w_{lij}^{\text{epoch}} = w_{lij}^{\text{epoch}-1} - \Delta_{lij}^{\text{epoch}} \cdot \text{sign}\left(\frac{\partial E}{\partial w_{lij}}(w^{\text{epoch}-1}, \vec{x}, \vec{t})\right).$$

The weight update depends only on the sign of the gradient, not on the steepness of the slope. The step size is given solely by the update value Δ_{lij}.

For the training the open source library FANN[1] is used. It implements the "improved RPROP algorithm without weight backtracking" (iRPROP−) [74]. The algorithm is

[1] https://github.com/libfann/fann

very similar to the original RPROP, except that after a sign change the slope $\frac{\partial E}{\partial w_{lij}}$ is artificially set to 0 in addition to reducing the update value. The effect is that the corresponding weight is not modified at all until one epoch later.

7.1.3. Validation

The danger in training on a limited set of training samples is the phenomenon of *overfitting*. It occurs when the network has more degrees of freedom than data samples to learn from. In principle, the network can perfectly approximate all data points, but the resulting function is most likely not a good approximation of the underlying distribution from which the data samples were generated. At some point in the training, the network will start to learn only the given examples "by heart" and lose its capability for generalization, that is the performance for new data points will decrease.

To avoid this effect, the error function is monitored on an independent data set, called the *validation set*, which is not considered for calculating the weight updates. As soon as the error on the validation set increases, the training is stopped. The optimal set of weights is found in the epoch with the lowest validation error.

The importance of overfitting in the training depends strongly on the amount of training data. Figure 7.4 shows the evolution of the errorfunction eq. (7.4) during the training for the training set and the validation set. The validation set consists of 1000 data samples. The size of the training set is derived from the number of degrees of freedom, which is equal to the number of weights given by eq. (7.3), with a variable scaling factor from 0.1 to 10. The overfitting effect is seen clearly for a scaling factor < 1, that is for more degrees of freedom than training samples. Although the training error becomes very small, the validation error quickly starts to increase. For a larger scaling factor, the minimum in the validation error is less prominent. Instead, both the training error and the validation error grow flat. The lowest validation error among the five trainings is achieved for the highest amount of training data.

For the z-vertex trigger training samples are relatively easy to produce. In the present studies, networks are trained on simulated tracks, with the true track parameters as targets. If necessary, the networks can later be retrained on real data, for example to adapt to changing background conditions, with target values provided by the offline reconstruction. The largest training set in fig. 7.4 uses $\approx 30\,000$ samples. The trigger rate of the first level trigger will be 30 kHz, with an additional reduction factor of ≈ 3 at the high level trigger [37]. With this rate, the training data for a single network can be collected in less than a minute. Even considering a prescaling factor for background tracks, training data should be available in abundance.

In the following studies, the size of the training set is therefore fixed at 10 times the number of weights. For this scaling factor the training error and validation error in fig. 7.4 are almost identical, that is the system is sufficiently overconstrained that overfitting is not a problem. Nevertheless, the validation set is monitored to stop the

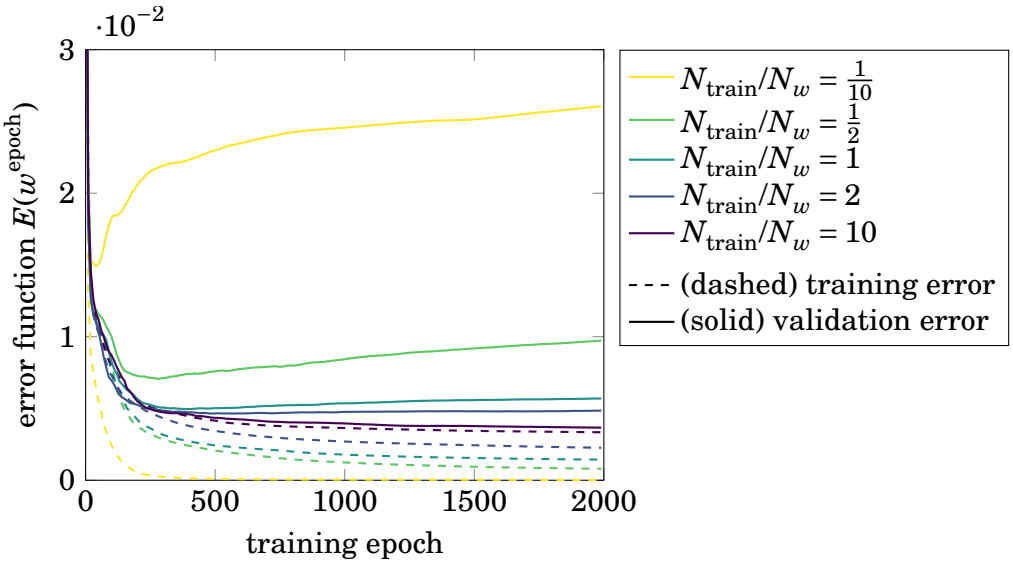

Figure 7.4.: Depending on the ratio between the number of training samples N_{train} and the degrees of freedom N_w, overfitting can occur. Although the error on the training set (dashed lines) goes down, the error on the independent validation set (solid lines) increases.

training when there is no more improvement. The stopping criterion is given by the condition

$$E(w^{\text{epoch}}, \vec{x}_{\text{valid}}, \vec{t}_{\text{valid}}) > E(w^{\text{epoch}-500}, \vec{x}_{\text{valid}}, \vec{t}_{\text{valid}}).$$

This means either that the validation error is increasing or that the fluctuations are larger than the gain, indicating that the training is stagnating. The interval of 500 epochs is chosen to avoid breaking off the training early due to small fluctuations. After 10000 epochs the training is stopped even if the stopping condition is not met.

Another issue that can occur in the training is that the error surface may have many local minima. The existence of multiple minima is certain, since the neural network function is symmetric under permutation of nodes in the same hidden layer. In addition to this ambiguity, there may exist local minima that are not also global minima, as illustrated in fig. 7.5. Depending on the initial weights, the training may converge to such a local minimum that is not a global optimum.

Figure 7.6a shows the distribution of optimal validation errors for 500 identical training runs that differ only in the set of initial weights. The results are within 25 % of the median, so no training run fails completely, but there is a significant spread, which hints that the training is stuck in a suboptimal local minimum. To increase the likelihood of finding a global optimum, the training is repeated several times with different initial weights, selecting the weight set which gives the best result among all runs.

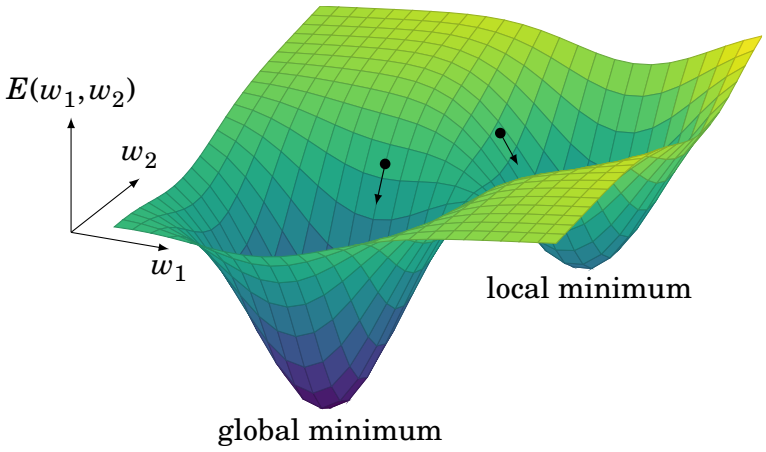

Figure 7.5.: Example error surface in two dimensions. Depending on the initial weights, the training may converge to a local minimum instead of the global one.

To roughly estimate the required number of repetitions, such a retraining is simulated by repeatedly drawing N_{run} values from the distribution in fig. 7.6a and selecting the best value of each bunch. Figure 7.6b shows the relative improvement of the median and the quartiles of each distribution. Not only the average validation error is

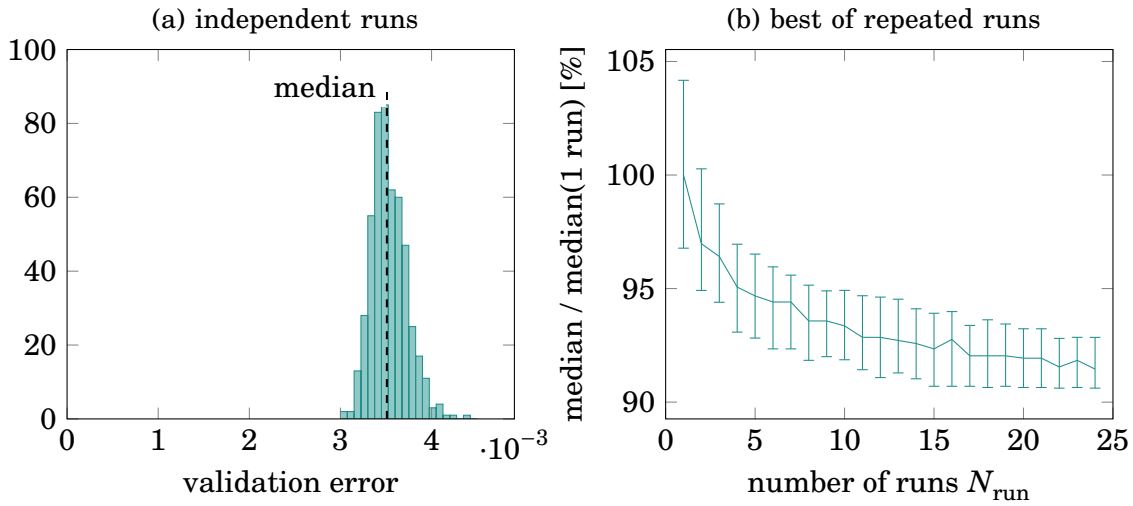

Figure 7.6.: (a) Distribution of optimal validation errors in 500 training runs with different initial weights. (b) Relative improvement by drawing N_{run} times from distribution (a) and keeping the best value. The line shows the median of 500 such tries, the intervals show the quartiles of each distribution, all normalized to the median for a single run.

reduced by the repetition, but also the spread of the distribution. The latter is important in optimization studies, where different neural network setups are compared to each other.

In a real training, the performance of the MLP is tested on a third data set of 5000 samples, called the *test set*, instead of comparing the validation error. For the optimization studies in this chapter, each training is repeated 5 times, which already gives a significant improvement. The final setup is trained 10 times to achieve optimal results.

7.1.4. Evaluation

To measure the performance of the neural network, the tracks are matched to simulated particles by the same algorithm that was used to evaluate the track finding, as described in section 5.2.2. Then the z-vertex estimate of the neural network z_{NN} is compared with the true z-vertex z_{MC} of the matched particles. Two types of quality measures can be defined: the z-vertex resolution and the positive rate for a given cut on the z-vertex.

Figure 7.7 shows a typical distribution of deviations between the estimated and true z-vertex for an example network. The z-vertex resolution can be derived from the width of the distribution. Since the distribution is dominated by outliers, the 95 % trimmed standard deviation $\sigma_z^{(95)}$ is chosen as a robust estimate for the z-vertex resolution (see appendix A.2). The resolution is useful to compare the performance of different setups without selecting a fixed cut.

The final quality measure for the z-vertex trigger is the efficiency for particles from the interaction region and the rejection rate for particles from outside of the interaction region. The final cut on the z-vertex can only be decided after the algorithm is optimized and the resolution is known. To allow a comparison of the efficiency already during the optimization studies, a z-vertex cut of 6 cm is imposed. This number is motivated from the recorded distribution of z-vertices in Belle, which is shown in fig. 3.9, and from the studies in [4, 68], which showed that a z-vertex resolution of 1 cm to 2 cm is possible under ideal conditions. The positive rate of the z-vertex trigger is defined as

$$\text{positive rate} = \frac{\text{number of tracks with } \left|z_{\mathrm{NN}}\right| \leq 6\,\mathrm{cm}}{\text{number of tracks found by the track finder}}.$$

The efficiency is given by the positive rate for particles with $z_{\mathrm{MC}} = 0\,\mathrm{cm}$. A special test sample with single muon tracks from the interaction point is generated to measure the efficiency. The momentum distribution is uniform in φ_0, $\cos\theta_0$ and $p_\perp^{-1}$, in other words it is uniform in the solid angle and the track curvature.

The total rejection rate depends on the distribution of background tracks. To get an estimate for the rejection, the positive rate depending on z_{MC} is studied. For this purpose, single muon tracks are generated with a uniform distribution of z-vertices in

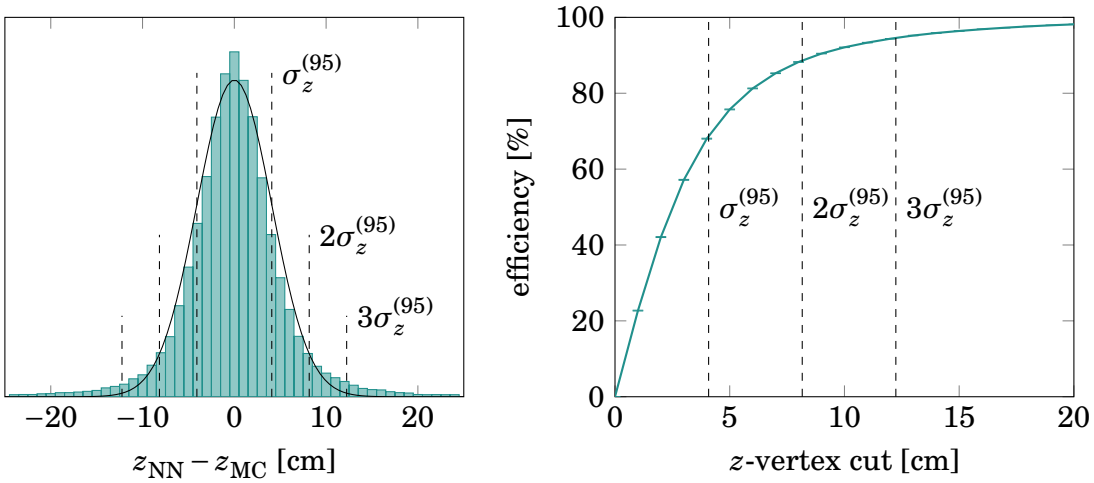

Figure 7.7.: Left: Typical shape of the z-vertex error distribution for the neural network estimate. The width is estimated by the 95 % trimmed standard deviation $\sigma_z^{(95)}$. A normal distribution with the same width is shown for comparison. Right: The z-vertex trigger efficiency with this neural network, depending on the z-vertex cut. For a cut of $3\sigma_z^{(95)}$ the efficiency is $(94.59 \pm 0.05)\,\%$.

the range $[-180\,\mathrm{cm}, 240\,\mathrm{cm}]$. The momentum distribution is again uniform in the solid angle and the track curvature. The same distribution is used to generate the training data, although different events are used in the training and evaluation. Figure 7.8 shows the z-vertex distribution of tracks that are found by the 2D track finding. The rejection of background tracks is most important for regions with a high 2D track finding rate. Therefore, a *weighted positive rate* ε is introduced, which is proportional to the z-vertex trigger positive rate and to the 2D track finding rate. For a given interval $[z - \Delta z, z + \Delta z]$ the weighted positive rate is defined as

$$\varepsilon(z) = \frac{\text{number of tracks with } \left|z_{\mathrm{NN}}\right| \leq 6\,\mathrm{cm} \text{ and } z_{\mathrm{MC}} \in [z - \Delta z, z + \Delta z]}{\text{number of found tracks with } z_{\mathrm{MC}} \in [-\Delta z, \Delta z]}. \tag{7.6}$$

In other words, the weighted positive rate is normalized not to the total rate in the investigated bin, but to the total rate at $z_{\mathrm{MC}} = 0\,\mathrm{cm}$. For $z = 0$ the efficiency is obtained again.

Three types of uncertainties need to be considered in the evaluation. The first are statistical uncertainties from the limited amount of testing data. The formulas for the statistical uncertainty of the resolution and the positive rate are given in appendix A. The data samples generated for the evaluation contain 190 000 events from the interaction point and another 135 000 events from the full z-vertex range.

The second type are systematic uncertainties, which originate for example from misalignment of the detector. They can be estimated by shifting the input values of the

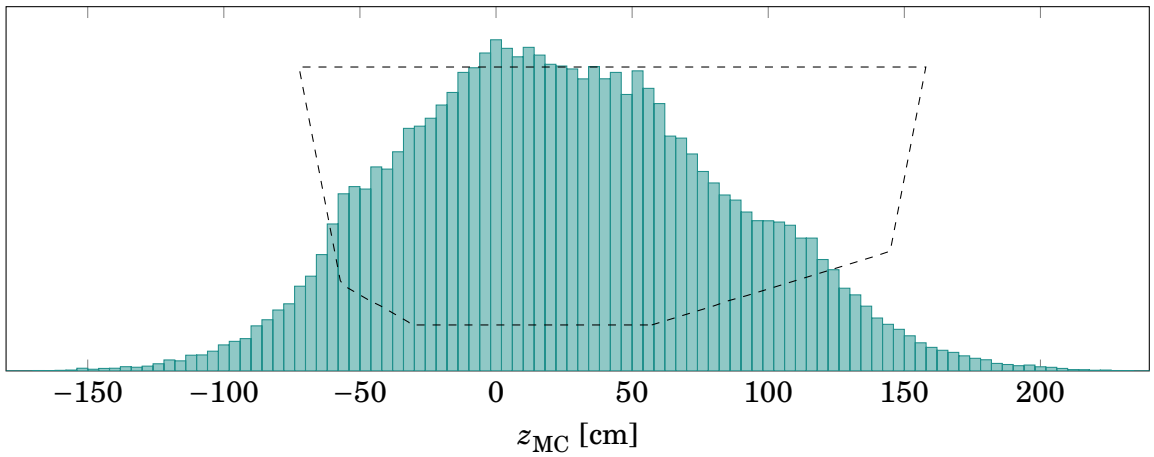

Figure 7.8.: z-vertex distribution of tracks that are found by the 2D track finder. The generated distribution is uniform in z_{MC}. To provide a reference for the length scale, the dashed line shows the outline of a side view of the CDC.

network according to their uncertainty, feeding them into the network and comparing the results to the original output values [75]. Ideally, this procedure should be based on a comparison of the simulation with experimental data, so a proper estimation of the systematic uncertainties is not possible yet.

The third type of uncertainty is due to imperfect training. As shown in fig. 7.6b, there is a spread of $\pm 2\,\%$ in the validation error for different initial weight sets. For the performance evaluation of the final z-vertex trigger this uncertainty is irrelevant. The neural network can be considered as a given function, whose accuracy can be evaluated without knowing how the function was obtained. However, when comparing different neural network setups, the training uncertainty must be taken into account. For example, if a network with two hidden layers achieves better results than a network with one hidden layer, the improvement might be caused by better initial weights instead of the different structure. One setup is significantly superior to another only if the improvement is larger than the fluctuations between repeated training runs.

Unfortunately, to measure the training uncertainty precisely, hundreds of trainings would be required for each setup. Therefore, only the statistical uncertainty is given explicitly in the following studies.

7.2. Input and target

In addition to the structure of the network, the format of the inputs and targets needs to be defined. The input values must be derived in some way from the track segment hits and the 2D track estimates. The output values must give the z-vertex and possibly the polar angle of each track. An additional constraint is given by the structure of

the MLP, which requires a fixed number of inputs and outputs. These requirements leave several possibilities for the input and target representation, which can have a considerable influence on the performance. The following section explains the selected representation and discusses some of the alternatives.

7.2.1. Input representation

The early studies about the z-vertex trigger in [4] and [68] use a very simple input representation, where each track segment is represented by an input node of the MLP. The input value is given by the drift time of the respective track segment, scaled to a range of $[-1, 1]$, with 1 corresponding to a drift time of 0 ns (hit directly at the wire) and -1 corresponding to the maximal drift time (hit far away from the wire). For track segments without hit the input value is set to -1, so "no hit" is interpreted as "far away from the track".

The obvious disadvantage of this method is a huge number of input nodes, most of which will only get the default input of -1. To cope with this problem, the idea is to divide the phase space into $\mathcal{O}(10^6)$ small sectors with 20 to 30 track segments each. For each sector an expert MLP has to be trained. For a given track the correct sector is chosen with the information from the 2D finder and a lookup table, then the corresponding MLP estimates the z-vertex.

The method was tested for a few representative example sectors to prove that the z-vertex can be estimated with the desired accuracy [4] and that the method is robust in the presence of background hits [68]. However, there are two inherent problems. The first is the difficulty to find the correct sector for a given track. The best results were achieved only with some sectorization not only in φ_0 and $p_\perp$, but also in the polar angle θ_0, which is not known without an additional preliminary 3D track estimate. The second problem is the challenge of training a huge number of neural networks and storing the weights for each sector on the FPGA.

To circumvent these problems, a completely different input representation is introduced in this thesis, which makes use of the knowledge about the shape of the tracks and their symmetry to define higher level inputs in addition to the raw drift times. The reconstruction is symmetric in φ, in the sense that all dependence on the azimuth angle can be removed by measuring all angles relative to the 2D track parameter φ_0. However, since the wire positions are discrete, the detector is not completely symmetric in φ. In particular, the relative position of wires in different superlayers to each other depends on φ. Therefore, to exploit this symmetry it is necessary to give up the topological input representation of one input node per track segment.

Instead, a tuple of inputs is used for each track segment hit, which gives not only the drift time but also the position of the track segment. To fulfill the demand of a fixed number of input nodes, a single hit per superlayer is selected. For each hit three input values are defined (see fig. 7.9):

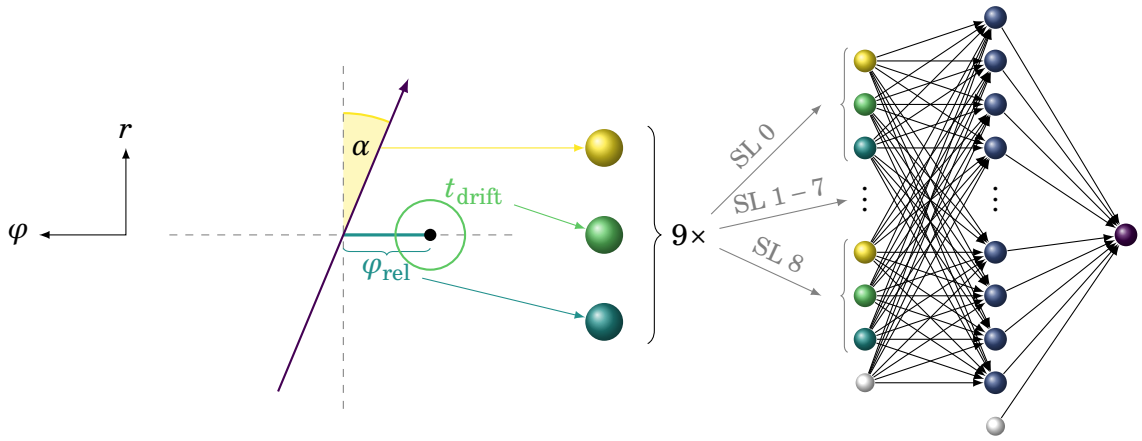

Figure 7.9.: Illustration of the input values for the neural network. For each superlayer a tuple of $(\varphi_{\text{rel}}, \pm t_{\text{drift}}, \alpha)$ is passed to three input nodes, giving 27 input nodes in total.

1. $\varphi_{\text{rel}} = \varphi_{\text{wire}} - \varphi'_{\text{cross}}$:
   The azimuthal position of the wire is given relative to the crossing point of the 2D track with the layer, which is calculated according to eq. (6.4). For stereo layers, φ_{rel} is required to calculate the z coordinate of the hit (compare eq. (6.7)). For axial layers, it is related to the error of the 2D track estimate.

2. $\pm t_{\text{drift}}$:
   The drift time is given as a separate input, so it is up to the neural network to find the nonlinear function for the drift length and to apply corrections depending on the crossing angle. The left/right state is given as a sign. For unknown left/right state the sign and therefore the drift time input is 0.

3. α:
   The crossing angle as defined in eq. (6.4) is needed for the extrapolation to the z-vertex (compare eq. (6.11)) and for the precise calculation of the hit position (compare eq. (6.6)). In addition, it should allow a correction of the approximation that was made in calculating the crossing point in eq. (6.4).

The radial position is not explicitly given to the network, as it is constant for each layer. Both first and second priority hits are used, which leads to a small uncertainty in the layer radius and the wire length.

The idea of this representation is to follow the analytical reconstruction described in chapter 6 as far as possible. At the point where approximations become necessary, the neural network takes over. Note that the network receives input from axial layers as well as from stereo layers. This compensates the fact that the 2D track parameters are taken directly from the track finder, without improvement by a fit which includes the drift times.

Table 7.1.: Comparison of the z-vertex resolution in small sectors for different input representations [76]. The tested z-vertex range is $[-10, 10]$cm. The momentum is limited to $\varphi_0 \in [180°, 181°]$ for rep1/rep2 and $\varphi_0 \in [135°, 225°]$ for rep3/rep4, $\theta_0 \in [50°, 60°]$ and $p_\perp^{-1} \in [1.95, 2](\mathrm{GeV}/c)^{-1}$ in the low $p_\perp$ sector and $p_\perp^{-1} \in [0.2, 0.25](\mathrm{GeV}/c)^{-1}$ in the high $p_\perp$ sector. See text for an explanation of the different input representations.

	input values	high p_T Δz	inputs	low p_T Δz	inputs
rep1	1 per TS (t)	1.40 cm	17	1.89 cm	31
rep2	2 per SL ($t, \varphi_{\mathrm{wire}}$)	1.10 cm	18	1.50 cm	18
rep3	2 per SL ($t, \varphi_{\mathrm{rel}}$)	1.18 cm	18	1.32 cm	18
rep4	3 per SL ($t, \varphi_{\mathrm{rel}}, \mu$)	1.15 cm	27	1.18 cm	27

A comparison of this high level input representation with the topological input from the early studies was published in [76]. Table 7.1 shows the results of the study, which compares the z-vertex resolution (defined here as the normal standard deviation) in two small sectors for four different input representations. “rep1” is the topological input representation. “rep2” uses one hit per superlayer, but does not directly make use of the 2D track parameters, taking the absolute position φ_{wire} as input in addition to the drift time. “rep3” replaces the absolute position by φ_{rel}. “rep4” adds a third input for each superlayer, namely the helix parameter $\mu = \frac{2\alpha}{\omega}$ from eq. (6.9), which was later replaced by the crossing angle α to get the final input representation. The advantage of the crossing angle is that it carries information about the charge of the track, otherwise it is proportional to μ and the resulting z-vertex resolution is very similar. The comparison shows that a high level input representation is clearly superior to simply using the raw drift times, especially for low transverse momentum. In addition, the network structure is independent of the sector and the total number of sectors can be strongly reduced.

Note that the sector size in the study in table 7.1 is unrealistically small without preliminary parameter estimates. The resolution of the track finder would allow sectors of $\Delta p_\perp^{-1} \approx 0.5(\mathrm{GeV}/c)^{-1}$ and $\Delta\varphi_0 \approx 5°$, assuming a $\pm 3\sigma$ interval (compare table 5.4); θ_0 is not known at all.

7.2.2. Hit selection

The disadvantage of the high level input representation is the need to select exactly one hit per superlayer, so it is potentially more vulnerable to background hits. As in the analytical reconstruction, only four of the input hits carry 3D information, so the hit selection has a strong influence on the performance.

The hit selection consists of two steps. First, a list of hit candidates is defined

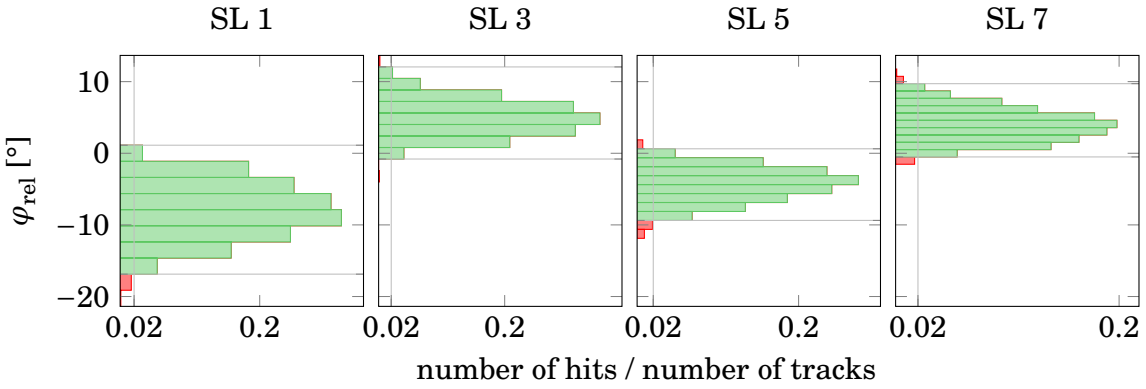

Figure 7.10.: To select the stereo hits, a relevant region around the 2D track is determined from histograms of φ_{rel}. The bin size corresponds to the distance between wires. All bins that have hits for at least 2 % of the tracks contribute to the relevant region (green). Hits outside of the relevant region (red) are rejected.

that are considered as related to the track. Then, for each superlayer one of the hit candidates is selected and the others are discarded. For the axial hits the selection is made by the track finder, as described in section 5.4.3. Hit candidates are defined as hits whose Hough curve crosses the cluster center; the selection is based on the priority position and the track segment ID.

For the stereo hits, a "relevant" region $[\varphi^{\mathrm{SL}}_{\mathrm{min}}, \varphi^{\mathrm{SL}}_{\mathrm{max}}]$ around the 2D track is defined. All hits with $\varphi^{\mathrm{SL}}_{\mathrm{min}} \leq \varphi_{\mathrm{rel}} \leq \varphi^{\mathrm{SL}}_{\mathrm{max}}$ are considered as hit candidates. The relevant region depends on the superlayer and is determined experimentally during the training, based on the procedure for selecting wires for the input nodes in the topological input representation in [4]. The first 1000 tracks in the training data are used to create one histogram for each superlayer, filled with φ_{rel} of all hits that are related to the true particle, as shown in fig. 7.10. The histograms are normalized to the number of test tracks. All bins exceeding a threshold of 2 % are considered as relevant and define the relevant region. Hits outside of this region are considered as belonging to a different track or as background. The relevant region depends on the stereo angle and on the range of θ_0 and z_0 in the training data, and also on the accuracy of the 2D track parameters.

In addition to the relevant φ_{rel} region, a time window is imposed. Only hits with a drift time $t_{\mathrm{drift}} \in [0\,\mathrm{ns}, t_{\mathrm{max}}]$ are considered as hit candidates, with $t_{\mathrm{max}} \approx 503\,\mathrm{ns}$ corresponding to 256 bins of the digitized timing in the trigger. The time window suppresses all background hits before and after the event and relies on the event time.

Among the hit candidates that pass this selection, hits with known left/right state are preferred over hits with unknown left/right state. If there are still several candidates, the hit with the shortest drift time is selected. This selection aims to suppress

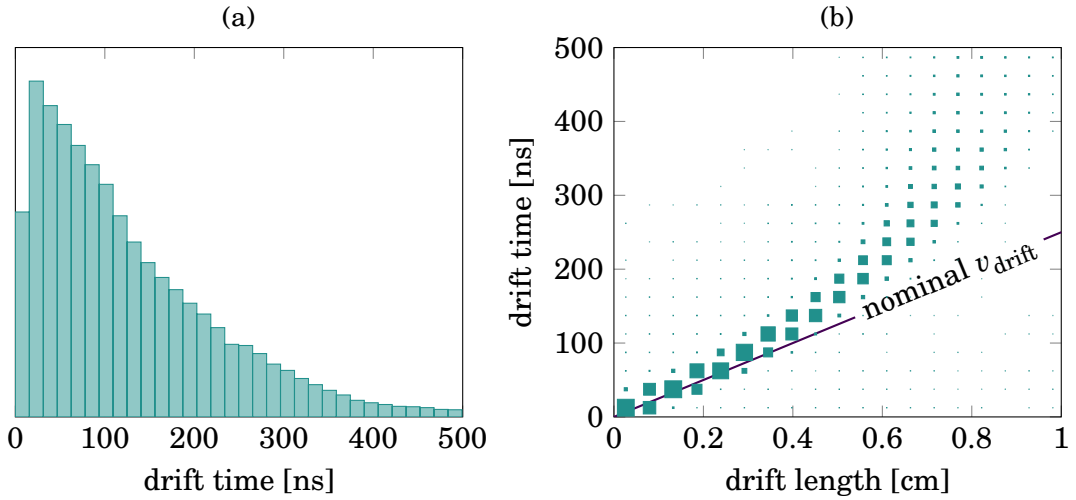

Figure 7.11.: (a) Drift time distribution of track segment hits. (b) Relation of drift time to drift length. The nominal drift velocity in the CDC of $v_{\text{drift}} = 40\,\mu\text{m}\,\text{ns}^{-1}$ gives a good approximation only for small drift times.

background hits and at the same time select the hits that contain the most useful information for the track reconstruction. The ratio of hits with known left/right state is $\approx 66\,\%$ for hits belonging to the particle and $\approx 34\,\%$ for background hits. In addition, the left/right sign is required for the drift time inputs, increasing the precision of the hit position by one order of magnitude. The drift time distribution for true hits is shown in fig. 7.11a. It peaks at short drift times, while the drift time distribution for background hits is flat. Therefore, a hit with short drift time is more likely to belong to the track than a hit with long drift time. In addition, the influence of nonlinearities in the drift velocity increases for longer drift times, so shorter drift times provide a slightly higher precision for the drift length (see fig. 7.11b).

Table 7.2 shows the selection efficiency and purity for the axial and stereo hits. The purity is defined as the fraction of candidates or selected hits that are related to a particle (“true hits” as opposed to background hits). The candidate efficiency is defined as the fraction of true hits that pass the candidate selection. The selection efficiency is defined as the number of true selected hits over the number of superlayers that have at least one true hit, since by definition at most one true hit per superlayer can be selected.

For the axial hits, the selection of the track finder can be combined with a time window cut and a relevant region, which is determined in the same way as for the stereo track segments. This additional selection decreases the hit efficiency, but increases the purity. In other words, hits that are likely to originate from background are deliberately ignored, even if there are no replacement hits. For superlayers without hit candidates, a default input tuple of $(0, 0, 0)$ is passed to the neural network.

Table 7.2.: Efficiency and purity of the hit selection of the track finder and the z-vertex trigger, measured on single tracks with background.

		candidates		selection	
hits	selection by	efficiency	purity	efficiency	purity
axial	finder	99.5 %	93.9 %	97.1 %	96.1 %
axial	finder + z trigger	—	—	96.6 %	98.2 %
axial	z trigger	99.5 %	97.1 %	98.5 %	98.4 %
stereo	z trigger	99.1 %	95.0 %	97.7 %	97.3 %

For comparison, the selection of axial hits without the track finder is also shown, using only the time window cut and the relevant region. It is clearly better than the hit selection from the finder, due to the additional information from the drift time and left/right state which is not used in the track finder. The direct comparison between axial and stereo hits with the same selection rules shows also that for stereo hits both the efficiency and the purity are $\approx 1\%$ worse than for axial hits. This is due to the larger relevant region caused by the stereo angle and to the higher background rate for stereo wires.

In the following, the relevant region and time window cuts are applied to both axial and stereo hits. For the purpose of the track matching in the evaluation of the neural network, all selected hits are related to the track.

7.2.3. Scaling

All input values are scaled before they are passed to the neural network to confine them to the interval $[-1,1]$. This decision is somewhat arbitrary, since any scaling factor can be compensated by an inverse scaling of the weights. However, for the fixed point arithmetic of the FPGA implementation it is convenient to have all input values in the same range.

All input values are limited by definition. The hit selection restricts φ_{rel} to the relevant region and the drift time $\pm t_{\text{drift}}$ to the interval $[-256,256]$. The crossing angle α is limited geometrically to $[-90°,90°]$. In fact a stricter bound for the crossing angle could be determined from the track segment finder acceptance, but for simplicity the maximal range is used for the scaling.

Since the scale factors are in principle arbitrary, they are chosen as powers of 2, so that the scaling corresponds to a simple bitshift for the symmetric intervals of the drift time and the crossing angle. This symmetric scaling is another advantage of using the crossing angle instead of the helix parameter μ in rep4 of table 7.1. The relevant region of the angle φ_{rel} is in general not symmetric (see fig. 7.10), so a shift is applied before the scaling. Note that this affects the interpretation of the default value 0 for missing hits, which now corresponds to the center of the relevant region.

7.2.4. Target representation

The aim of the z-vertex trigger is to reject tracks that do not originate from the interaction point. In principle, the output of the network could be binary: the track is either from the interaction point or from somewhere else. In this case a cut would have to be applied to the true z-vertex during the training to label the tracks either as signal or background. However, this approach is not very flexible, since the cut has to be decided beforehand. In addition, the background category would include tracks with both negative and positive z-vertex, which look very different in the detector.

Instead the z-vertex is directly used as target for the network. The output of the network is bounded by the activation function to $[-1,1]$, so the targets need to be scaled to that region. For this purpose a limited symmetric *target range* $[-z_{\max}, z_{\max}]$ is defined. Training, validation and test samples are taken only from the target range. To get the z-vertex in cm, the output value of the network is scaled back from $[-1,1]$ to the target range. For tracks with a z-vertex outside of the target range, the network should ideally return the range limit $\pm z_{\max}$. Evaluating the network on such tracks is equivalent to extrapolating the learned function of the network to regions that were not included in the training data. The best resolution is expected at the center of the target range, which should therefore correspond to the interaction point. This is why the target range is defined symmetric, although the track finder acceptance with respect to the z-vertex is asymmetric.

The z-vertex resolution depends quite strongly on the target range, as shown in fig. 7.12a, ranging from 2.45 cm for a target range of $[-10\,\text{cm}, 10\,\text{cm}]$ to 4.34 cm for a target range of $[-60\,\text{cm}, 60\,\text{cm}]$. A small target range allows the neural network to concentrate on subtleties in the track shape for this region, but the extrapolation to tracks far away from the target region becomes difficult. Therefore, there is a high false positive rate for background tracks, as shown in fig. 7.12b. To get an optimal background rejection, a target range of at least $[-50\,\text{cm}, 50\,\text{cm}]$ is required. For this target range, the average positive rate for $|z_{\text{MC}}| > 12\,\text{cm}$ (twice the z-vertex cut) is only $(0.70 \pm 0.02)\,\%$. However, the efficiency is also relatively low with $(81.91 \pm 0.09)\,\%$. For a larger target range, the resolution and efficiency decrease without further benefit in the rejection rate, so in the following studies the target range is set to $[-50\,\text{cm}, 50\,\text{cm}]$.

After the z-vertex is determined by the neural network, the only missing track

Table 7.3.: Comparison of the neural network performance with θ_0 estimate for two different target ranges and without θ_0 estimate.

