

Thesis for the aquisition of the academic degree
Master of Science

**Development of a real-time track
reconstruction for the proposed
LumiTracker detector**

Lukas Rolf
born in Münster

June 2022

Lehrstuhl für Experimentelle Physik V
Fakultät Physik
Technische Universität Dortmund

First corrector: Prof. Dr. Johannes Albrecht
Second corrector: Prof. Dr. Kevin Kröninger
Submission date: 1. June 2022

Abstract

Measuring the luminosity is a vital task performed at the LHCb experiment. The luminosity is used as feedback for the LHC and as input for offline analyses. The main goal of the proposed LumiTracker detector is to provide a measurement of luminosity by operating independently from the rest of the LHCb experiment. The LumiTracker measures luminosity by reconstructing and counting tracks and also measures the luminous region by extending these tracks to the interaction point. The reconstruction needs to be performed in real-time.

This thesis presents the pattern recognition and track fitting algorithms developed and implemented in a dedicated LumiTracker sequence in Allen. The LumiTracker sequence has a throughput of $(1.691 \pm 0.011) \times 10^7$ events/s with 14 threads on a Intel i7-8086k CPU. Therefore two of these CPUs are needed for real-time reconstruction for an effective bunch crossing rate of 30 MHz. The pattern recognition used finds $(96.84 \pm 0.16)\%$ of the tracks from the luminous region, while keeping the ghost and clone rate under 1%.

Kurzfassung

Die Messung der Luminosität ist eine wichtige Aufgabe des LHCb-Experiments. Die Luminosität wird als Feedback für den LHC und als Input für offline Analysen verwendet. Das Hauptziel des vorgeschlagenen LumiTracker-Detektors ist die Messung der Luminosität, indem er unabhängig vom Rest des LHCb-Experiments operiert. Der LumiTracker misst die Luminosität durch das Rekonstruieren und Zählen von Spuren und misst auch die Kollisionsregion, indem er diese Spuren bis zum Wechselwirkungspunkt verlängert. Die Rekonstruktion muss in Echtzeit durchgeführt werden.

In dieser Arbeit werden die Algorithmen zur Spurfindung und Spuranpassung vorgestellt, die für eine spezielle LumiTracker-Sequenz in Allen entwickelt und implementiert wurden. Die LumiTracker-Sequenz hat mit 14 Threads auf einer Intel i7-8086k CPU einen Durchsatz von $(1.691 \pm 0.011) \times 10^7$ Events/s. Daher sind zwei dieser CPUs für die Echtzeit-Rekonstruktion bei einer effektiven Eventrate von 30 MHz erforderlich. Die verwendete Mustererkennung findet $(96.84 \pm 0.16)\%$ der Spuren aus der Kollisionsregion, während die Ghost- und Klonrate unter 1% bleibt.

Contents

1	Introduction	1
2	The LHCb experiment	3
2.1	The LHC	3
2.2	The LHCb detector	4
2.2.1	Tracking	7
2.2.2	Luminosity measurements	8
3	The LumiTracker	10
3.1	Hardware	11
3.2	Software	12
3.2.1	High-level trigger 1	12
3.2.2	LumiTracker sequence	13
3.3	Simulation	14
4	Development of a pattern recognition	18
4.1	Search by Triplet	18
4.1.1	Track seeding	19
4.1.2	Track forwarding	20
4.1.3	Implementations under consideration	20
4.2	Parameter optimization	21
4.2.1	LumiTracker implementation	23
4.2.2	VELO implementation	25
4.2.3	Comparison between the two implementations	28
4.3	Results	30
4.3.1	Dependence on momentum	30
4.3.2	Dependence on origin vertex position	30
4.3.3	Dependence on occupancy	32
4.4	Conclusion	33
5	Development of the track fit	34
5.1	Track state definition	34
5.2	Kalman filter	34
5.2.1	Prediction step	35

5.2.2	Filter step	36
5.2.3	Performance evaluation	37
5.3	Application to the LumiTracker	38
5.3.1	Scattering	39
5.3.2	Measurements	42
5.3.3	Extrapolation to the IP	43
5.4	Tuning the fit inside the LumiTracker	44
5.4.1	Case with momentum	44
5.4.2	Case without momentum	46
5.4.3	Comparison	50
5.5	Tuning the extrapolation to the IP	53
5.6	Final results	56
6	Summary and Outlook	57

1 Introduction

Since ancient times, trying to understand the world around us has been a desire for humans. Through the ages, many models for the world have been developed. If a model's prediction did not match the observations, it was either discarded or improved. The best model we have today is the Standard Model (SM) of particle physics. It describes with huge success three of the four fundamental forces of nature: the strong, weak, and electromagnetic forces. The gravitational force is described today by the separate theory of general relativity. Unifying both theories is still a huge challenge. The Standard Model's incompleteness is evident, as it fails to describe gravity, the matter-antimatter asymmetry in the universe, dark matter, and neutrino oscillations. Nevertheless, it accurately describes particle interaction at a smaller scale. The idea is that the Standard Model is only a part of a whole and is missing physics of higher order, which would allow explaining the not yet understood phenomena. Many different experiments are performed to probe the Standard Model. One of these experiments is the LHCb experiment located at the Large Hadron Collider (LHC) at CERN in Switzerland.

This thesis is a contribution to the LHCb experiment and describes the process of implementing and tuning the track reconstruction for the proposed LumiTracker. The LumiTracker is a dedicated luminometer that would operate independently from the rest of the LHCb experiment. It would measure the luminosity at the LHCb interaction and the location of the collision region. The LHCb detector undergoes major upgrades before Run 3 to allow for a 40 MHz readout [1]. In addition, also the LHC injection complex is upgraded. Therefore, measuring the luminosity with different independent methods and detectors is necessary to allow cross-checks and robustness against systematic uncertainties.

For Run 3, the trigger system of LHCb is upgraded, and the hardware trigger is removed to allow for higher trigger yield in the fully hadronic decay modes [2] and, as a consequence, consists of a fully software-based high-level trigger [3]. In order to process events at 40 MHz and deal with higher track multiplicities due to the increased instantaneous luminosity of $2 \times 10^{33} \text{ cm}^{-2} \text{ s}^{-1}$ a fast track reconstruction in the first stage of the trigger (HLT1) is crucial. The HLT1 is implemented in the Allen project [3] which will perform a partial reconstruction for charged particles to identify events of interest [3]. As the Allen project already takes care of providing an

1 Introduction

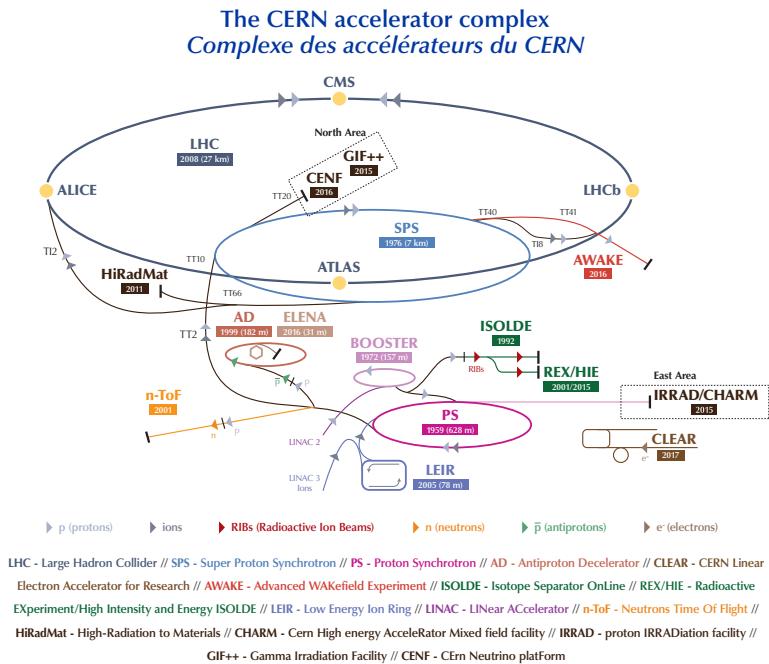
adequate framework for reconstruction tasks, it is a suitable choice as the baseline for the LumiTracker reconstruction.

This document is structured as follows. First, the LHCb experiment is discussed in chapter 2, and the need for different luminometers is explained. Then the LumiTracker detector is introduced in chapter 3 together with an overview of the Allen software and the simulation used for performance studies. The two main algorithms used in the software and their performance will then be shown in chapter 4 and 5. Finally, the results are summarized in chapter 6.

2 The LHCb experiment

The Large Hadron Collider beauty (LHCb) experiment is one of the four major physics experiments located at the Large Hadron Collider (LHC). The Large Hadron Collider itself is located near Geneva in Switzerland and operated by the European Organization for Nuclear Research (CERN). The experiments located at the LHC try to improve the understanding of particle physics and search for new physics beyond the Standard Model by studying the products of particle collisions. This chapter will give a brief overview of the LHC and the LHCb experiment.

2.1 The LHC



The LHC is a particle accelerator and collider and supersedes the Large Electron-Positron Collider (LEP). The LHC is located in a tunnel around 100 m below ground. The particle collider has a circumference of 26.7 km and was designed to collide protons at a center-of-mass energy of 14 TeV [5]. This makes it the largest particle collider in the world. Two proton beams are accelerated in opposing directions to energies of 7 TeV per beam. As the magnetic field of the magnets holding the particles on their circular path can not be arbitrarily varied, several pre-accelerators are necessary, with the last one being the Super Proton Synchrotron (SPS). The LHC, together with the pre-accelerators, are depicted schematically in Figure 2.1. After the injection to the LHC, both beams consist of around 3600 bunches, with up to 10^{11} protons each [6]. The bunches revolve around the LHC with a revolution frequency of 11 245 Hz. Fillings with heavy ions instead of protons are also possible. There are four different locations where the beams are crossed to collide the accelerated particles. These correspond to the locations of the four major experiments: A Toroidal LHC ApparatuS (ATLAS), Compact Muon Solenoid (CMS), Large Hadron Collider beauty (LHCb), and A Large Ion Collider Experiment (ALICE). The ATLAS [7] and CMS [8] experiments use general-purpose detectors. Their main goals are to study the properties of the Higgs boson and the top quark. The two collaborations independently observed the Higgs boson in 2012 [9],[10]. The LHCb [11] experiment was designed to study B physics and is described in more detail in section 2.2. The ALICE [12] experiment focuses on studying strongly interacting matter present in quark-gluon-plasma created at the extreme energy densities present for heavy-ion collisions. In Run 3 bunch crossings occur for all experiments every 25 ns corresponding to a bunch crossing rate of 40 MHz. Because of the complicated filling procedure, the filling is not homogeneous and empty bunches exist [13]. Therefore the effective bunch crossing rate is about 30 MHz.

2.2 The LHCb detector

The LHCb detector is a single-arm forward spectrometer designed to study hadrons containing beauty (b) and charm (c) quarks. The different layout from the other experiments is due to its optimization for particles containing these quarks. The $b\bar{b}$ quark pairs are primarily produced within a small angle to the beams. The covered pseudorapidity η is between two and five. Simulations show that over 24% of the $b\bar{b}$ -pairs fall within the detector acceptance. For Run 3 of the experiment, the detector is being upgraded.

The upgraded LHCb detector is shown in Figure 2.2 in the LHCb reference system. A global right-handed coordinate frame is defined with the z axis along the beamline and the y axis pointing upward. In this system, the interaction point (IP) is

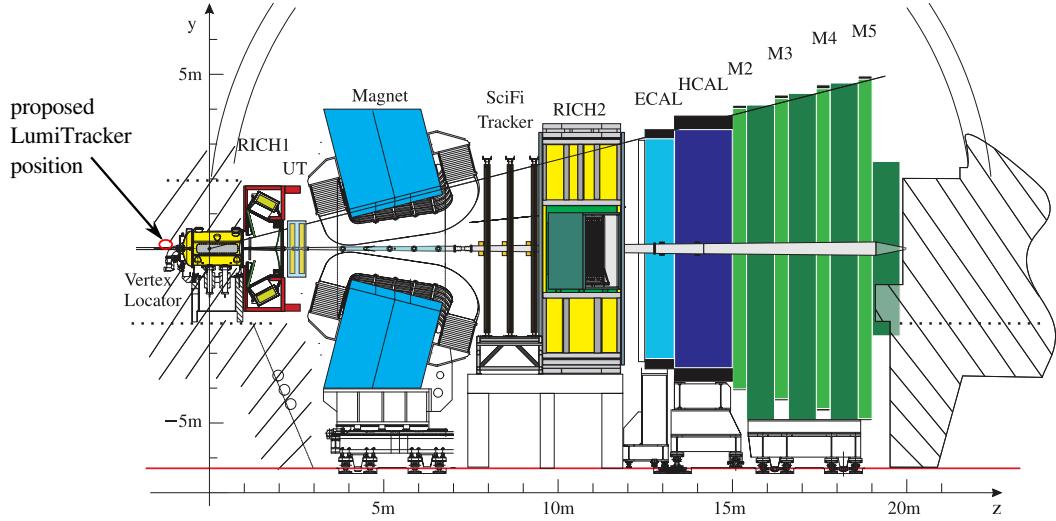


Figure 2.2: A schematic picture of the upgraded LHCb detector. The proposed location for the LumiTracker is marked in red. The position of the components is to scale depicted in the LHCb reference system. The not depicted x axis is perpendicular to the y - z plane.

located in the origin. This coordinate system is used throughout the thesis. The direction in z and $-z$ is referred to as downstream and upstream respectively. The upgraded detector will be taking data after the long shutdown 2 [14]. The multiple subdetectors and the location of the proposed LumiTracker are indicated in the figure. The LumiTracker is discussed in detail in chapter 3. The major components of the upgraded detector are briefly introduced in the following paragraphs, starting with the tracking detectors, then the magnet, and lastly the subdetectors for particle identification.

VELO One point of interest is the decay time of beauty hadrons. Therefore a high vertex reconstruction efficiency and good impact parameter resolution are necessary. This is achieved with the vertex locator (VELO) [14]. The impact parameter resolution achieved with the VELO is around $30\text{ }\mu\text{m}$ for particles with 1 GeV transverse momentum [14]. The primary vertex finding efficiency is around 90 % [3]. The VELO is also an essential component of the tracking system, which is further discussed in section 2.2.1. It features 52 modules with four hybrid pixel sensors each arranged in two halves of 26 modules each, which can be open up to 3 cm and closed around the beam. The detector surrounds the interaction region. It is located inside the VELO tank and is separated from the vacuum in the beam pipe by a around $150\text{ }\mu\text{m}$ thin aluminum foil. In the closed state, the sensors are as

close as 5.1 mm from beamline. In this state, the detector has the highest primary vertex resolution. The open state exists for unstable beam conditions, which for example, can be present during the injection of particles to the LHC and would damage the detector in the closed state.

UT The upstream tracker (UT) [15] is part of the tracking system upstream of the magnet. Its main purpose is to provide a position measurement of tracks before the magnet to make the extrapolation of tracks to the SciFi tracker behind the magnet as easy as possible. The detector is composed of four planes of silicon strips. The first and last plane have vertically orientated strips, while the middle planes are rotated by -5° and 5° respectively. The tilt in the middle layers allows to also measure the y coordinate.

SciFi The scintillating fiber (SciFi) tracker [15] is located downstream of the magnet. It is divided into three stations called T1, T2, and T3. Each of them is composed of four scintillating fiber layers. Two layers of the four layers for each station are tilted like for the UT. The light passing through the fibers after interaction with a charged particle is detected with silicon photomultipliers.