θ_0 target range	$\sigma_\theta^{(95)}$	$\sigma_z^{(95)}$	efficiency
$[0°, 180°]$	$(4.00 \pm 0.01)°$	$(4.06 \pm 0.01)\,\text{cm}$	$(82.61 \pm 0.09)\,\%$
$[30°, 127°]$	$(4.35 \pm 0.01)°$	$(4.17 \pm 0.01)\,\text{cm}$	$(82.98 \pm 0.09)\,\%$
no θ_0 estimate	—	$(4.11 \pm 0.01)\,\text{cm}$	$(81.91 \pm 0.09)\,\%$

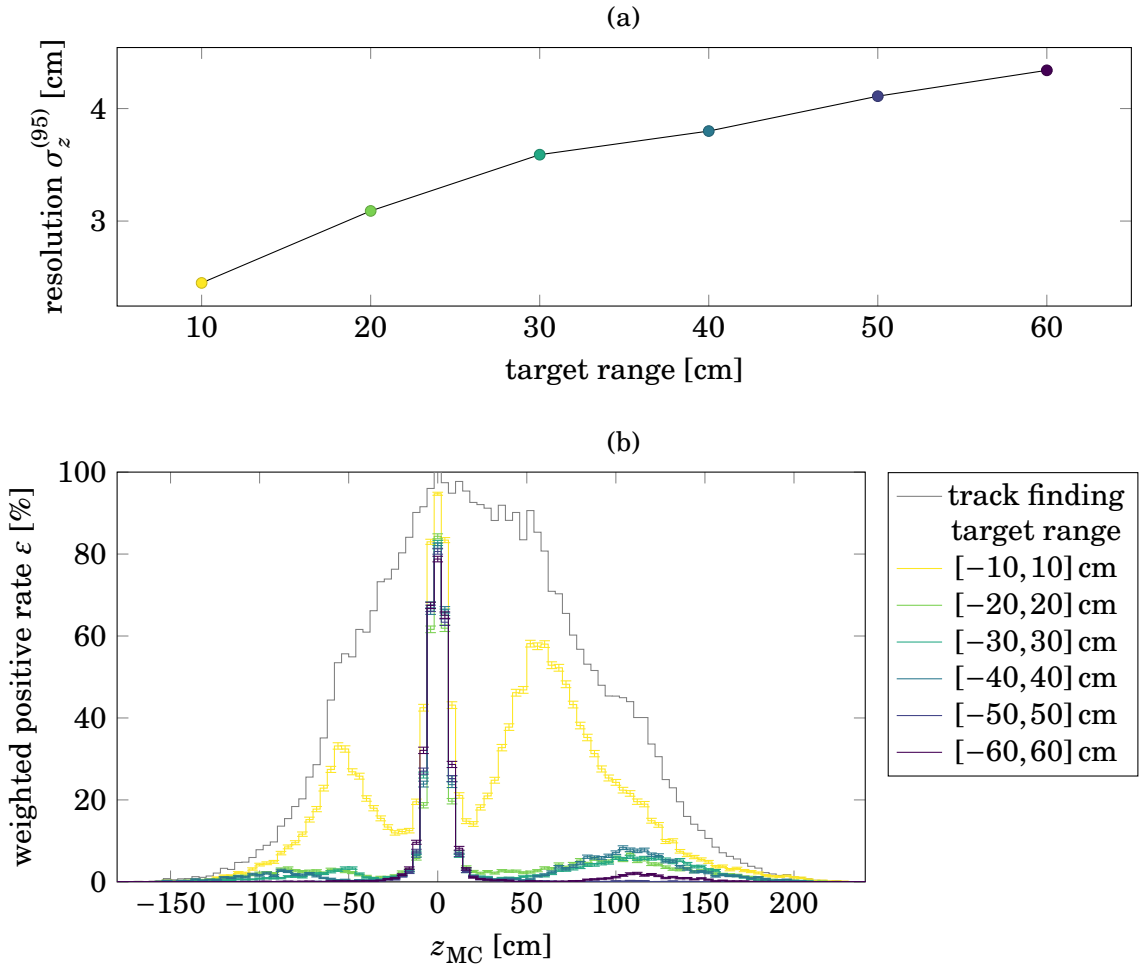

Figure 7.12.: (a) z-vertex resolution of a single neural network depending on the target range of the network, measured for tracks where z_{MC} is in the target range. (b) Weighted positive rate according to eq. (7.6) depending on the target range.

parameter is the polar angle θ_0. In the analytical reconstruction, the parameters θ_0 and z_0 are determined together with a fit of eq. (6.11). Similarly, a single neural network with two output nodes can determine both the polar angle and the z-vertex of a track. As for the z-vertex, a target range is defined for the polar angle to scale the network output. Table 7.3 shows the resolution of the polar angle that can be obtained by a neural network and the influence on the z-vertex estimate. Three networks are compared, one with the full target range of $\theta_0 \in [0°, 180°]$, one with the target range limited to the track finder acceptance of $\theta_0 \in [30°, 127°]$ (compare table 5.4) and one without θ_0 estimate. In contrast to the z-vertex, the resolution of the polar angle is better for the full target range. Although the z-vertex resolution is no longer the only

objective of the training, the efficiency for a z-vertex cut of 6 cm increases compared to a network with a single output. Therefore, the following studies always include an output node for θ_0 with a target range of $[0°, 180°]$, even if the polar angle is not explicitly evaluated.

7.3. Specialized networks

The results of the previous section show that a single neural network can estimate both z_0 and θ_0, but the resolution is not good enough to apply the target z-vertex cut of 6 cm. To improve the resolution, several expert networks can be trained, which focus on different track classes. This approach works as long as it is possible to determine the track class before the neural network is executed.

7.3.1. Missing hits

The input representation of the neural network expects exactly one hit per superlayer. However, it is not guaranteed that there is at least one hit candidate in each superlayer. Reasons for missing hits include long drift times outside of the considered time window for one event and tracks that pass only a part of the CDC. In addition, the hit selection efficiency is not perfect (compare table 7.2), so sometimes hits are simply missed.

Since the number of input values is built into the structure of the MLP, the three input values for a superlayer with a missing hit are set to $(0,0,0)$. Although one might naively expect that this effectively turns off the corresponding inputs, this input tuple has a well defined meaning. For an axial superlayer, the default tuple corresponds to a hit directly on the track. For a stereo superlayer, it corresponds to a hit in the center of the wire. For all layers the last value in the default tuple implies a straight track.

Instead of using default values with an unintended interpretation, it is better to train a dedicated expert network for the case of missing hits, which has less input nodes in the first place. However, it is not enough to train one network for tracks with 9 hits, one for tracks with 8 hits and so on. The geometrical configuration of the different superlayers is only learned implicitly, so it is not possible to randomly exchange inputs from different superlayers depending on which hit is missing. Instead one expert network is required for each possible combination of missing hits. This would mean 2^9 experts, but most of the combinations are very unlikely.

First, different combinations of missing stereo hits are examined. Axial hits are only required for small corrections, so in this test the input values for missing axial hits are still replaced by the default tuple. In principle $2^4 = 16$ combinations of missing stereo hits are possible. However, at least two stereo hits are needed to determine the two unknown track parameters z_0 and θ_0. Errors in the individual hit positions can only by compensated for an overdetermined system, that is for three or four stereo hits. Figure 7.13 shows the z-vertex distribution depending on the number and superlayer

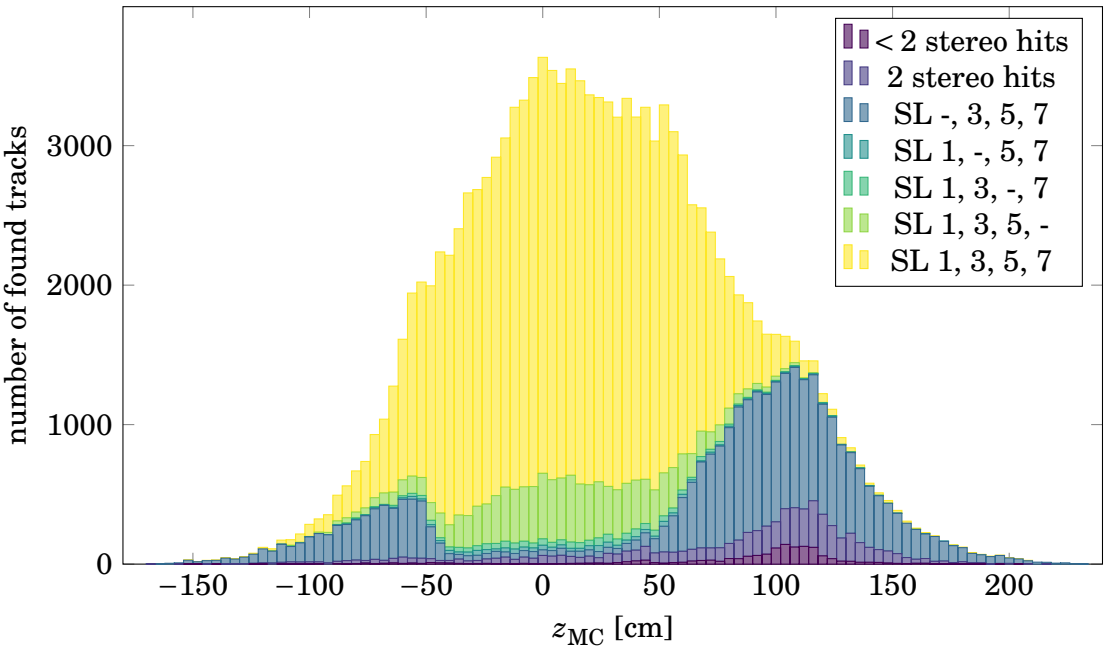

Figure 7.13.: Number of selected stereo hits for a track, depending on the true z-vertex. For two or less stereo hits, the distributions for different superlayer combinations are summed.

combination of selected stereo hits. In the critical region around the interaction point, most tracks have either four or three stereo hits, where the outermost stereo superlayer is most likely to have no hit, as expected from the track shape. However, there is also a finite probability to miss one or more of the inner stereo superlayers.

Five dedicated expert networks are trained for the case of four stereo hits and all cases of three stereo hits. To collect a sufficient amount of training samples for rare hit combinations, hits are deliberately ignored. Thus a track with four stereo hits can be used to produce one training sample for each of the five experts. This means also that the z-vertex distribution in the individual training samples does not correspond to the individual distributions in fig. 7.13, but is approximately the same for all experts. The network structure is in fact unchanged, with 27 input nodes, 81 hidden nodes and missing input values replaced by the default tuple. The important difference between training an expert or a general network is to turn off the unused input nodes already during the training, so that they do not affect the weight updates.

Table 7.4 summarizes the frequency of different stereo hit combinations for tracks from the interaction point and their respective efficiency for a general network and for five separate expert networks. Tracks with less than three selected hits are not handled at all by the ensemble of experts, so the efficiency for these cases is 0. Despite these small losses, the total efficiency increases by 4.5 % for the ensemble of experts.

As a second step, networks can be specialized to different combinations of missing

Table 7.4.: Comparison of the efficiency of a general neural network for different combinations of missing stereo hits with several expert networks trained specifically for a certain hit combination. For ≤ 2 hits no experts are trained and all tracks are rejected.

stereo hits	fraction of events	efficiency (general network)	efficiency (expert network)
SL 1, 3, 5, 7	82.3 %	(88.46 ± 0.08) %	(91.28 ± 0.07) %
SL 1, 3, 5, -	12.8 %	(59.9 ± 0.3) %	(72.9 ± 0.3) %
SL 1, 3, -, 7	1.2 %	(52.7 ± 1.0) %	(83.4 ± 0.8) %
SL 1, -, 5, 7	0.9 %	(58.2 ± 1.2) %	(89.5 ± 0.7) %
SL -, 3, 5, 7	1.3 %	(41.1 ± 1.0) %	(65.5 ± 1.0) %
2 stereo hits	1.4 %	(32.1 ± 0.9) %	0 %
< 2 stereo hits	0.1 %	(6.7 ± 2.0) %	0 %
any	100 %	(82.61 ± 0.09) %	(87.11 ± 0.08) %

Table 7.5.: Comparison of the efficiency of expert networks trained for specific axial hit combinations (see text). Only tracks with four selected stereo hits are used in this test.

axial hits	fraction of events	efficiency (1 network)	efficiency (6 networks)	efficiency (2 networks)
≥ 4 hits	80.5 %	(91.57 ± 0.07) %	(91.89 ± 0.07) %	(91.73 ± 0.07) %
≤ 3 hits	1.7 %	(77.8 ± 0.7) %	0 %	(81.5 ± 0.7) %
any	82.3 %	(91.28 ± 0.07) %	(89.95 ± 0.08) %	(91.51 ± 0.07) %

axial hits. This is tested for the most frequent case of four selected stereo hits. Table 7.5 compares the efficiency for three different setups. The first is the setup from table 7.4, which is specialized only to the combination of stereo hits, using axial hits if they are present and otherwise replacing them with the default tuple. The second contains six networks for each axial hit combination with at least four hits. The improvement by this specialization is minimal. The gain for tracks that can be handled is smaller than the training uncertainty and less than the loss by tracks that cannot be handled.

The third setup contains two networks, specialized to whether there is a hit in the last axial superlayer (which is the most likely to be missing for tracks from the interaction point). For all other axial superlayers, the default tuple is used if there is no hit, so there are no additional losses from tracks that cannot be handled. However, again the total gain compared to a single network is not significant. Similar results were obtained for tracks with three selected stereo hits in superlayers 1, 3 and 5.

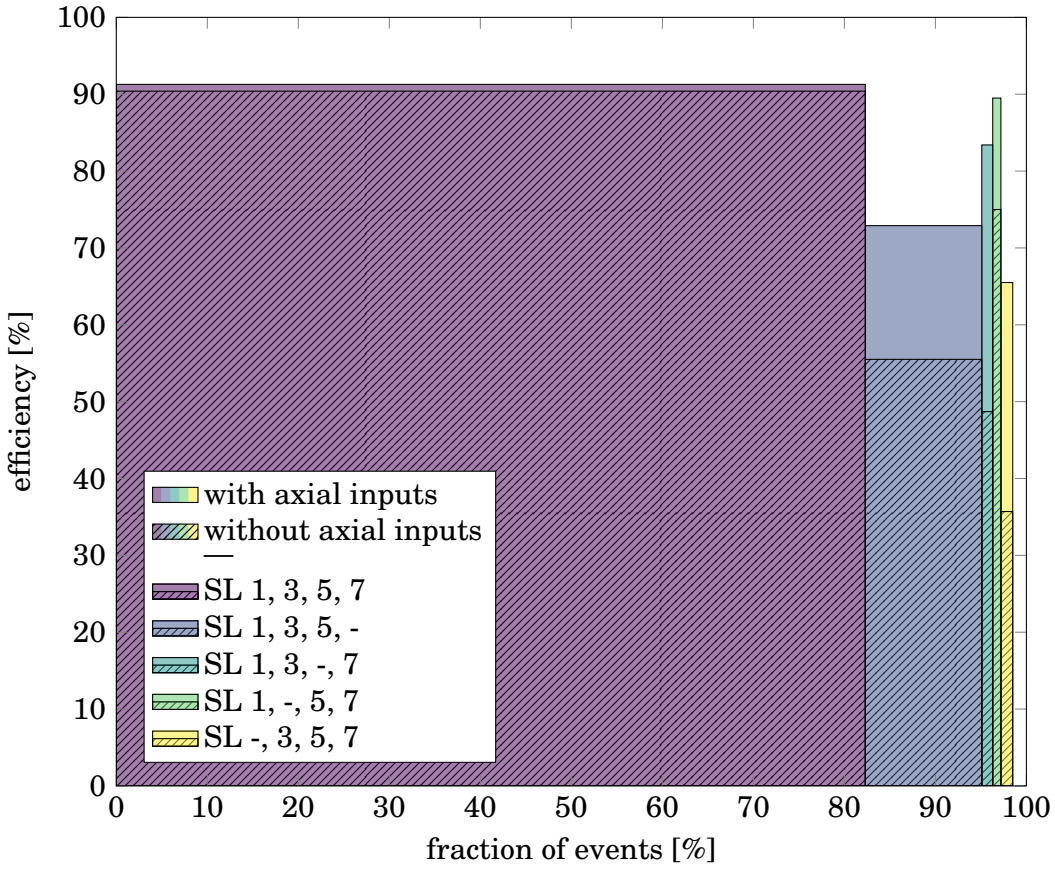

Figure 7.14.: Comparison of the z-vertex trigger efficiency with and without axial inputs, depending on the number of selected stereo hits. The width of each bar gives the fraction of events with the corresponding stereo hit combination. The area is proportional to the contribution to the total efficiency. The missing 1.5 % correspond to tracks with less than three stereo hits.

These results seem to indicate that the axial inputs might not be required at all, so another set of five networks is trained which do not use any axial hits. This setup does not require the default tuple. However, the total efficiency without axial inputs is only $(83.14 \pm 0.09)\,\%$, which is significantly less than with axial inputs and default values. The individual efficiencies for different stereo hit combinations are shown graphically in fig. 7.14. For tracks with missing stereo hits, there is a significant efficiency loss for the setup without axial inputs. In contrast, for tracks with all four stereo hits the efficiency is almost unchanged.

The intention of providing axial hits as input is to allow drift time corrections to the 2D track parameters and thus the crossing positions of stereo wires with the track. Apparently four stereo hits are enough to compensate small errors in the individual

crossing positions without axial inputs. With only three hits the precision of each crossing point is more important and the additional information from axial inputs gives a significant improvement.

To summarize, training dedicated expert networks for different combinations of missing stereo hits gives a clear improvement of the performance. The total efficiency increases from $(82.61 \pm 0.09)\,\%$ to $(87.11 \pm 0.08)\,\%$, while the average z-vertex resolution decreases from $(4.06 \pm 0.01)\,\mathrm{cm}$ to $(3.44 \pm 0.01)\,\mathrm{cm}$. Specialized networks for different combinations of missing axial hits give no significant improvement, although the axial inputs are important for the reconstruction. The reason is that for axial hits the default tuple has a sensible meaning: it can be interpreted as "no corrections", which is identical to not using the axial input at all. For stereo hits this interpretation does not work, so the default inputs are harmful instead of simply ignored.

7.3.2. Sectorization in transverse momentum

Another possibility is to train specialized networks for different ranges of the track parameters, called *sectors*. Such a sectorization was essential for the topological input representation in [4]. For the high level input representation introduced in section 7.2.1 it is no longer necessary, but it might still help to have dedicated expert networks for extreme parameter regions.

In chapter 6 it was shown that the expected accuracy of the 3D track reconstruction is better for straight tracks. Although the neural network receives information about the track curvature in form of the crossing angle α, different approximations may be necessary for high and low transverse momentum.

Figure 7.15 shows the dependence of the z-vertex resolution and efficiency on the transverse momentum. The resolution varies by a factor of 2, from $(2.59 \pm 0.02)\,\mathrm{cm}$ for high $p_\perp$ to $(5.24 \pm 0.04)\,\mathrm{cm}$ for low $p_\perp$. Accordingly, the efficiency for the default z-vertex cut of 6 cm is only $(67.4 \pm 0.3)\,\%$ for low $p_\perp$, but $(93.5 \pm 0.2)\,\%$ for high $p_\perp$.

To see whether a sectorization in $p_\perp$ improves this behavior, networks are trained for data samples with fixed transverse momentum. This case corresponds to maximal sectorization, which would require an infinite number of expert networks and a perfect sector selection to be applied to the full momentum range. Both positively and negatively charged tracks are used in the study, with separate expert networks trained for each charge to ensure that each network can focus on one track shape of fixed signed curvature. The results of these experts are included in fig. 7.15. Despite the idealization of infinitely small sectors, the accuracy of the z-vertex estimate hardly improves. It is not to be expected that a more realistic sectorization will give better results.

Clearly experts for different $p_\perp$ sectors cannot solve the problem of $p_\perp$ dependent resolution. One solution to get a more uniform efficiency would be to set different z-vertex cuts depending on the transverse momentum of each track. However, this leads also to a higher false positive rate from background tracks with low transverse

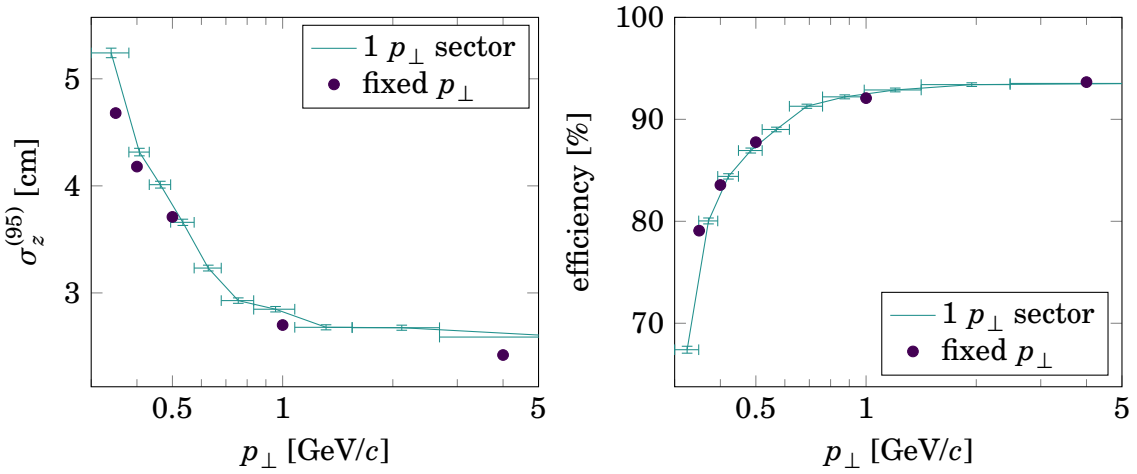

Figure 7.15.: Resolution and efficiency depending on the transverse momentum. The solid line shows the results for a networks trained for the full transverse momentum range. The dots show the results for separate networks trained and tested on tracks with fixed transverse momentum.

momentum, so the signal to background ratio is not necessarily better. For simplicity, a fixed z-vertex cut independent of the track parameters is kept.

7.3.3. Sectorization in polar angle

Another possibility is to introduce several sectors in the polar angle θ_0. As mentioned before, the polar angle is not known before it is estimated by the neural network, so in the presented setup it is not possible to find the correct θ_0 sector. However, an upgrade of the track finding to three dimensions could provide an estimate for θ_0 in the future [65]. With this initial estimate, it would be possible to select an expert network for a limited θ_0 sector with a width of $\approx 19°$, which corresponds to five sectors in the acceptance of the 2D finder.

In the following studies, the polar angle is taken from the true momentum of the particle. The sectors are defined relative to the track finding acceptance of $\theta_0 \in [30°, 127°]$ (compare table 5.4). This region is divided into five sectors. To ensure that all tracks can be handled, the first (last) sector is enlarged to encompass also the extreme forward (backward) region of 0° to 30° (127° to 180°). Figure 7.16 shows the z-vertex resolution and efficiency for different sectorization of the polar angle. First, it can be observed that the resolution is best in the central region and gets worse in the forward and backward region, as the errors of the z-vertex extrapolation due to parallax increase for flat polar angles. Second, the performance improves significantly with sectorization. The general dependence on the polar angle is unchanged, but the average z-vertex resolution for five sectors is (2.746 ± 0.007) cm, compared to

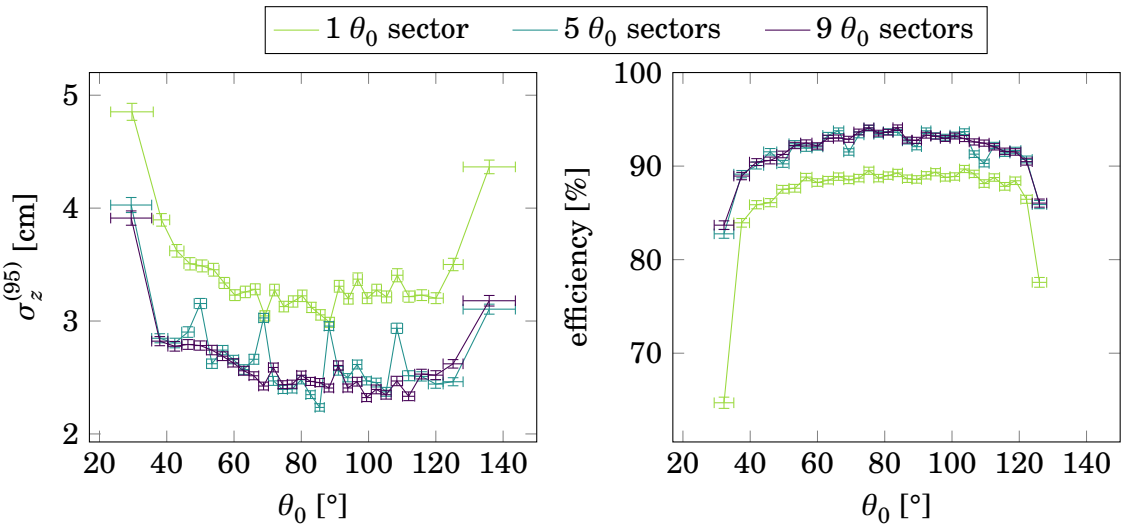

Figure 7.16.: Resolution and efficiency depending on the polar angle. Three sectorization setups are compared: one with a single network for the full polar angle range, one with five sectors of width $\Delta\theta_0 = 19.4°$ and one with nine overlapping sectors of the same width.

(3.44 ± 0.01) cm for a single sector. However, around the sector edges the resolution is significantly worse. This is a side effect of the training, which tends to distort the results near the edge of the trained region. The effect is expected to get even worse when the sector is selected by an estimate of θ_0 instead of the true value, as tracks close to the sector edge may be assigned to the wrong sector.

To remedy this problem, sectors are defined with an overlap, as proposed in [4]. This leads to nine sectors with the same width as in the five sector setup, as illustrated in fig. 7.17. Once the training is finished, each sector is redefined to handle only the center of the training region. With this definition, the fluctuations in the resolution disappear. The average resolution and efficiency is only slightly better, but this is likely due to the idealized sector selection. As soon as misassignment of sectors becomes a problem, an overlap in the sector definition is expected to be essential for the performance.

Although an estimate for the polar angle is required as input in this setup, the polar angle is still also estimated by the neural network. There are two possibilities for the target range: it can be set either to the sector range or left at the full range of $[0°, 180°]$. The networks in fig. 7.16 use the second option for simplicity, that is the unscaling of the network output does not depend on the sector. However, with a limited target range the θ_0 resolution might improve, as the neural network can focus better on the given sector.

Table 7.6 compares the resolution of both z_0 and θ_0 for different sectorization and target ranges. With a limited target range the polar angle resolution improves, but

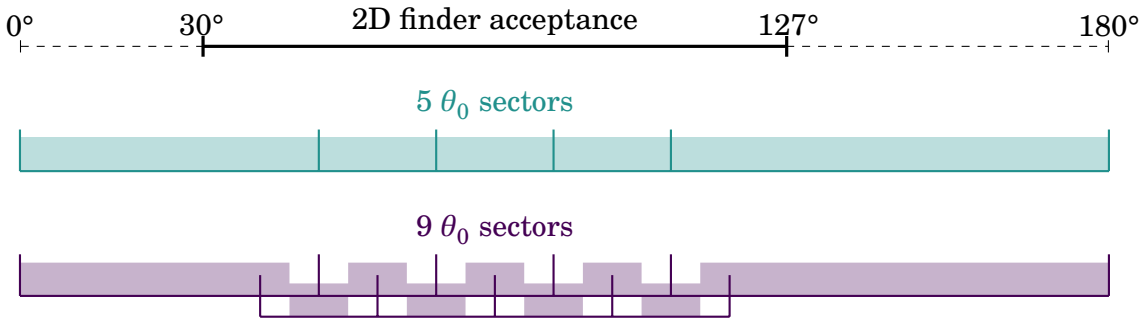

Figure 7.17.: Illustration of the polar angle sectorization in fig. 7.16. The 2D finder acceptance is divided into five sectors. The outer two sectors are enlarged to handle also the occasional tracks outside of the acceptance region. To avoid edge effects, four additional overlap sectors are defined. The shaded region shows the range that is handled by a given sector after the training is finished.

at the same time the z-vertex resolution gets worse. The reason for this effect is found in the definition of the error function that is minimized during the training (see eq. (7.4)). The error function is a sum over the deviation of all output values from their respective target. So when the relative scaling between the two outputs θ_0 and z_0 is changed, the main objective during the training shifts from one to the other. In the case of a limited polar angle target range, an error in the polar angle output gets more weight in the error function than an error in the z-vertex output.

In summary, specialized networks for different polar angle sectors achieve a significantly higher performance compared to a general network. The initial θ_0 estimate effectively removes one degree of freedom from the reconstruction task, so the improvement is not surprising. In total, the tested setup consists of 45 expert networks (9 overlapping polar angle sectors and 5 experts for missing stereo hits in each sector).

Table 7.6.: Neural network performance for different polar angle sectorization and target range. “Full” target range means an output range of $[0°, 180°]$ for all sectors. “Limited” target range means that the output range is identical to the sector range during the training.

sectors	target range	$\sigma_\theta^{(95)}$	$\sigma_z^{(95)}$	efficiency
1	full	$(3.43 \pm 0.01)°$	$(3.44 \pm 0.01)\,\mathrm{cm}$	$(87.11 \pm 0.08)\,\%$
5	full	$(2.609 \pm 0.007)°$	$(2.746 \pm 0.007)\,\mathrm{cm}$	$(91.74 \pm 0.06)\,\%$
5	limited	$(2.479 \pm 0.007)°$	$(2.843 \pm 0.008)\,\mathrm{cm}$	$(91.59 \pm 0.06)\,\%$
9	full	$(2.340 \pm 0.006)°$	$(2.638 \pm 0.007)\,\mathrm{cm}$	$(91.91 \pm 0.06)\,\%$
9	limited	$(2.304 \pm 0.006)°$	$(2.697 \pm 0.007)\,\mathrm{cm}$	$(91.82 \pm 0.06)\,\%$

Unfortunately, such a sectorization is not possible without an additional 3D track finding step, which will probably not be included in the trigger at the start of the experiment. The potential upgrade of the track finding is further discussed in [66], together with detailed studies about a more realistic polar angle sectorization. The present studies are continued with only five expert networks for missing stereo hits.

7.4. Network structure

The function of the neural network is defined by the structure of the network and by the individual weights. The structure of the MLP has to be defined by hand. So far all studies used the same network structure with one hidden layer and a factor of three times more hidden nodes than input nodes. This structure gave good results in the preliminary studies, although the choice was somewhat arbitrary, with very similar results obtained for a factor of 2 to 5 [68].

The weights are set automatically during the training, with the initial values generated randomly in the range $[-0.1, 0.1]$, which is again an arbitrary choice. In the following section, both the structure and the initial weight range are scanned to make sure that the MLP is used to full capacity.

7.4.1. Size and number of hidden layers

From the parallelization point of view, a single hidden layer is preferable, which leaves the number of hidden nodes as the only free parameter. This study has been done before in [68], but after changing the input representation, the target range and the sectorization, it is not obvious if the size is still optimal.

On the other hand, current research in machine learning focuses more and more on so-called “deep networks”, which are characterized by several successive hidden layers [77]. Deep network architectures play an important role for tasks like image recognition, where each layer extracts more abstract features than the layer before, leading from simple edges to complicated structures.

Figure 7.18 shows the z-vertex resolution and efficiency of MLPs with different numbers of hidden layers and nodes. For simplicity, the same layer size is chosen for all hidden layers within each MLP. Networks with less than ≈ 20 hidden nodes or ≈ 500 weights give very bad results. Above that threshold, the gain for a single layer network is very slow, with occasional fluctuations that are probably due to the random initial weights. The networks with more than one hidden layer achieve a better z-vertex resolution. However, the efficiency does not improve together with the resolution. This indicates either that the shape of the distribution changes rather than the total width, or that the networks behaves differently around the interaction point, where the efficiency is measured.

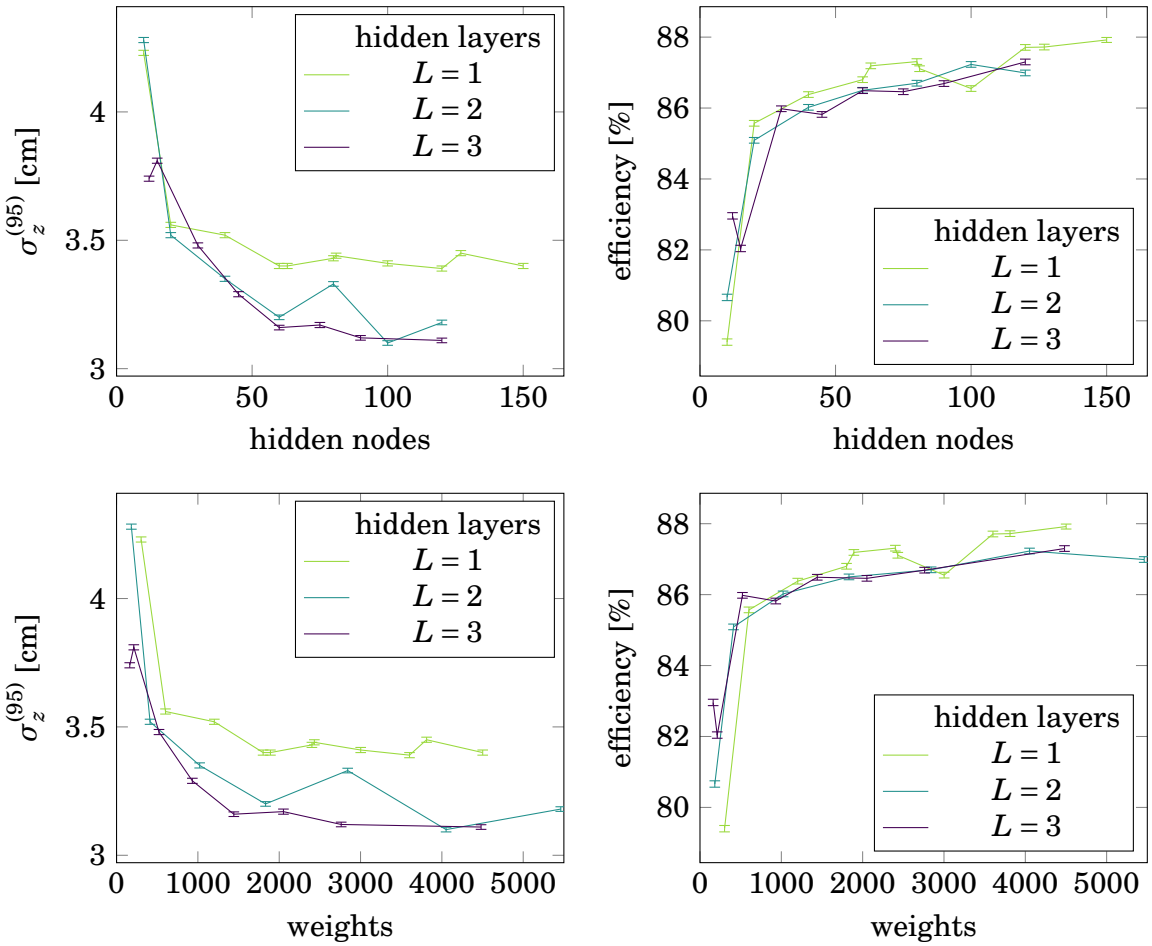

Figure 7.18.: Top row: average z-vertex resolution and efficiency depending on the total number of nodes in all hidden layers. Bottom row: same results depending on the total number of weights (given by eq. (7.3)).

Figure 7.19 shows the z-vertex error distribution for three example networks with the same total number of nodes, but different structure. For tracks from the full target range, the distribution for the networks with two and three hidden layers is more narrow than for a single hidden layer, giving a smaller width $\sigma_z^{(95)}$. In contrast, for tracks from the interaction point, the distributions are shifted against each other, while the width of the three distributions is almost identical, with $\sigma_z^{(95)}(\mathrm{IP}) = 3.08\,\mathrm{cm}$ to $3.11\,\mathrm{cm}$. The ratio between the different distributions shows an excess of entries in the left tail for the networks with two and three hidden layers, which is responsible for the lower efficiency.

So although the resolution over the full target range improves for the deeper structures, the performance for the critical region around the interaction point decreases. It is possible that the performance could be improved by a more complex training

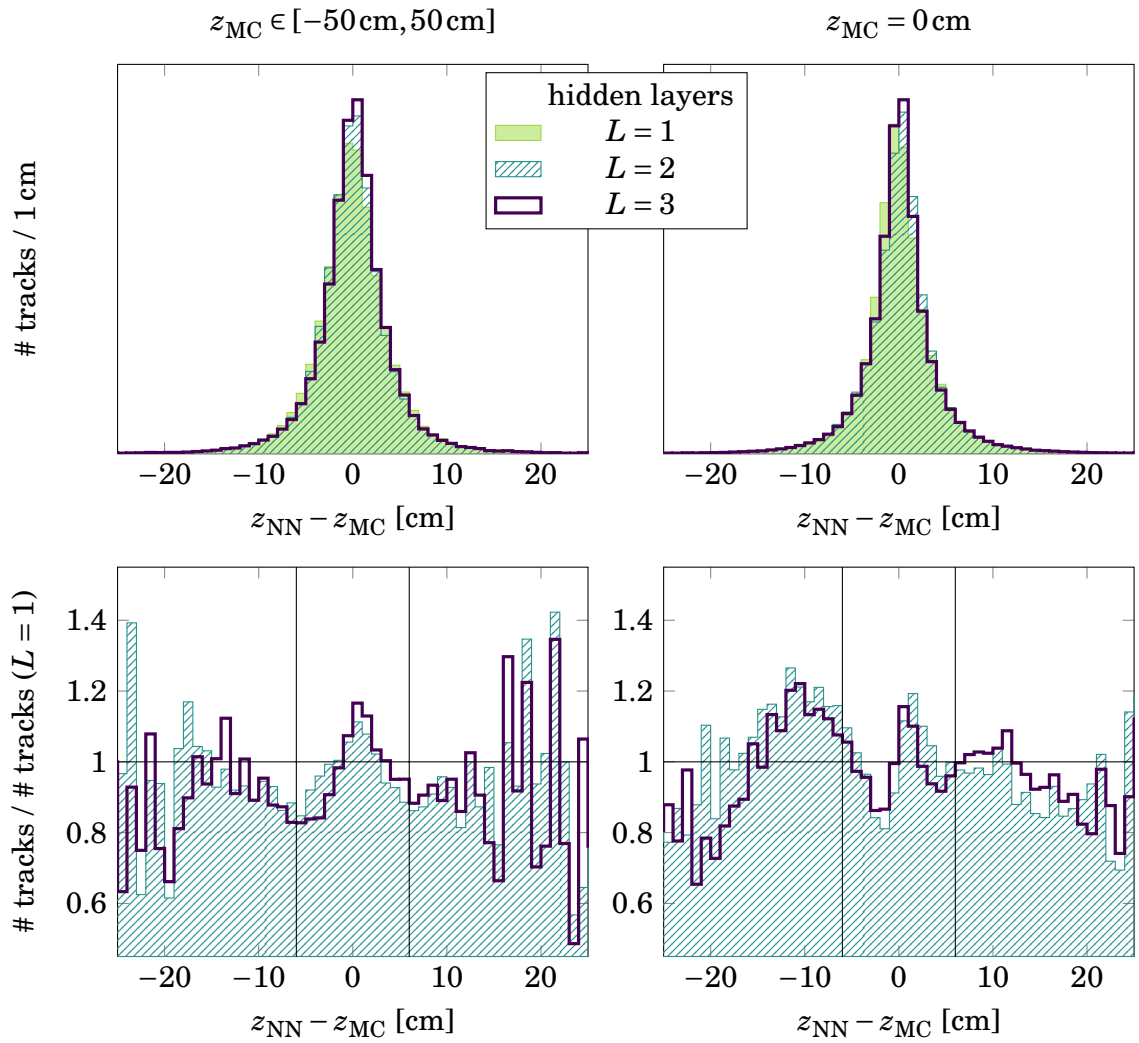

Figure 7.19.: Top row: z-vertex error distribution for three networks with different layer structure. All networks have the same total number of 60 hidden nodes, evenly distributed to $L \in \{1,2,3\}$ hidden layers. Bottom row: ratio of distributions for 2 and 3 hidden layers over the distribution for 1 hidden layer. The default z-vertex cut of 6 cm is shown for comparison. Left column: distributions for the full z-vertex range. Right column: distributions for tracks from the interaction point.