Dipole magnet The magnet is important for the estimation of the particle momentum and the determination of its charge. The bending of the trajectories of the particles allows an estimation of the momentum of the particle. The magnet has an integrated magnetic field of 4 T m along the z axis and can be operated in both polarities *MagUp* and *MagDown* with the main component of the magnetic field pointing up and down respectively. The traversing particle's trajectories are therefore bent in the x - z plane.

RICH1 and RICH2 The ring imaging Cherenkov (RICH) detectors RICH1 and RICH2 are primarily used to measure particle velocity. If particles travel through a material faster than the speed of light in the material, a cone of photons is radiated at an angle $\theta = \cos^{-1}(1/(n\beta))$. The refractive index of the material is denoted as n and the velocity of the particle divided by the speed of light in a vacuum as β . For RICH1 and RICH2, C_4F_{10} and CF_4 are used as radiators respectively [16]. Together with the momentum estimate from the tracking systems, the velocity estimate allows the identification of particles over the mass.

ECAL and HCAL The calorimeter measures the energy deposit of particles. The LHCb detector has both an electronic calorimeter (ECAL) and a hadronic calorimeter (HCAL) [16]. The ECAL is composed of alternating layers of scintillators and lead absorbers and measures the energy of electrons and photons and contributes to their identification. The HCAL is similar in design and uses iron as an absorber instead. It can measure the energy of hadrons.

Muon system As the muons traverse the whole calorimeter, a separate system specially designed for muons is needed for their identification. The muon detector is located downstream of the calorimeter and is composed of four muon stations named M2 to M5. Between the stations, iron walls are placed as absorbers. For the muon system, a separate reconstruction of muon tracks is performed. It allows for a resolution in the transverse particle momentum of around 20 % [17].

2.2.1 Tracking

For a good estimation of the kinematic quantities of a particle, their track reconstruction is essential. Therefore the experiment is equipped with dedicated tracking subdetectors. These are depicted in Figure 2.3 with the addition of the proposed LumiTracker. Their goal is to reconstruct the trajectories of the particles through the detector. It is, therefore, necessary to find for each particle a pattern in the hits in the detectors and assign them to the corresponding tracks accordingly. A good pattern recognition groups the hits produced by the same particle, minimizing the wrongly assigned hits. Tracks, which only traverse the VELO, are called VELO tracks. These are primarily used for the reconstruction of the primary vertices. If a track traverses only the SciFi detector, it is called a T track. T tracks can be used for the RICH2 reconstruction. If a track traverses all the tracking subdetectors, it is called a long track. Long tracks have the most accurate momentum estimate and are the most useful for physics. Tracks only traversing the VELO and UT are called upstream tracks. These have a reduced momentum resolution as they do not have hits after the magnet and are used in the RICH1 reconstruction. Lastly, downstream tracks only have hits in the UT and SciFi. They allow the reconstruction of particles that decay outside the VELO.

The proposed LumiTracker is completely separate from the rest of the tracking detectors. Different from the other tracking detectors, it is located upstream of the interaction point. Tracks traversing the LumiTracker are called lumi tracks in this document. The lumi tracks are located far from the magnet and have no momentum estimate.

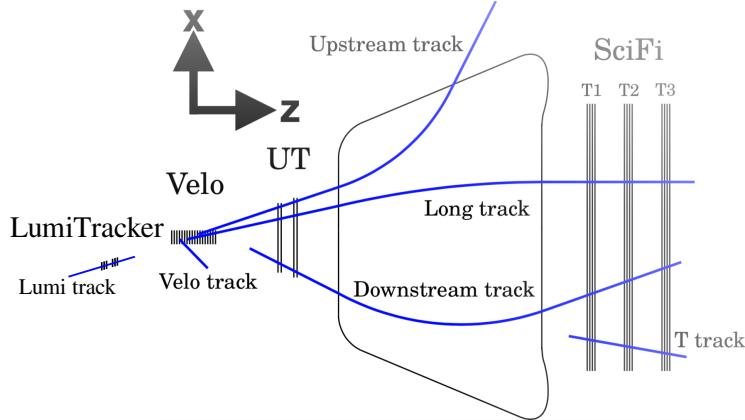


Figure 2.3: Sketch of the the upgraded LHCb tracking system with the addition of the LumiTracker. The different types of tracks are depicted. The sketch is adapted from [3] and not to scale.

2.2.2 Luminosity measurements

Besides the collision energy, the luminosity is the most important measure of the performance of a collider, as it is directly proportional to the number of collisions. High statistics, which come with a high number of collisions, are important for the study of rare decays and precision measurements. At LHCb the instantaneous luminosity of $2 \times 10^{33} \text{ cm}^{-2} \text{ s}^{-1}$ is expected after the long shutdown 2 [14]. The number of particle collisions per bunch crossing follows a Poisson distribution and varies between bunch crossings. The average instantaneous luminosity for relativistic particles and small crossing angles can be expressed as [18]:

$$\langle L_{\text{inst}} \rangle = N_1 N_2 f_r 2c \int \rho_1(x, y, z, t) \rho_2(x, y, z, t) dx dy dz dt, \quad (2.1)$$

where N_1 and N_2 denote the number of particles in the colliding bunches, f_r the revolution frequency of the beams and ρ_1 and ρ_2 the spatial particle density functions. A precise measurement of luminosity is needed for many analyses. This type of measurement is called offline luminosity measurement. It is used, for example, in the estimation of the proton-proton cross-section [19]. A measurement of the luminosity is furthermore also needed for the stable operation of the LHC. The luminosity at the LHCb experiment is leveled, which means that the luminosity is kept to an average of about 5.5 visible collisions in upgrade conditions. The luminosity can be adjusted by offsetting the beams or defocusing them [20]. To adjust for changes, the LHC needs to be provided with measurements of the luminosity at the LHCb interaction point every few seconds. This measurement is referred to as online luminosity measurement.

The luminosity is determined by continuously monitoring processes, referred to as counters, whose rate is proportional to luminosity. Several counters are used, from different subdetectors, for example, the number of VELO tracks and vertices. A dedicated luminometer has been installed in Run 3, called PLUME (Probe for LUminosity MEasurement) [21]. PLUME uses Cherenkov radiation to measure the luminosity based on the logZero method [21]. The LumiTracker is an additional dedicated luminometer proposed to be installed during Run 3 of data taking. Using a completely different technology than PLUME, it will provide complementary and additional information. All these counters, including the dedicated luminometer, provide a relative measurement of luminosity that needs to be calibrated. For this, two different methods can be used. The first one is unique to the LHCb experiment and is called “beam-gas imaging” [22]. It allows the measurement of the particle beam profiles for both beams by injecting noble gas into the beam area. The VELO can be used to precisely measure the distribution of the interactions between the beams and the gas molecules. The “van der Meer scan” [23] is also used by other experiments and uses the possibility of moving the beams. The relative collision rate is scanned over the x - y plane to measure the overlap integral.

3 The LumiTracker

The LumiTracker detector is a proposed luminometer to be installed during Run 3 of data taking and was designed to measure the online luminosity at the LHCb interaction point using the track counting method. In addition, it also should contribute to the offline measurement of luminosity. The LumiTracker is designed as a mini-telescope of hybrid silicon pixel sensors. The proposal can be found in more detail in reference [24]. The track counting method exploits the linear dependence between the mean luminosity per bunch and the mean number of reconstructed tracks:

$$\langle L_{\text{bunch}} \rangle = \frac{A \langle n_{\text{tracks}} \rangle f_r}{\sigma_{\text{vis}}}, \quad (3.1)$$

where the revolution frequency f_r for the LHC is 11 245 Hz. The constant proportionality factor A/σ_{vis} is determined in dedicated runs with the van der Meer scan [23]. Reconstructing tracks also allows studying the luminous region through the estimated track origin. In principle, all reconstructed tracks, including tracks that originated from the luminous region and tracks due to the interaction with the material of the LHCb detector and beam pipe, can be used for the luminosity measurement. Nevertheless, movements of the luminous region lead to different interactions with the material and influence the number of tracks from material interactions. This means that the proportionality factor in equation (3.1) has a dependence on the position of the luminous region due to the varying fraction of material tracks. Hence the best stability for the luminosity measurement is obtained for a high percentage of tracks coming directly from the luminous region. As the LumiTracker is also able to estimate the position of the luminous region, the effect can be partially corrected. There are several requirements to be fulfilled for an online luminometer for LHCb. From the point of view of software design, the most important ones are the need to provide the estimate of luminosity to the LHC every few seconds, per bunch, and to be able to operate independently from the LHCb all the time while still being integrated into its readout system.

The proposed location of the LumiTracker is upstream of the VELO: the conceptual design can be seen in Figure 3.1. To be noted that only a few of the sensitive planes are depicted in the design together with their readout electronics. The mechanics and cooling infrastructure are not indicated. The LumiTracker has a length of around 25 cm. The six planes are arranged into two arms with a spacing of 100 mm

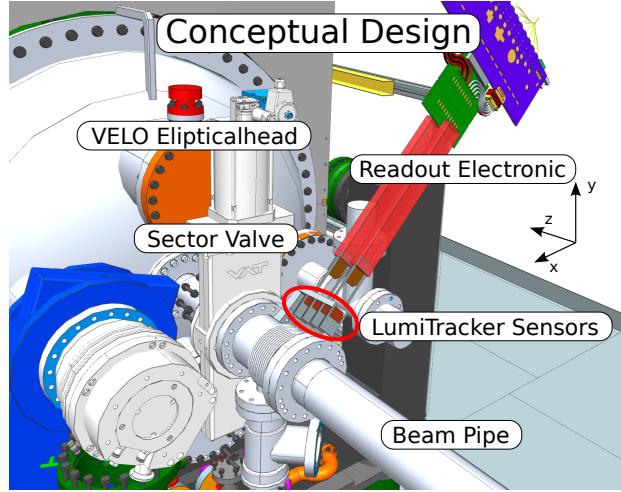


Figure 3.1: Conceptual design of the LumiTracker. The active area of the LumiTracker sensors is indicated in dark red. While only four depicted planes are depicted, the current design uses six planes equally divided into two arms.

between the arms and 30 mm between the planes in each arm. The planes are staggered in order to point to the interaction region. This staggering is done so that particles coming from the luminous region traverse as many planes as possible. The planes are also rotated by 45° around the z axis. The sensitive area of the LumiTracker is located between 120 mm and 160 mm away from the z axis and between -1361 mm and -1138.8 mm along the z axis.

3.1 Hardware

Each plane of the LumiTracker consists of a 200 μm thick n-on-p silicon sensor bump-bonded to three 200 μm thick VeloPix ASICs. The specifications of the VeloPix ASIC are listed in Table 3.1. Each silicon sensor has 256×768 pixels with a pitch of 55 μm . The resolution of the sensor depends on several factors, including the incidence angle of the particle, the bias voltage, and the irradiation level. As a reference, the resolution of single hit clusters at perpendicular incidence for a full depletion is around 15 μm [25]. The cluster finding efficiency is above 99 %. Because of the distance of the LumiTracker to the IP, it is expected that cooling of the detector with dry air to a temperature slightly below room temperature is sufficient [24]. For efficient cooling, the detector layers will be surrounded by an airtight box through which air of 15 °C is blown. As the temperature of the air is above the dew point, condensation is avoided. The sensor technology is the same

Table 3.1: Specifications of the VeloPix ASIC.

pixel size	$55 \times 55 \mu\text{m}^2$
matrix size	256×256
thickness	$(200 \pm 20) \mu\text{m}$
timestamp resolution	25 ns
Time over Threshold	low rate only
peak pixel hit rate	900 MHz
power consumption	< 3 W/ASIC
radiation hardness	> 400 MRad
single event upset robust	yes
number of links	4
bit rate per link	5.12 Gbps

as for the VELO detector. Hence the readout electronics of the LumiTracker are completely based on the one developed for the VELO upgrade [14]. The electronic is for the LumiTracker read out at the bunch-crossing frequency of 40 MHz.

3.2 Software

Here the software, for the LumiTracker track reconstruction, used for both offline luminosity measurement and as feedback for the LHC is discussed. The reconstruction of the tracks per each collision event will be performed in the Event Builder servers in real-time similar to the new High-Level Trigger 1 (HLT1). The Event Builder servers receive the data from the subdetectors and combine the data fragments from a single bunch crossing into one event and group around 1000 of them into multi-events packets [3].

3.2.1 High-level trigger 1

The main goal of the trigger is to filter out events of interest and persist them for later processing stages. This high-level trigger 1 for the upgraded LHCb experiment is implemented in the Allen project [3] as a sequence of algorithms. The Allen project features a partial event-reconstruction for the HLT1 sequence. The reconstruction is a huge computational challenge, as nearly 30 million events need to be processed per second (see 2.1). Therefore the Allen project is optimized for computing performance. This is also a focus of the LumiTracker reconstruction. Allen is implemented mostly in C++ and CUDA [26]. CUDA is Nvidia's API for developing GPU accelerated

software. In the CUDA programming model, algorithms running on the GPU are implemented as kernels. These are executed as a grid of blocks of threads. Each thread can be used to perform a part of the calculations. Synchronization barriers

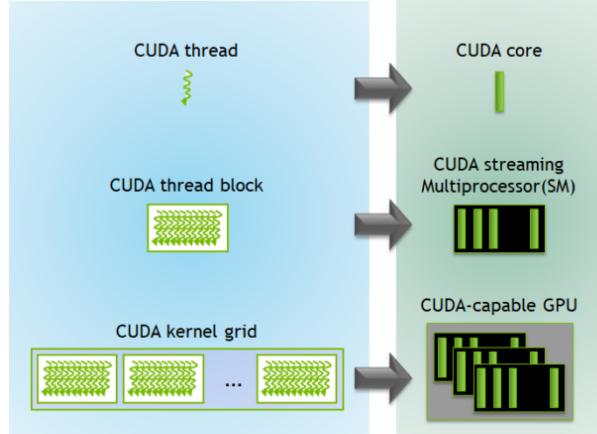


Figure 3.2: The different components of the CUDA programming model and the corresponding components on the GPU [27].

cause all threads in a block to be finished before the execution is resumed. The CUDA streaming multiprocessors depicted in Figure 3.2 have fast shared memory which can be used by all threads inside one block. If work on the same data is required, it is preferred to do it in the same block. Most of the time, events can be processed completely independently from each other, and therefore a separate block is used for each event. The algorithms are then parallelized for one event using the threads of one block.

The Allen project also uses multiple CPU threads, each steering a CUDA stream. A CUDA stream is a sequence of CUDA operations executing in issue order. A configurable number of events is processed in parallel on every stream. This number is also called the number of events in a slice. The entire sequence of algorithms is performed for every slice.

3.2.2 LumiTracker sequence

For the LumiTracker reconstruction a dedicated LumiTracker sequence has been implemented. This sequence is similar in structure to the already implemented VELO reconstruction sequence and is depicted in Figure 3.3. First, the input with the encoded clusters is decoded into hits. This is followed by a pattern recognition algorithm, which finds hits belonging to the same track. The implementation and

performance are in detail discussed in 4. Finally, a track fit is performed. The implementation, tuning, and results are discussed in chapter 5. The output of the LumiTracker sequence is evaluated in a separate custom C++ program afterward for the use of this thesis.

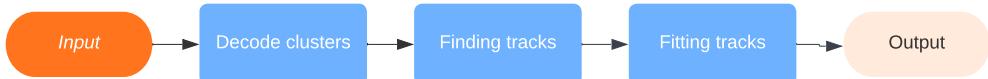


Figure 3.3: A simplified view of the different steps performed in the LumiTracker sequence.