Table 7.7.: Comparison of the performance of networks with $2^n - 1$ hidden nodes, which are optimal for the summation in the output layer (see text).

hidden nodes	resolution	efficiency
63	(3.40 ± 0.01) cm	(87.19 ± 0.08) %
127	(3.45 ± 0.01) cm	(87.72 ± 0.08) %
255	(3.51 ± 0.01) cm	(87.38 ± 0.08) %

algorithm. However, the runtime of the MLP also increases with the number of layers, since consecutive layers cannot be parallelized. Therefore, the deeper structures are not investigated further and the network structure with a single hidden layer is kept. The size of 81 nodes seems reasonable, but a moderate increase can still improve the performance.

To choose the size of the hidden layer, the latency of the algorithm is considered. In a fully parallelized implementation, the multiplications of each layer happen in parallel, followed by the summation of the weighted inputs and the bias in a binary tree. The number of steps in this summation is given by $\log_2(N+1)$ for a layer with N inputs. From this naive calculation, a hidden layer size of $N = 2^n - 1$ is optimal. Table 7.7 shows the resolution and efficiency for a hidden layer size of 63, 127 and 255. There is a small but significant gain for 127 hidden nodes as opposed to the smaller network with 63 hidden nodes, which is also visible in fig. 7.18. The next step to 255 hidden nodes gives no improvement.

In fact the latency of the neural network is more complicated to estimate, as the size of the FPGA does not allow a full parallelization. More details are given in appendix B.2. Nevertheless, the setup with 127 hidden nodes is used in the following studies. Since the algorithm is rather insensitive to the exact number of nodes, adjusting the hidden layer size to the demands of the hardware is simple and does not require any reoptimization.

7.4.2. Weight range

In principle, the training should move the weights quickly towards the minimum in the cost function, whatever initial values were chosen. However, the range of initial values could matter for the previously discussed issue of local minima. Supposing that the minima are more loosely spaced than the initial weight range, the training would always converge to the local minimum closest to the starting point, never finding the global minimum.

An analysis of the optimized weights shows that the most extreme weights are almost two orders of magnitude larger than the initial range of $[-0.1, 0.1]$. The median weight magnitude is much smaller, but still outside of the initial range. Apparently, there is no minimum within the initial region, so the training algorithm does not get

Table 7.8.: Extreme values and median for the trained weights w_{final}, depending on the range of the initial random weights, together with the performance of the resulting networks.

$\lvert w_{\text{initial}}\rvert$	$w_{\text{final}} \in$	median($\lvert w_{\text{final}}\rvert$)	resolution	efficiency
≤ 0.001	$[-56.1, 53.3]$	0.25	(3.39 ± 0.01) cm	(86.84 ± 0.08) %
≤ 0.01	$[-96.9, 85.9]$	0.26	(3.49 ± 0.01) cm	(87.38 ± 0.08) %
≤ 0.1	$[-29.7, 88.0]$	0.25	(3.45 ± 0.01) cm	(87.72 ± 0.08) %
≤ 1	$[-241.1, 47.8]$	0.46	(3.48 ± 0.01) cm	(86.42 ± 0.08) %
≤ 10	$[-206.6, 207.4]$	2.43	(4.08 ± 0.01) cm	(78.15 ± 0.09) %

stuck, but moves outside the initial range towards the minimum in the cost function, as it should.

Nevertheless, the training is repeated with different initial ranges, to check how much influence the initial range has on the training. The results are shown in table 7.8 for ranges that differ by up to two orders of magnitude from the default initial range. For smaller initial weights there is no big difference. The median of the weights is almost identical, and the difference of the performance is within the fluctuations observed in the previous studies (compare fig. 7.18). However, for larger initial weight ranges the behavior changes. The final weights also increase, while the efficiency decreases.

The lesson of this study is that the training algorithm tends to move the weights towards more extreme values. Starting with very small weights, some weights are strongly increased as needed, while the others are fine tuned with respect to each other. On the other hand, when starting with large weights, it is more difficult to find a well balanced optimum. Therefore, the default initial range of $[-0.1, 0.1]$ is used again in the following studies.

7.5. Robustness of the neural network estimate

With the definition of input and target values, the ensemble of expert networks and the structure of the networks, all free parameters of the algorithm are fixed. The next step is to estimate the robustness of the z-vertex estimate with respect to errors in the input values.

7.5.1. Influence of background hits

An important source of errors are background hits, which can lead to distorted or wrong input values for the network. All previous optimization studies included background hits according to the nominal background level, so the estimate for the resolution

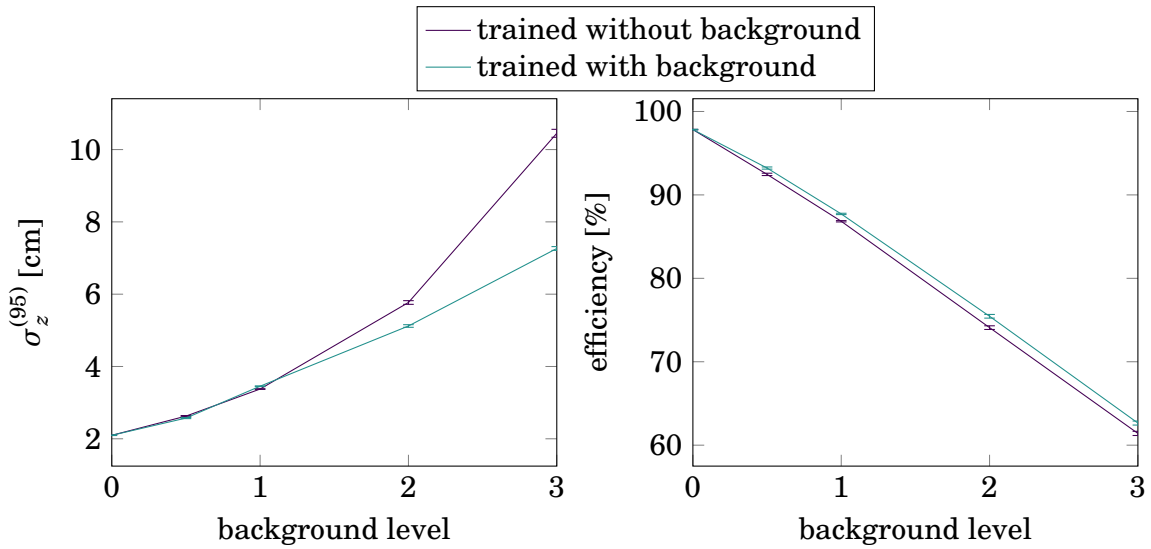

Figure 7.20.: Resolution and efficiency of the z-vertex trigger depending on the background level relative to the simulated nominal background. The networks are either trained and tested with the same background level, or trained without background.

and efficiency should be realistic. However, a realistic background simulation is very difficult, so it is not clear whether the estimated background level is correct. The advantage of the neural networks is that they can adapt to changing background conditions. Once the experiment has started, the networks can be trained on real data which contains the actual background.

To test this scenario, networks are trained and tested with different levels of simulated background. In the simulation, the background level is controlled by scaling the amount of background hits that are mixed into each event. The results are shown in fig. 7.20. Compared to earlier studies [68], the z-vertex trigger is rather sensitive to the background level, although training under the appropriate background condition helps, especially for higher background levels. The efficiency without background is $(97.83 \pm 0.03)\,\%$. The loss with increasing background is approximately linear, with 10 % to 12 % decrease for each integral factor in the background level.

The strong sensitivity is probably a consequence of the high level input representation, which uses more information from the track segments and 2D track parameters, and of the hit selection, which requires exactly one hit per superlayer. Background hits can have the following effects in the track trigger:

1. Background track segments: In the simplest case, additional track segment hits appear, which consist entirely of background hits. Although the track segment shape suppresses noise from isolated hits, track segment hits can be caused for example by particles that are scattered into the CDC under flat angles and cross

only a few layers. Background track segments can distort the shape of the 2D track estimates. In the z-vertex trigger, they should ideally be rejected by the hit selection.

2. Noisy track segments: The other possibility is that track segments include both "real" hits (caused by a trackable particle) and background hits. The position of such track segments is correct, so the 2D track finding is not affected. However, the drift time may be wrong, depending on whether the priority wire signal is caused by a background hit or a real hit. Even if the drift time is correct, the hit pattern is changed by the background contribution, so the left/right state may be wrong. This kind of noise cannot be rejected in the hit selection, unless there are additional clean hits in the same superlayer.

Based on these considerations, the track segment hits that are selected in the neural network are sorted into five quality classes:

clean	All hits in the track segment are related to a particle.
noisy	There are some background hits in the track segment, but the priority hit is related to a particle.
wrong time	Some hits in the track segment are related to a particle, but the priority timing signal is caused by a background hit.
background	All hits in the track segment are background hits.
missing	There is no hit candidate for a superlayer.

The quality classes are ranked, that is a noisy hit is considered as worse than a clean hit, a wrong time hit is considered as worse than a noisy hit and so on. To study the impact of different background types on the neural network results, a track quality class is defined as the lowest quality class among the selected stereo hits. Figure 7.21 shows the frequency of different track quality classes and their respective efficiency for the nominal background level.

Several observations can be made about the different background effects. When a background hit is selected, the efficiency corresponds to randomly guessing the z-vertex within the target range of the network. However, background hits are selected in only 0.7 % of the tracks, since they are efficiently rejected by the hit selection. About half of the losses are due to noisy or wrong time hits, which are much harder to reject than background hits. In fact, noisy hits give the correct position and drift time, so they should not be rejected at all. Instead, a more reliable estimate for the left/right state is required.

The rest of the losses are due to tracks with missing hits, which are more vulnerable to background effects in the remaining three hits. Note that with background the number of missing hits increases by $\approx 2\,\%$. This can be caused either by wrong time hits with negative drift time, which "hide" the real hit, or by a distortion of the 2D track such that the hits are not in the relevant region around the track.

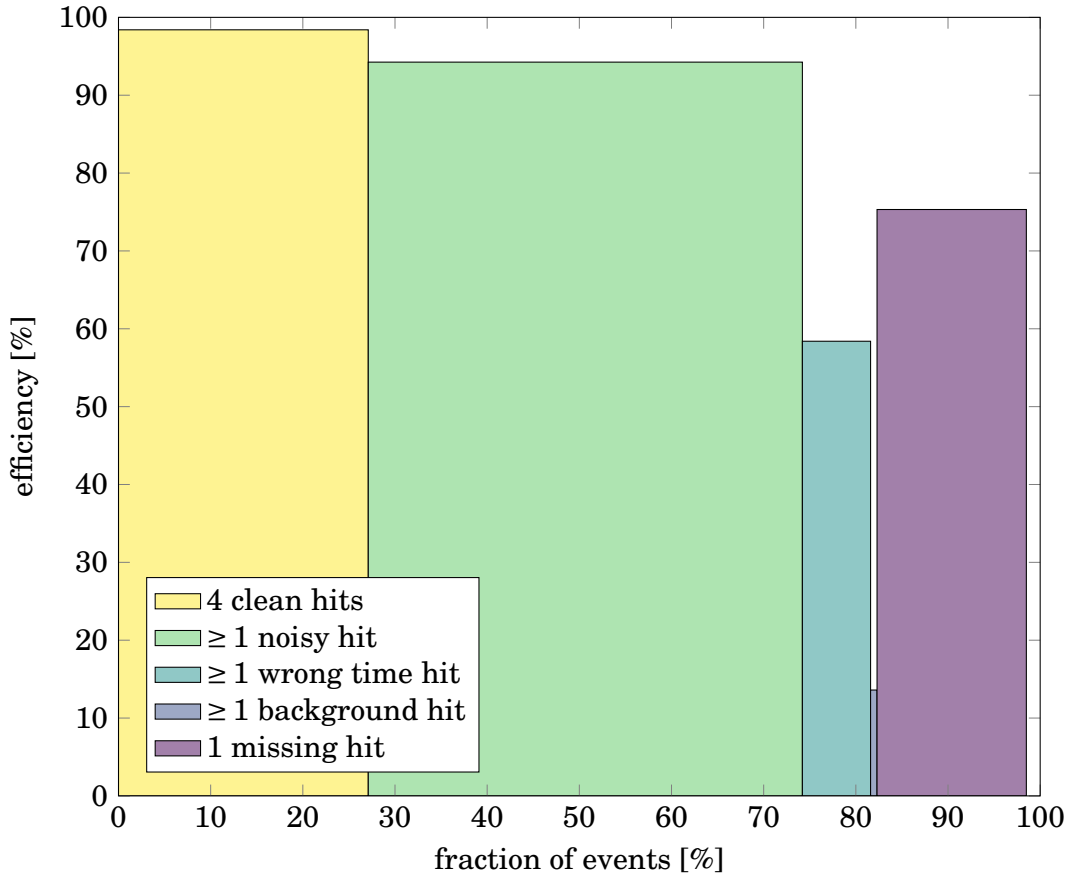

Figure 7.21.: z-vertex trigger efficiency depending on the quality of the selected stereo hits (see text). The track quality class is defined as the quality class of the worst hit. The width of each bar gives the fraction of events with the corresponding quality class. The missing 1.5 % correspond to tracks with less than three stereo hits.

7.5.2. Optimized left/right lookup table for background noise

Table 7.9 shows the fraction of hits with correct left/right state (*correctness*, normalized to the number of hits with decided left/right state) and the fraction of hits with undecided left/right state (*undecided fraction*, normalized to the total number of hits) for events with and without background. As expected, with background the undecided fraction is higher, and even for the hits with decided left/right state the correctness is lower. This difference is the cause for the lower z-vertex efficiency of tracks with noisy hits.

The default left/right lookup table of the track segment finder is trained on events without background, by counting the fraction of hits with true left/true right for each hit pattern (see section 4.1.1). To improve the efficiency for noisy and wrong time track

Table 7.9.: Correctness and undecided fraction for the default left/right lookup table. Clean hits and noisy hits are labeled as “MC” (correct drift time); wrong time hits and background hits are labeled as “Bkg” (meaningless drift time).

background level	MC correct	MC undecided	Bkg undecided
0	$(97.21 \pm 0.02)\,\%$	$(26.01 \pm 0.04)\,\%$	—
1	$(95.03 \pm 0.02)\,\%$	$(33.82 \pm 0.04)\,\%$	$(65.88 \pm 0.06)\,\%$

segment hits, the lookup table should be trained with the appropriate background level. A perfect lookup table would give the correct left/right state also for noisy hits, and undecided for background hits and wrong time hits, where the left/right state has no meaning. This would allow to reject wrong time hits and improve the results for tracks with noisy hits.

The condition for defining a hit pattern as left or right in the default lookup table is given by eq. (4.1):

$$\text{left/right state} = \begin{cases} \text{left} & \text{if } n_L > p \cdot (n_L + n_R) + 3\sigma \\ \text{right} & \text{if } n_R > p \cdot (n_L + n_R) + 3\sigma \\ \text{undecided} & \text{otherwise}, \end{cases}$$

where n_L (n_R) is the number of hits with true left (right) state for the given hit pattern. The threshold p is set to 0.8. When the lookup table is trained with background, some hits have no meaningful true left/right state. Therefore, a third hit count n_B is introduced, which is the number of wrong time hits and background hits for the given hit pattern. This count is included in the left/right condition together with a new threshold parameter b:

$$\begin{aligned} \text{left/right state} &= \begin{cases} \text{left} & \text{if } n_L > p \cdot (n_L + n_R) + 3\sigma \text{ and } n_B \leq b \cdot n_{\text{Total}} \\ \text{right} & \text{if } n_R > p \cdot (n_L + n_R) + 3\sigma \text{ and } n_B \leq b \cdot n_{\text{Total}} \\ \text{undecided} & \text{otherwise}, \end{cases} \\ n_{\text{Total}} &= (n_L + n_R + n_B). \end{aligned} \tag{7.7}$$

In words, the original condition is applied only if the fraction of background hits n_B is below the background threshold b. Otherwise the pattern is defined as background and the left/right state is set to undecided.

Figure 7.22 shows the correctness and the undecided fraction for different thresholds p and b. Clearly there is a trade-off between correctness (which is best for high p) and undecided fraction (which is best for low p). However, there is a region $0.7 \leq p \leq 0.8$ where both the correctness and the undecided fraction are better than with the default

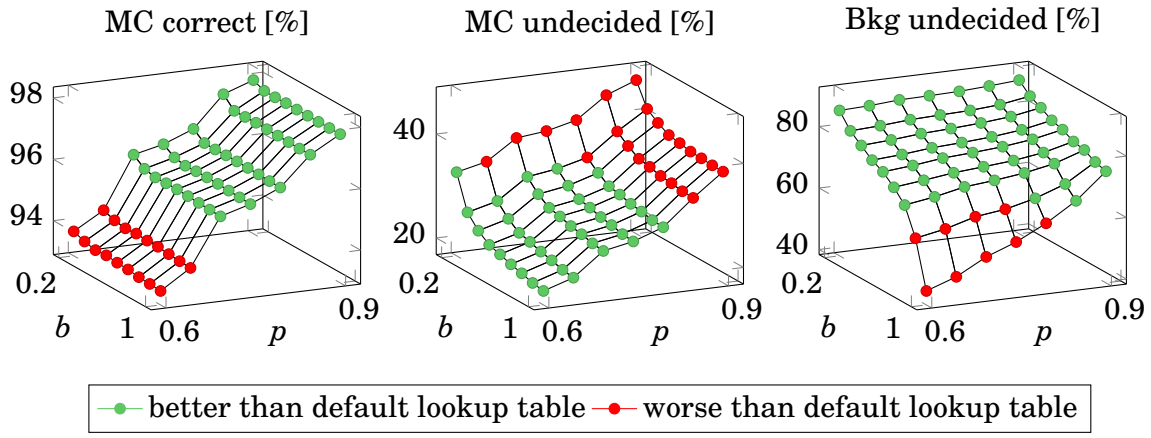

Figure 7.22.: Correctness and undecided fraction for lookup tables trained with different thresholds p and b (see eq. (7.7)). For clean hits and noisy hits ("MC") a high correctness and low undecided fraction is desired. For wrong time hits and background hits ("Bkg") a high undecided fraction is desired.

lookup table. A very low background threshold ($b < 0.4$) leads to a high undecided fraction for clean hits and noisy hits, as too many patterns are classified as background. A very high background threshold ($b > 0.8$) leads to a low undecided fraction for wrong time hits and background hits, so they cannot be identified and rejected.

It is not obvious which parameter is most important for the z-vertex reconstruction: a high correctness, a low undecided fraction or an efficient background rejection. To find the best trade-off, the z-vertex trigger is tested with different lookup tables. The results are shown in fig. 7.23, both for the network trained with the default lookup table and for networks that are specifically trained with the tested lookup table. Naturally the results are better for the case where training data and test data match. However, testing all lookup tables with the same network removes one source of uncertainty, which is the training fluctuations. Therefore, the dependence of the z-vertex efficiency on the thresholds p and b is more clear in the test without retraining.

The region of $0.7 \le p \le 0.8$ and $0.4 \le b \le 0.8$, where all quality figures of the lookup table are better than for the default lookup table, is scanned completely. A clear optimum is found for $p = 0.7, b = 0.8$ in both tests, that is for the lowest possible undecided fraction. In the test with a single network, there is a clear gradient with respect to both p and b. In the test with retraining, the gradient is obscured by training fluctuations, but the same tendency is visible.

Based on these results, the region around the optimum is scanned. In this region either the correctness or the background rejection are worse than for the default lookup table, but the overall performance might still be better. However, no impro-

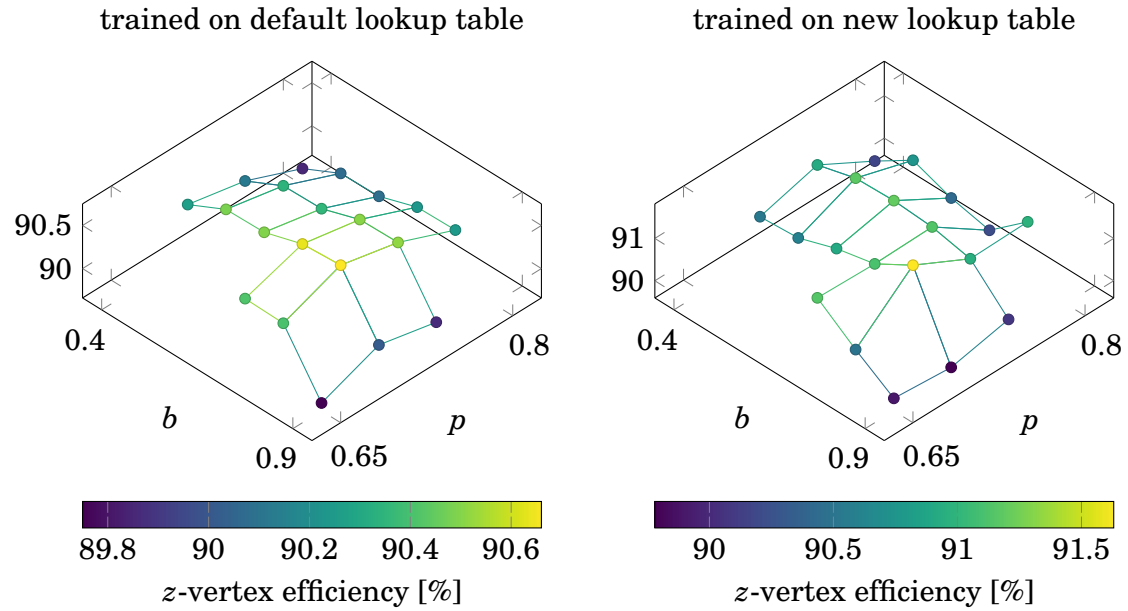

Figure 7.23.: z-vertex efficiency for different left/right lookup tables. The lookup tables are generated according to eq. (7.7) with threshold parameters p and b. Left: results for a network trained with the default lookup table. Right: results for networks trained with the respective tested lookup table.

vement is found, so the lookup table with $p = 0.7, b = 0.8$ is selected as optimal. The z-vertex efficiency for the network trained with this lookup table is $(91.63 \pm 0.06)\,\%$, which is $\approx 4\,\%$ better than with the default lookup table. The z-vertex resolution is $(2.922 \pm 0.008)\,\mathrm{cm}$.

Figure 7.24 once again shows the dependence of the z-vertex trigger performance on the background level. For each background level, a dedicated lookup table is trained. Both the efficiency and the resolution show a clear improvement, especially for high background level.

It is not expected that the optimal threshold values p and b depend strongly on the background, so the selected thresholds of $p = 0.7, b = 0.8$ were used for all background levels. However, if the actual background level in the experiment differs strongly from the simulation and it becomes necessary to train a new lookup table, the region around the selected thresholds should be scanned again to make sure that the threshold values are still optimal.

7.5.3. Event time jitter

Besides the somewhat uncertain background level, there is one other factor that has been ignored so far, namely the event time. To transform drift times into drift lengths,

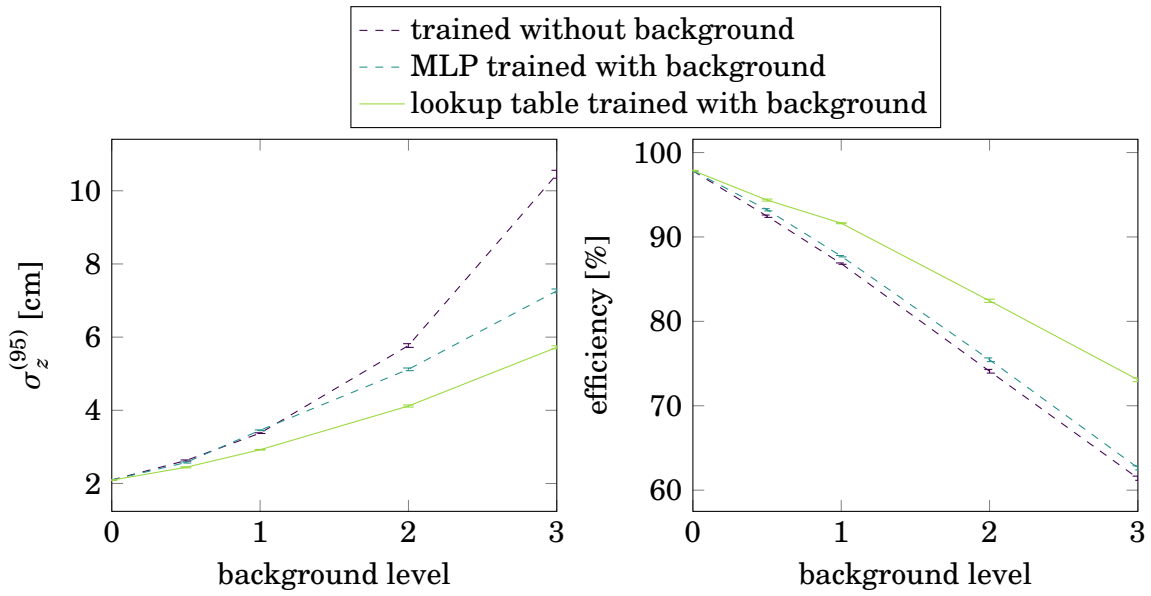

Figure 7.24.: Resolution and efficiency of the z-vertex trigger depending on the background level relative to the simulated nominal background. The dashed curves are identical to fig. 7.20. The solid curve shows the performance for the case where also the left/right lookup table is trained with the appropriate background level.

it is necessary to know the production time of the particle. An estimate is given by the event time finder, which monitors a histogram of the fastest drift time values in each track segment. An event is marked by a sudden increase of hits, so the event time is determined by the first timing bin in the histogram that exceeds a certain threshold of hits.

The event time finder is still being developed, so there is no reliable simulation yet. Therefore, the true event time has been used in the previous studies, which is of course an idealization. In the following, the sensitivity of the z-vertex estimate on an error in the event time is estimated by adding a random time offset to all hits in an event. Since this offset simulates an incorrect event time estimate, it is called T_0. A uniform distribution within a given interval is used to generate one T_0 for each event.

Figure 7.25 shows the performance of the z-vertex trigger for different T_0 intervals. First, symmetric intervals are tested, that is T_0 is drawn from an interval $[-\Delta T_0/2, \Delta T_0/2]$, based on the assumption that the event time finder will not be biased. The resolution is quite insensitive to a wrong event time, with $\approx 1.2\,\text{mm}$ resolution loss for each increase of ΔT_0 by 20 ns. In contrast, the efficiency losses are quite large, with $\approx 1.8\,\%$ decrease for each 20 ns increase of ΔT_0. However, most of these losses are not due to a wrong z-vertex prediction, but because of too many missing hits that slip out of the drift time window and are rejected in the hit selection.

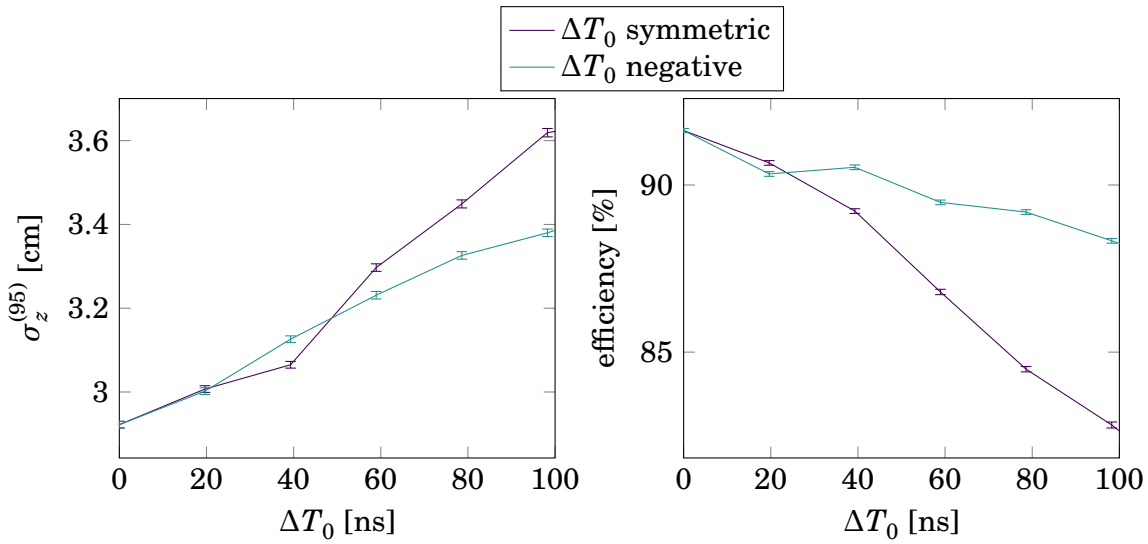

Figure 7.25.: z-vertex trigger performance for different random event time offsets. Symmetric intervals are defined as $[-\Delta T_0/2, \Delta T_0/2]$. Negative intervals are defined as $[-\Delta T_0, 0]$.

Therefore, a second test is performed where T_0 is purely negative, that is the event time interval is defined as $[-\Delta T_0, 0]$. Such an interval is still realistic, because the z-vertex trigger can simply subtract an offset from the estimated event time when calculating the drift times. In other words, if the uncertainty of the event time estimate is known, the event time can be deliberately shifted by that uncertainty. For purely negative event time intervals the efficiency is much better, with only $\approx 0.7\,\%$ loss for each 20 ns increase of ΔT_0.

Note that the tested event time intervals are much larger than the expected uncertainty of T_0. The event time finder aims at an uncertainty of ± 8 ns or better. If this goal can be achieved, the performance losses of the z-vertex trigger are negligible. However, another concern is the efficiency of the event time finder. The histogram method requires a certain number of hits within the same timing bin, but it is not

Table 7.10.: z-vertex trigger performance without initial event time estimate. The event time is either determined from the hit candidates of each track, or the drift times are ignored.

T_0 definition	resolution	efficiency
no drift times	(6.24 ± 0.01) cm	$(62.21 \pm 0.11)\,\%$
shortest drift time	(3.74 ± 0.01) cm	$(84.67 \pm 0.08)\,\%$
shortest axial drift time	(3.84 ± 0.01) cm	$(74.32 \pm 0.10)\,\%$

guaranteed that any bin exceeds the threshold. Especially for low multiplicity events, it can happen that no event time is found. In that case, there are two options for the z-vertex trigger: either set all drift time inputs to 0, or try to determine an estimate for the event time for each track, based on the shortest drift time among all hit candidates. In any case, separate networks should be used to handle events without event time.

Table 7.10 shows the performance of the z-vertex trigger without drift times and with an event time determined from the hit candidates. Even without using any drift times, the resolution increases only by a factor of ≈ 2, although the uncertainty of the individual hit coordinates increases by one order of magnitude. For the case with a track-based event time estimate, two versions are compared, one using the shortest drift time of all hit candidates, the other using only the axial hit candidates to determine the event time. The axial hits have a higher selection purity, so the probability for getting a fake event time from background is lower. On the other hand, there is a chance to lose some stereo hits due to the time window cut. In consequence, the efficiency is significantly better when using all hit candidates for the event time estimate, although the resolution of the two cases is comparable.

Another possibility is to use the event time estimate of another subtrigger, such as the TOP trigger, which is expected to provide the most precise event time estimate. Determining the event time is essential for sending the trigger signal at the correct time, so some estimate must always be available. Therefore, the true event time is used again in the following studies.

7.6. Fixed point calculation

The last step in defining the neural network is to determine the numeric precision that is required in the calculation. So far all simulations were done with 32 bit floating point numbers. In the hardware, only operations on fixed point numbers are possible, that is the number of digits (*bitwidth*) before and after the decimal point must be known precisely. Since the resources on the FPGA are limited, the precision should not be higher than necessary. In the following studies, the bitwidth of the non-integer variables in the z-vertex trigger is varied and the resulting resolution is compared to the results without rounding. The requirement is that the resolution losses from rounding should be one order of magnitude smaller than the ideal resolution.

7.6.1. Floating point variables

In the following, the z-vertex trigger algorithm is once again summarized to identify all intermediate variables that need to be rounded. For each track, the z-vertex trigger needs to execute the following steps:

Calculate track dependent variables: The input representation as defined in section 7.2.1 requires the crossing point φ'_{cross} of the 2D track with the layer and the crossing angle α. Both of these values depend only on the 2D track parameters and on layer dependent geometrical constants, but not on the selected hits. Therefore, they are precalculated for the first and second priority layer of all superlayers. The relevant formulas are given by eq. (6.4):

$$\alpha = \arcsin\left(\frac{1}{2} \cdot r_{\text{wire}} \cdot \omega\right), \tag{7.8}$$

$$\varphi'_{\text{cross}} = \varphi_0 - \alpha\,,$$

$$\text{ID}_{\text{cross}} = \varphi'_{\text{cross}} \cdot \frac{N_{\text{wire}}}{2\pi} \equiv (\varphi_0 - \alpha) \cdot s_\varphi\,. \tag{7.9}$$

In the last line, the crossing point is scaled from radian to the number of wires in the respective layer by the layer-dependent scaling factor s_φ. With this definition, there is no more need to calculate the wire positions φ_{wire} of the individual hits in radian. Instead, the wire number within the superlayer can be directly used. Since there are in general more hits than layers, scaling the 2D crossing points is more efficient than scaling the individual hits. Otherwise the two options are equivalent.

The arcsin() function in eq. (7.8) is implemented as a lookup table. For the wire radius there are 18 possible values (first and second priority for each superlayer). The track curvature estimate ω from the 2D track finding ist also discrete, with 67 possible values. The output of the lookup table is rounded to a variable bitwidth b_α. For eq. (7.9) three more bitwidth variables are defined: b_φ gives the initial bitwidth of the azimuth angle φ_0. The constant scaling factor s_φ is stored with a bitwidth b_s. The final result for the crossing point ID_{cross} is rounded to a bitwidth b_{ID}.

Calculate hit dependent variables: The hit selection requires the relative wire position $\varphi_{\text{rel}} = \varphi_{\text{wire}} - \varphi'_{\text{cross}}$ and the drift time. In accordance with eq. (7.9), the relative wire position is replaced by a relative wire number $\text{ID}_{\text{rel}} = \text{ID}_{\text{wire}} - \text{ID}_{\text{cross}}$. For hits in the first priority layer, the wire number ID_{wire} is identical with the track segment number. Wires in the second priority layer are shifted by a half cell compared to the first priority cell. Therefore, the variable ID_{wire} is always given by an integer or half-integer number and does not require any rounding. The relative wire position thus has the same bitwidth as the 2D crossing point. Similarly, both the priority timing signal and the event time are received as integer values, so the drift time also requires no rounding.

The hit selection is based on logical operations that can be implemented precisely in the hardware. The scaling of the input values $(\text{ID}_{\text{rel}}, \pm t_{\text{drift}}, \alpha)$ is defined such that it corresponds to a bitshift, so this operation can in principle also be performed without rounding. However, for the network calculation it is convenient to define a single precision for all nodes. Therefore, the inputs are rounded to a bitwidth of b_n after the scaling.

Run the neural network: The neural network function is given by eq. (7.2):

$$y_j = \tanh\left(\sum_{i=0}^{N} \frac{x_i \cdot w_{ij}}{2}\right).$$

This function needs to be calculated for each hidden and output node. The input values x_i for the output layer are given by the output values y_j of the hidden layer. For simplicity, all layers are calculated with the same precision, which implies that the bitwidth of the output values is b_n. The bitwidth of the weights b_w is set independently. The activation function tanh() is implemented as a lookup table. The input for this lookup table, that is the weighted sum of the inputs, is rounded down to a bitwidth b_{LUT}. The rounding introduces a bias, which needs to be compensated in the calculation of the lookup table. Details are given in appendix B.3.

7.6.2. Required precision after decimal point

In total, seven separate bitwidth parameters have been defined for the z-vertex trigger. The rounding errors depend on the number of digits after the decimal point, which is optimized in the following. Since φ_0 appears only in eq. (7.9) in a sum with the crossing angle α, b_α and b_φ are optimized together. All other parameters are scanned independently, starting with 20 bit after the decimal point for all bitwidth parameters.

Two types of errors can occur due to rounding. The first type are simple rounding errors, that is the z-vertex estimate z_{fix} obtained with the fixed point calculation is not identical to the z-vertex estimate z_{float} obtained with the precise floating point calculation. Rounding errors are measured by the standard deviation σ_z^{fix} of $z_{\mathrm{fix}} - z_{\mathrm{float}}$. The second type of error can occur in the hit selection, if hits move into or out of the relevant φ_0 region due to rounding errors in $\mathrm{ID}_{\mathrm{cross}}$. If the selected hits in the floating point and fixed point calculation are not identical, the z-vertex results are not comparable. Therefore, the hit selection errors are not included in σ_z^{fix} but counted separately as $N_{\Delta\mathrm{hit}}$.

With 20 bit precision for all parameters, the difference between floating point and fixed point calculation is $\sigma_z^{\mathrm{fix}} = 2\,\mu\mathrm{m}$. The number of hit selection errors is $N_{\Delta\mathrm{hit}} = 15$ for a test sample of 127 000 tracks. Starting from this point, the bitwidth parameters are varied one by one to find the minimal precision that keeps the rounding errors σ_z^{fix} below $\approx 1\,\mathrm{mm}$. The parameters that appear late in the algorithm are optimized first to reduce the effects of rounding error propagation. After each step, the optimized bitwidth parameter is fixed at the selected value, so the rounding errors gradually increase with each optimization step.

Figure 7.26 shows the results of the optimization process. The number of hit selection errors is not affected by rounding in the network calculation, so it is not shown for the first three steps. For most parameters the optimum is quite clear, with a sharp increase of the rounding errors if the precision is reduced further. The bitwidth

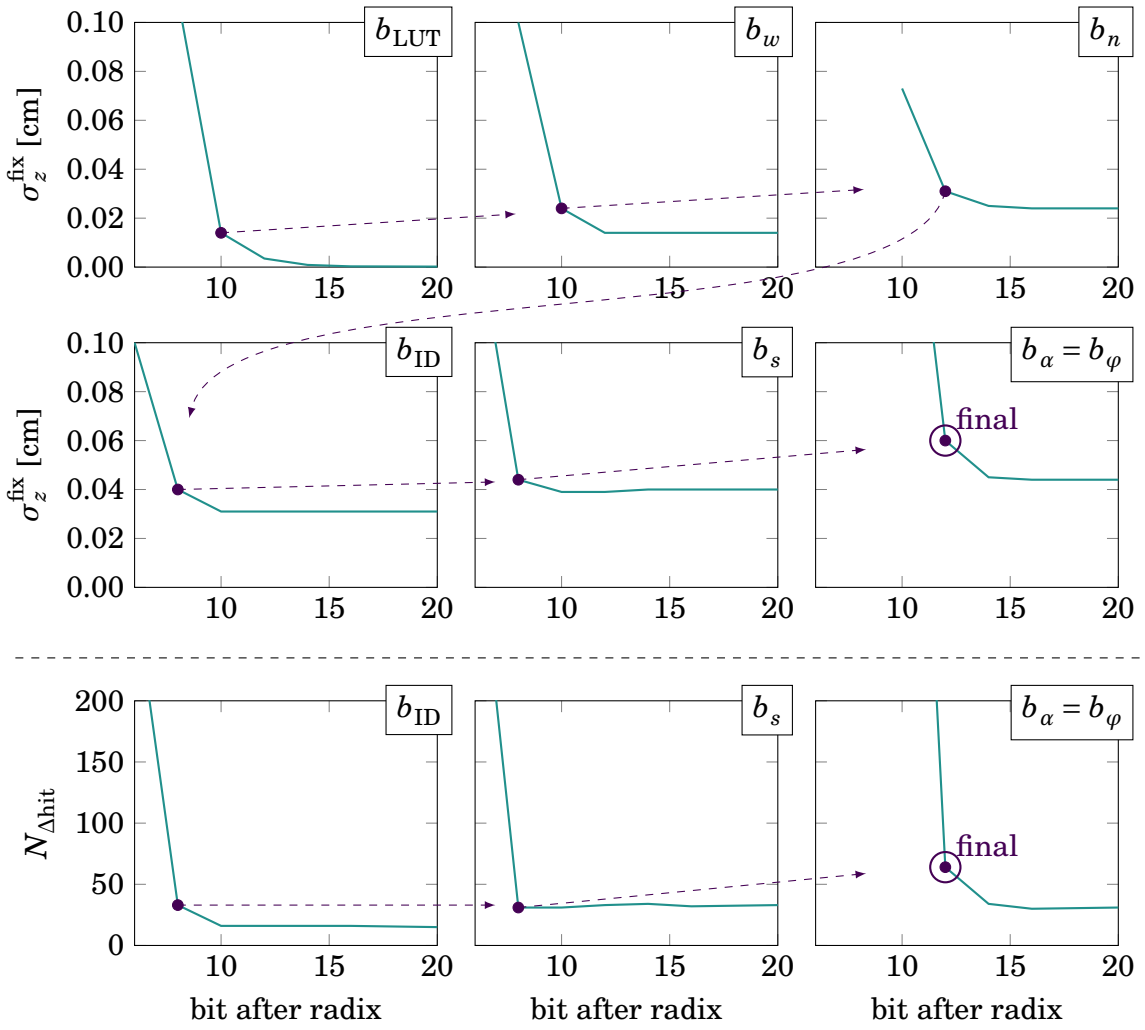

Figure 7.26.: Successive optimization of the bitwidth parameters b_i (see text). The upper two rows show the z-vertex difference between floating point and fixed point calculation. The bottom row shows the number of hit selection errors.

b_n of the network nodes might be decreased by 1 or 2 bit if necessary. For the final choice of parameters, the difference between floating point and fixed point calculation is $\sigma_z^{\mathrm{fix}} = 0.6\,\mathrm{mm}$ for z_0 and $\sigma_\theta^{\mathrm{fix}} = 0.08°$ for the polar angle. The number of hit selection errors is $N_{\Delta\mathrm{hit}} = 64$.

Figure 7.27 shows a direct comparison of the z-vertex trigger performance for floating point and fixed point calculation with the final choice of bitwidth parameters. The resolution with fixed point calculation is $\sigma_z^{(95)} = (2.924 \pm 0.008)\,\mathrm{cm}$. The efficiency for a z-vertex cut of 6 cm is $(91.62 \pm 0.06)\,\%$. Both results agree with the floating point performance within the statistical uncertainties.

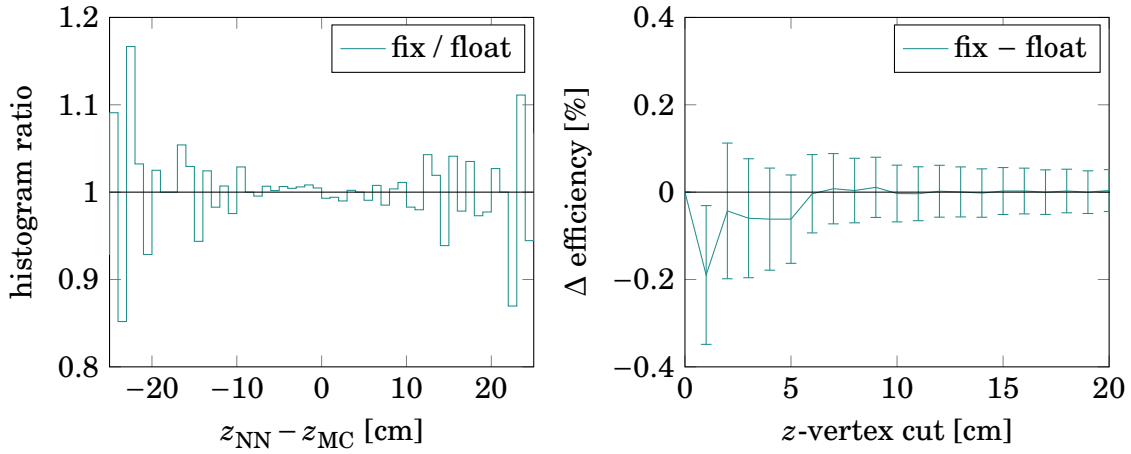

Figure 7.27.: Difference of the z-vertex trigger performance with floating point and fixed point calculation. Left: ratio of the z-vertex error distributions. Right: difference of the efficiency depending on the z-vertex cut.

7.6.3. Total number of digits

The number of digits before the decimal point is given by the maximal parameter range of the respective variable. The parameter ranges and the resulting bitwidths are summarized in table 7.11. Most parameters are limited by definition. The maximum range of the lookup table is derived in appendix B.3. In total, the lookup table contains 9937 bins. Note that the intermediate values for the weighted sum require a larger range, which depends on the number of inputs and on the magnitude of the weights. Only after the full sum is calculated, it can be cut off to 14 bit for the lookup.

The most difficult parameter to estimate are the bitwidths required for the weights. According to table 7.8 half of the weights are below 0.25, but a few individual weights can be much larger. Checking the weights of the tested networks, there are two extreme weights of −472.4 and 134.7, while all other weights are within ±40. Looking deeper into the network structure, it turns out that the two extreme weights lead to the same hidden node. Furthermore, the connections of that hidden node to the output

Table 7.11.: Summary of required bitwidth for the different variables of the z-vertex trigger.

	b_α	b_φ	b_s	b_{ID}	b_n	b_w	b_{LUT}
bit after decimal	12	12	8	8	12	10	10
maximum	$\frac{\pi}{2}$	π	61.1	288	1	≈ 64	9.7
signed	✓	✓	—	✓	✓	✓	—
total bit	14	15	14	18	13	17	14

layer are very weak. In other words, the extreme weights seem to have almost no effect on the final result. After tentatively setting the two extreme weights to ± 64, the resolution and efficiency are unchanged within the uncertainties. So for the network at hand, 6 bit before the decimal point are enough for the weights.

The trouble is of course that the networks are not necessarily final. They are intended to be retrained with real data once the experiment is running. Checking the weight range by hand after every training and making adjustments is not a practical procedure. Simply cutting off large weights after the training without a careful check of the performance is not an option either, so a hard limit on the weights is included already in the training. In each training epoch, the weights are limited to $[-63, 63]$ after the weight update. Thus, the network can compensate the cut-off by adjusting other weights if necessary.

The final networks are trained with this weight limit and 10 training runs for each network. With fixed point calculation, they achieve a z-vertex resolution of $\sigma_z^{(95)} = (2.910 \pm 0.008)\,\text{cm}$, a polar angle resolution of $\sigma_\theta^{(95)} = (3.133 \pm 0.009)°$ and an efficiency of $(91.46 \pm 0.06)\,\%$ for a z-vertex cut of 6 cm. Compared to the networks trained without a weight limit, which had a resolution of $\sigma_z^{(95)} = (2.922 \pm 0.008)\,\text{cm}$ and an efficiency of $(91.63 \pm 0.06)\,\%$), the performance is almost identical within the statistical uncertainties. Therefore, these networks are selected for the efficiency studies in chapter 8.

7.7. Training on reconstructed tracks

So far, the targets for the neural network training were taken from the true particles, which are known for simulated events. However, if there are differences between the simulation and the experiment, the networks may implicitly learn some features of the simulation and not work properly in the real experiment. Therefore, the plan is to retrain the networks on reconstructed tracks when experimental data is available. This will also allow the networks to compensate imperfections of the detector that might not be modeled correctly in the simulation.

On the other hand, training on reconstructed tracks means that the target values for the network are no longer perfect, but distorted by the resolution of the tracking. In the following, the consequences of training on reconstructed tracks instead of true particles are tested.

7.7.1. Training on single tracks

To get a fair comparison with the training on true particles, single muons are generated with the same distribution as in the previous studies, that is with a uniform distribution in the solid angle, the track curvature and the z-vertex. The standard track reconstruction consists of four steps [34]: track finding in the CDC, track finding

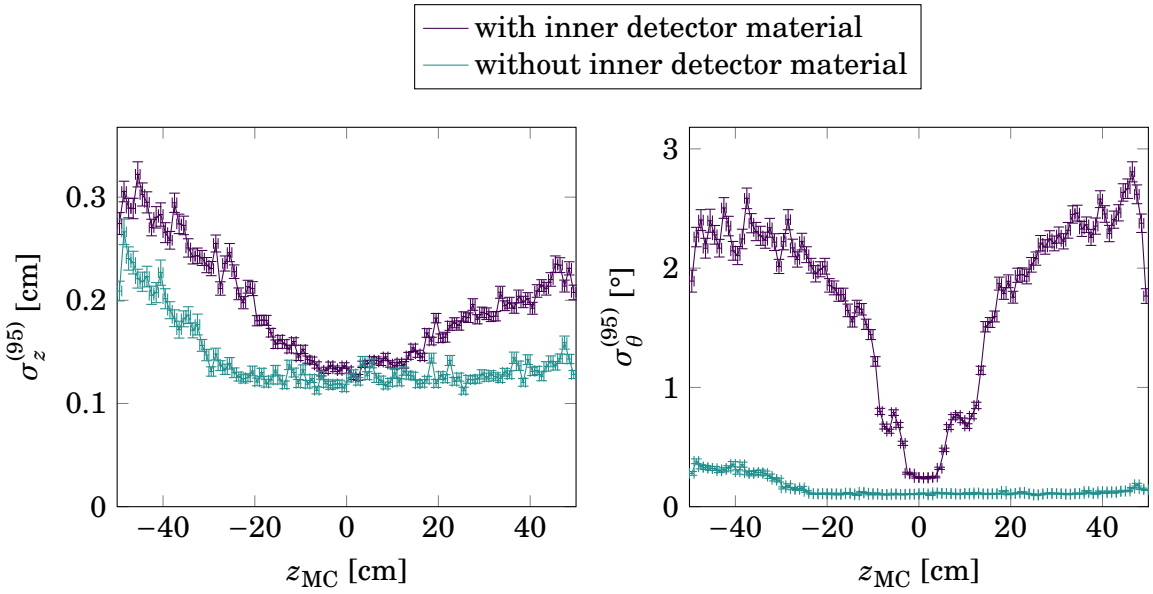

Figure 7.28.: Resolution of reconstructed tracks. Scattering in the inner detector increases the difference between reconstructed tracks in the CDC and the true momentum at the production vertex.

in the VXD, merging the tracks from the CDC and from the VXD and finally fitting the combined tracks. However, the VXD track finding works only very close to the interaction point ($|z_0| \lessapprox 2\,\mathrm{cm}$). Therefore, the VXD track finding and the track merging are skipped for the neural network training. The target values are provided by the CDC track finding and the subsequent fit.

The reconstructed tracks are matched to the trigger tracks by an algorithm based on the respective hit relations, analogous to the matching with simulated particles described in section 5.2.2. The hit relations of the reconstructed tracks are also used to determine the relevant φ_{rel} region for the neural network input (compare fig. 7.10). The target values for z_0 and θ_0 are obtained from the point of closest approach of the reconstructed track to the z-axis.

Figure 7.28 shows the resolution of the z-vertex and the polar angle obtained in the reconstruction. The z-vertex resolution is on the order of mm, which is one order of magnitude better than the resolution of the z-vertex trigger, but the polar angle resolution shows a strong dependence on the z-vertex, with an increase by one order of magnitude for tracks not from the interaction point. However, this increase vanishes when the material of the beam pipe and the inner detector is excluded from the simulation. This means that the difference between reconstructed tracks and true particles is not due to errors in the reconstruction, but reflects actual changes in the particle trajectory, which arise from scattering. The momentum of the true particles is taken at the production vertex, while the reconstructed tracks extrapolate the

Table 7.12.: Comparison of the z-vertex and the polar angle of two sets of neural networks (NN) with true particles (MC) and reconstructed tracks (reco). One set of networks is trained with true particles as targets, the other with reconstructed tracks.

compare	NN trained on	$\sigma_z^{(95)}$	$\sigma_\theta^{(95)}$
NN − MC	MC	(2.911 ± 0.008) cm	$(3.140 \pm 0.009)°$
NN − MC	reco	(3.000 ± 0.008) cm	$(3.123 \pm 0.009)°$
NN − reco	MC	(2.956 ± 0.008) cm	$(2.530 \pm 0.008)°$
NN − reco	reco	(3.038 ± 0.009) cm	$(2.507 \pm 0.008)°$

trajectory in the CDC to the z-axis. Since the z-vertex trigger uses only hits from the CDC, the extrapolated values from the reconstruction might actually give better targets.

The resolution without the inner detector material gives an estimate for the error of the reconstruction. The z-vertex resolution is (0.1375 ± 0.0005) cm and the polar angle resolution is $(0.1344 \pm 0.0006)°$. For $z_0 < -30$ cm the errors start to increase, as tracks no longer pass the full detector, but the resolution is still one order of magnitude better than the resolution of the z-vertex trigger.

Table 7.12 compares the resolution of networks trained on true particles with networks trained on reconstructed tracks. The difference in the z-vertex resolution is $\mathcal{O}(1\,\text{mm})$, which corresponds to the errors of the reconstruction. For the polar angle resolution, the network trained on reconstructed tracks is even slightly (though not significantly) better. The neural network results can also be directly compared to the reconstructed tracks. For the z-vertex, the results are very similar to the resolution with respect to the true particles. For the polar angle, the resolution with respect to the reconstructed tracks is significantly better, confirming that the reconstructed tracks give a better estimate for the trajectory within the CDC.

7.7.2. Cosmic ray test

The first real test of the trigger hardware will be performed with cosmic rays. In contrast to the expected machine background, cosmic rays do not necessarily pass the detector at the beamline, so special networks need to be prepared for the altered conditions. For this purpose, cosmic rays are simulated with the CRY generator [78]. Particles are generated on the surface of a large cube with an edge length of 50 m. The angular distribution of cosmic rays is proportional to $\cos^2(\theta')$ [11], where θ' is the angle to the vertical axis, that is the y-axis of the Belle II coordinate system. To get particles that pass the Belle II detector, the CRY generator is configured to skip particles that do not cross a volume of $|x| < 4.2\,\text{m}$, $|y| < 4.2\,\text{m}$ and $|z| < 4.2\,\text{m}$ around

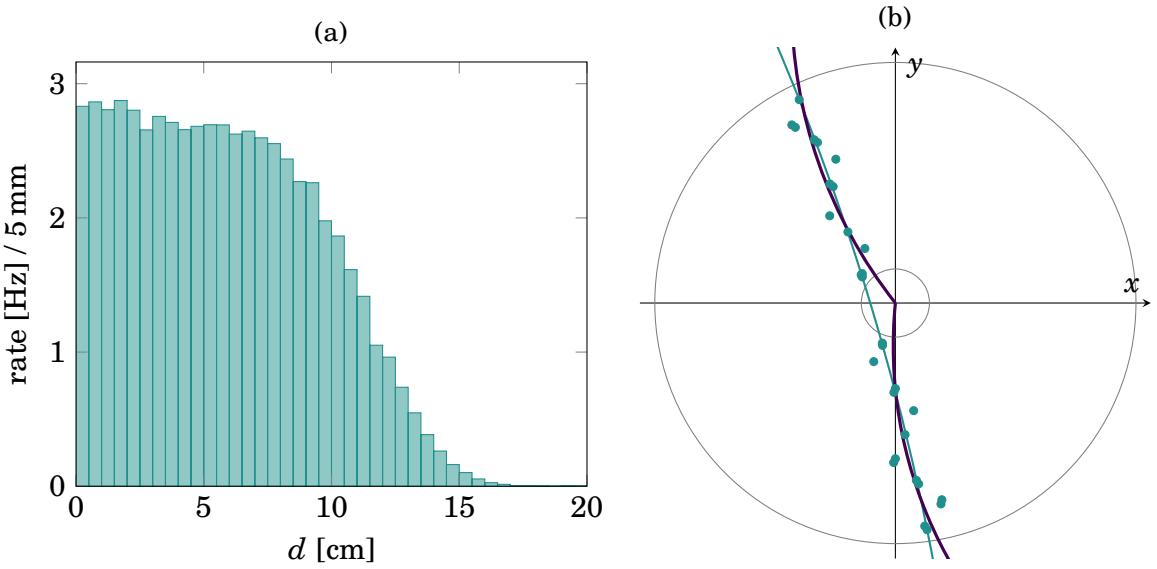

Figure 7.29.: (a) Track finding rate depending on the transverse distance d between the z-axis and the track. (b) Example event with large d. In the trigger, two tracks through the origin are found, with an increased curvature instead of the displacement.

the origin. These borders are chosen such that they contain the full volume of the magnetic field map used in the simulation. For all particles within this volume, the detector simulation, the track reconstruction and the trigger simulation up to the track finding are executed. The detector simulation includes all detector components except for the vertex detector, which is not installed in the cosmic ray test. The reconstructed tracks are used as reference for the trigger test. As shown in the previous section, they give a good estimate for the trajectory in the CDC, while the true particle position and momentum are only known at the production vertex, which is 50 m from the detector. Extrapolating the true particle to the beamline is not trivial, since particles pass the inhomogenous magnetic field outside of the solenoid. In addition, the true particles are of course not known in the real experiment.

The reconstructed tracks are evaluated at the point of closest approach to the z-axis. The transverse distance d of this point to the z-axis is the most important characteristic of a cosmic particle track. In the trigger, the Hough transformation of the track finder assumes $d = 0$. Nevertheless, tracks with displacements up to $d = \pm 15\,\mathrm{cm}$ can be found in the trigger, as shown in fig. 7.29a. The displacement is compensated by a shift in φ_0 and $p_\perp$, which increases with d. An example for a track with large d is shown in fig. 7.29b. Typically two tracks are found for each particle, one for the upper half and one for the lower half of the detector. According to the simulation, events with at least one track in the trigger are expected with a rate of

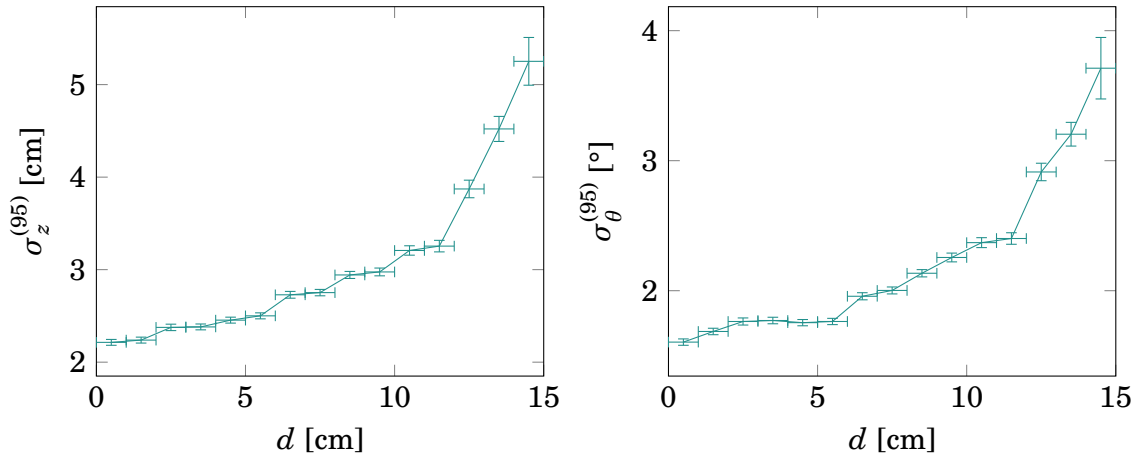

Figure 7.30.: Neural network resolution for cosmic particles, depending on the transverse distance d between the z-axis and the track.

41.25 Hz.

The neural networks need to account for the displacement d to correctly reconstruct the z-vertex. This is achieved simply by training special networks with tracks from the cosmic ray simulation. The input representation and the network structure are unchanged. Information about the displacement is already available from the distance between the axial hits and the 2D track estimate. Figure 7.30 shows the resolution depending on the displacement d. For tracks close to the z-axis, the resolution is better than for the full experiment, due to the absence of background hits. For increasing d, the resolution gets worse, but a reasonable estimate and thus a validation of the z-vertex trigger on cosmic rays is still possible.

8. Trigger Efficiency

In the following chapter, the performance of the track trigger is studied, both independently and in combination with the ECL trigger. The independent track trigger is based on the track multiplicity, combined with a veto on tracks not from the interaction point. To suppress also the uninteresting QED processes, a tentative veto on Bhabha events is derived from the track trigger, based on the track parameter distribution. This veto is then improved by adding the ECL trigger, which allows to identify electrons by matching tracks and clusters that correspond to the same particle. The final efficiency for selected event channels is given for the combination of the proposed vetos and compared to the expected background rate. The event channels selected for the evaluation include generic and specific B and τ events and are described in section 8.2.

8.1. Trigger objects

Before testing the trigger on full events, the two separate 3D reconstruction methods (3D fitter and neural z-vertex trigger) are compared one last time on single tracks. Based on these tests, the various estimates for the track parameters are combined to a single 3D track class to simplify the decision logic. Then, the matching procedure between tracks and calorimeter clusters is described, which is essential for the identification of electrons in Bhabha scattering events. The combined 3D tracks and the track-cluster matches are the basis for the trigger logics tested in this chapter.

8.1.1. Track classes

The various modules of the track trigger output four classes of trigger tracks:

1. *2D finder tracks:* The output of the track finding contains discrete parameter estimates for $p_\perp$ and φ_0. All following modules are based on the track finding, so the 2D finder information is present for all tracks.

2. *neural network tracks:* The neural network adds z_0 and θ_0 estimates to the 2D finder tracks. The 2D track parameters are not altered. Since neural networks are defined only for the case of three or more stereo hits, not every track contains the neural network estimates.

3. *2D fitter tracks:* The fitter first improves the 2D track parameter estimates by including the drift times. Only two axial hits are required for the fit, which is lower than the track finder requirement, so nearly all tracks contain results from the 2D fit. The fit is missing for the rare cases where the hit selection of the track finding fails, for example if the estimated cluster center is not a peak candidate.

4. *3D fitter tracks:* The 3D fitter is based on the 2D fitter results and estimates z_0 and θ_0. In contrast to the neural network, the 3D fitter works already for two stereo hits. In the neural network, tracks with only two stereo hits are skipped, because they are likely to originate from background (compare fig. 7.13). Note that the stereo hit selection of the neural network and the 3D fitter are different, so tracks with a neural network estimate but no 3D fitter result may also occur.

For the track trigger decision, the various track stages are combined to only two track classes: *found tracks* and *3D tracks*. Found tracks are the direct results of the track finding, that is their efficiency is not influenced by the requirements of the following modules. 3D tracks are all tracks that contain any 3D track parameter information. For the 3D tracks, the redundant track parameter information is combined.

For the 2D track parameters φ_0 and $p_\perp$, the 2D fitter results are used when present. The 2D finder estimates are only used when there is no result from the 2D fitter. For the 3D track parameters θ_0 and z_0, different options exist, which have to be evaluated against each other. One possibility is to prefer either the neural network results or the 3D fitter results and use the other method as fallback. This option makes sense if one algorithm is clearly superior to the other one. Another option is to take an average of the two estimates, which improves the results if the errors of the two algorithms are uncorrelated. Last, for the z-vertex veto either the smaller or larger z_0 estimate can be chosen, which is equivalent to requiring that at least one respectively both z_0 estimates are within the z-vertex cut.

Figure 8.1 shows the error distributions of the two methods, as well as the correlation of the errors. Compared to the neural network, the distribution of the 3D fitter is more

Table 8.1.: Resolution and z-vertex efficiency for different strategies to combine the results of the neural network and the 3D fitter (see text).

combination	$\sigma_z^{(95)}$	$\sigma_\theta^{(95)}$	efficiency
network	$(2.985 \pm 0.008)\,\mathrm{cm}$	$(3.195 \pm 0.010)°$	$(92.24 \pm 0.06)\,\%$
3D fitter	$(4.122 \pm 0.017)\,\mathrm{cm}$	$(3.964 \pm 0.013)°$	$(85.13 \pm 0.08)\,\%$
average	$(3.123 \pm 0.011)\,\mathrm{cm}$	$(3.333 \pm 0.010)°$	$(89.91 \pm 0.07)\,\%$
minimum	$(3.559 \pm 0.013)\,\mathrm{cm}$	—	$(95.69 \pm 0.05)\,\%$
maximum	$(3.301 \pm 0.011)\,\mathrm{cm}$	—	$(81.68 \pm 0.09)\,\%$

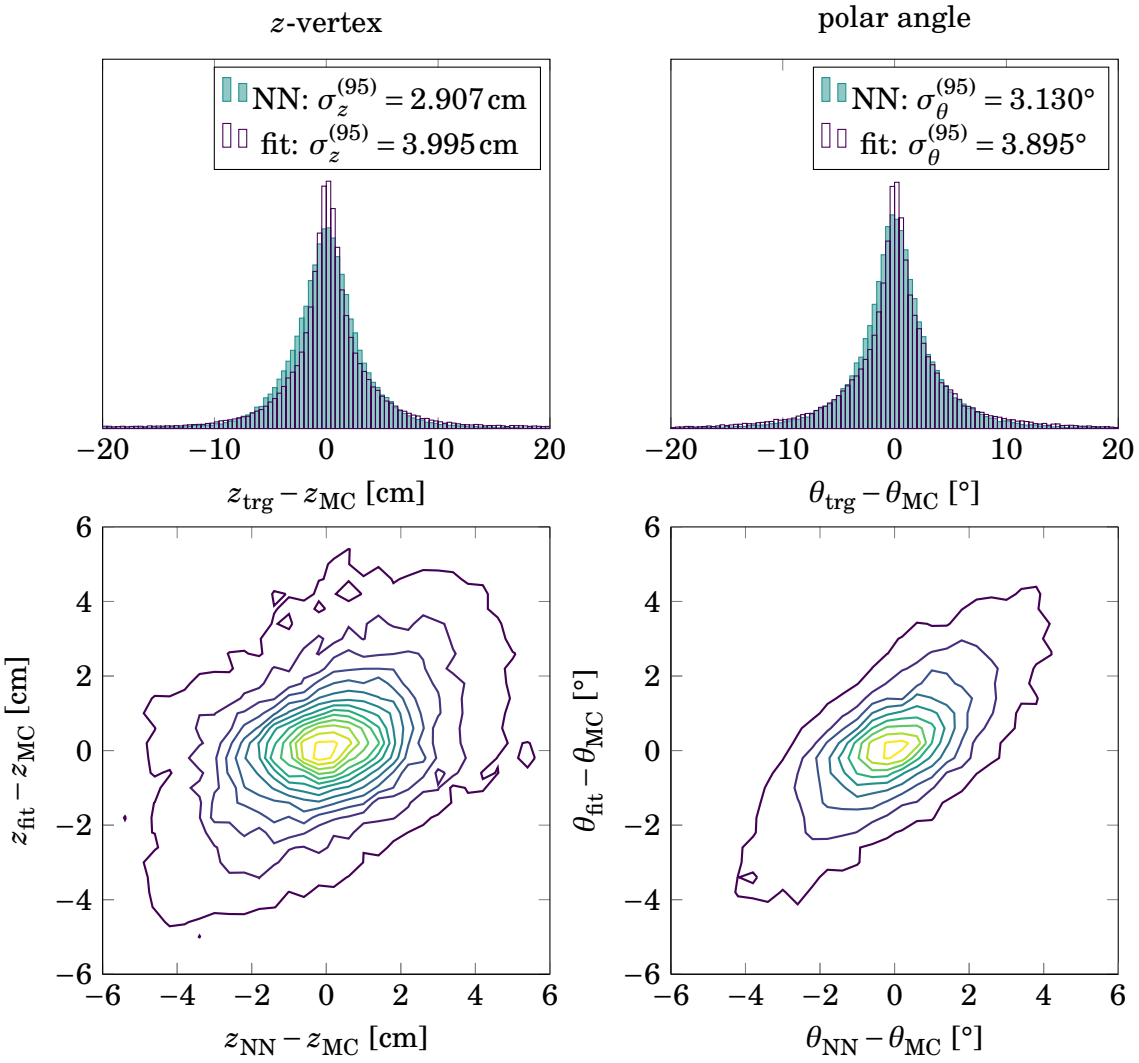

Figure 8.1.: Top row: error distributions for the neural network and the 3D fitter for tracks that have results from both algorithms. Bottom row: Correlation between the errors of the neural network and the 3D fitter.

narrow in the center, but has more outliers. Accordingly, the resolution of the 3D fitter (as given by the trimmed standard deviation) is worse than for the neural network. The polar angle errors have a stronger correlation than the z-vertex errors, with correlation coefficients of 0.466 for θ_0 and 0.265 for z_0. Table 8.1 shows the resolution and efficiency of the combined 3D tracks for different combination strategies. The best resolution is obtained when the neural network results are preferred, with the 3D fitter results used only as fallback. The average of both methods is worse than the neural network estimate, despite the relatively low correlation, since the resolution difference between the neural network and the 3D fitter is too large.

By taking the minimum of the two z-vertex estimates, the efficiency can be improved by $\approx 3.5\,\%$. However, the false positive rate will increase likewise, so this option cannot be selected without first checking the impact on the background rate. Therefore, in the following studies both the *network* option and the *minimum* option are applied and compared.

8.1.2. Matching with calorimeter clusters

As explained in section 4.2.2, the matching between tracks and clusters is based on an extrapolation of the track to the radius r_{ECL} of the calorimeter. Both transverse and longitudinal coordinates are required for the matching, so it is based on the 3D tracks. The longitudinal coordinate $z(r)$ was already derived in eq. (6.11) as

$$z(r_{\mathrm{ECL}}) = z_0 + \cot\theta_0 \cdot \frac{2}{\omega} \arcsin\frac{r_{\mathrm{ECL}}\omega}{2},$$

where ω, θ_0 and z_0 are the track parameters. For the transverse distance, it is most convenient to calculate the azimuthal coordinate $\varphi(r)$, which can be derived from the helix parametrization eq. (6.9):

$$\begin{aligned}\tan\varphi(\mu) = \frac{y}{x} &= \frac{\cos\left(\omega\mu - \varphi_0\right) - \cos\varphi_0}{\sin\left(\omega\mu - \varphi_0\right) + \sin\varphi_0} = \tan\left(\varphi_0 - \frac{\omega\mu}{2}\right)\\ \Rightarrow \varphi(r_{\mathrm{ECL}}) &= \varphi_0 - \arcsin\frac{r_{\mathrm{ECL}}\omega}{2}.\end{aligned}$$

In the second line, eq. (6.10) has been used to express φ in terms of the radius instead of the helix parameter μ. The cluster position is given in cartesian coordinates (X,Y,Z) (denoted by capital letters to distinguish them from the track coordinates). The transverse and longitudinal distances are then given by

$$\begin{aligned}\Delta r &= \sqrt{\left(r_{\mathrm{ECL}} \cdot \cos\varphi(r_{\mathrm{ECL}}) - X\right)^2 + \left(r_{\mathrm{ECL}} \cdot \sin\varphi(r_{\mathrm{ECL}}) - Y\right)^2},\\ \Delta z &= \left|z(r_{\mathrm{ECL}}) - Z\right|.\end{aligned}$$

To allow proper matching also in the endcap sections of the calorimeter, the radius r_{ECL} is calculated from each cluster position as $r_{\mathrm{ECL}} = \sqrt{X^2 + Y^2}$. This allows to apply the same algorithm in the full parameter range.

A track and a cluster are matched if both Δr and Δz are below certain thresholds. Figure 8.2 shows the distributions of Δr and Δz, using the *network* option for the 3D tracks. It was measured on single electron tracks from the interaction point with a uniform momentum distribution from 0.1 GeV/c to 7 GeV/c and a uniform solid angle distribution. For comparison, the distributions of the transverse and longitudinal distance between the cluster position and the true particle position are also shown. For the transverse distance, the uncertainty of the extrapolated track position is negligible

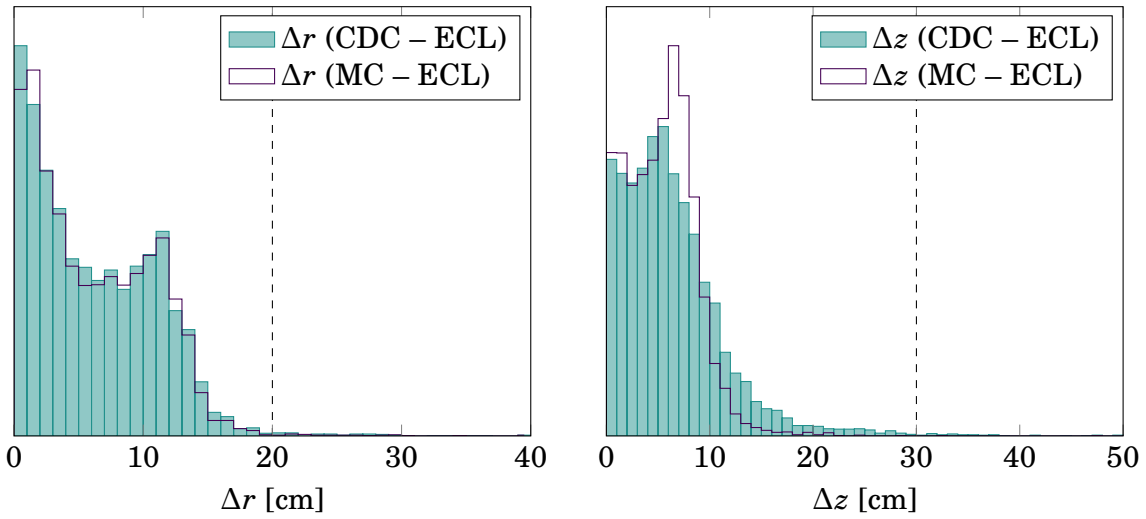

Figure 8.2.: Distribution of distances between CDC tracks and ECL clusters and between true particle positions and ECL clusters. The dashed lines show the selected thresholds for the track-cluster matching. The secondary peaks are a consequence of the discrete cluster positions.

compared to the uncertainty of the cluster position. For the longitudinal distance, the error of the track extrapolation distorts the distribution, so a larger matching threshold is required. From fig. 8.2, thresholds of 20 cm for Δr and 30 cm for Δz are selected for the following studies.

8.2. Tested event types

Two types of signal events are tested, namely $B\overline{B}$ events and $\tau^+\tau^-$ events. The focus is set on events with low track multiplicity. A physical motivation for the selected test channels was given in section 3.3. In the following, they are once more summarized and some technical details are given.

The signal efficiency is compared to the background rate. Two important sources of background have been introduced in section 3.4 and are considered in this chapter: machine background, which includes all tracks that do not originate from e^+e^- collisions, and Bhabha scattering, which dominates the e^+e^- cross section.

8.2.1. B events

$B\overline{B}$ events are simulated with the EvtGen generator [79]. The trigger efficiency generally depends on the selected decay channel of the signal side B meson. Therefore, two exemplary cases are studied: generic $B\overline{B}$ events and invisible B decays. Generic

$B\overline{B}$ events contain both B^+B^- and $B^0\overline{B}^0$ events, with decays according to the normal branching ratios. This corresponds to the most general case that can be studied and gives an average trigger efficiency.

Invisible B decays refer to neutral $B^0\overline{B}^0$ events, with one B^0 decaying to two neutrinos and the other decaying generically. The neutrinos act as placeholders for any particles that are not visible in the Belle II detector. Thus, the efficiency on the invisible B sample gives a lower limit for the trigger efficiency on signals with few or no charged tracks. Examples for such invisible signals are given in section 3.3.2.

8.2.2. Tau events

$\tau^+\tau^-$ events are generated with the KKMC generator [80]. As for $B\overline{B}$ events, both generic decays and a specific decay channel are studied. A particularly interesting case are lepton flavor violating decays, which are forbidden in the standard model, as discussed in section 3.3.1. Among the various possible decay modes, two are of special interest [34]. The decay $\tau \to 3\mu$ is easy to distinguish from background and gives valuable hints to distinguish between different lepton flavor violating models. The decay $\tau \to \mu\gamma$ has the largest branching fraction, if the lepton flavor violating processes are induced by one-loop diagrams including new heavy particles. From the point of view of the track trigger, the process with only one charged particle in the final state is more challenging. Therefore, the decay $\tau \to \mu\gamma$ is selected for the trigger efficiency studies.

8.2.3. Background events

Machine background is simulated with the SAD software [81], which tracks single particles around the storage ring and records the loss positions of particles that hit the beampipe or other machine elements. Particles that are lost in the vicinity of the detector are used as input for the simulation of background events in the detector. The initial distribution of the energy and momentum of the particles in the storage ring is simulated separately for each background type: Touschek scattering, elastic beam-gas scattering or inelastic beam-gas scattering.

For Bhabha scattering, the BHWide generator [82] is used for most of the phase space. Since the BHWide generator cannot be used for scattering angles $\theta \to 0$, the specialized BBBrem generator [83] is used for the extreme forward region of scattering angles $\theta < 0.5°$. Both generators produce the radiative Bhabha scattering process $e^+e^- \to e^+e^-\gamma$.