3.3 Simulation

The implemented LumiTracker sequence has been tested on simulated data. The simulation was performed for the upgraded LHCb beam conditions listed in Table 3.2. The standard LHCb simulation package GAUSS v53r1 is used. For the simulation of nominal pp collisions at 14 TeV the backend PYTHIA [28] and LHCb configuration [29] were used. Interactions of the created particles with the material of the detector were simulated through the GEANT4 toolkit [30]. If a particle interaction would lead to a signal in the sensitive planes of the LumiTracker, the position of the particle in the middle of the plane is taken as the true hit position. As the digitization of the LumiTracker hits has not been implemented yet, the position of the hits in the x and y direction of the LHCb coordinate system was smeared with a Gaussian of 15 μm width, to simulate the detector resolution. The sensor and the 3 ASICs are modeled as two blocks of silicon with a thickness of 0.2 mm each. Details of the mechanics and infrastructure of the LumiTracker will be implemented at a later stage. To be noted, that the simulation is missing part of the material in between the interaction region and the LumiTracker, in particular, the VELO elliptical head and the valves that can be seen in Figure 3.1. Hence the studies presented need to be repeated with the full updated detector description. The geometry description used in the simulation is depicted in Figure 3.4. For the studies presented in this thesis, about 10^5 simulated events are used.

As the LumiTracker is located about 1.3 m away from the interaction point, the expected occupancy inside one LumiTracker sensor is significantly smaller than for one VELO sensor. The overall occupancy in the LumiTracker and the occupancy per sensor is depicted in Figure 3.5 per collision event. The overall distribution

Table 3.2: Upgrade I nominal beam conditions used in simulation.

Beam energy	7 TeV
Number of bunches colliding	2400
Half internal angle	$\pm 135 \mu\text{rad}$
Half external angle	$-250 \mu\text{rad}$
Luminosity	$2 \times 10^{33} \text{ cm}^{-2} \text{ s}^{-1}$
ν (# interactions per crossing)	7.6
μ (# visible interactions per crossing)	5.5
Bunch z RMS	90 mm
β^*	2.79 m
Emittance	$3.8 \mu\text{m}$
z RMS luminous region	63 mm
x, y RMS luminous region	27 μm

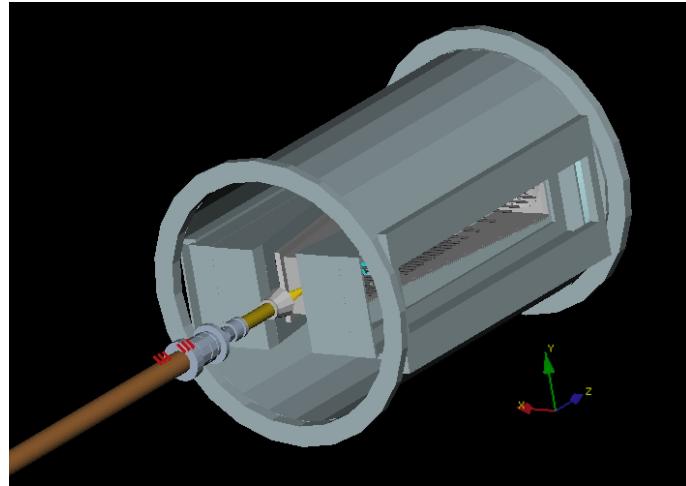


Figure 3.4: The material used in the simulation displayed through Panoramix [31]. The planes of the LumiTracker are depicted in red.

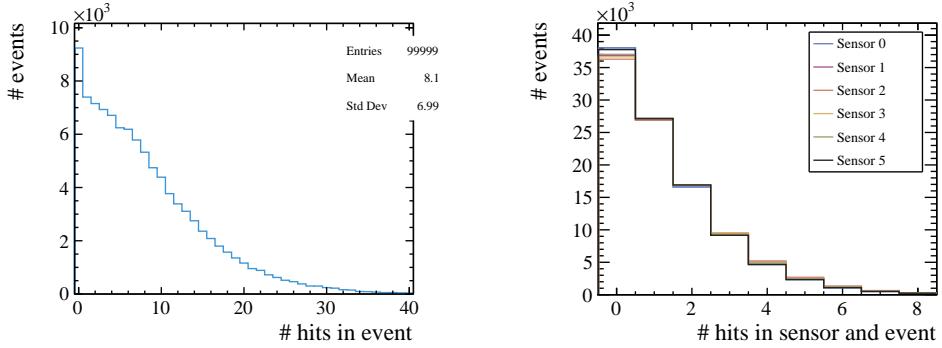


Figure 3.5: The overall occupancy in the detector (left) and the occupancy in the different sensors (right) per collision event.

of hits in the sensor closest to the IP is depicted in Figure 3.6. It can be seen that the distribution inside the sensor is nearly uniform, with a slightly higher number of hits closer to the beamline. This is the expected behavior as the particle density decreases with the distance from the beamline and IP. The luminous region

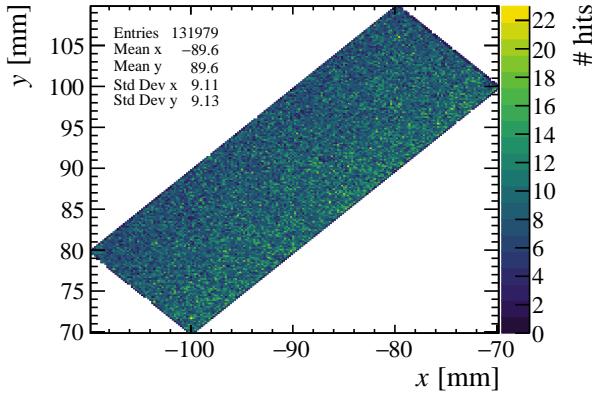


Figure 3.6: The distribution of hits inside the LumiTracker sensor closest to the IP.

is defined from $z = -150$ to $z = 150$. Particles having their origin vertex in this range are denoted as particles from the luminous region and all other particles as particles from material interactions. As the reconstruction of tracks from the luminous region is a priority of the LumiTracker, the distributions of the origin vertices along the beamline are depicted for particles from the luminous region and from material interactions in Figure 3.7. The distribution of primary vertices is expected to be Gaussian. The depicted distribution of origin vertices for particles

3.3 Simulation

from the luminous region is comparable, with only some small spikes caused by material interaction. In addition, the left side of the distribution has a larger tail as also particles from decays are present. In the distribution of the origin vertex location for particles from material interactions, the position of material is clearly visible as spikes in the distribution.

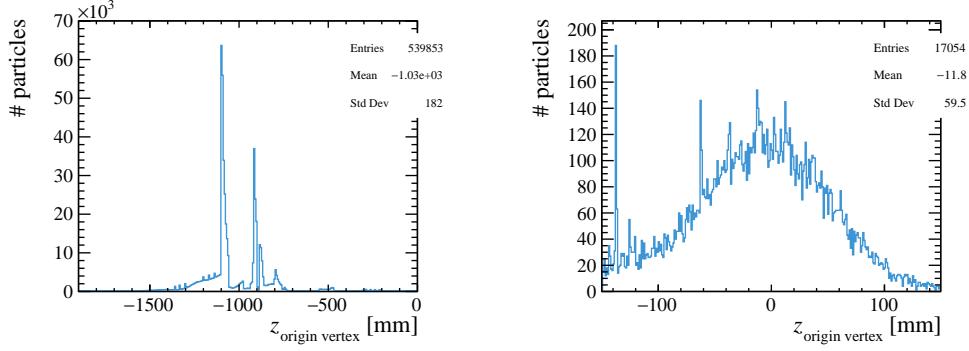


Figure 3.7: The distribution of the origin vertex location along the z axis for all particles (left) and for particles from the luminous region (right).

4 Development of a pattern recognition

In the case of track reconstruction, pattern recognition means the process of finding hits in the different stations of the tracking detectors that originated from the same particle, hence exhibiting a pattern. The pattern recognition in the LumiTracker detector is based on the *Search by Triplet* algorithm [32], adapted in order to account for the fact that the LumiTracker is located about 1.3 m away from the interaction point and its specific layout.