All types of background are studied in detail in [66, 84]. The details about the background simulation are also described there. In the following studies, the background rates for different trigger conditions have been extracted from [84]. In addition, a special test sample of Bhabha events was prepared with a limited polar angle range, where both the electron and the positron are constrained to be within $[15°, 165°]$ in the

center of mass frame. This limited test sample is used to study the parameter distribution of tracks from Bhabha events and to derive cuts for a Bhabha veto. The resulting rejection rates are extrapolated to the background rates from the full background sample.

8.2.4. Event rates

At the goal luminosity of $8 \times 10^{35}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$, the rate of desirable physics processes is about 4.5 kHz, assuming 100 % trigger efficiency [37]. This includes $B\overline{B}$ events, $\tau^+\tau^-$ events, $\mu^+\mu^-$ events and hadron production from the continuum. The data acquisition requires a maximum average trigger rate below 30 kHz, so the total background rate should not exceed 25 kHz to avoid saturating the trigger with background events.

The following studies consider only trigger conditions that include the track trigger. However, there will also be independent calorimeter triggers that will inevitably increase not only the efficiency, but also the background rate. Therefore, the following optimizations aim at a background rate of about 15 kHz, while keeping the efficiency as high as possible.

8.3. Pure track trigger

The task of the global decision logic is to combine the individual trigger objects to a simple yes/no decision. Different trigger conditions can be defined and combined by an *or*-logic, in other words a trigger signal is sent if one of the trigger conditions is fulfilled. In the following, trigger conditions are compared that include only information from the track trigger.

8.3.1. Multiplicity trigger

The most simple track trigger condition is based on the track multiplicity, that is the number of found tracks in an event. Figure 8.3 shows the track multiplicity per event for $B\overline{B}$ events. The true number of charged particles is shown for comparison, counting only particles within the polar angle acceptance of the CDC of $[17°, 150°]$. In addition to the polar angle cut, particles are required to be produced before they enter the CDC and to not decay before they have left the CDC. With this limitation, the average number of charged particles produced in a $B\overline{B}$ event is 8.1 (≈ 4 from each B meson). The average number of found tracks is 6.1 for generic $B\overline{B}$ events and 2.7 for invisible B decays. The track multiplicity for generic $B\overline{B}$ events is slightly higher than twice the multiplicity for invisible B decays, since the probability of finding additional ghost tracks increases with the number of tracks. The true particles that are not found are mostly due to too low transverse momentum or too flat polar angle, as shown in the bottom row of fig. 8.3, where a particle is considered as found if it can be matched

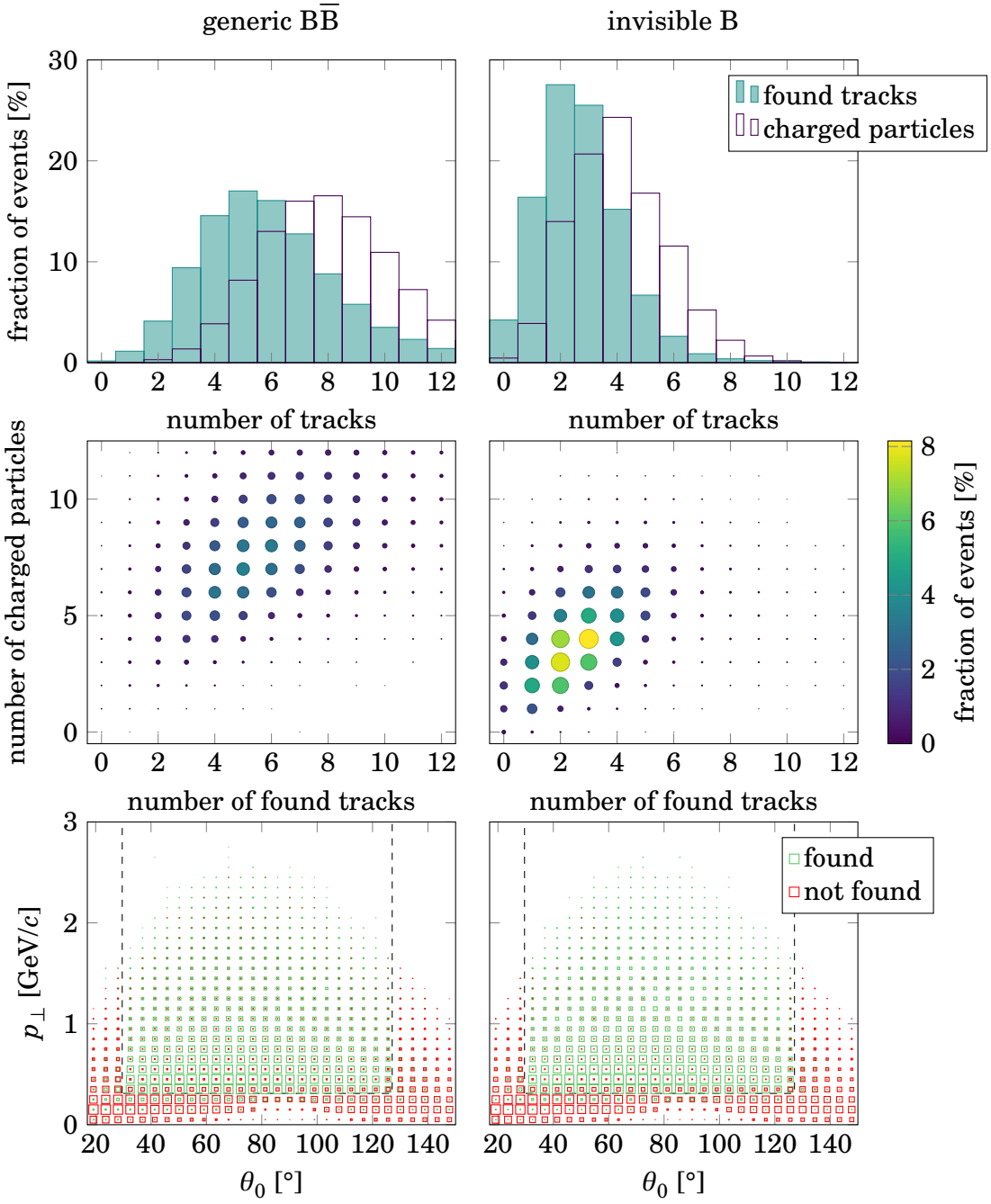

Figure 8.3.: Track multiplicity for generic $B\overline{B}$ events (left column) and invisible B decays (right column). Top row: number of found tracks and true number of charged particles that cross the CDC. Middle row: Correlation between number of found tracks and true number of particles. Bottom row: Parameter distribution of true particles, depending on whether the particle is found. Within the dashed lines, a finding efficiency above 50 % is expected.

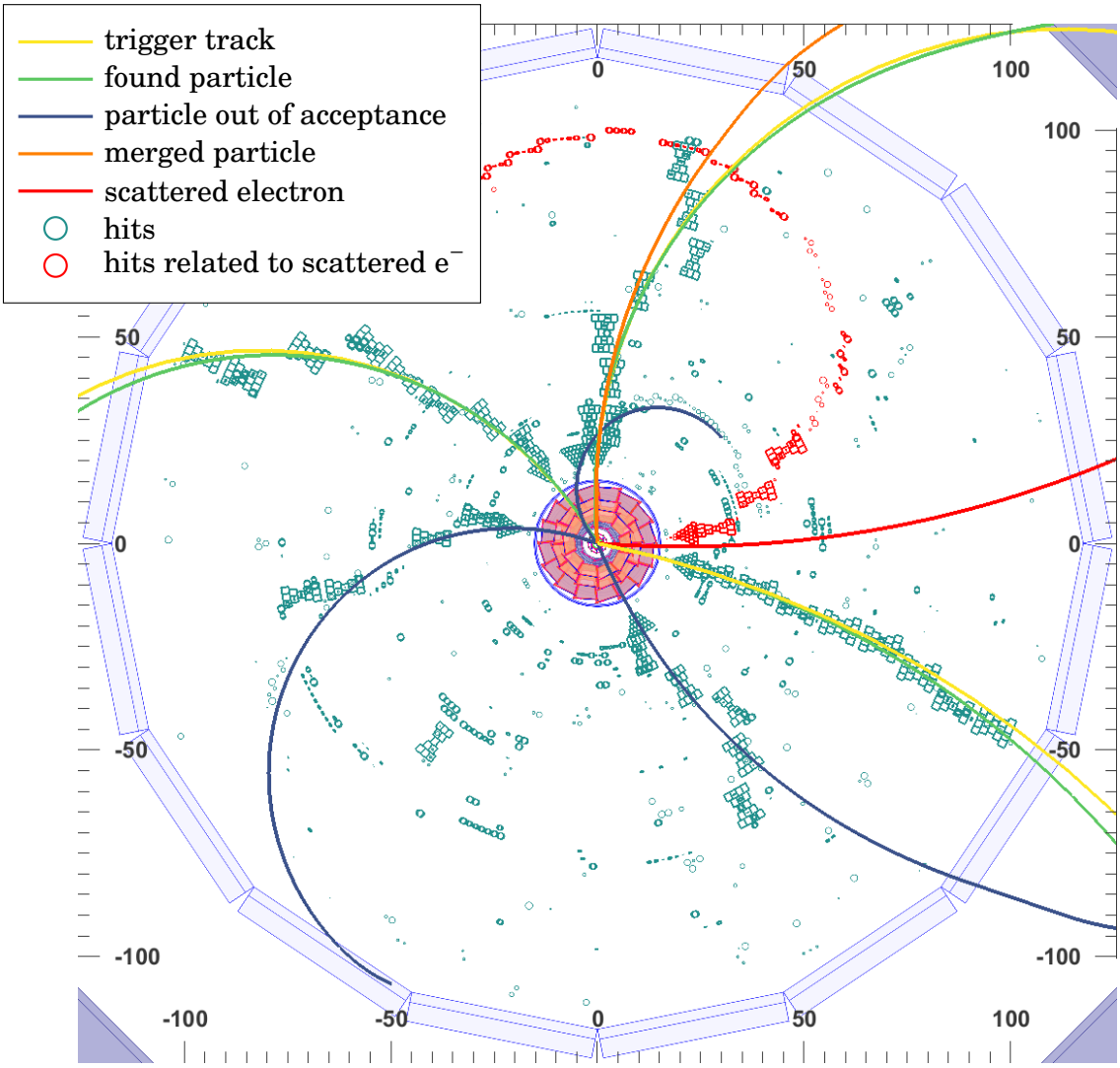

Figure 8.4.: Event display of a $B\overline{B}$ event where two particles generated within the track finding acceptance are not found: one particle is merged to another particle (orange), one electron is scattered out of the acceptance (red).

to a track. Missing particles within the acceptance of the track finding are either due to merging of two particles, or because an electron changed its trajectory due to scattering in the beampipe or vertex detector (the track parameters in fig. 8.3 are taken from the generated particles, before the simulation of the detector). Figure 8.4 shows an example event where both cases occur.

The track multiplicity for $\tau^+\tau^-$ events is shown in fig. 8.5. 85 % of the τ lepton decays have a single charged particle in the final state ("1-prong"), 15 % have three charged particles ("3-prong") and only 0.1 % have five or more charged particles in the final state [11]. Therefore, the track multiplicity is shown separately for 1-prong and

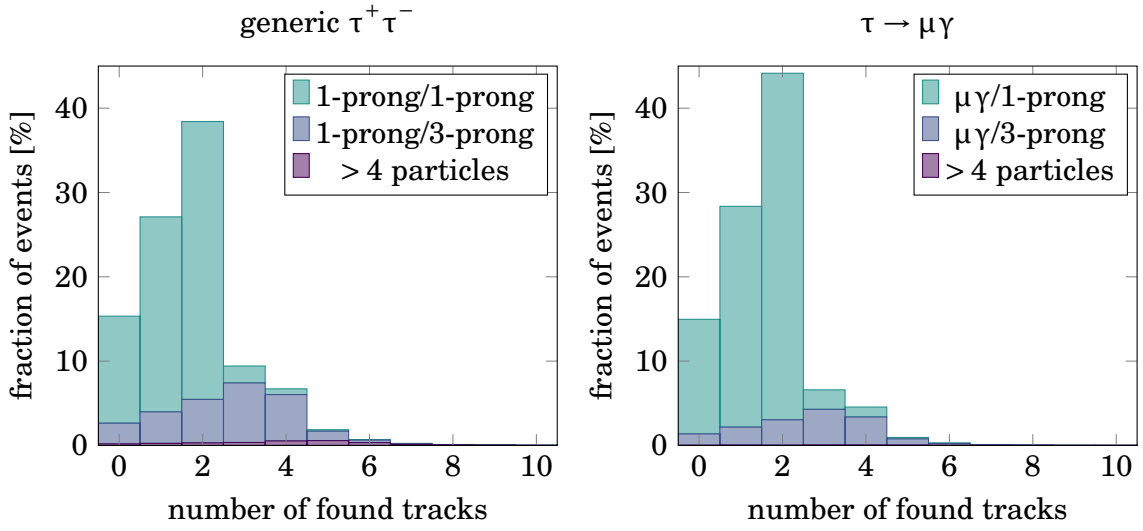

Figure 8.5.: Track multiplicity for $\tau^+\tau^-$ events with generic decays (left) or a decay to $\mu\gamma$ (right), depending on the number of charged particles in the final state of each τ decay.

3-prong decays, regardless of whether the particles cross the CDC or not. The average number of found tracks is 1.4 for 1-prong/1-prong events and 2.6 for 1-prong/3-prong events. The main difference between generic τ decays and the specific decay of one τ to $\mu\gamma$ is the probability to produce 3-prong decays. In addition, the momentum distribution of the muons from $\tau \to \mu\gamma$ decays differs from generic 1-prong decays, since the $\tau \to \mu\gamma$ decay always contains only two particles. Figure 8.6 shows the parameter distribution of the true charged particles for 1-prong, 3-prong and $\tau \to \mu\gamma$ decays. For 1-prong and $\tau \to \mu\gamma$ decays, particles are missing if they are out of the acceptance of the track finding or if they do not fully cross the CDC. For 3-prong decays, there are also 7.6 % losses due to merged particles. In addition, the average momentum of each particle is lower in 3-prong decays, so there are also higher losses due to too low transverse momentum. Note that some particles out of the acceptance are counted as merged, if they are close to a found track.

Table 8.2 shows the trigger efficiencies that can be achieved with a pure multiplicity trigger, depending on the number of required tracks. For generic $B\overline{B}$ events, a good efficiency is achieved already for a three track trigger, where the background rate is very low. In contrast, for invisible B decays and $\tau^+\tau^-$ events, it is essential to trigger also events with two tracks or even only one track. However, the background rate for a two track trigger is already very high. To enable a one track trigger, further constraints on both one track and two track events are necessary.

In the following, both a z-vertex veto and a Bhabha veto are applied to one and two track events in order to get rid of the dominant background sources. The detailed studies of the parameter distributions are carried out only for some of the tested event

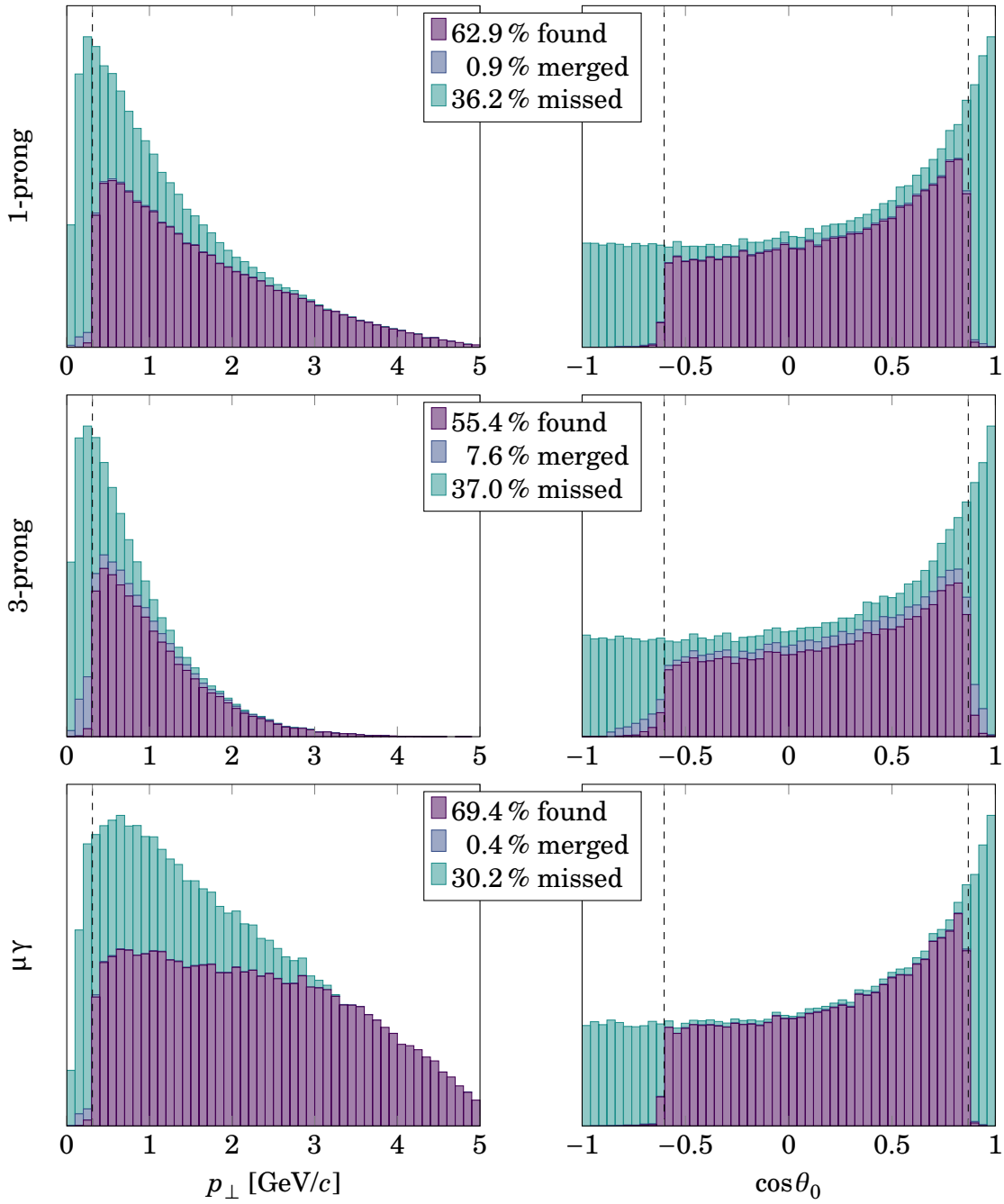

Figure 8.6.: Parameter distribution of true particles for generic $\tau^+\tau^-$ events, shown separately for particles from 1-prong and 3-prong τ decays, and parameter distribution of true muons from $\tau \to \mu\gamma$ decays. Within the dashed lines, a finding efficiency above 50 % is expected. Missed particles within the $p_\perp$ acceptance are mostly out of the θ_0 acceptance, and vice-versa.

Table 8.2.: Efficiency and background rate for different track multiplicity requirements (background rates taken from [84]).

	≥ 1 tracks	≥ 2 tracks	≥ 3 tracks
generic $B\overline{B}$	$(99.83 \pm 0.01)\,\%$	$(98.70 \pm 0.04)\,\%$	$(94.58 \pm 0.07)\,\%$
invisible B	$(95.77 \pm 0.06)\,\%$	$(79.38 \pm 0.13)\,\%$	$(51.84 \pm 0.16)\,\%$
generic $\tau^+\tau^-$	$(84.67 \pm 0.11)\,\%$	$(57.56 \pm 0.16)\,\%$	$(19.13 \pm 0.12)\,\%$
$\tau \to \mu\gamma$	$(85.04 \pm 0.11)\,\%$	$(56.68 \pm 0.16)\,\%$	$(12.53 \pm 0.10)\,\%$
background	121.5 kHz	15.7 kHz	0.5 kHz

channels. For the z-vertex veto, there is no strong dependence on the event type, so only the $\tau \to \mu\gamma$ events, where a low multiplicity trigger is most important, are studied in detail. For the Bhabha veto, $B\overline{B}$ events and $\tau^+\tau^-$ events have different signatures, so both need to be studied separately. Here, the two specific decay channels have been selected for the detailed studies. In both cases, the final efficiencies for all event types are checked after the cuts have been selected.

8.3.2. z-vertex veto

About 2/3 of the background is due to tracks that do not come from the interaction point [84]. This background is referred to as *reducible background* from the point of view of the z-vertex veto, in contrast to *irreducible background* from the interaction point that has to be suppressed by an independent veto. As explained in section 3.4, irreducible background is caused by uninteresting QED processes, which are produced in the e^+e^- collision, but should not be triggered. According to the definition in [84], all background tracks with a true z-vertex $\left|z_{\mathrm{MC}}\right| \leq 1\,\mathrm{cm}$ are considered as irreducible.

Figure 8.7a shows the efficiency per track for $\tau \to \mu\gamma$ events with one or two found tracks, depending on the z-vertex cut. Both the *network* option and the *minimum* option for the combined 3D tracks are shown. As expected, the *minimum* option leads to a better efficiency for the same z-vertex cut. However, the false positive rate of the reducible background also inevitably increases. Figure 8.7b shows the weighted positive rate as defined in eq. (7.6) for both options. Although the positive rate is evaluated on single muon tracks from a particle gun instead of the actual background tracks, it clearly shows the difference between the *minimum* and the *network* option: while the *minimum* option leads to a higher efficiency for tracks from the interaction point, the false positive rate falls much slower beyond the cut than for the *network* option. The background studies in [84] do not include the 3D fitter, so the *minimum* option is not further investigated here. However, it would be worthwhile to compare the trade-off between efficiency and background rate for both options in the future.

Figure 8.8 shows the efficiency per event separately for one track and two track

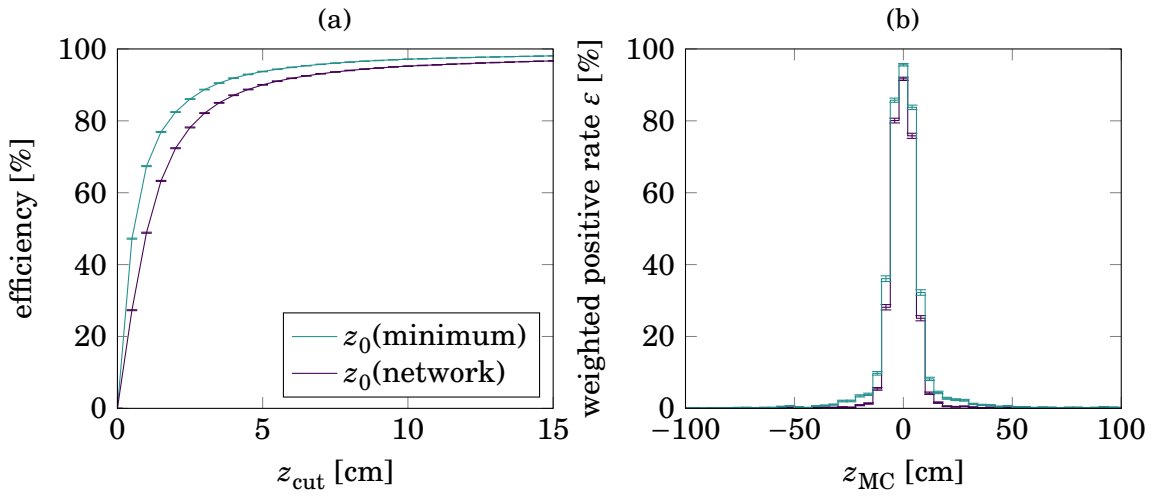

Figure 8.7.: (a): efficiency per track (number of tracks with $|z_0| \leq z_{\text{cut}}$ over total number of tracks) for $\tau \to \mu\gamma$ events with one or two found tracks. (b): weighted positive rate for a cut at 6 cm, according to the definition in eq. (7.6).

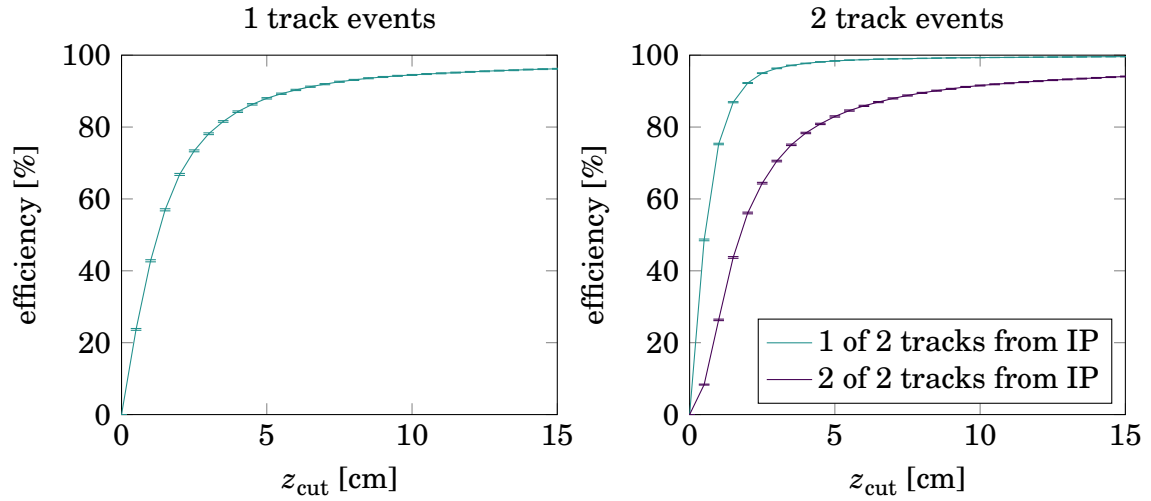

Figure 8.8.: Efficiency per event for $\tau \to \mu\gamma$ events with one or two tracks (number of events with one/two tracks with $|z_0| \leq z_{\text{cut}}$ over total number of events with one/two tracks).

Table 8.3.: Efficiency and background rate after the z-vertex veto (background rates taken from [84]). IP tracks are defined as tracks with $|z_0| \leq z_{\text{cut}}$. The total efficiencies and background rates include also the ≥ 3 track rates without additional constraints from table 8.2.

$z_{\text{cut}} = 10\,\text{cm}$:	1 IP track	2 tracks (≥ 1 IP)	total
generic $B\overline{B}$	$(0.93 \pm 0.03)\,\%$	$(3.85 \pm 0.06)\,\%$	$(99.35 \pm 0.03)\,\%$
invisible B	$(14.27 \pm 0.11)\,\%$	$(26.62 \pm 0.14)\,\%$	$(92.73 \pm 0.08)\,\%$
generic $\tau^+\tau^-$	$(24.87 \pm 0.14)\,\%$	$(37.88 \pm 0.15)\,\%$	$(81.88 \pm 0.12)\,\%$
$\tau \to \mu\gamma$	$(26.80 \pm 0.14)\,\%$	$(43.83 \pm 0.16)\,\%$	$(83.17 \pm 0.12)\,\%$
reducible bkg.	5.4 kHz	0.7 kHz	6.1 kHz
irreducible bkg.	25.5 kHz	10.2 kHz	35.7 kHz
$z_{\text{cut}} = 5\,\text{cm}$:	1 IP track	2 tracks (≥ 1 IP)	total
generic $B\overline{B}$	$(0.81 \pm 0.03)\,\%$	$(3.55 \pm 0.06)\,\%$	$(98.94 \pm 0.03)\,\%$
invisible B	$(12.59 \pm 0.10)\,\%$	$(25.30 \pm 0.14)\,\%$	$(89.72 \pm 0.10)\,\%$
generic $\tau^+\tau^-$	$(22.82 \pm 0.13)\,\%$	$(37.26 \pm 0.15)\,\%$	$(79.21 \pm 0.13)\,\%$
$\tau \to \mu\gamma$	$(24.96 \pm 0.14)\,\%$	$(43.42 \pm 0.16)\,\%$	$(80.91 \pm 0.12)\,\%$
reducible bkg.	1.8 kHz	0.5 kHz	2.3 kHz
irreducible bkg.	16.1 kHz	9.5 kHz	25.6 kHz

events. For one track events, the definition of the z-vertex veto is obvious: the event passes the veto if the z-vertex of the track is within the cut. For two track events, there are two possibilities: either the z-vertices of both tracks have to be within the cut, or at least one of them. If both tracks are required to be within the cut, the event efficiency is approximately given by the square of the per track efficiency. If only one track is required to be within the cut, the event efficiency is much higher. The background studies in [84] suggest that the reducible background rate for two track events is relatively low and can be cleanly suppressed. Therefore, it should be realistic to require only one track within the z-vertex cut.

Table 8.3 shows the trigger efficiencies for the different track multiplicities after the z-vertex veto, as well as the reducible and irreducible background rates, for cuts of 5 cm and 10 cm (in [84] the background rates are given in steps of 5 cm for the z-vertex cut). With the 5 cm cut, the efficiency of the low multiplicity channels is 2 % to 3 % lower than with the 10 cm cut, but the background rate is also 14 kHz lower. Note, however, that most of the background reduction for the tighter cut corresponds to irreducible background, in other words it is a side effect of the lower efficiency for tracks from the interaction point. The final z-vertex cut should probably be chosen somewhere between the cuts shown here, depending on how much of the irreducible

background can be suppressed, and on how much background comes from the other subtriggers.

8.3.3. Bhabha identification

The irreducible background is almost entirely due to Bhabha scattering, so the next step is to identify and reject Bhabha events. In the following, the possibilities of a Bhabha identification based only on the track trigger are investigated. Without information from the calorimeter, it is not possible to identify electrons, but there are still differences in the track parameters that allow to distinguish Bhabha events from $B\overline{B}$ or $\tau^+\tau^-$ events. In section 8.4, the calorimeter information will be added to show how a combination of different subtriggers can improve the trigger decision. The Bhabha veto studies are loosely based on [58].

Bhabha events are characterized by two particles of opposite charge that are emitted approximately back-to-back in the center of mass system and carry almost the total energy of the incoming particles. In radiative Bhabha events, some energy is carried away by the emitted photon, but typically it has very low energy. In contrast, $B\overline{B}$ events and $\tau^+\tau^-$ events contain more than two particles, so the energy of each particle is lower and the momentum is not correlated. The back-to-back condition is illustrated in fig. 8.9 for the center of mass system. Both the sum of the polar angles and the difference between the azimuth angles give 180°. In the laboratory system, the sum of the polar angles becomes smaller due to the boost, but the two polar angles are still correlated. Therefore, all cuts are applied in the laboratory system.

The first condition for the Bhabha identification is an event with exactly two found tracks of opposite charge, both of which have also a 3D reconstruction (otherwise the polar angle and the total momentum cannot be determined). Figure 8.10 shows the distribution of the back-to-back variables for Bhabha events, invisible B events and $\tau \to \mu\gamma$ events that pass this first criterion. Based on these distributions, cuts of $[165°, 190°]$ for $\left|\varphi_0^+ - \varphi_0^-\right|$ and $[130°, 175°]$ for $\theta_0^+ + \theta_0^-$ are selected. Note that the slight

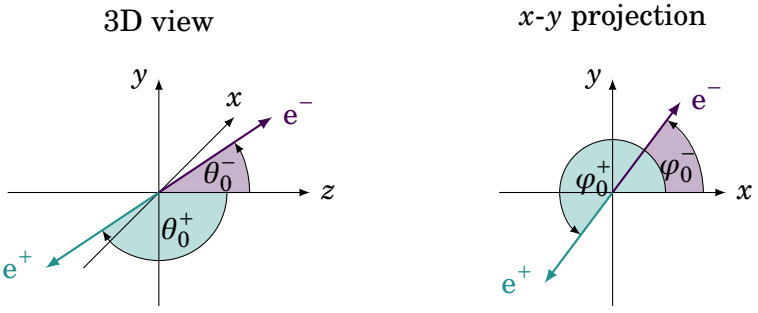

Figure 8.9.: Relation of the polar and azimuth angles of particles that are emitted back-to-back. In the center of mass system, $\theta_0^+ + \theta_0^- = 180°$ and $\left|\varphi_0^+ - \varphi_0^-\right| = 180°$.

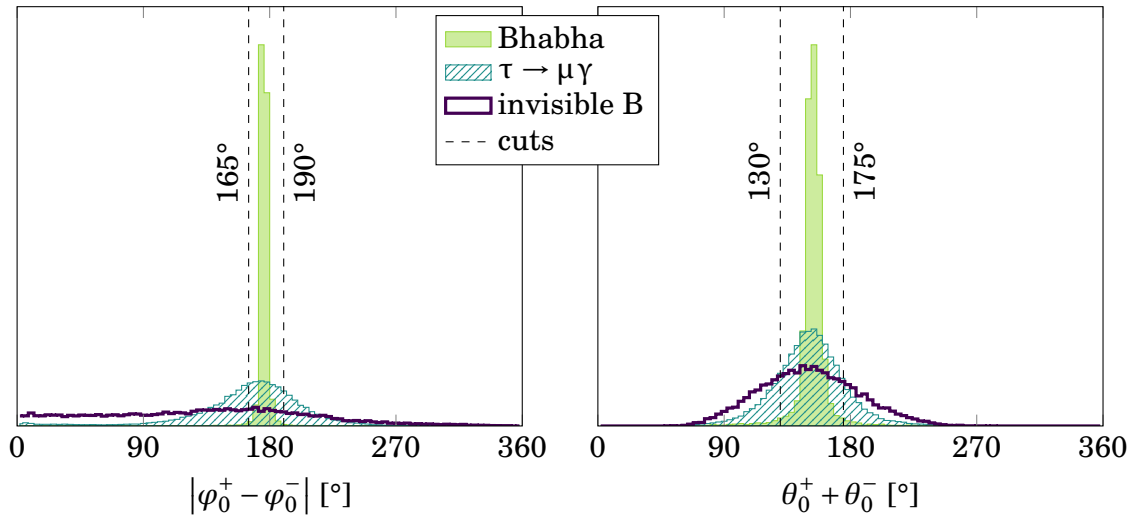

Figure 8.10.: Distribution of the back-to-back variables for events with two 3D tracks of opposite charge. The dashed lines show the selected cuts.

asymmetry in the $|\varphi_0^+ - \varphi_0^-|$ distribution is due to the modulus function.

The opposite charge requirement and the back-to-back cuts already achieve a very good separation between Bhabha events and B events. To distinguish Bhabha events from τ events, the total momentum of the tracks is considered next. Figure 8.11 shows the total momenta $p^\pm = p_\perp^\pm / \sin\theta_0^\pm$ of the two particles for Bhabha events and $\tau \to \mu\gamma$ events. As expected, the particles in the τ events have lower momenta, so a reasonable separation can be achieved. With cuts of $p^+ \geq 2\,\mathrm{GeV}/c$, $p^- \geq 3\,\mathrm{GeV}/c$ and $p^+ + p^- \geq 7\,\mathrm{GeV}/c$, only 5 % of the $\tau \to \mu\gamma$ events are misidentified as Bhabha events, while 81 % of the Bhabha events are correctly identified.

However, there is one class of events that cannot be identified by the back-to-back condition and the momentum cuts, namely $\mu^+\mu^-$ events. Although these events are not the primary focus of the present studies, they are necessary for alignment and should not be rejected by a Bhabha veto. Since muons don't decay within the detector, $\mu^+\mu^-$ events exhibit the same back-to-back signature with high momenta as Bhabha events. However, the polar angle distribution of Bhabha events is different from $\mu^+\mu^-$ events, due to the t-channel contribution that is not present in $\mu^+\mu^-$ production (see fig. 3.10 in section 3.4). Figure 8.12 shows the polar angle distributions separately for the positively and the negatively charged particles. For Bhabha events, the electron is preferably emitted in the forward direction, while the positron is emitted in the backward direction. For $\mu^+\mu^-$ events, the distribution is almost flat and independent of the charge (the distribution of the true polar angle before any filtering would follow a $(1 + \cos^2\theta_0)$ distribution). This allows some separation between Bhabha events and $\mu^+\mu^-$ events, although due to the overlap between the distributions, it is not possible to rescue all $\mu^+\mu^-$ events.

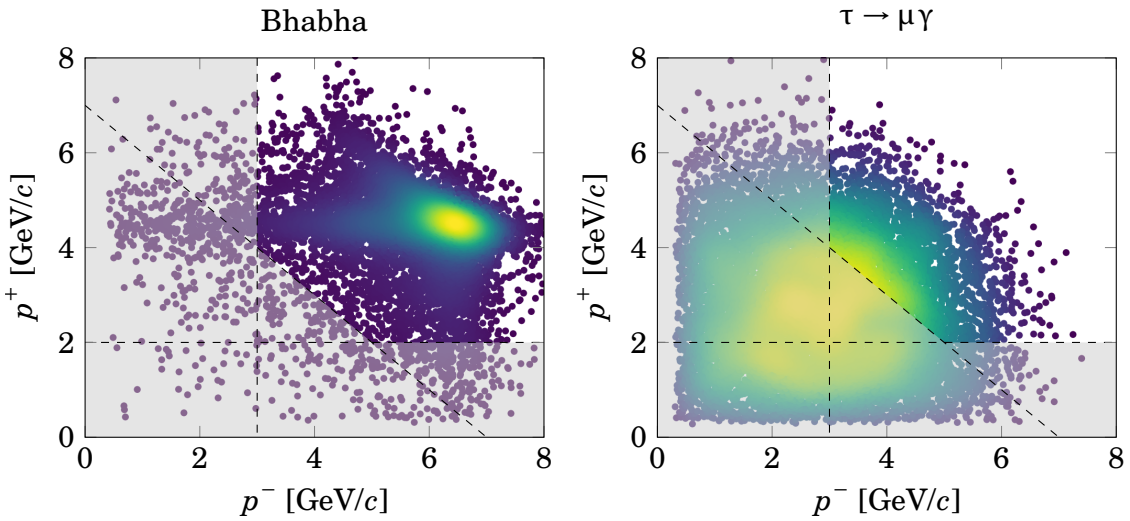

Figure 8.11.: Total momenta of the positively and negatively charged particles in events that pass the back-to-back cuts. The color shows the point density. The dashed lines show the selected cuts of $p^+ \geq 2\,\text{GeV}/c$, $p^- \geq 3\,\text{GeV}/c$ and $p^+ + p^- \geq 7\,\text{GeV}/c$. The shaded area shows the combination of the cuts.

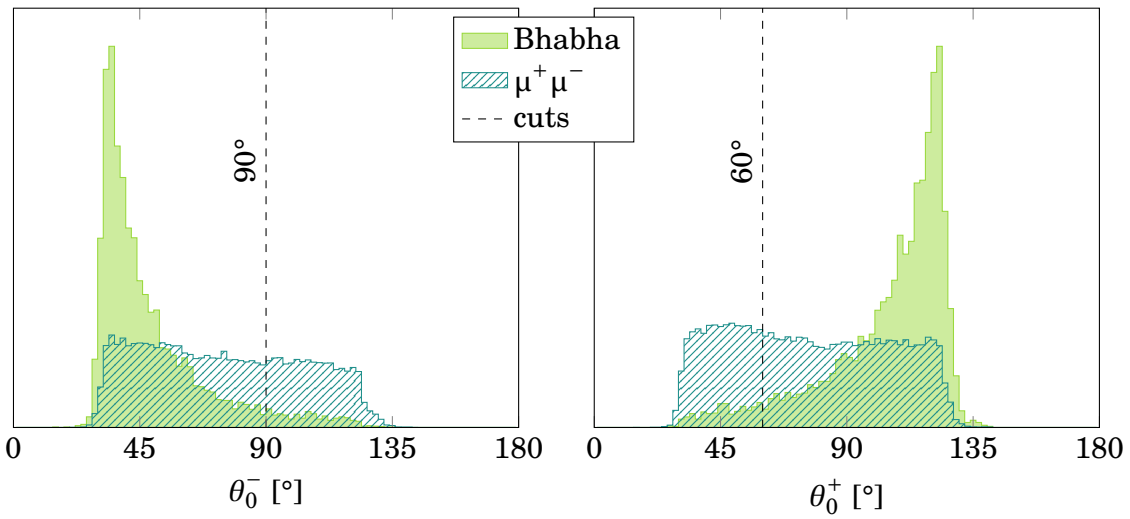

Figure 8.12.: Polar angle distribution of Bhabha events and $\mu^+\mu^-$ events that pass the back-to-back and momentum cuts. The dashed lines show the selected cuts.

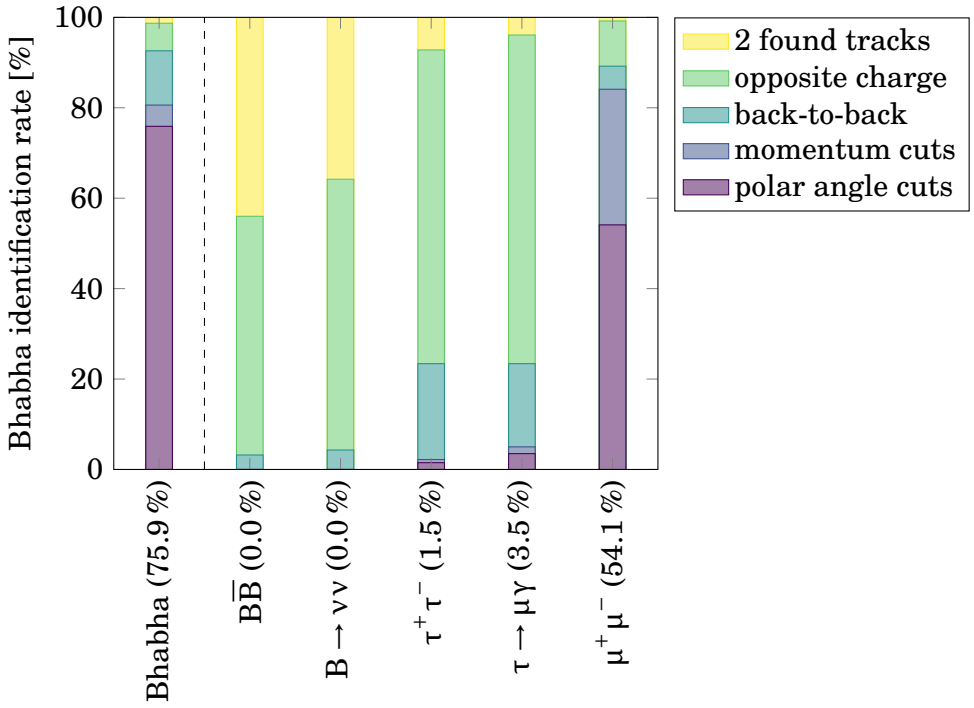

Figure 8.13.: Fraction of events that pass the successive cuts for the Bhabha identification (normalized to the number of events with exactly two found tracks). The numbers in brackets give the final (mis)identification rate.

The resulting event fractions identified as Bhabha events after the various cuts are shown in fig. 8.13. About 75.9 % of the Bhabha events can be rejected, while keeping a very low misidentification rate for $B\overline{B}$ events and $\tau^+\tau^-$ events. For $\mu^+\mu^-$ events, there is still a misidentification rate of 54.1 %, which is hard to improve without calorimeter and/or KLM information. The momentum and polar angle cuts have been chosen somewhat arbitrarily and might be optimized further. However, the present results should be sufficient as a reference for a Bhabha veto that includes the calorimeter.

The next question is whether some kind of Bhabha veto is also possible for events with only one track. If it is possible to sufficiently reduce the background rate, the trigger rate for low multiplicity events could be significantly increased. However, with only one track the opposite charge criterion and the back-to-back cuts do not work. The only information available is the charge of the track, the polar angle and the total momentum. The best separation power is given by the polar angle, which is strongly charge dependent for Bhabha events, but not for the signal events. The polar angle distributions for one track events are shown in fig. 8.14. As for the case with two tracks, some separation is possible, although it is far from perfect. A further improvement is possible with a cut on the total momentum. Figure 8.15 shows the momentum distribution, again depending on the charge of the track. Note that tracks with low

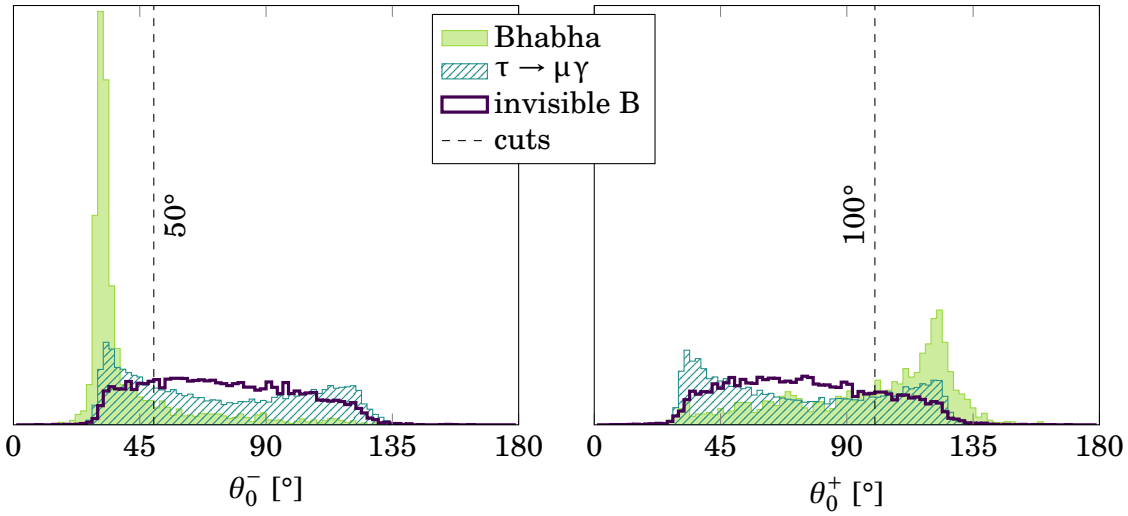

Figure 8.14.: Polar angle distribution for events with one found track, depending on the charge of the track. The dashed lines show the selected cuts.

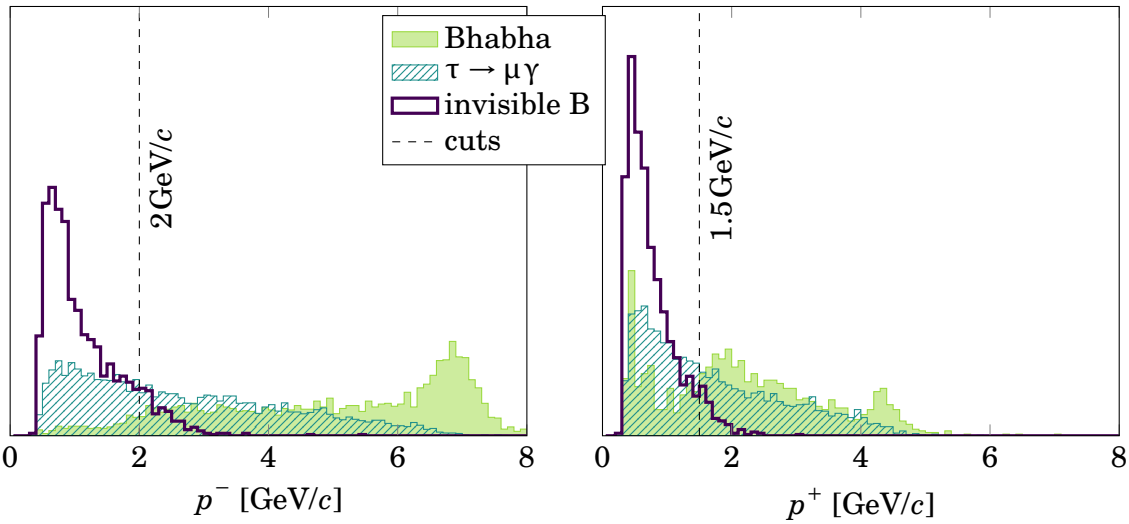

Figure 8.15.: Momentum distribution for events with one found track that passes the polar angle cut, depending on the charge of the track. The dashed lines show the selected cuts.

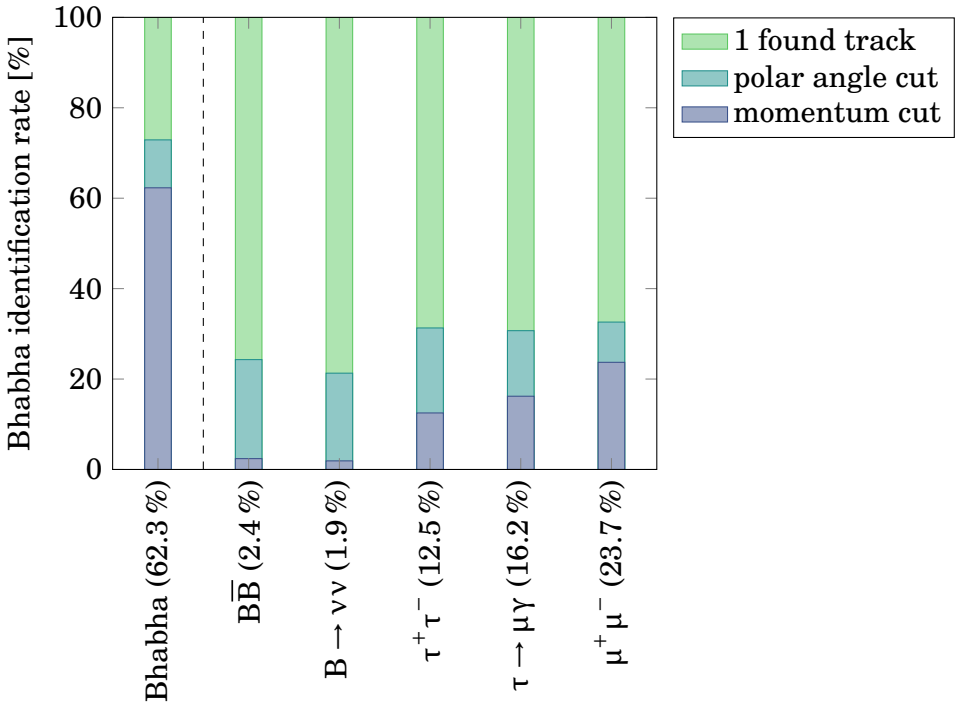

Figure 8.16.: Fraction of events that pass the successive charge dependent cuts for the single track Bhabha identification (normalized to the number of events with exactly one found track). The numbers in brackets give the final (mis)identification rate.

momentum are much more likely for Bhabha events with one track than for Bhabha events with two tracks. This means that one track Bhabha events often include a high energy photon. Figure 8.16 shows the resulting event fractions identified as Bhabha events. As in the two track case, the cuts might be optimized further, but will be replaced by a calorimeter based criterion anyway.

Table 8.4 shows the final efficiency and background rate that can be obtained using only the track trigger information, after the z-vertex veto of 5 cm or 10 cm and the Bhabha veto derived above. The irreducible background rates have been scaled with the Bhabha identification rates obtained in fig. 8.13 and fig. 8.16, while the reducible background rates were taken without modification from table 8.3. Therefore, the background rates given in table 8.4 are only rough estimates for the true background rate that can be expected. With the looser z-vertex cut of 10 cm, the total background rate of 18.2 kHz is higher than the goal of 15 kHz. With the tighter cut, the background rate of 10.7 kHz is well below the goal, so either the Bhabha veto or the z-vertex cut could be relaxed. With the present cuts, the final trigger rate for τ events is $\approx 10\,\%$ lower than for a one track trigger without vetos, but still $\approx 20\,\%$ higher than for a two track trigger without vetos, which just barely meets the goal for the background rate.

Table 8.4.: Efficiency and background rate after the z-vertex veto and the Bhabha veto without calorimeter. The irreducible background rate is extrapolated from table 8.3, using the Bhabha identification rates in fig. 8.13 and fig. 8.16. The total efficiencies and background rates include also the ≥ 3 track rates without additional constraints from table 8.2.

$z_{\text{cut}} = 10\,\text{cm}$:	1 IP track	2 tracks (≥ 1 IP)	total
generic $B\overline{B}$	$(0.91 \pm 0.03)\,\%$	$(3.84 \pm 0.06)\,\%$	$(99.33 \pm 0.03)\,\%$
invisible B	$(14.01 \pm 0.11)\,\%$	$(26.62 \pm 0.14)\,\%$	$(92.46 \pm 0.08)\,\%$
generic $\tau^+\tau^-$	$(21.63 \pm 0.13)\,\%$	$(37.29 \pm 0.15)\,\%$	$(78.06 \pm 0.13)\,\%$
$\tau \to \mu\gamma$	$(22.38 \pm 0.13)\,\%$	$(42.27 \pm 0.16)\,\%$	$(77.18 \pm 0.13)\,\%$
reducible bkg.	5.4 kHz	0.7 kHz	6.1 kHz
irreducible bkg.	9.6 kHz	2.5 kHz	12.1 kHz
$z_{\text{cut}} = 5\,\text{cm}$:	1 IP track	2 tracks (≥ 1 IP)	total
generic $B\overline{B}$	$(0.79 \pm 0.03)\,\%$	$(3.55 \pm 0.06)\,\%$	$(98.92 \pm 0.03)\,\%$
invisible B	$(12.34 \pm 0.10)\,\%$	$(25.30 \pm 0.14)\,\%$	$(89.48 \pm 0.10)\,\%$
generic $\tau^+\tau^-$	$(19.85 \pm 0.13)\,\%$	$(36.67 \pm 0.15)\,\%$	$(75.65 \pm 0.14)\,\%$
$\tau \to \mu\gamma$	$(20.81 \pm 0.13)\,\%$	$(41.87 \pm 0.16)\,\%$	$(75.20 \pm 0.14)\,\%$
reducible bkg.	1.8 kHz	0.5 kHz	2.3 kHz
irreducible bkg.	6.1 kHz	2.3 kHz	8.4 kHz

8.4. Combined track and calorimeter trigger

In the following section, the information of the calorimeter trigger is combined with the track information in order to improve the one track and two track triggers. The consequence of this is the possibility to identify electrons and thus a more efficient Bhabha veto. Standalone calorimeter triggers, which are based on the number and energy of calorimeter clusters, are also possible, but not investigated here.

8.4.1. Two track trigger

The Bhabha veto for a two track trigger already gave reasonable results for a pure track trigger, but can be improved by a proper electron identification. Electrons and positrons are characterized by a charged track with a corresponding calorimeter cluster, whose energy E matches the total momentum p of the track. Therefore, they can be identified with a cut on the ratio E/p, which is approximately 1 for electrons and positrons. However, the particle identification alone is not enough to identify Bhabha events, since electrons and positrons can also be produced in $B\overline{B}$ events and $\tau^+\tau^-$ events. To identify Bhabha events, both the electron and the positron must have

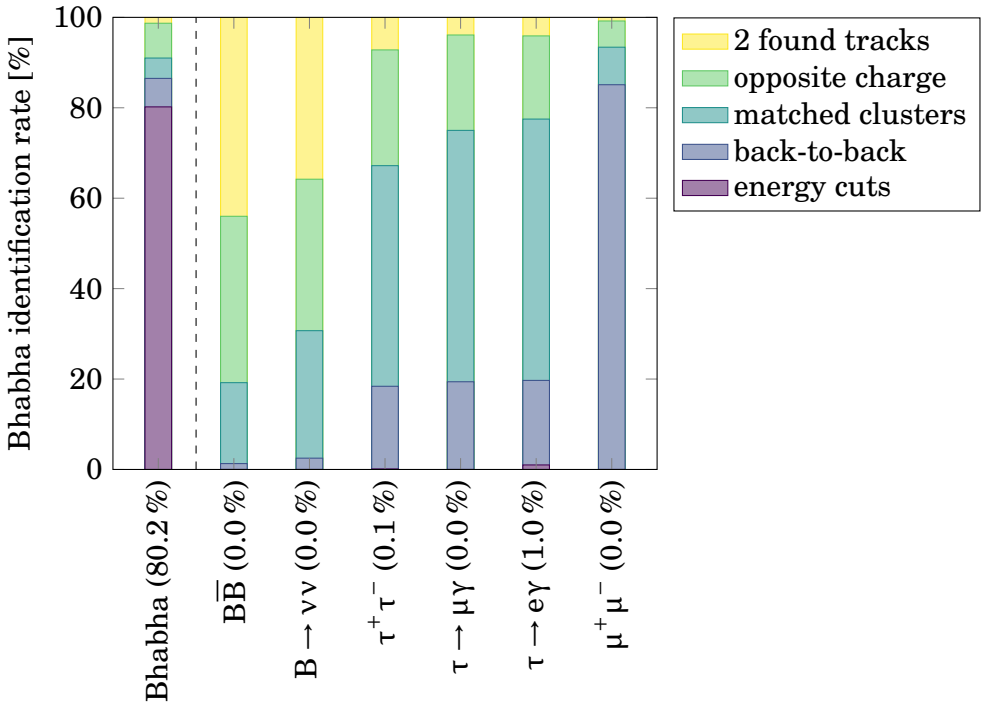

Figure 8.17.: Fraction of events that pass the successive cuts for the Bhabha identification with calorimeter (normalized to the number of events with exactly two found tracks). The numbers in brackets give the final (mis)identification rate.

high energy. In fact, the energy requirement alone is enough, since a matched cluster with high energy implies also a high ratio E/p.

The combined Bhabha veto starts with the same requirements as for the pure track trigger: two 3D tracks of opposite charge that fulfill the back-to-back requirement. In addition, both tracks must be matched to a calorimeter cluster. The momentum cuts are replaced by cuts on the cluster energy. Figure 8.17 shows the fraction of events identified as Bhabha events, using the same energy thresholds that were applied to the total momentum before. An additional lepton flavor violating τ decay is shown, namely $\tau \to e\gamma$, which is similar to $\tau \to \mu\gamma$ but always contains an electron. The misidentification rate for all signal processes is very low. However, the Bhabha identification rate of about 80.2 % is also relatively low.

To increase the Bhabha identification rate, the cuts on the energy, the back-to-back variables or the distance between track and cluster used for the matching can be relaxed (the latter was defined in section 8.1.2). Figure 8.18 shows the cluster energy for Bhabha events and $\tau \to e\gamma$ events, which have the highest misidentification rate. The distribution indicates that the losses are not an issue of too tight cuts, but of individual clusters with very low energy. As shown in fig. 8.19a, these low energy

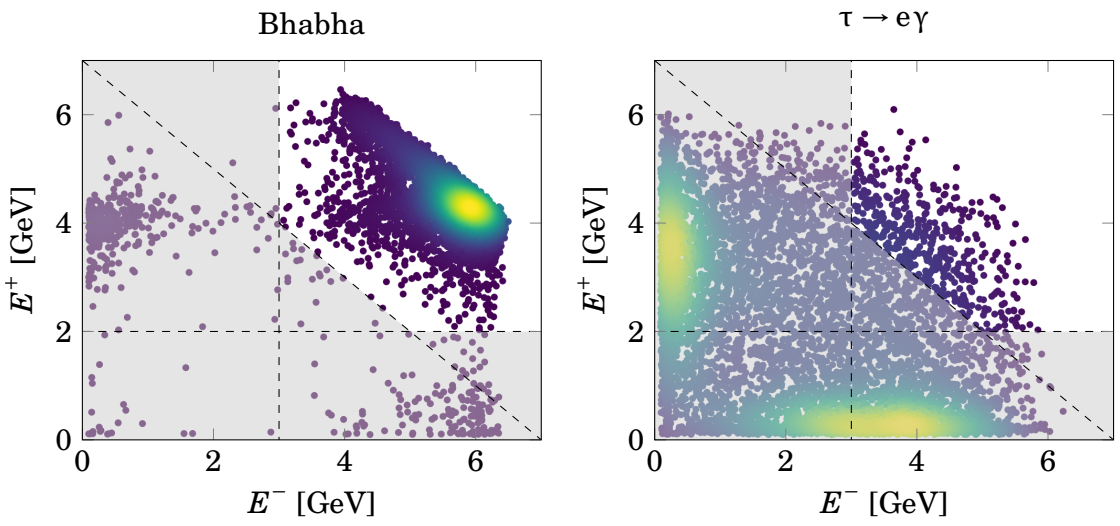

Figure 8.18.: Energy of clusters matched to the positively and negatively charged particles in events that pass the back-to-back cuts. The color shows the point density. The dashed lines show the selected cuts of $E^+ \geq 2\,\text{GeV}$, $E^- \geq 3\,\text{GeV}$ and $E^+ + E^- \geq 7\,\text{GeV}$. The shaded area shows the combination of the cuts.

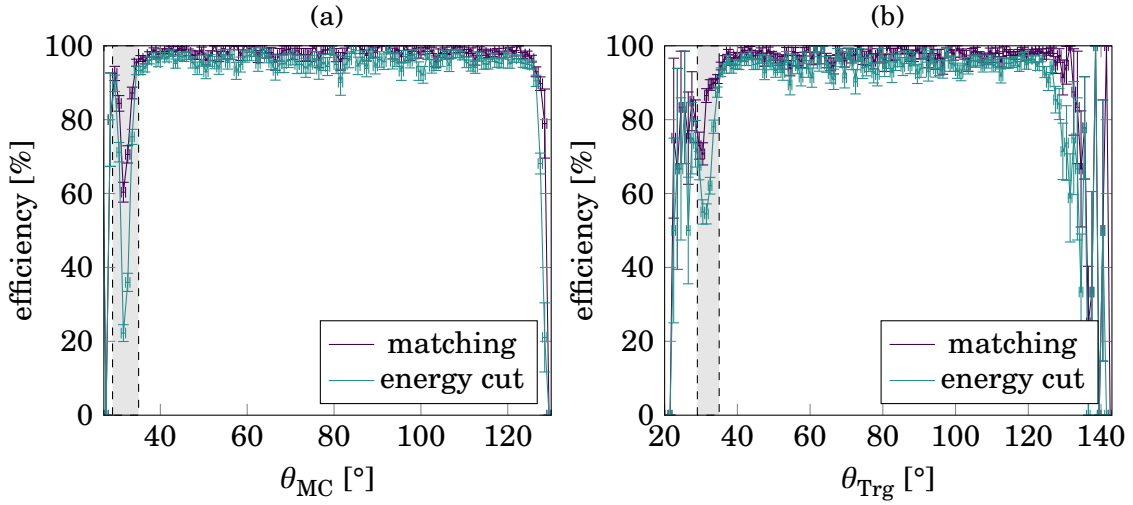

Figure 8.19.: (a): matching efficiency and efficiency of energy cut for tracks in Bhabha events with two oppositely charged 3D tracks, depending on the true polar angle. (b): same efficiencies, depending on the estimated polar angle of the 3D track. For the energy cut efficiency, a charge dependent cut of $E^+ \geq 2\,\text{GeV}$ or $E^- \geq 3\,\text{GeV}$ is applied. The shaded area marks the region $\theta_0 \in [29°, 35°]$.

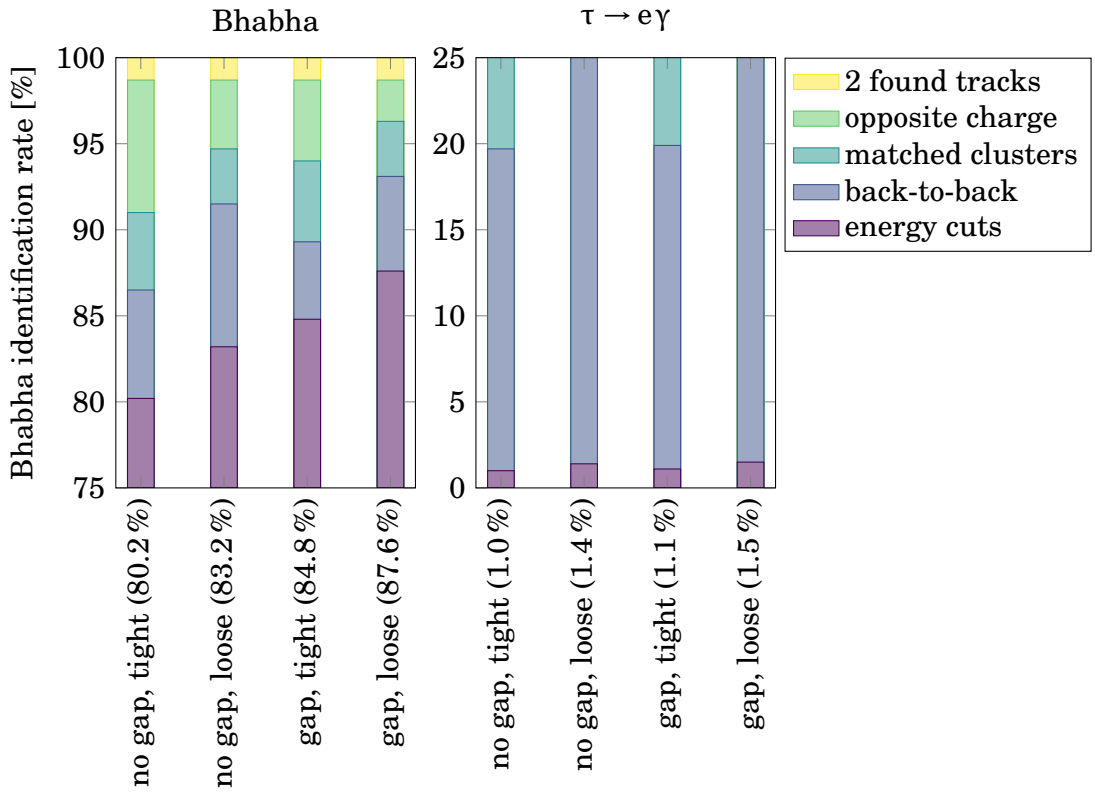

Figure 8.20.: Bhabha identification rate with calorimeter for different cut combinations (normalized to the number of events with exactly two found tracks). The numbers in brackets show the final (mis)identification rates. Tight cuts: matching cuts of $\Delta r \leq 20\,\text{cm}$, $\Delta z \leq 30\,\text{cm}$ and back-to-back cuts of $165° \leq \Delta\varphi_0 \leq 190°$, $130° \leq \sum\theta_0 \leq 175°$. Loose cuts: matching cuts of $\Delta r \leq 30\,\text{cm}$, $\Delta z \leq 50\,\text{cm}$ and back-to-back cuts of $160° \leq \Delta\varphi_0 \leq 195°$, $120° \leq \sum\theta_0 \leq 185°$. For separate gap treatment, the total momentum is used instead of the energy if $\theta_0 \in [29°, 35°]$.

clusters occur mostly around the gap between the barrel and endcap parts of the calorimeter, at a polar angle region of approximately $[29°, 35°]$. This region also shows a large matching inefficiency. To avoid these effects, the region around the gap can be handled separately: for tracks with a polar angle close to the gap, no related cluster is searched and the total momentum is taken instead of the energy. In fig. 8.19a, only the gap on the forward side is visible, which is relevant for electrons, so for positrons no special treatment is applied. This ensures also that the particle identification works even close to the gap, since one high energy cluster is always required. Figure 8.19b shows the matching efficiency and energy cut efficiency depending on the estimated polar angle. The distribution is distorted by the uncertainty of the polar angle, but the gap is still visible.

The cuts on the back-to-back variables and the matching distance can be relaxed in addition (or alternatively) to the separate gap treatment. The resulting (mis)identification rates for Bhabha events and $\tau \to e\gamma$ events are shown in fig. 8.20. The Bhabha identification rate can be increased by 7.4 %, while the misidentification rate for $\tau \to e\gamma$ events is still only 1.5 % (for the other signal events, the misidentification rate is around 0.1 % or smaller). The separate treatment of tracks close to the calorimeter gap makes the trigger condition more complicated, but gives a better trade-off between Bhabha identification and signal misidentification than simply relaxing the cuts.

8.4.2. One track trigger

The single track Bhabha veto without the calorimeter was based on charge dependent cuts on the polar angle and the total momentum. One way to improve it is to replace the total momentum cut by a cut on the energy of a matched cluster. In addition, the calorimeter has a larger acceptance region than the track trigger, so it is possible to detect the lost electron/positron as an additional unmatched ("neutral") cluster. This allows to apply a back-to-back criterion to the opening angle between the track and the cluster, as well as a cut on the energy of the additional cluster. For this kind of Bhabha identification, at least one neutral cluster is required. The number of neutral clusters in events with exactly one found track is shown in fig. 8.21a for Bhabha events, $\tau \to \mu\gamma$ events and invisible B decays.

For the back-to-back condition, the cluster closest to $\varphi_0^{\text{track}} - 180°$ is searched. Note that if the cluster was caused by a charged particle, which is out of the track finding acceptance, the cluster position is shifted due to the track curvature. However, since the curvature is not known, this shift cannot be taken into account. Figure 8.21b shows the transverse distance $\Delta\varphi_0$ between the track and the closest cluster. Both the signal events and the Bhabha events peak at 0°, since by definition the closest cluster is chosen if several are present. For Bhabha events, there is an additional small peak at $\Delta\varphi_0 \approx -180°$, which corresponds to a photon from final state radiation that is emitted collinear to the track. A moderate cut of $\left|\Delta\varphi_0\right| \leq 50°$ is chosen, but mostly the purpose of the back-to-back criterion is to select a cluster on which further cuts can be applied. Note that a cut on $\theta_0^{\text{cluster}} + \theta_0^{\text{track}}$ does not make sense. If the electron and positron were emitted back-to-back in θ_0, either both or none should be in the acceptance region of the track finding. In one track events, the correlation between the polar angles is no longer valid, most often because of a photon from initial state radiation that carries away some of the longitudinal momentum.

Instead, separate cuts are applied to θ_0^{track} and $\theta_0^{\text{cluster}}$, depending on the charge of the track. The polar angle distribution of the track is shown in fig. 8.14 and was used already for the single track Bhabha veto without calorimeter. Since there are now additional variables available for the Bhabha identification, the cuts on θ_0^{track} are relaxed to $\theta_0^{-,\text{track}} \leq 100°$ and $\theta_0^{+,\text{track}} \geq 50°$. Figure 8.22 shows the polar angle

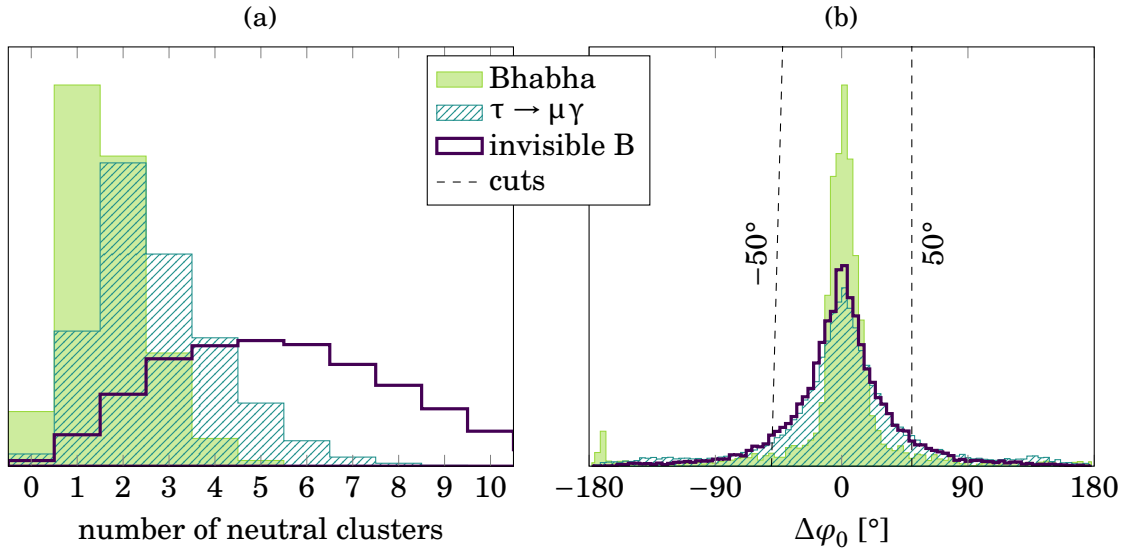

Figure 8.21.: (a): number of neutral (unmatched) clusters in events with one found track. (b): distance $\Delta\varphi_0 = (\varphi_0^{\text{cluster}} - \varphi_0^{\text{track}}) - 180°$ between the track and the most back-to-back neutral cluster (with minimal $\Delta\varphi_0$). The dashed lines show the selected cuts.

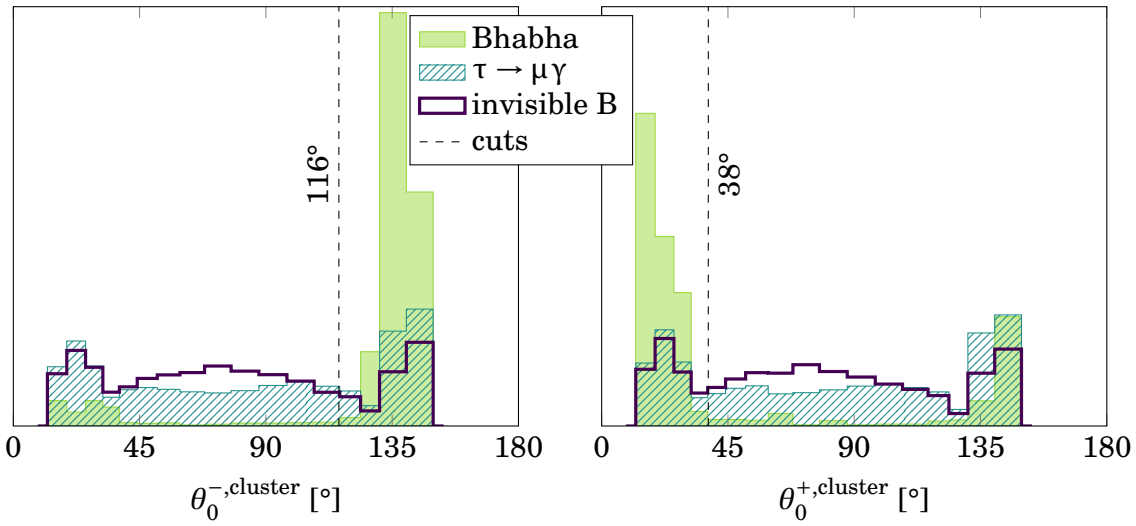

Figure 8.22.: Polar angle distribution of the back-to-back cluster in events with one found track, depending on the charge of the track. The dashed lines show the selected cuts. Note that non-uniform bins are chosen such that each bin corresponds to one ring of trigger cells.

distribution of the back-to-back cluster, also depending on the charge of the track. Again, there is a very strong charge dependence for Bhabha events, and the back-to-back cluster appears mostly in the region outside of the track finding acceptance. In contrast, the back-to-back clusters in the signal events show a more uniform polar angle distribution, since they are often caused by genuinely neutral particles. Cuts of $\theta_0^{-,\text{cluster}} \geq 116°$ and $\theta_0^{+,\text{cluster}} \leq 38°$ are selected, which corresponds to a cut between the fourth and fifth ring of ECL trigger cells (counting from the forward or backward direction).

The last variables that can be used in the Bhabha identification are the energies of the track and the back-to-back cluster. The matching efficiency in single track Bhabha events is relatively low, with 84.5 %, since many of the tracks are within the gap of the calorimeter (compare fig. 8.14 and fig. 8.19). Therefore, a matched cluster is not strictly required. When there is no matched cluster for the track, the total momentum is used instead of the energy. As for the two track Bhabha veto, the total momentum is also used if the polar angle of the track is within the calorimeter gap of $[29°, 35°]$. Figure 8.23 shows the resulting energy distributions for this definition of the track energy. Note that instead of the $\tau \to \mu\gamma$ channel, the $\tau \to e\gamma$ channel is shown again, since the electron causes higher cluster energies. For Bhabha events, the energy of the additional cluster peaks at $\approx 4\,\text{GeV}$ or $\approx 6\,\text{GeV}$, depending on the charge of the track, while the energy distribution of the track is less pronounced, as was already observed for the total momentum (compare fig. 8.15). However, there is also a significant number of events with very low energy in the additional cluster. Therefore, the best identification is obtained by requiring that either the track or the cluster energy are above a certain threshold. Charge dependent thresholds of $E^{-,\text{cluster}} \geq 2\,\text{GeV}$ / $E^{-,\text{track}} \geq 5\,\text{GeV}$ and $E^{+,\text{cluster}} \geq 5\,\text{GeV}$ / $E^{+,\text{track}} \geq 1\,\text{GeV}$ are selected. For invisible B events, the track and cluster energies are typically below 2 GeV, so the misidentification rate after the energy cut is very small. For $\tau \to e\gamma$ events, there is a remaining irreducible overlap. Note that in fig. 8.23 both signal event types show a charge dependent energy distribution. This is a consequence of the charge dependent polar angle cuts, which select negatively charged particles in the forward direction and positively charged particles in the backward direction. Because of the boost, the kinematic limit on the energy is higher for forward particles than for backward particles.

The final (mis)idenfication rates after all cuts are shown in fig. 8.24, normalized to the number of events with one found track. The last decision is how to handle events with one track but no additional clusters. Either such events are accepted by the trigger, since they do not match the Bhabha signature to cause a veto, or they are rejected to suppress more of the background. The latter is equivalent to a “one track, one cluster” trigger condition instead of the simple track multiplicity requirement. In either case, the trade-off between high Bhabha identification and low signal misidentification is improved compared to the Bhabha veto without calorimeter (compare fig. 8.16).