4.1 Search by Triplet

The Search by Triplet algorithm takes as input hits from the six different planes for each collision event with the goal of finding and grouping together hits produced by the same particle. It is assumed that the charged particles traverse the LumiTracker in nearly straight lines since no magnetic field is present and scattering occurs only at the different planes. The scattering consists of two alternating steps:

- The first step is called the *track seeding* and finds triplets from three neighboring planes. A *triplet* is a set of three hits, one for each of these sensors.
- The second step is called *track forwarding* and extrapolates the triplets and tracks to the next plane to search for additional hits. A *track* denotes triplets that have been assigned at least one additional hit.

To filter in both steps the hits considered for a track or triplet, search windows are used. A *search window* denotes, in this context, a cut on a variable to decide beforehand if a hit should be further considered. The implementation for the LumiTracker starts with track seeding for the three planes furthest away from the interaction point and then progresses towards it. The Search by Triplet algorithm is performed in parallel for multiple events, as they can be considered entirely independent.

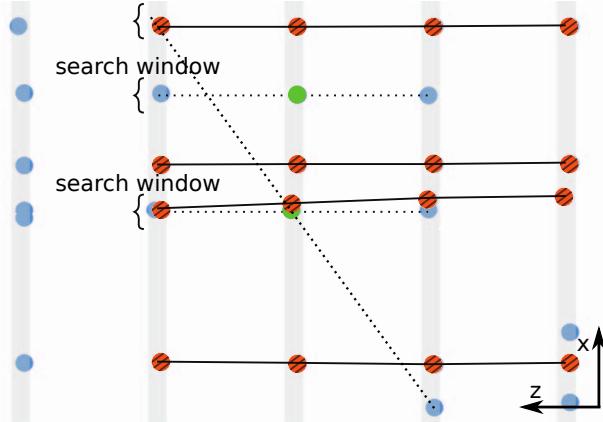


Figure 4.1: The procedure of finding triplets is depicted. The schematic is adapted from [32]. The hits marked in red are already part of a track and therefore not considered in the seeding. The seeding is performed for all the hits in a plane marked in green. The linear extrapolation is depicted with dotted lines, and the search windows are indicated with brackets.

4.1.1 Track seeding

Given hits in three neighboring sensors, the track seeding finds the “best” triplets of hits. Figure 4.1 depicts the process of finding triplets. Only hits that are not part of an already found track are used for the track seeding. For a hit in the middle sensor of the three neighboring planes, all pair combinations with hits in the upstream plane are considered and extrapolated to the next plane downstream. Only hits within a specified search window $\sqrt{\Delta x^2 + \Delta y^2} < R_{\text{search}}$ around the extrapolated position are taken into account. Here x and y denote the position of the hit in the LHCb reference frame (see section 2.2). The hit closest to one of the extrapolated positions is added to the other two hits to form a triplet. The R_{search} parameter accounts for the uncertainty of the measured hit position and the possible scattering of the particle.

The track seeding step can be parallelized. As the usage of the same hit in the previous plane or next plane is allowed for different triplets, the work for each hit in the middle plane is done in parallel. Synchronization of the threads afterward is necessary to keep the track forwarding separate. Because of the low occupancy in the LumiTracker (see 3.3), it is worth considering removing the inter-event parallelization completely and processing more events in parallel instead.

4.1.2 Track forwarding

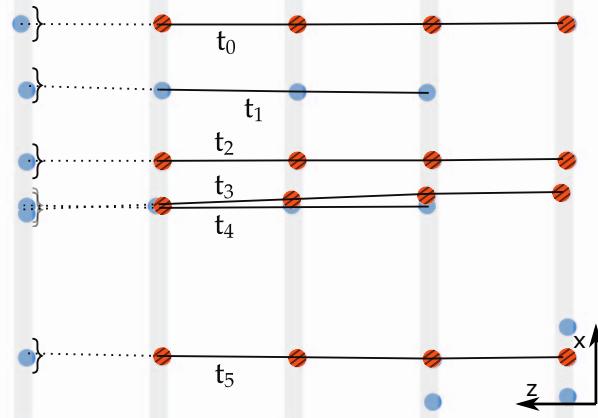


Figure 4.2: The process of extending the tracks and triplets to the next plane is depicted. The schematic is adapted from [32]. The line segments used for the linear extension are denoted as t_0 to t_5 , and the search window is depicted as a bracket.

Existing triplets and tracks are then extrapolated to the next plane downstream. This is done by using a straight-line extension of the last two hits in the track or triplet for a position estimate in the next plane. The nearest hit inside the search window is then added to the track. Hits can again be added to multiple tracks. If no hit is within the search window of the next plane, hits in the next plane downstream are considered until the maximally allowed number of skipped modules n_{skip} is exceeded. This parameter is used to tune the pattern recognition to optimize the physics performance. The process of track forwarding is depicted in Figure 4.2.

As each track can be forwarded independently, this is done in parallel. Synchronization of the threads is necessary afterward, as all added hits need to be flagged before the next track seeding step is performed. It is again worth considering removing the parallelization because of the low occupancy inside the LumiTracker.

4.1.3 Implementations under consideration

In this thesis, two different implementations of the Search by Triplet algorithm are investigated. The one described in the previous sections is referred to as LumiTracker implementation. The second one is closer to the implementation used for the VELO detector and is, therefore, referred to as the VELO implementation.

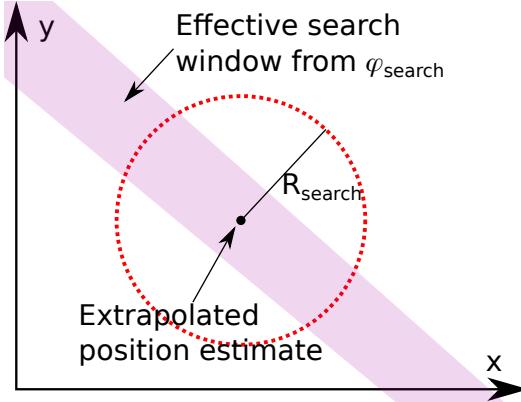


Figure 4.3: Depicted are the two different search windows R_{search} and φ_{search} . Only hits that are inside both search windows at the same time are considered.

The main difference in the VELO implementation with respect to the LumiTracker one is that the hits are sorted beforehand in $\varphi = \tan^{-1}(y/x)$ for each plane, with x and y coordinates of the hits in the LHCb reference frame. An additional search window in φ , denoted as φ_{search} , is introduced and depicted together with the search window R_{search} in Figure 4.3. The search window φ_{search} can be implemented through a binary search, as the hits are sorted in φ . Differently from the LumiTracker implementation, where all hit pairs are considered for the triplet, here only a definite number of combinations, specified by the n_{seeding} parameter is examined. The used seeding candidates are determined by a pendulum search in φ around the φ position of the hit in the middle plane. The LumiTracker implementation does not use φ in any way.

4.2 Parameter optimization

The two different pattern recognition algorithms have been implemented, and in each case, the parameters have been tuned to achieve optimal physics and computing performance. For the physics performance, the reconstructed tracks from the track reconstruction are compared to the simulated particles from the simulation. The quantities from the simulation which are not available in the reconstruction are denoted with a preceding “true”. Details of the simulation can be found in section 3.3. The physics performance is evaluated based on the following figures of merit:

- The *reconstructed tracking efficiency* is defined as the ratio between matched

particles and reconstructible particles:

$$\epsilon_{\text{reconstructed}} = N_{\text{matched particles}} / N_{\text{reconstructible}}. \quad (4.1)$$

A particle is considered matched if it shares at least 4 hits with a track. The track is then a matched track. A particle is denoted as reconstructible if at least hits in 4 different planes of the LumiTracker exist.