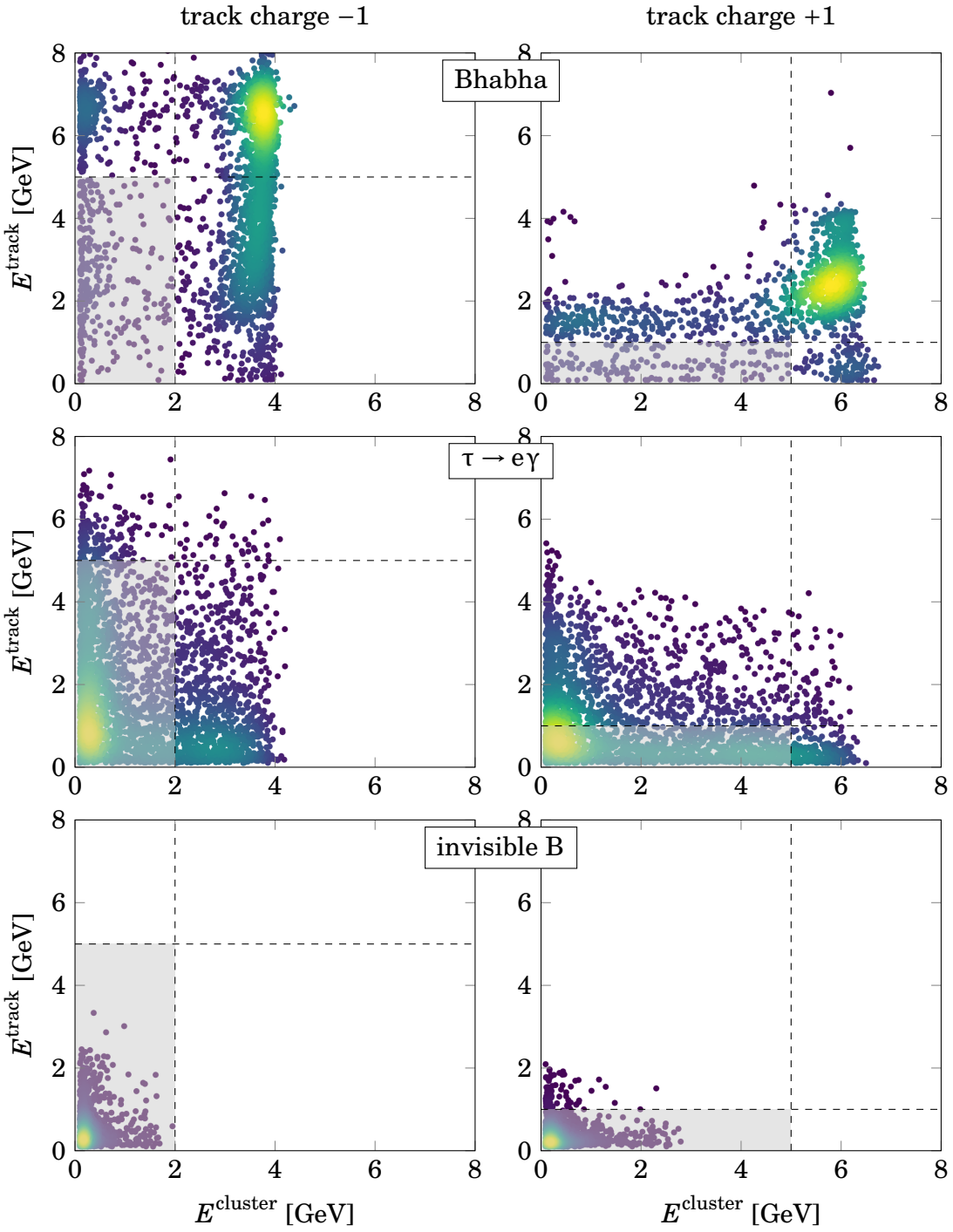

Figure 8.23.: Energy of the track and the back-to-back cluster in single track events that pass the back-to-back and polar angle cuts, depending on the charge of the track. The color shows the point density. The dashed lines show the selected cuts. The shaded area shows the combination of the cuts.

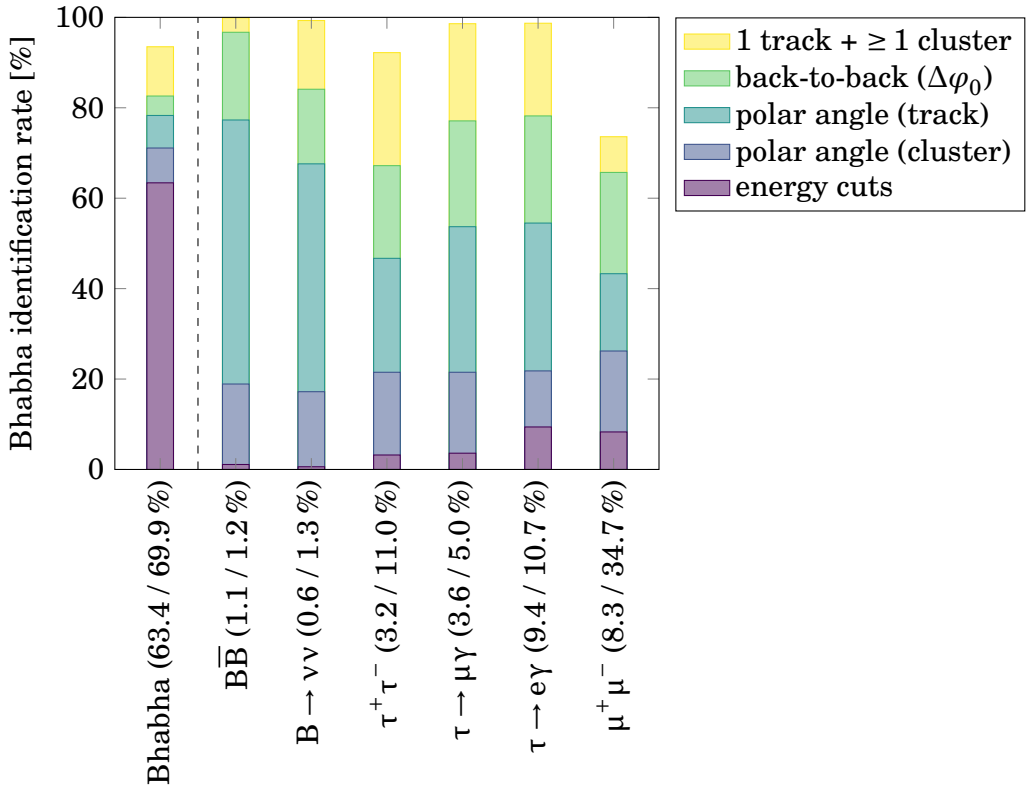

Figure 8.24.: Fraction of events that pass the successive cuts for the single track Bhabha identification with calorimeter (normalized to the number of events with exactly one found track). The numbers in brackets show the final veto rates for the cases where events without neutral clusters are triggered/vetoed.

8.4.3. Summary

Table 8.5 shows the efficiency and background rate that is obtained from the combination of a track multiplicity trigger with all suggested vetos. For single track events, at least one additional neutral cluster is required. The background rates from table 8.3 have again been scaled with the Bhabha identification rates to obtain the total background rate. With this estimate, the background rate for a 10 cm z-vertex cut is now 15.1 kHz, which agrees with the goal set at the beginning of the efficiency studies. The efficiencies for invisible B events and $\tau \to \mu\gamma$ events are only 3.2 % lower than with a pure one track trigger. For generic $\tau^+\tau^-$ events, the efficiencies are 5.7 % lower, due to the requirement of an additional neutral cluster in one track events.

For comparison, the Bhabha veto proposed in [58] reaches a Bhabha identification rate of only 82.3 % for two track events and 44.0 % for one track events, leading to a total background rate of 22.2 kHz for the 10 cm cut or 13.0 kHz for the 5 cm cut. The

Table 8.5.: Efficiency and background rate after the z-vertex veto and the Bhabha veto with calorimeter. The irreducible background rate is extrapolated from table 8.3, using the Bhabha identification rates in fig. 8.20 (loose cuts, separate gap treatment) and fig. 8.24. The total efficiencies and background rates include also the ≥ 3 track rates without additional constraints from table 8.2.

$z_{\text{cut}} = 10\,\text{cm}$:	1 IP track, ≥ 1 clusters	2 tracks (≥ 1 IP)	total
generic $B\overline{B}$	$(0.92 \pm 0.03)\,\%$	$(3.85 \pm 0.06)\,\%$	$(99.34 \pm 0.03)\,\%$
invisible B	$(14.09 \pm 0.11)\,\%$	$(26.62 \pm 0.14)\,\%$	$(92.55 \pm 0.08)\,\%$
generic $\tau^+\tau^-$	$(22.03 \pm 0.13)\,\%$	$(37.83 \pm 0.15)\,\%$	$(79.00 \pm 0.13)\,\%$
$\tau \to \mu\gamma$	$(25.48 \pm 0.14)\,\%$	$(43.82 \pm 0.16)\,\%$	$(81.83 \pm 0.12)\,\%$
reducible bkg.	5.4 kHz	0.7 kHz	6.1 kHz
irreducible bkg.	7.7 kHz	1.3 kHz	9.0 kHz
$z_{\text{cut}} = 5\,\text{cm}$:	1 IP track, ≥ 1 clusters	2 tracks (≥ 1 IP)	total
generic $B\overline{B}$	$(0.80 \pm 0.03)\,\%$	$(3.55 \pm 0.06)\,\%$	$(98.93 \pm 0.03)\,\%$
invisible B	$(12.41 \pm 0.10)\,\%$	$(25.30 \pm 0.14)\,\%$	$(89.55 \pm 0.10)\,\%$
generic $\tau^+\tau^-$	$(20.16 \pm 0.13)\,\%$	$(37.21 \pm 0.15)\,\%$	$(76.50 \pm 0.13)\,\%$
$\tau \to \mu\gamma$	$(23.73 \pm 0.13)\,\%$	$(43.41 \pm 0.16)\,\%$	$(79.66 \pm 0.13)\,\%$
reducible bkg.	1.8 kHz	0.5 kHz	2.3 kHz
irreducible bkg.	4.8 kHz	1.2 kHz	6.0 kHz

main differences are that the Bhabha veto in [58] uses no separate gap treatment and no charge dependency. For the one track trigger, the ECL Bhabha veto is applied, which searches for high energy back-to-back clusters without considering the tracks. In addition, one matched track with a high energy cluster is required. Compared to the Bhabha veto proposed here, the Bhabha misidentification rates for two track events are higher, while those for one track events are lower. Due to the necessary tighter z-vertex cut, the total efficiencies are slightly lower (77.03 % for generic $\tau^+\tau^-$ events and 80.07 % for $\tau \to \mu\gamma$ events).

Of course, the derived cuts for the z-vertex veto and the Bhabha veto are not necessarily final and should be verified both in a more detailed background study and with real data. However, the studies here show that a single track trigger is feasible, unless the background multiplicities deviate strongly from the simulation. The vetos for the background suppression are only possible with the full 3D track reconstruction from the z-vertex trigger.

9. Conclusion

Within this thesis, two major components of the track trigger of the Belle II experiment were optimized and evaluated: the 2D track finding algorithm and the z-vertex trigger, both based on the information from the central drift chamber (CDC). The main challenges are the limited latency and computing resources at the first level trigger, requiring fast and simple algorithms, and the data reduction by the track segment finder, which produces hits in only nine distinct layers. A summary of the developed track trigger algorithms was published in [85].

The task of the track finding algorithm is to separate tracks and to assign track segments from the axial CDC layers to the tracks, using a Hough transformation in the transverse plane of the detector. In the classical definition of the Hough transformation, an ambiguity exists for the charge of the found tracks. This ambiguity was avoided by a suitable choice of the track parametrization, which covers only tracks going outward from the interaction point. This definition also solved the problem of frequent clone tracks for particles with high transverse momentum, which was present in the previous version of the track finding algorithm. After defining the parametrization, the free parameters of the Hough transformation were scanned and optimized with respect to high track finding efficiency, low clone rate and high resolution of the track parameters. For the search of crossing points in the Hough plane, a clustering algorithm is necessary. Building on the previous version of the track finding, a parallel clustering algorithm was defined, which uses only local information to find connected cells in a limited region. The final track finding algorithm is sensitive to transverse momenta above $0.31\,\mathrm{GeV}/c$ and a polar angle region of $[29.6°, 127.1°]$.

The z-vertex trigger works on individual tracks to reconstruct the polar angle and the z-vertex of each track, giving a full set of parameters for a three dimensional track together with the transverse momentum and the azimuth angle, which were determined by the track finding algorithm. Artificial neural networks are trained on single tracks propagated through the full detector simulation of Belle II to estimate the polar angle and the z-vertex as continuous parameters. An input representation for the neural networks was developed, which combines the information from the track finding with the track segment hits from the four stereo CDC layers to 27 input values. The analytical reconstruction procedure is followed as far as possible without making approximations, then all nonlinear corrections are delegated to the neural network. This strategy allows to make optimal use of the knowledge about the track shape without oversimplifying the reconstruction.

For tracks with missing stereo hits in one layer, separate networks were trained,

improving the average z-vertex resolution from (4.06 ± 0.01) cm to (3.44 ± 0.01) cm. A similar specialization was also tested for different regions of the parameter space. For networks specialized to a particular track curvature, no significant improvement was found. Specialized networks for different polar angle regions gave promising results, if a suitable algorithm can be found that selects the correct network for each track. The structure of the networks was also briefly scanned, showing that the performance is relatively insensitive to the size of the hidden layer. Compared to earlier studies, a moderate increase to 127 hidden nodes was chosen, but the final size should be determined by the constraints of the hardware.

After the optimization, the background robustness of the z-vertex trigger was studied. The z-vertex trigger can adapt to changing background conditions by training the networks with data that includes the appropriate amount of background hits, but the performance still decreases significantly with increasing background. After analyzing the causes for these losses, the left/right lookup table of the track segment finder, which gives important input information for the z-vertex trigger, was also retrained with data including background hits. With the improved input, a z-vertex resolution of (2.922 ± 0.008) cm was obtained for the nominal background level, compared to (2.090 ± 0.005) cm without background hits. Finally, the z-vertex trigger algorithm was converted to fixed point calculation, as required by the hardware. The final networks achieve a resolution of (2.910 ± 0.008) cm for the z-vertex and $(3.133 \pm 0.009)°$ for the polar angle.

In the last part of the thesis, the trigger efficiency of $B\overline{B}$ and $\tau^+\tau^-$ events was studied, with the lepton flavor violating decay channel $\tau \to \mu\gamma$ as an explicit example, and compared to simulated background rates, which are studied in detail in [66]. The z-vertex trigger suppresses background tracks from outside of the interaction point sufficiently that a single track trigger is feasible. However, there is an important background from QED events, most importantly from Bhabha scattering. To reduce these uninteresting events and keep the maximal trigger rate even with a single track trigger, the z-vertex veto is combined with suitable veto conditions. Several possible Bhabha vetos with and without the addition of the calorimeter trigger were proposed. Without calorimeter information, an efficiency of 75.20 % to 77.18 % can be obtained for $\tau \to \mu\gamma$ events, with a corresponding background rate of 10.7 kHz to 18.2 kHz, where the given ranges cover different z-vertex cuts. When including the calorimeter, the efficiency for $\tau \to \mu\gamma$ events is increased to 81.83 %, while the background rate is 15.1 kHz. For comparison, without the 3D track reconstruction of the z-vertex trigger, the track trigger would need to demand at least two tracks, combined with a Bhabha veto given by the calorimeter trigger. The best track trigger efficiency that could be expected in this case for $\tau \to \mu\gamma$ events would be 56.7 %, assuming a very pure Bhabha veto. A pure track trigger without calorimeter would have to demand at least three tracks, leading to an efficiency of only 12.53 % for $\tau \to \mu\gamma$ events, which is six times lower than the pure track trigger with 3D track information.

The Belle II experiment is scheduled to observe the first e^+e^- collisions in 2018.

In the commissioning phase of the experiment, the trigger conditions will be tuned. The neural networks can be retrained on real data, to adapt to the real background situation and if necessary learn corrections for misalignment of the detector. The target values for this retraining can be provided by the offline track reconstruction, which is one order of magnitude more precise than the first level trigger. Such a training on reconstructed tracks instead of the true particles was successfully demonstrated within this thesis. The cuts derived for the z-vertex veto and the Bhabha veto may have to be varied, depending on how close the background simulation is to the real experiment and on the background rate from the other subtriggers. In addition to changing the trigger cuts, some parameters of the track trigger itself could be varied. One possibility is to lower the number of hits required for the track finding and the z-vertex trigger in order to increase the acceptance of the track trigger. The ramifications of such a change are discussed in appendix C. Another variation would be the integration of a 3D track finding, as proposed in [65]. In any of these scenarios, the z-vertex trigger will be essential in suppressing background tracks and opening the trigger for low multiplicity physics.

A. Definition of Quality Measures

Two different types of quality measures are used frequently throughout this work. The first are *rates*, such as the efficiency or the clone rate. The second are *resolutions*, which evaluate the deviation of an estimated parameter from the true value. The following sections define these quantities and the uncertainty of each.

A.1. Statistical uncertainty of the rate

An efficiency-like parameter or rate can be defined both track by track or event by event. Tracks or events are sorted into two categories, for example triggered and not triggered. The rate for category 1 is given by the ratio

$$\epsilon = \frac{N_1}{N_1 + N_2}.$$

A rough estimate for the uncertainty is given by counting error $\Delta N_i = \sqrt{N_i}$. The uncertainty of the rate can be calculating by propagating the uncertainties:

$$\begin{aligned}
\Delta\epsilon &= \sqrt{\left(\frac{\partial\epsilon}{\partial N_1}\cdot\Delta N_1\right)^2 + \left(\frac{\partial\epsilon}{\partial N_2}\cdot\Delta N_2\right)^2} \\
&= \sqrt{\left(\frac{N_2}{(N_1+N_2)^2}\right)^2\cdot N_1 + \left(\frac{N_1}{(N_1+N_2)^2}\right)^2\cdot N_2} \\
&= \sqrt{\frac{N_1\cdot N_2}{(N_1+N_2)^3}}.
\end{aligned}$$

A.2. Trimmed standard deviation

A resolution is usually defined as the standard deviation of the differences between estimated and true parameter. For a normal distribution of errors this gives a good estimate for the width of the distribution. However, outliers can strongly influence the standard deviation. A more robust quality measure is the *trimmed standard deviation*. In general, a trimmed estimator is obtained by discarding a given percentage of the lowest and highest values and calculating the estimator on the remaining middle of

the data. An extreme example is the median, which is the maximally trimmed version of the mean.

The trimmed standard deviation is defined as

$$\sigma^{(p)} = \sqrt{\frac{1}{p \cdot N} \sum_{i=(1-p)/2 \cdot N}^{(1+p)/2 \cdot N} (x_i - \overline{x})^2},$$

where the x_i are sorted, p is the fraction of values that remain after removing the tails and $\overline{x}$ is the trimmed mean calculated on the same fraction of values.

A.3. Statistical uncertainty of the standard deviation

The variance of a finite sample is given by

$$\mathrm{Var}(x) = s^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \overline{x})^2.$$

The sample variance is an approximation of the variance σ^2 of the underlying distribution from which the samples were drawn. The variance of the sample variance gives a measure for the spread of the variance of different samples drawn from the same distribution. It is given by [86]

$$\mathrm{Var}(s^2) = \frac{1}{N} \left(\mu_4 - \frac{N-3}{N-1} s^4 \right),$$

where the fourth moment μ_4 is given by

$$\mu_4 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \overline{x})^4.$$

The statistical uncertainty of the sample variance is given by $\sqrt{\mathrm{Var}(s^2)}$. For a large sample size N the standard deviation is $\sigma \approx \sqrt{s^2}$, so the statistical uncertainty of the standard deviation is found by propagating the uncertainty to

$$\Delta\sigma = \frac{\sqrt{\mathrm{Var}(s^2)}}{2\sigma}.$$

The trimmed version of this formula is used as an estimate for the trimmed standard deviation.

B. Firmware Implementation Details

The following descriptions are based on private communication with the experts working on the FPGA implementation. The 2D track finding is implemented by Tzu-An Sheng from the National Taiwan University. The z-vertex trigger is implemented by Steffen Bähr from the Karlsruhe Institute of Technology.

B.1. Clustering of Hough cells

A logical overview over the clustering algorithm is given in section 5.4.

The track finding is performed on four separate boards which cover a square of the parameter space each (90° in φ_0). To avoid edge effects, a small overlap is required. On the left edge one additional column of 2×2 squares is needed to check if a square fulfills the cluster seed condition. On the right edge two additional columns are needed to ensure that for clusters with a seed in the rightmost regular column the full cluster area can be searched. The Hough plane for each board therefore contains 46×34 cells or 23×17 squares.

The clustering procedure is performed in parallel for all squares. First the cluster seed condition is checked for all squares except the three overlap columns: a seed needs to be active (≡ contain at least one peak candidate) and be disconnected from its left and bottom neighbors. Then a binary pattern of connected squares is built within the clustering rectangle. The cluster seed is the bottom left corner; the top right corner square is found according to the hierarchy defined in fig. 5.20 as the first square that is active. The corner cells within each corner square are determined similarly from the binary pattern within the square. A lookup table is used to derive the center cell from the corners. The result of the algorithm is a matrix of possible cluster centers with the center of each found track turned on.

For a larger clustering rectangle the complexity of building the cluster of connected squares increases, as well as the number of possible top right corners and the size of the lookup table for finding the cluster center. Note also that in contrast to the simulation, the clustering is performed for all squares, whether they contain peak candidates or not. If the cluster seed is not active, the clustering logic does not return a cluster center, but it still requires resources on the FPGA. The maximum cluster area that is feasible to implement in hardware has to be determined experimentally by the Taiwan group.

B.2. Latency of a neural network

The Multi Layer Perceptron (MLP) is introduced in section 7.1.1. The setup in the z-vertex trigger consists of an three layers. The values in the hidden layer and the output layer are calculated according to eq. (7.2):

$$y = \tanh\left(\sum_{i=0}^{N} \frac{x_i \cdot w_i}{2}\right).$$

The multiplications are performed in the FPGA by dedicated calculation units. The standard trigger board UT3 (Universal Trigger Board 3) has ≈ 800 of these multipliers. However, a network with 27 input nodes and 127 hidden nodes (the selected setup in section 7.4) requires 3429 multiplications for a fully parallelized calculation of the hidden layer. For a pipelined operation, the calculation of the output layer for one track has to be performed in parallel to the calculation of the hidden layer for the next track, so the total number of multiplications is even higher. Since this exceeds the resources of the board, the calculation has to be split into several packets.

The minimum number of packets for a given number of hidden nodes can be estimated from the available number of multipliers. A network with 27 input nodes, N hidden nodes and 2 output nodes requires $(27 \cdot N + N \cdot 2)$ multiplications. These multiplications are split into x packets with $(\frac{27}{x} \cdot N + \frac{N}{x} \cdot 2) \leq 800$ each, where the fractions have to be rounded up. So considering only the number of available multipliers, the maximal number of hidden nodes that can be calculated with a given number of packets are

packets x	1	2	3	4	5
hidden nodes N	27	53	82	106	125

The summation for each packet can be executed directly after the multiplication in the same clock, as long as the number of summands n is not too high. The summation happens in a binary tree, so basically the complexity is given by $\log_2(n)$, although the bitwidth of intermediate results also has some influence. Two of these trees have to fit on the board together, one for the hidden layer and one for the output layer. If the summation does not fit into one clock cycle, one can either increase the number of packets or reduce the clock frequency.

So although some general considerations about the size of the network are possible, the actual setup always has to be tested to make sure that it fits into the FPGA with the expected latency. In addition, the optimal numbers may change with a future upgrade of the trigger board, the UT4. The chosen setup in section 7.4 is therefore not necessarily final.

B.3. Neural network activation function

To approximate nonlinear functions, a neural network requires a nonlinear activation function. For the z-vertex trigger, the activation function is the hyperbolic tangent $\tanh\left(\frac{x}{2}\right)$. In the FPGA, it is implemented as a lookup table with N_{LUT} bins of size $2^{-b_{\mathrm{LUT}}}$ and an output precision of b_n (the notation follows the definition in section 7.6). Since the tanh() function is symmetric, the lookup table is defined only for positive values. The output is given by $\tanh\left(\frac{x}{2}\right) \approx \mathrm{LUT}(|x|) \cdot \mathrm{sign}(x)$.

The range of the lookup table is given by the condition

$$\tanh\left(\frac{x_{\max}}{2}\right) \overset{!}{>} 1 - \frac{\Delta y}{2}$$
$$x_{\max} = 2 \cdot \operatorname{arctanh}\left(1 - \frac{\Delta y}{2}\right),$$

where $\Delta y = 2^{-b_n}$ is the difference between two output values. In other words, for $x > x_{\max}$ the output of the lookup table is always 1, so there is no need for storing it in the table. The number of bins in the lookup table is therefore given by

$$N_{\mathrm{LUT}} = \mathrm{ceil}\left(2 \cdot \operatorname{arctanh}\left(1 - 2^{-(b_n+1)}\right) \cdot 2^{b_{\mathrm{LUT}}}\right).$$

The argument x of the activation function is rounded down before the lookup, that is the input precision of the lookup table is less than the bitwidth of x. To keep the rounding errors symmetric, the output of the lookup table is defined as

$$y_i = \mathrm{round}\left(\tanh\left(\frac{1}{2} \cdot \left(x_i + \frac{1 - 2^{-\Delta b}}{2^{b_{\mathrm{LUT}}+1}}\right)\right) \cdot 2^{b_n}\right),$$

where x_i is the lower edge of a lookup table bin, y_i is the output of that bin and Δb is the number of bits that are cut away from x when rounding the input. The purpose of the offset is to shift x_i to the average of all values that are covered by the given bin, as illustrated in fig. B.1.

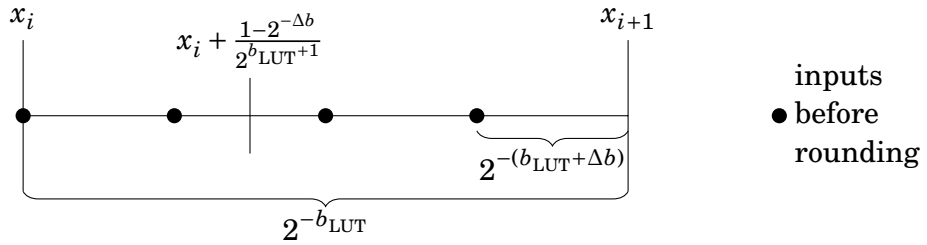

Figure B.1.: Illustration of the lookup table bin definition for $\Delta b = 2$.

C. Track Trigger Setup for Short Tracks

The following sections discuss an alternative of the trigger setup derived in chapters 5 and 7, which was optimized for the best possible z-vertex resolution. The studies in chapter 8 give an estimate for the efficiency that can be achieved with this setup. However, the final trigger settings will be tuned during the initial phase of the actual experiment and adapted to changing background and luminosity. One possibility to increase the acceptance of the track finding is a short track finder, which can find also tracks with only three axial hits. To provide a 3D reconstruction for those short tracks, the z-vertex trigger is also changed to handle tracks with only two stereo hits.

C.1. Short track finding

Especially for $\tau^+\tau^-$ events, many charged particles are not found simply because they are out of the acceptance region of the track finder. This acceptance region can be increased by lowering the peak threshold to 3 (compare fig. 5.11), which means that only three axial hits are required to find a track. However, this leads to two problems: first, both the background rate and the clone rate increase. Second, the cluster size increases, which leads to an increased probability for overflows from the clustering rectangle and thus to a larger bias in the track parameters. Despite these problems, the total efficiency may increase, if the background rate can be kept low enough.

Alternatively to simply lowering the peak threshold, a special short track condition can be introduced, which is fulfilled for tracks with hits in the first three axial superlayers. In the Hough plane, this is realized by defining Hough cells with lines from the first three axial superlayers as peak candidates. For the clustering, they are treated the same as peak candidates with at least four lines.

Table C.1 shows the resolution and bias of the track parameters as well as the acceptance and clone rate for the different track finding options. Both the lower peak threshold and the short track condition increase the polar angle acceptance by about 6° in both directions. In $p_\perp$, the efficiency is above 50 % for the full nominal acceptance of $p_\perp \geq 0.3\,\mathrm{GeV}/c$. Comparing the lower peak threshold and the short track condition, the lower peak threshold shows a slightly higher efficiency, but also a significantly higher clone rate and bias. Therefore, the short track condition is used in the following studies.

Table C.1.: Comparison of the standard track finding (4 hits) with two alternatives for short track finding (3 hits or additional short track condition requiring hits in the 3 innermost axial superlayers, see text).

peak condition	4 hits	3 hits	short tracks
threshold for efficiency $\geq 50\,\%$			
$p_\perp$ / GeV/c	0.310 ± 0.001	0.3	0.3
θ / °	$[29.6 \pm 0.2,$ $127.1 \pm 0.2]$	$[23.7 \pm 0.2,$ $133.7 \pm 0.2]$	$[23.7 \pm 0.2,$ $133.7 \pm 0.2]$
efficiency inside of 50 % threshold			
	$(96.02 \pm 0.07)\,\%$	$(98.17 \pm 0.05)\,\%$	$(97.21 \pm 0.06)\,\%$
Bhabha clone rate	$(0.16 \pm 0.03)\,\%$	$(2.98 \pm 0.08)\,\%$	$(1.55 \pm 0.06)\,\%$
resolution			
$\Delta\varphi_0$ / °	0.818 ± 0.003	1.496 ± 0.012	0.949 ± 0.004
$\Delta p_\perp^{-1}$ / (GeV/c)$^{-1}$	0.0790 ± 0.0003	0.1669 ± 0.0023	0.1429 ± 0.0005
bias			
$\Delta\varphi_0$ / °	0.009 ± 0.003	-0.902 ± 0.005	-0.118 ± 0.003
$\Delta p_\perp^{-1}$ / (GeV/c)$^{-1}$	-0.0014 ± 0.0003	-0.1134 ± 0.0006	-0.0365 ± 0.0005

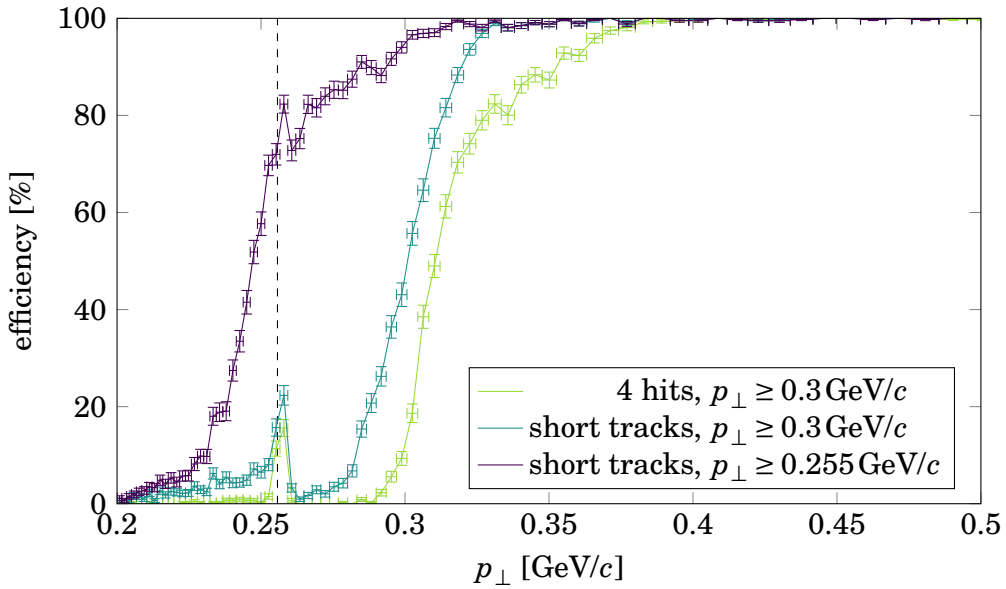

Figure C.1.: Track finding efficiency measured on single muons for the standard track finding (4 hits), with an additional short track condition and with an enlarged Hough plane starting from $p_\perp = 0.255\,\mathrm{GeV}/c$. The dashed line shows the minimum transverse momentum required to leave the CDC.

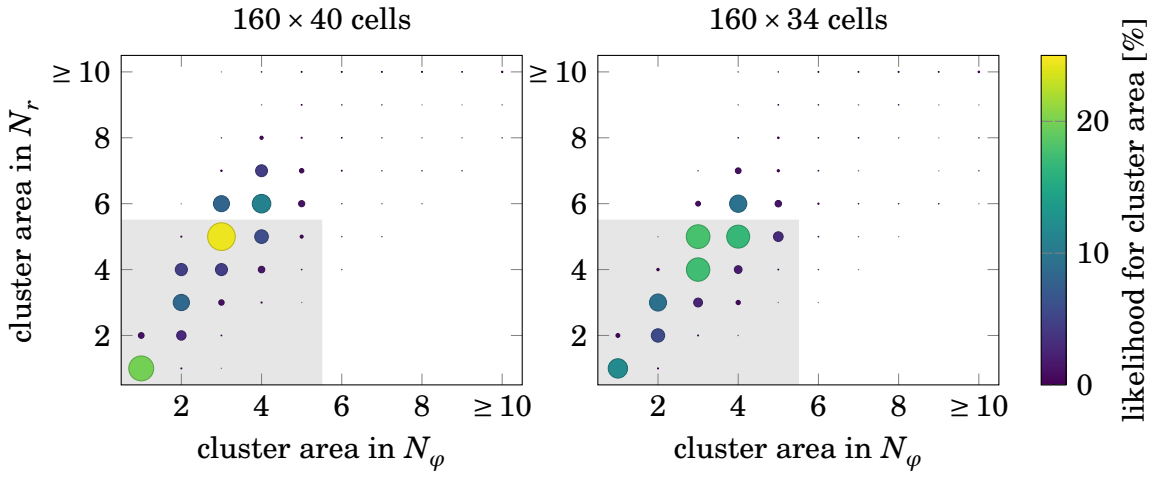

Figure C.2.: Percentages of different cluster areas with short track finding in a Hough grid covering a region of $p_\perp \geq 0.255\,\mathrm{GeV}/c$, measured on single muons without background hits. Very large clusters correspond to curling tracks. The gray area marks the clusters that are guaranteed to be covered by a clustering area of 3×3 squares.

In addition to the short track condition, the Hough plane could be enlarged to lower $p_\perp$ in order to further increase the acceptance. For example, with 160×40 cells of the same size as the standard Hough plane, a $p_\perp$ region down to $0.255\,\mathrm{GeV}/c$ can be covered. This border coincides with the minimum transverse momentum required to leave the CDC in radial direction. Tracks with $p_\perp < 0.255\,\mathrm{GeV}/c$ can curl in the CDC (depending on the polar angle). Figure C.1 shows the track finding efficiency for short track finding with the standard Hough plane and an enlarged Hough plane. Enlarging the Hough plane leads to a significant increase in the $p_\perp$ acceptance.

However, increasing the number of cells in the Hough plane is nontrivial on the hardware and takes more resources. Another possibility is to keep the number of Hough cells unchanged when enlarging the Hough plane limits, leading to a larger cell size $\Delta\frac{1}{r}$. This affects the cluster shape, but since the cluster size is already changed by the short track condition, it is not obvious if the cell size of the standard track finding is still optimal. As shown in fig. C.2, there are less cluster overflows with only 160×34 Hough cells. Also the clone rate and the bias are lower, as shown in table C.2, while the resolution is not affected by the change in the cell size.

Table C.3 shows the trigger efficiencies for a pure multiplicity trigger with short track finding, both with and without enlarging the Hough plane. For $\tau^+\tau^-$ events, the efficiencies increase significantly (about 6 % for a one track trigger) compared to the standard track finding (compare table 8.2). Enlarging the Hough plane gains only a few percent, as most lost tracks in $\tau^+\tau^-$ events are due to too flat polar angles rather than too low transverse momentum. For invisible B events, the angular distribution

Table C.2.: Clone rate, resolution and bias for short track finding in a Hough grid covering a region of $p_\perp \geq 0.255\,\mathrm{GeV}/c$. The resolution is measured on tracks with $p_\perp \geq 0.3\,\mathrm{GeV}/c$, to avoid curling tracks and to keep the results comparable with table C.1.

Hough cells	160×40	160×34
Bhabha clone rate	$(1.65 \pm 0.06)\,\%$	$(1.18 \pm 0.05)\,\%$
resolution		
$\Delta\varphi_0$ / °	0.937 ± 0.004	0.939 ± 0.004
$\Delta p_\perp^{-1}$ / $(\mathrm{GeV}/c)^{-1}$	0.1426 ± 0.0005	0.1440 ± 0.0006
bias		
$\Delta\varphi_0$ / °	-0.120 ± 0.003	-0.060 ± 0.003
$\Delta p_\perp^{-1}$ / $(\mathrm{GeV}/c)^{-1}$	-0.0400 ± 0.0005	-0.0258 ± 0.0005

is more uniform, so there is a larger gain when enlarging the Hough plane. However, for a one track trigger the efficiency is already very high with the standard Hough plane limits, so the gain is again negligible.

An estimate for the background rate with short track finding is not yet available. From the momentum distributions in [84], it is expected that the reducible background increases strongly for lower transverse momenta, while the angular distribution is relatively uniform. On the other hand, the Bhabha cross section increases for flat polar angles. Comparing table C.3 and table 8.5, it is clear that the short track finding only gives an advantage if a single track trigger is still possible, in other words if the increasing background rate can be sufficiently suppressed.

C.2. z-vertex trigger for short tracks

To suppress the additional background expected for the short track finding, the z-vertex trigger also needs to be adapted to handle short tracks. First, the neural networks need to be retrained with tracks found by the short track finding. Second, the z-vertex trigger should handle at least some tracks with only two stereo wires. Figure C.3 shows the true z-vertex distribution for tracks found with the short track finding, depending on the combination of stereo hits associated with the track. Combinations with more than two stereo hits are not shown separately, since they were already analyzed in chapter 7 (see fig. 7.13). For tracks with two stereo hits, the most frequent combination are hits in the two innermost stereo superlayers, as expected when the short track finding requires also hits from the innermost superlayers. Another frequent combination are hits in the two outermost stereo superlayers, but this case occurs only for tracks far away from the interaction point. All other combinations are very rare (less than 1 % for tracks from the interaction point). Therefore, only

Table C.3.: Efficiency with short track finding, depending on the required track multiplicity.

	≥ 1 tracks	≥ 2 tracks	≥ 3 tracks
standard Hough plane limits ($p_\perp \geq 0.3\,\mathrm{GeV}/c$), 160×34 cells			
generic $B\overline{B}$	$(99.96 \pm 0.01)\,\%$	$(99.60 \pm 0.02)\,\%$	$(97.94 \pm 0.04)\,\%$
invisible B	$(97.80 \pm 0.05)\,\%$	$(86.86 \pm 0.11)\,\%$	$(64.48 \pm 0.15)\,\%$
generic $\tau^+\tau^-$	$(90.70 \pm 0.09)\,\%$	$(66.64 \pm 0.15)\,\%$	$(24.07 \pm 0.14)\,\%$
$\tau \to \mu\gamma$	$(91.09 \pm 0.09)\,\%$	$(66.12 \pm 0.15)\,\%$	$(16.53 \pm 0.12)\,\%$
enlarged Hough plane limits ($p_\perp \geq 0.255\,\mathrm{GeV}/c$), 160×34 cells			
generic $B\overline{B}$	$(99.986 \pm 0.004)\,\%$	$(99.78 \pm 0.01)\,\%$	$(98.85 \pm 0.03)\,\%$
invisible B	$(98.58 \pm 0.04)\,\%$	$(90.47 \pm 0.09)\,\%$	$(72.02 \pm 0.14)\,\%$
generic $\tau^+\tau^-$	$(91.75 \pm 0.09)\,\%$	$(69.14 \pm 0.15)\,\%$	$(25.59 \pm 0.14)\,\%$
$\tau \to \mu\gamma$	$(91.79 \pm 0.09)\,\%$	$(68.15 \pm 0.15)\,\%$	$(17.49 \pm 0.12)\,\%$

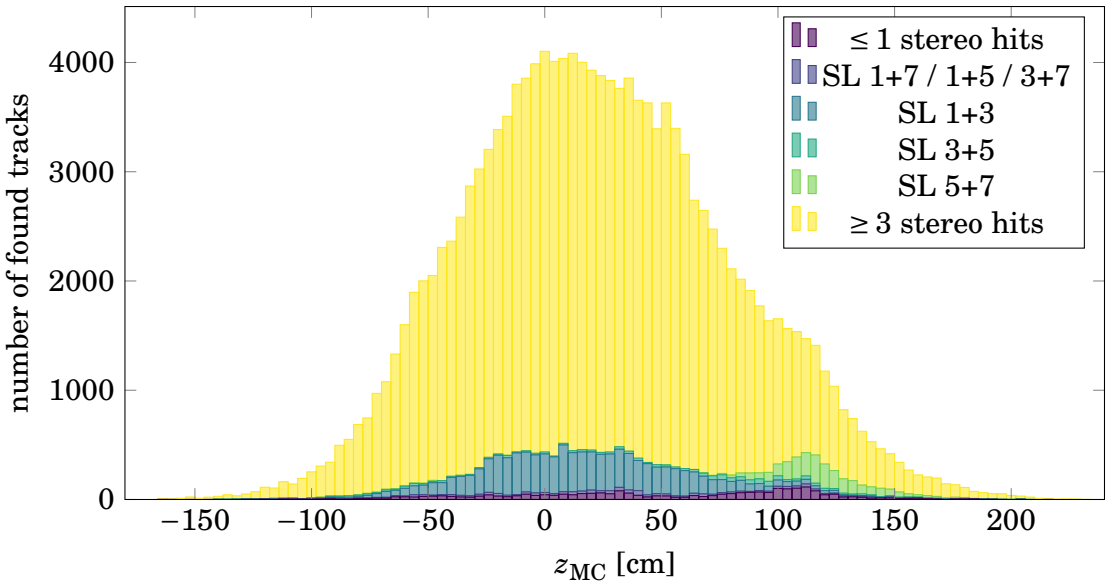

Figure C.3.: Number of selected stereo hits for a track, depending on the true z-vertex. For some combinations, the distributions are summed: for more than two stereo hits (standard z-vertex trigger), for less than two stereo hits (untrackable) and for two stereo hits that are not in neighboring superlayers (very rare).

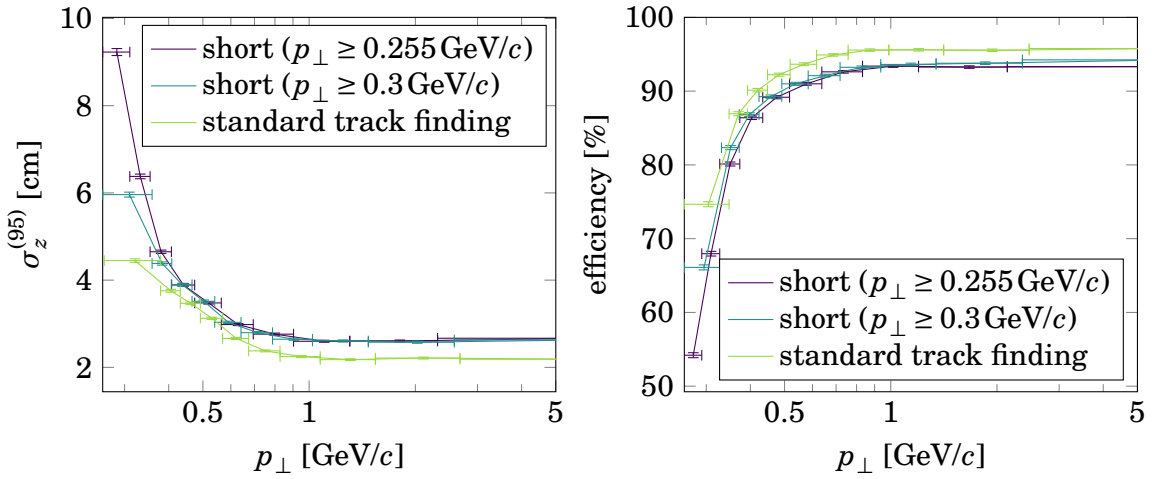

Figure C.4.: Resolution and efficiency with a z-vertex cut of 6 cm, depending on the transverse momentum and on the track finding setup. The efficiency is normalized to the number of tracks found with the respective setup.

one additional expert network is added, specialized to tracks with hits in the two innermost stereo superlayers, leading to a total of six networks.

Figure C.4 shows the resolution and efficiency for the different track finding setups, tested on single muons with $p_\perp \geq 0.26\,\mathrm{GeV}/c$. For the two short track finding setups, six expert networks are trained, including the case of two stereo hits in superlayers 1 and 3. For the standard track finding, there are only five expert networks for three or more stereo hits. As expected, the performance with short track finding is worse than with the standard track finding over the full $p_\perp$ range. On the one hand, the resolution of the 2D track parameters is worse than for the standard track finding, on the other hand, tracks with two stereo hits contribute to the total resolution. For very low $p_\perp$, the errors increase sharply, continuing the tendency observed already with the standard track finding. For transverse momenta below 0.3 GeV/c a z-vertex resolution of (9.23 ± 0.08) cm is obtained. Note that the difference between the two short track finding setups is caused by the different acceptance: with Hough plane limits of $p_\perp = 0.255\,\mathrm{GeV}/c$, the test sample contains more low $p_\perp$ tracks than with Hough plane limits of $p_\perp = 0.3\,\mathrm{GeV}/c$.

Figure C.5a shows the efficiency per track after a z-vertex veto for $\tau \to \mu\gamma$ events with one or two tracks found by the short track finding with standard Hough plane limits. Compared to the standard track trigger, the efficiency is 1 % to 3 % lower, assuming that the same z-vertex cut is chosen. It is likely that a tighter cut will be required to cope with the increasing background rate, so the actual losses will probably be higher. However, the total efficiency per event increases, as shown in fig. C.5b. The reason is the higher track multiplicity due to the increased track finding acceptance. Note, however, that the background multiplicity could also increase. If that is the

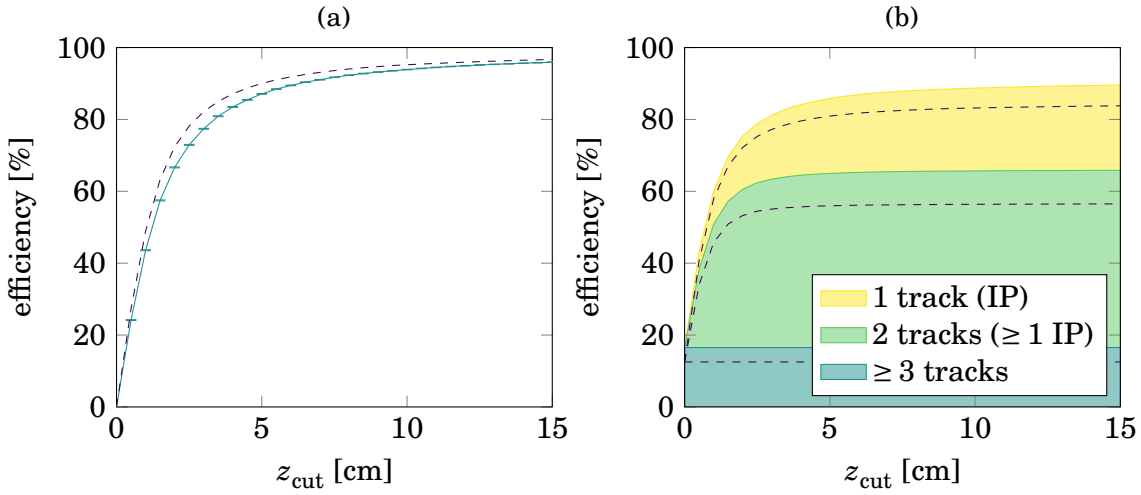

Figure C.5.: Efficiency for $\tau \to \mu\gamma$ events with the short track trigger, depending on the z-vertex cut. The dashed lines show the corresponding efficiencies for the standard track trigger. (a): efficiency per track for events with one or two found tracks. (b): total efficiency per event, with the contributions from events with one, two and more tracks shown separately.

case, a z-vertex veto for three track events might be required, or it could be necessary to demand both tracks in two track events within the z-vertex cut. Clearly, a full background simulation with the short track trigger is needed for a final evaluation.

C.3. Bhabha identification for short tracks

In principle, the Bhabha identification developed in section 8.4 should work also with the short track finding setup. However, due to the increase of the polar angle acceptance, the backward calorimeter gap is now within the track finding acceptance. In fact, covering the insensitive region of the calorimeter in the track trigger is a major advantage of the short track finding, as it leads to a more uniform efficiency of the full trigger system, but it also means that the backward gap has to be taken into account in the Bhabha identification. Figure C.6 shows the matching efficiency and the efficiency of the energy cut in the Bhabha identification for tracks obtained with the short track finding (with Hough plane limits of $p_\perp \geq 0.3\,\text{GeV}/c$). To account for the low cluster efficiency in the gap region of $[124°, 133°]$, the total momentum of positively charged tracks is taken instead of the energy and no matched cluster is required, in analogy to the separate gap treatment of negatively charged tracks in the forward gap region.

Figure C.7 shows the two track Bhabha (mis)identification rates for short track

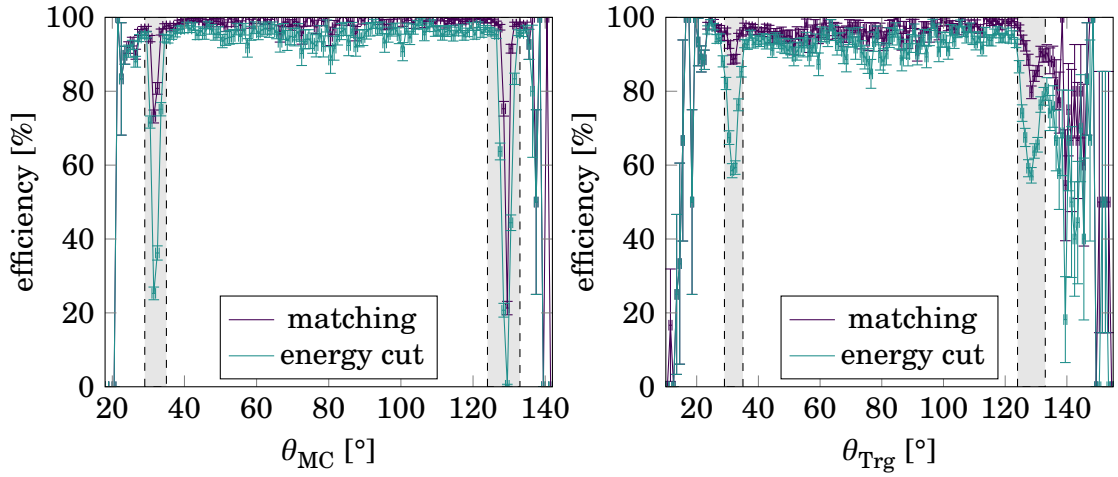

Figure C.6.: Matching efficiency and efficiency of energy cut for tracks in Bhabha events with two oppositely charged 3D tracks obtained with short track finding, depending on the true polar angle (left) and the estimated polar angle of the 3D track (right). For the energy cut efficiency, a charge dependent cut of $E^+ \geq 2\,\mathrm{GeV}$ or $E^- \geq 3\,\mathrm{GeV}$ is applied. The shaded areas mark the regions $\theta_0 \in [29°, 35°]$ and $\theta_0 \in [124°, 133°]$.

finding, using the loose matching and back-to-back cuts defined in fig. 8.20 and separate gap treatment for both the forward and the backward gap. Compared to the standard track finding, the Bhabha identification rate of 81.2 % is rather low. The main reason for this is the reduced efficiency of the matching and the energy cut. Even excluding the gap regions, the reduced polar angle resolution of the short z-vertex trigger makes the matching more difficult, especially in the extreme forward and backward regions. Note also that the total momentum is less precise than the energy of a matched cluster. In addition, 3.3 % of the Bhabha events are not identified because the tracks have the same charge. These events are mainly due to the increased clone rate (compare table C.1), that is two tracks are found for one particle while the other is out of the acceptance. The misidentification rates are still very low. The largest misidentification rate of 2.5 % is found for $\mu^+\mu^-$ events. In the misidentified events, both tracks are in the calorimeter gap, so the calorimeter is not available for particle identification.

For single track events, the cuts selected in section 8.4 are also mostly kept, except for the addition of the backward gap and the cut on the polar angle of the neutral cluster. For Bhabha events, the neutral cluster is mostly caused by a charged track outside of the track finding acceptance, so the cut can be tightened to $\theta_0^{-,\mathrm{cluster}} \geq 124°$ and $\theta_0^{+,\mathrm{cluster}} \leq 32°$, which corresponds to a cut between the third and fourth ring of ECL trigger cells. The polar angle distribution and the new cuts are shown in fig. C.8. The Bhabha (mis)identification rates with this selection are shown in fig. C.9. Compared

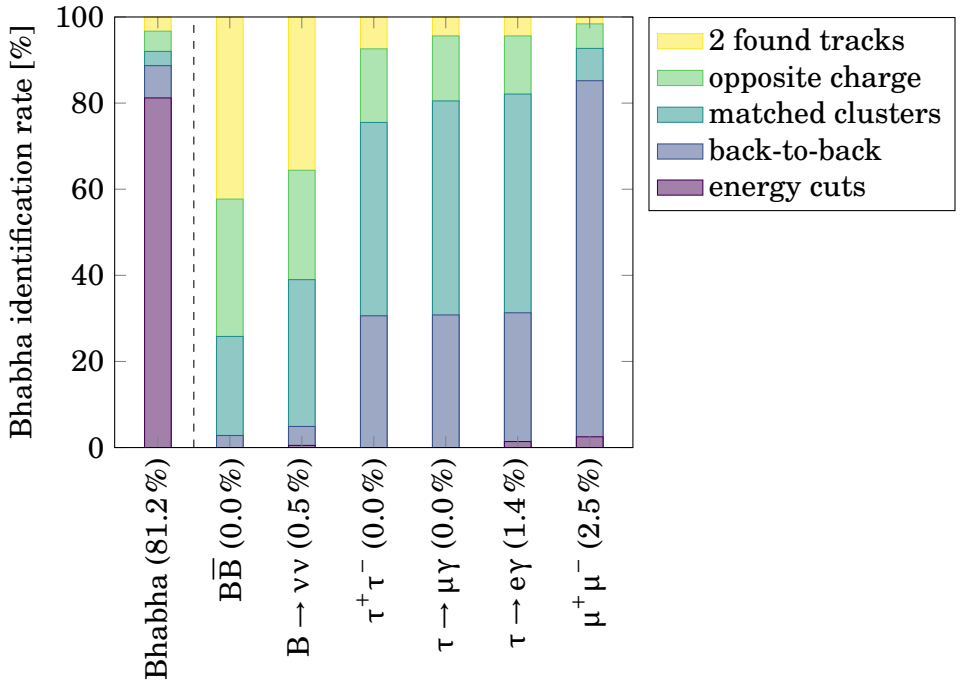

Figure C.7.: Fraction of events that pass the successive cuts for the Bhabha identification with calorimeter (normalized to the number of events with exactly two tracks found with short track finding). The numbers in brackets give the final (mis)identification rate.

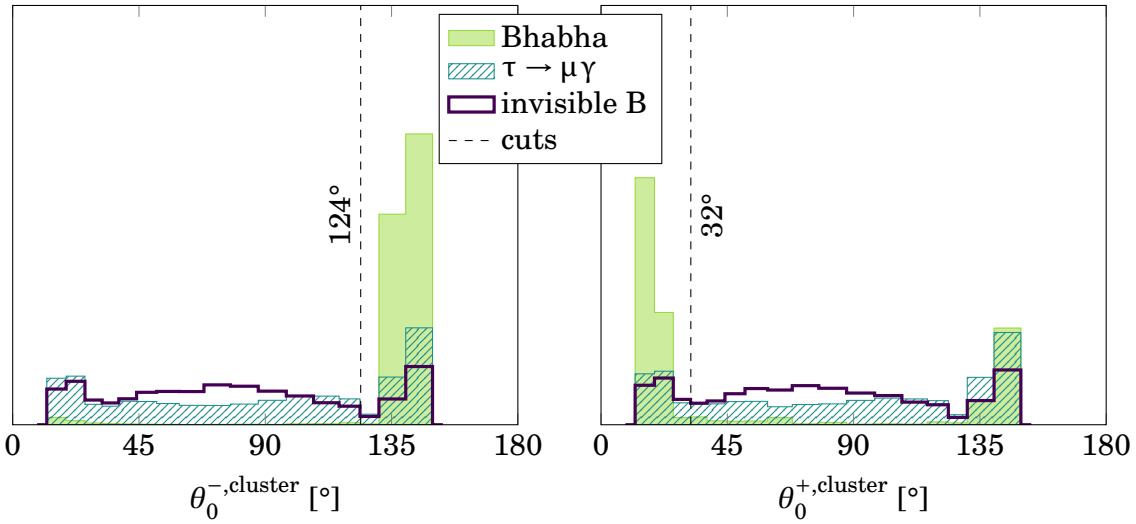

Figure C.8.: Polar angle distribution of the back-to-back cluster in events with one track found with short track finding, depending on the charge of the track. The dashed lines show the selected cuts. Note that non-uniform bins are chosen such that each bin corresponds to one ring of trigger cells.

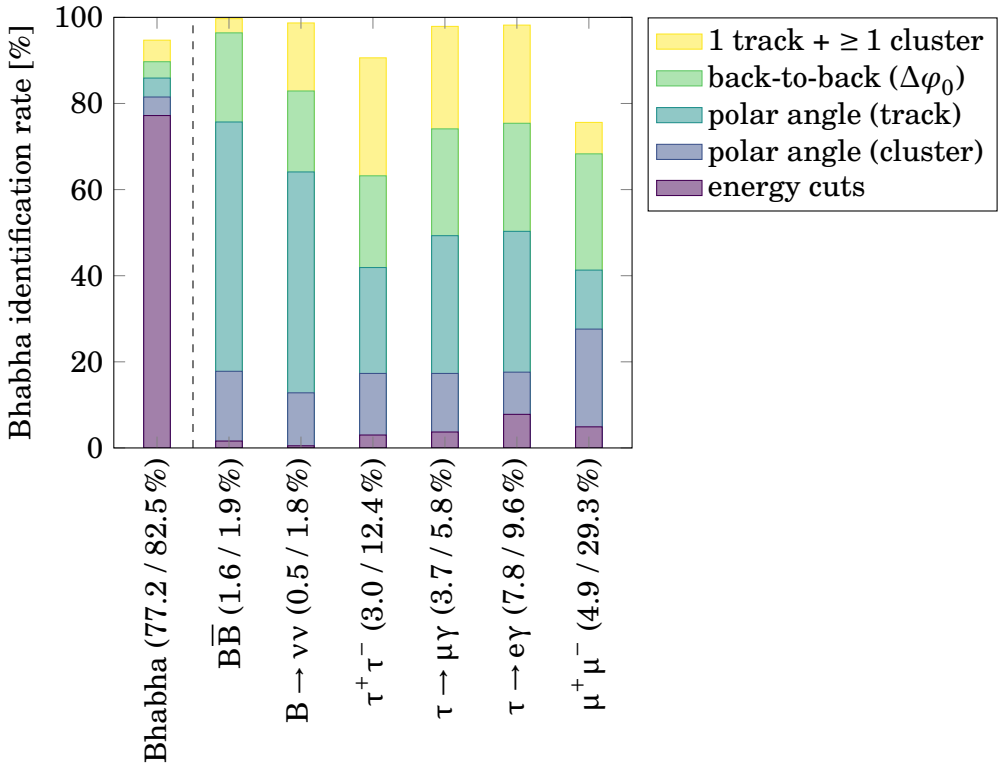

Figure C.9.: Fraction of events that pass the successive cuts for the single track Bhabha identification with calorimeter (normalized to the number of events with exactly one track found with short track finding). The numbers in brackets show the final veto rates for the cases where events without neutral clusters are triggered/vetoed.

to the standard tracking, the single track Bhabha identification rate increases. The main reason is that there are less single track Bhabha events in the gap region of the calorimeter, which is now covered by the two track trigger. The signal misidentification

Table C.4.: Fraction of triggered Bhabha events after the Bhabha veto, relative to the total number of generated Bhabha events. The generated events cover a polar angle region of $\theta_0^* \in [15°, 165°]$, where θ_0^* is the polar angle of the electron/positron in the center of mass frame.

track finding	1 track	2 tracks	≥ 3 tracks	total
standard	1.75 %	1.18 %	0.22 %	3.15 %
short ($p_\perp \geq 0.3\,\mathrm{GeV}/c$)	1.80 %	2.63 %	0.76 %	5.19 %
short ($p_\perp \geq 0.255\,\mathrm{GeV}/c$)	1.97 %	2.89 %	0.70 %	5.56 %

rates change only by $\mathcal{O}(1\,\%)$, except for $\mu^+\mu^-$ events, where the misidentification rate decreases due to the better particle identification.

To obtain an estimate for the total background rate from Bhabha events, table C.4 shows the fraction of triggered Bhabha events for the limited test sample that was used to derive the Bhabha vetos. With the short track trigger, the triggered Bhabha rate increases by a factor of 1.6 compared to the standard track trigger, or a factor of 1.8 if the Hough plane is enlarged for low $p_\perp$. Again, a full background simulation is needed to evaluate the final background rate after the Bhabha veto and the z-vertex veto.

Bibliography

[1] A. D. Sakharov. Violation of CP invariance, C asymmetry, and baryon asymmetry of the universe. *Soviet Physics Uspekhi*, 34(5):392, 1991.

[2] J. H. Christenson, J. W. Cronin, V. L. Fitch, and R. Turlay. Evidence for the 2π decay of the K_2^0 meson. *Phys. Rev. Lett.*, 13:138–140, Jul 1964.

[3] M. Kobayashi and T. Maskawa. CP-Violation in the Renormalizable Theory of Weak Interaction. *Progress of Theoretical Physics*, 49(2):652–657, 1973.

[4] S. Skambraks. Use of Neural Networks for Triggering in Particle Physics. Diploma thesis, Ludwig-Maximilians-Universität, München, 2013.

[5] M. Tomoto et al. Z-trigger system with the BELLE Central Drift Chamber. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 447(3):416 – 423, 2000.

[6] C. L. Cowan, F. Reines, F. B. Harrison, H. W. Kruse, and A. D. McGuire. Detection of the Free Neutrino: a Confirmation. *Science*, 124(3212):103–104, 1956.

[7] R. N. Cahn and G. Goldhaber. *The experimental foundations of particle physics*. Cambridge University Press, 2nd edition, 2011.

[8] M. Breidenbach, J. I. Friedman, H. W. Kendall, E. D. Bloom, D. H. Coward, H. DeStaebler, J. Drees, L. W. Mo, and R. E. Taylor. Observed Behavior of Highly Inelastic Electron-Proton Scattering. *Phys. Rev. Lett.*, 23:935–939, Oct 1969.

[9] E. Fermi. Versuch einer Theorie der β-Strahlen. I. *Zeitschrift für Physik*, 88(3):161–177, 1934.

[10] N. Cabibbo. Unitary Symmetry and Leptonic Decays. *Phys. Rev. Lett.*, 10:531–533, Jun 1963.

[11] C. Patrignani et al. Review of Particle Physics. *Chin. Phys. C*, 40(10):100001, 2016 and 2017 update. available at `http://pdg.lbl.gov/2017/`.

[12] S. L. Glashow, J. Iliopoulos, and L. Maiani. Weak Interactions with Lepton-Hadron Symmetry. *Phys. Rev. D*, 2:1285–1292, Oct 1970.

[13] J. J. Aubert et al. Experimental Observation of a Heavy Particle J. *Phys. Rev. Lett.*, 33:1404–1406, Dec 1974.

[14] J. E. Augustin et al. Discovery of a Narrow Resonance in e^+e^- Annihilation. *Phys. Rev. Lett.*, 33:1406–1408, Dec 1974.

[15] M. L. Perl et al. Evidence for Anomalous Lepton Production in e^+e^- Annihilation. *Phys. Rev. Lett.*, 35:1489–1492, Dec 1975.

[16] S. W. Herb et al. Observation of a Dimuon Resonance at 9.5 GeV in 400 GeV Proton-Nucleus Collisions. *Phys. Rev. Lett.*, 39:252–255, Aug 1977.

[17] F. Abe et al. Observation of Top Quark Production in $p\overline{p}$ Collisions with the Collider Detector at Fermilab. *Phys. Rev. Lett.*, 74:2626–2631, Apr 1995.

[18] S. Abachi et al. Search for High Mass Top Quark Production in $p\overline{p}$ Collisions at $\sqrt{s} = 1.8$ TeV. *Phys. Rev. Lett.*, 74:2422–2426, Mar 1995.

[19] T. D. Lee and C. N. Yang. Question of Parity Conservation in Weak Interactions. *Phys. Rev.*, 104:254–258, Oct 1956.

[20] C. S. Wu, E. Ambler, R. W. Hayward, D. D. Hoppes, and R. P. Hudson. Experimental Test of Parity Conservation in Beta Decay. *Phys. Rev.*, 105:1413–1415, Feb 1957.

[21] R. L. Garwin, L. M. Lederman, and M. Weinrich. Observations of the Failure of Conservation of Parity and Charge Conjugation in Meson Decays: the Magnetic Moment of the Free Muon. *Phys. Rev.*, 105:1415–1417, Feb 1957.

[22] T. D. Lee, R. Oehme, and C. N. Yang. Remarks on Possible Noninvariance under Time Reversal and Charge Conjugation. *Phys. Rev.*, 106:340–345, Apr 1957.

[23] D. Boutigny et al. The BABAR Physics Book: Physics at an Asymmetric B Factory. In *Workshop on Physics at an Asymmetric B Factory (BaBar Collaboration Meeting) Pasadena, California, September 22-24, 1997*, 1998.

[24] I.I. Bigi and A.I. Sanda. *CP Violation*. Cambridge Monographs on Particle Physics, Nuclear Physics and Cosmology. Cambridge University Press, 2009.

[25] A. Rasin. Diagonalization of quark mass matrices and the Cabibbo-Kobayashi-Maskawa matrix. arXiv:hep-ph/9708216, 1997.

[26] L. Wolfenstein. Parametrization of the Kobayashi-Maskawa Matrix. *Phys. Rev. Lett.*, 51:1945–1947, Nov 1983.

[27] K. Prothmann. *Measurement of Branching Ratios and CP Asymmetries for the decays $B^0 \to \pi^+\pi^-$, $B^0 \to K^+\pi^-$, $B^0 \to K^+K^-$*. PhD thesis, Ludwig-Maximilians-Universität, München, 2012.

[28] O. Steinkamp. *Experiments in Beauty*. Habilitationsschrift, University of Zürich, 2015.

[29] I. Adachi et al. Precise Measurement of the CP Violation Parameter $\sin 2\phi_1$ in $B^0 \to (c\bar{c})K^0$ Decays. *Phys. Rev. Lett.*, 108:171802, Apr 2012.

[30] J. Charles et al. Current status of the standard model CKM fit and constraints on $\Delta F = 2$ new physics. *Phys. Rev. D*, 91:073007, Apr 2015. Updated results and plots available at `https://ckmfitter.in2p3.fr`.

[31] J. W. Flanagan et al. Letter of Intent for KEK Super B Factory. Part III: Accelerator Design. Tech. rep. KEK Report 04-4, 2004.

[32] T. Abe et al. Achievements of KEKB. *Progress of Theoretical and Experimental Physics*, 2013(3):03A001, 2013.

[33] Y. Ohnishi et al. Accelerator design at SuperKEKB. *Progress of Theoretical and Experimental Physics*, 2013(3):03A011, 2013.

[34] The Belle II collaboration and the B2TiP theory community. The Belle II Physics Book. in preparation (draft available at `https://confluence.desy.de/display/BI/B2TiP+ReportStatus`).

[35] Jolanta Brodzicka et al. Physics Achievements from the Belle Experiment. *PTEP*, 2012:04D001, 2012. arXiv:1212.5342.

[36] K. Wille. *Physik der Teilchenbeschleuniger und Synchrotronstrahlungsquellen: Eine Einführung*. Teubner Studienbücher Physik. Vieweg+Teubner Verlag, 2nd edition, 1996.

[37] T. Abe et al. Belle II Technical Design Report. arXiv:1011.0352, 2010.

[38] M. Venturini and W. Kozanecki. The hourglass effect and the measurement of the transverse size of colliding beams by luminosity scans. In *PACS2001. Proceedings of the 2001 Particle Accelerator Conference (Cat. No.01CH37268)*, volume 5, pages 3573–3575 vol.5, 2001.

[39] W. Herr and T. Pieloni. Beam-Beam Effects. In *CAS - CERN Accelerator School: Advanced Accelerator Physics Course*, pages 431–459, 2014. arXiv:1601.05235.

[40] Belle II Detector 3D model. `https://www.belle2.org/archives/`.

[41] A. Abashian et al. The Belle detector. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 479(1):117 – 232, 2002. Detectors for Asymmetric B-factories.

[42] F. Müller. *Characterization and Optimization of the Prototype DEPFET Modules for the Belle II Pixel Vertex Detector*. PhD thesis, Ludwig-Maximilians-Universität, München, 2017.

[43] T. Geßler, W. Kühn, J. S. Lange, Z. Liu, D. Münchow, B. Spruck, and J. Zhao. The ONSEN Data Reduction System for the Belle II Pixel Detector. *IEEE Transactions on Nuclear Science*, 62(3):1149–1154, June 2015.

[44] K. Nishimura. The time-of-propagation counter for Belle II. In *Proceedings of the 7th International Workshop on Ring Imaging Cherenkov Detectors (RICH 2010)*, volume A639, pages 177–180, 2011.

[45] V. Aulchenko et al. Electromagnetic calorimeter for Belle II. *Journal of Physics: Conference Series*, 587(1):012045, 2015.

[46] T. Aushev et al. Physics at Super B Factory. arXiv:1002.5012, 2010.

[47] Y. Fukuda et al. Evidence for Oscillation of Atmospheric Neutrinos. *Phys. Rev. Lett.*, 81:1562–1567, Aug 1998.

[48] Q. R. Ahmad et al. Measurement of the Rate of $\nu_e + d \rightarrow p + p + e^-$ Interactions Produced by ^{8}B Solar Neutrinos at the Sudbury Neutrino Observatory. *Phys. Rev. Lett.*, 87:071301, Jul 2001.

[49] Z. Maki, M. Nakagawa, and S. Sakata. Remarks on the Unified Model of Elementary Particles. *Progress of Theoretical Physics*, 28(5):870, 1962.

[50] B. Pontecorvo. Neutrino Experiments and the Problem of Conservation of Leptonic Charge. *Soviet Journal of Experimental and Theoretical Physics*, 26:984, May 1968.

[51] J. P. Lees et al. Improved limits on B^0 decays to invisible ($+\gamma$) final states. *Phys. Rev. D*, 86:051105, Sep 2012.

[52] C. Bernardini, G. F. Corazza, G. Di Giugno, G. Ghigo, J. Haissinski, P. Marin, R. Querzoli, and B. Touschek. Lifetime and Beam Size in a Storage Ring. *Phys. Rev. Lett.*, 10:407–409, May 1963.

[53] H. Nakano et al. Beam Background Simulation for SuperKEKB/Belle-II. In *Proceedings of the 2nd International Particle Accelerator Conference, IPAC 2011, San Sebastian, Spain*, volume C110904, pages 3700–3702, Sep 2011.

[54] A. Moll. *Comprehensive study of the background for the Pixel Vertex Detector at Belle II*. PhD thesis, Ludwig-Maximilians-Universität, München, 2015.

[55] 11th background campaign of the Belle II collaboration.

[56] I. S. Lee. Status on ECL trigger. slides at 26th B2GM. `https://kds.kek.jp/indico/event/23336/session/19/contribution/322`.

[57] I. S. Lee. private communication.

[58] C. Li and J. Yin. Study of GRL&GDL. slides at 26th B2GM. `https://kds.kek.jp/indico/event/23336/session/19/contribution/335`.

[59] A. Moll. The Software Framework of the Belle II Experiment. *Journal of Physics: Conference Series*, 331(3):032024, 2011.

[60] M. Schnell. *Development of an FPGA-based Data Reduction System for the Belle II DEPFET Pixel Detector*. PhD thesis, Rheinische Friedrich-Wilhelms-Universität, Bonn, 2015.

[61] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111 – 122, 1981.

[62] P. V. C. Hough. Machine Analysis Of Bubble Chamber Pictures. In *Proceedings, 2nd International Conference on High-Energy Accelerators and Instrumentation, HEACC 1959: CERN, Geneva, Switzerland, September 14-19, 1959*, volume C590914, pages 554–558, 1959.

[63] R. O. Duda and P. E. Hart. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Commun. ACM*, 15(1):11–15, Jan 1972.

[64] K. Chen. Updated 2D Tracker TSIM Design for Central Drift Chamber (CDC) at Belle-II. Master's thesis, Fu Jen Catholic University, Taiwan, 2015.

[65] S. Skambraks, S. Neuhaus, and C. Kiesling. The Neuro-Z-Vertex Trigger of the Belle II Experiment. *EPJ Web Conf.*, 127:00016, 2016.

[66] S. Skambraks. *Machine learning based tracking at the first trigger level of the Belle II experiment*. PhD thesis, Ludwig-Maximilians-Universität, München, in preparation.

[67] Software Track Matching Status. internal documentation of the Belle II collaboration, available at `https://confluence.desy.de/display/BI/Software+TrackMatchingStatus`.

[68] F. Abudinén. Studies on the neural z-Vertex-Trigger for the Belle II Particle Detector. Master's thesis, Ludwig-Maximilians-Universität, München, 2014.

[69] Y. Iwasaki. private communication.

[70] R. Rojas. *Neural Networks – A Systematic Introduction*. Springer-Verlag, 1996.

[71] A. N. Kolmogorov. On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition. *Dokl. Akad. Nauk USSR*, 114:953 – 956, 1957.

[72] R. Hecht-Nielsen. Theory of the backpropagation neural network. In *1989 IJCNN, International Joint Conference on Neural Networks*, pages 593–605, 1989.

[73] M. Riedmiller. Advanced supervised learning in multi-layer perceptrons – from backpropagation to adaptive learning algorithms. *Computer Standards & Interfaces*, 16(3):265 – 278, 1994.

[74] C. Igel and M. Hüsken. Improving the Rprop Learning Algorithm. In *Proceedings of the Second International Symposium on Neural Computation*, pages 115 – 121, 2000.

[75] J. Zimmermann. *Statistical Learning in High Energy and Astrophysics*. PhD thesis, Ludwig-Maximilians-Universität, München, 2005.

[76] S. Neuhaus, S. Skambraks, F. Abudinén, Y. Chen, M. Feindt, R. Frühwirth, M. Heck, C. Kiesling, A. Knoll, S. Paul, and J. Schieck. A neural network z-vertex trigger for Belle II. *Journal of Physics: Conference Series*, 608(1):012052, 2015.

[77] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436 – 444, 2015.

[78] C. Hagmann, D. Lange, and D. Wright. Cosmic-ray shower generator (CRY) for Monte Carlo transport codes. In *2007 IEEE Nuclear Science Symposium Conference Record*, volume 2, pages 1143–1146, Oct 2007.

[79] D. J. Lange. The EvtGen particle decay simulation package. *Nucl. Instrum. Meth.*, A462:152–155, 2001. BEAUTY2000, Proceedings of the 7th Int. Conf. on B-Physics at Hadron Machines.

[80] S. Jadach, B.F.L. Ward, and Z. Was. The precision Monte Carlo event generator KK for two-fermion final states in e^+e^- collisions. *Computer Physics Communications*, 130(3):260–325, 2000.

[81] K. Hirata. An introduction to SAD. In *Proceedings of the Second Advanced ICFA Beam Dynamics Workshop*, CERN 88-04, 1988.

[82] S. Jadach, W. Placzek, and B.F.L. Ward. BHWIDE 1.00: O(α) YFS exponentiated Monte Carlo for Bhabha scattering at wide angles for LEP1/SLC and LEP2. *Physics Letters B*, 390(1):298–308, 1997.

[83] R. Kleiss and H. Burkhardt. BBBREM: Monte Carlo simulation of radiative Bhabha scattering in the very forward direction. *Comput. Phys. Commun.*, 81:372–380, 1994.

[84] S. Skambraks. Background Suppression with the Belle II Neural Network Trigger. slides at 18th International Workshop on Advanced Computing and Analysis Techniques in Physics Research. `https://indico.cern.ch/event/567550/contributions/2629586`.

[85] S. Neuhaus, S. Skambraks, and C. Kiesling. Track vertex reconstruction with neural networks at the first level trigger of Belle II. *EPJ Web Conf.*, 150:00009, 2017.

[86] E. Cho. Variance of sample variance with replacement. *IJPAM*, 52(1):43–47, Mar 2009.

Acknowledgements

First I wish to thank my supervisor, Professor Christian Kiesling, who supported me throughout my work. Besides guiding me in my project and fighting for funding, he also taught me a lot about particle physics in general. Thanks for sharing your experience and enthusiasm!

Thanks also to Professor Alois Knoll from the computer science department of the TU Munich, who initially gave me the opportunity to do my PhD in his group despite not having studied computer science, but physics. He then introduced me to Christian and the neurotrigger project, so I ended up handing in the thesis at the physics department after all, but I am very greatful for his support.

My next thanks goes to my colleague Sebastian Skambraks, who introduced me to the project and helped to get me started. During the last four years, he developed the z-vertex trigger together with me and gave helpful feedback and ideas. In particular, thanks for letting me borrow the results of your background studies.

Of course, all the ideas and algorithms we came up with would be useless without someone implementing them in the trigger hardware. Special thanks to Steffen Bähr and Yang Chen, for bringing the z-vertex trigger into reality and patiently explaining the possibilities and impossibilities of FPGAs. Thanks also to Tzu-An Sheng, for re-designing the track finder at a fairly late stage in the development to include my suggestions.

A big thanks goes also to the rest of the CDC trigger group. For someone working primarily with simulations, the technical discussions were sometimes hard to understand, but certainly interesting. Thanks to all of you for the many helpful explanations.

Finally, I want to thank the whole Belle II group at the Max Planck Institute, for the great time I had during my PhD studies. Thanks especially to Luigi Li Gioi and Fernando Abudinén for their help in understanding B physics, and to Felix Müller for sharing all his experience in writing a thesis (about one year ahead of me) and his preparation notes for the defense. Thanks also to Miro, Tanja, Viktor, Katha and everyone else in the defense-preparation group.

Last but not least, I want to take this opportunity to thank my parents for their support throughout my studies. Thanks to my father for encouraging me in my choice of subject and sharing my interest, and thanks to my mother for her patience during many long discussions at dinner.