- The *ghost rate* is determined by:

$$\text{ghost rate} = N_{\text{ghosts}} / N_{\text{reconstructed}}, \quad (4.2)$$

where $N_{\text{reconstructed}}$ is the overall number of reconstructed tracks and N_{ghosts} is the number of tracks that have less than four hits from a common particle.

- The *clone rate* is given by the ratio between the number of clone tracks and the number of matched tracks:

$$\text{clone rate} = N_{\text{clones}} / N_{\text{matched tracks}}. \quad (4.3)$$

A clone track is when a track shares more than a certain amount of its hits with another track. In this case the limit was set to 70 %.

- The *hit purity* for a matched track is the fraction of true hits belonging to the matched particle:

$$\text{hit purity} = \frac{\# \text{ true hits on track}}{\text{total } \# \text{ of hits on matched track}}. \quad (4.4)$$

- The *hit efficiency* is the efficiency of assigning hits from the matched particle to the track:

$$\text{hit efficiency} = \frac{\# \text{ true hits on track}}{\text{total } \# \text{ of hits on matched particle}}. \quad (4.5)$$

- The *material fraction* of matched tracks is the fraction of tracks not from the luminous region. It is given by:

$$\text{material fraction} = \frac{N_{\text{material}}}{N_{\text{matched tracks}}}. \quad (4.6)$$

The computing performance is evaluated through the number of processed events per second also called throughput for the whole sequence.

For the LumiTracker, particles from the luminous region are of increased interest, as stated in chapter 3. The tracking efficiency, hit efficiency, and hit purity used

for the tuning are therefore calculated from luminous region tracks and particles. The goal is to have a high reconstructed tracking efficiency of at least 95 % while also minimizing the material fraction of tracks, ghost rate, and clone rate. A high hit purity and a hit efficiency of above 90 % are targeted, as they are important for good results for the track fit. For the ghost and clone rate a value of under 1 % is desirable.

In the following, the pattern recognition algorithm parameters are introduced, and their optimizations are performed for both the LumiTracker and the VELO implementations. Finally, the results are compared, and one of the implementations is chosen for the LumiTracker sequence.

4.2.1 LumiTracker implementation

As previously introduced, the LumiTracker implementation has two parameters R_{search} and n_{skip} . The value for the R_{search} parameter is chosen based on the trend of the graphs in 4.4, where each of the figures of merit is studied as a function of the parameter value. The ghost rate is for nearly all values of R_{search} under the 1 % threshold. Hence it is not relevant for the optimization of the search window. The hit purity is also for all considered search windows higher than 99 %. Therefore the choice of the search window is based on balancing the tracking efficiency and hit efficiency against the material fraction of tracks and clone rate. To meet the previously stated goals for the efficiencies while minimizing the material fraction and clone rate, the search window of $R_{\text{search}} = 0.15$ mm is chosen and indicated as a line in the figure.

The n_{skip} parameter is optimized by fixing the R_{search} parameter to 0.15 mm. The n_{skip} parameter does not have much of an impact on the different performance measurements. This is shown for the hit efficiency and material fraction in Figure 4.5 as an example. A value of 1 has then been chosen for n_{skip} in order to account for a possible missing hit. With this, both parameters of the LumiTracker implementation are chosen, and the physics performance is summarized in Table 4.1.

Table 4.1: The performance measurements for the tuned parameters $R_{\text{search}} = 0.15$ mm and $n_{\text{skip}} = 1$ for the LumiTracker implementation. The corresponding ghost rate is $(0.36 \pm 0.05) \%$ and the material fraction $(28.93 \pm 0.35) \%$.

selection	recon. eff. [%]	hit pur. [%]	hit eff. [%]	clone rate
all	80.69 ± 0.28	99.916 ± 0.010	89.78 ± 0.12	0.84 ± 0.07
lumi. region	96.84 ± 0.16	99.936 ± 0.009	91.35 ± 0.13	0.52 ± 0.07

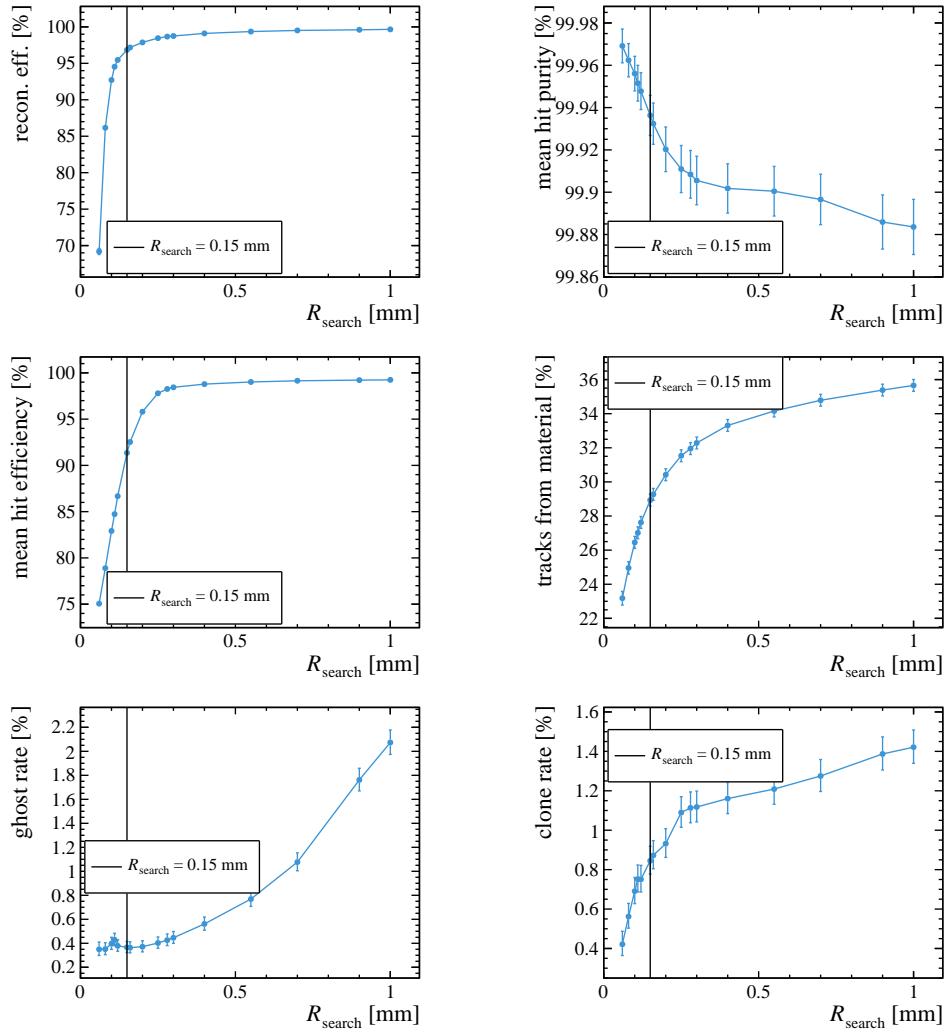


Figure 4.4: All six performance measurements against the used search window R_{search} . The chosen value $R_{\text{search}} = 0.15$ mm is indicated by a black line.

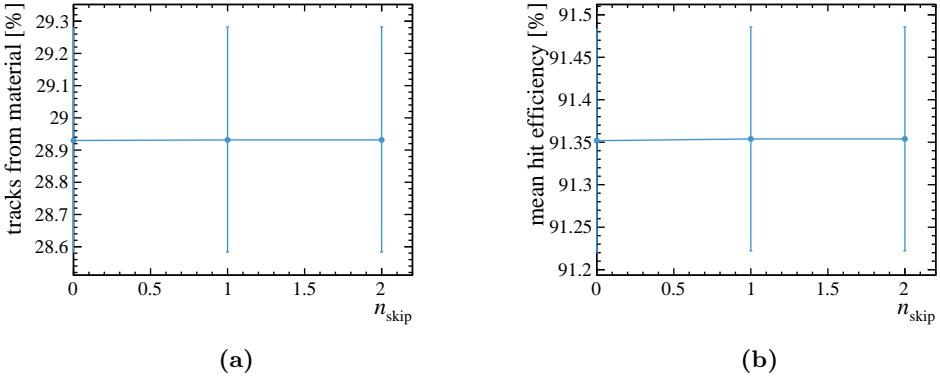


Figure 4.5: The material fraction (a) and the hit efficiency (b) against the n_{skip} parameter.

4.2.2 VELO implementation

In the VELO implementation there are four parameters to tune: R_{search} and n_{skip} , as for the LumiTracker implementation, and additionally, φ_{search} and n_{seeding} . The interplay between the radial and angular search windows has been investigated by studying the various figures of merit as a function of tracking efficiency while varying the values of these parameters. This is shown in Figure 4.6. It can be seen that the overall behavior for both search windows is similar, with the R_{search} parameter overall yielding better results. It was therefore decided to leave the search window at the previously chosen value as optimized for the LumiTracker implementation and use the parameter φ_{search} only to optimize the computing performance. The throughput on the CPU as function of the φ_{search} is shown in Figure 4.7a. As can be seen, a smaller value of φ_{search} yields a better computing performance, as expected as fewer combinations of hits need to be iterated over. The physics performance takes priority over the computing performance. Hence the parameter is set to the smallest possible value without impacting the physics performance. This is when taking into account the nearest distance of the sensitive area of the LumiTracker $R_{\text{active}} = 120 \text{ mm}$ to the z-axis and the chosen search window $R_{\text{search}} = 0.15 \text{ mm}$ calculated as

$$\varphi_{\text{search}} = \tan^{-1} \left(\frac{R_{\text{search}}}{R_{\text{sensitive}}} \right) \approx 1.25 \times 10^{-3} \text{ rad.} \quad (4.7)$$

For this value, all hits inside the radial search window are also contained in the search window in φ , as the effective search area depicted in Figure 4.3 in φ contains the search area defined by R_{search} in the relevant region of the sensitive area of the sensors.

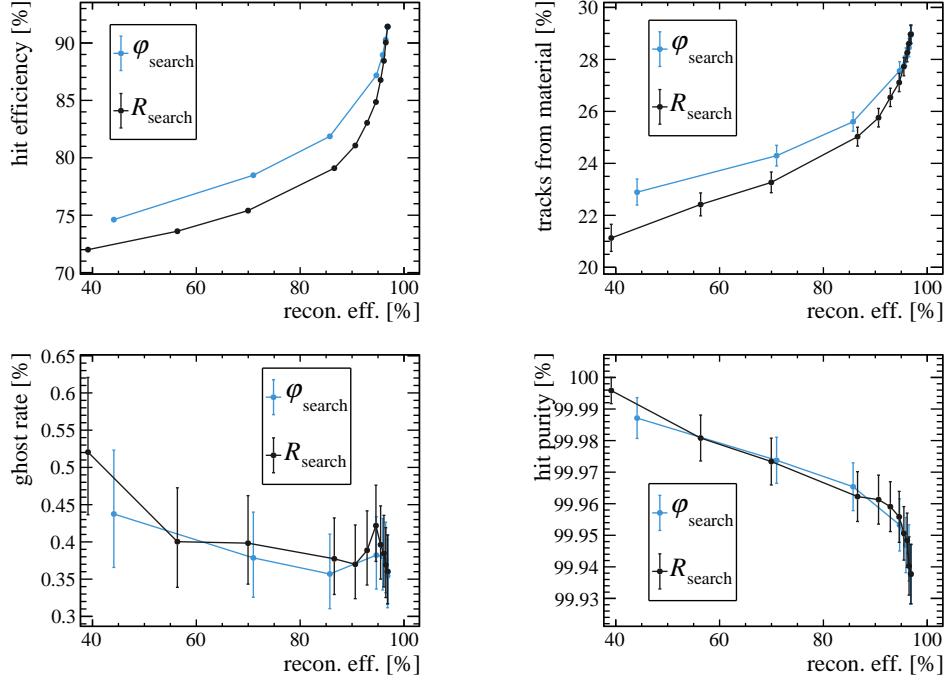


Figure 4.6: The other performance measurements except the clone rate against the reconstructed tracking efficiency for tracks from the luminous region.

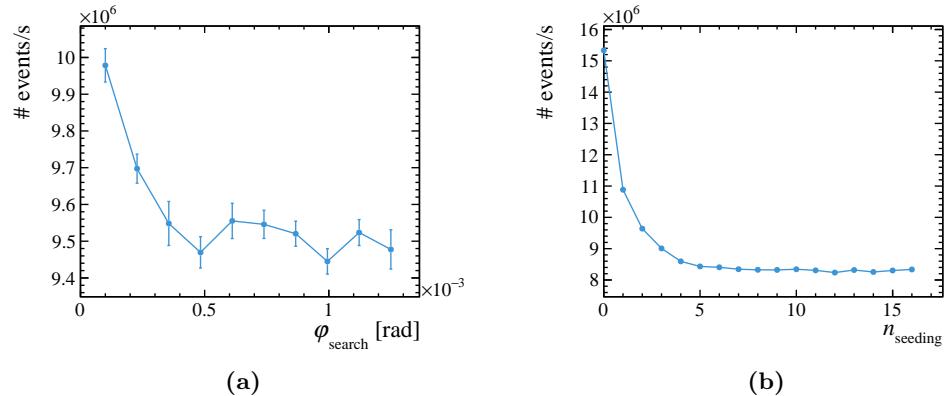


Figure 4.7: The computing performance on CPU in processed events per second against the φ_{search} (a) and n_{seeding} (b) parameter.

With all other parameters set to the determined values, the n_{seeding} parameter is investigated. Decreasing this parameter improves the computing performance, as depicted in Figure 4.7b, but below three, it also impacts the physics performance negatively. Therefore $n_{\text{seeding}} = 3$ is chosen. The finally chosen parameter set for the VELO implementation is listed in Table 4.2 and the corresponding physics performance is summarized in Table 4.3.

Table 4.2: The final parameter choice for the VELO implementation.

R_{search}	φ_{search}	n_{skip}	n_{seeding}
0.15 mm	1.25×10^{-3} rad	1	3

Table 4.3: The performance measurements for the tuned parameters listed in Table 4.2 for the VELO implementation. The corresponding ghost rate is $(0.37 \pm 0.05)\%$ and the material fraction $(28.92 \pm 0.35)\%$.

selection	recon. eff. [%]	hit pur. [%]	hit eff. [%]	clone rate [%]
all	80.61 ± 0.28	99.917 ± 0.009	89.84 ± 0.12	0.83 ± 0.07
lumi. region	96.78 ± 0.16	99.935 ± 0.010	91.42 ± 0.13	0.51 ± 0.07

The impact on the physics performance of the search windows is easily understood, as they limit how much the reconstructed tracks are allowed to scatter. This results in the suppression of low momentum particles. The reason for the n_{skip} parameter is also apparent, as it allows tracks that do not have hits in some sensors due to the detection efficiency not being 100 %. The physical interpretation of the n_{seeding} parameter is more complicated. Therefore it was investigated how the parameter choice influences which tracks are reconstructed. As a pendulum search around the φ position of the hit in the middle plane (see 4.1.3) is used to determine the candidates, the difference in phi $\Delta\varphi$ between hits in neighboring planes from the same particle was plotted against the origin vertex z position for matched particles. The hits in neighboring planes are in the following called hit pairs. To be able to better interpret the results, the hit pair distribution is first shown for all particles and all reconstructible particles from the simulation in Figure 4.8.

The luminous region can be clearly seen for $z_{\text{origin vertex}}$ between -150 mm and 150 mm as a concentration of hit pairs with small $\Delta\varphi$. The larger tail in $\Delta\varphi$ for particles from the luminous region is caused by multiple Coulomb scattering in the material. The vertical bands visible in the plot are caused by particles interacting with the material of the beam pipe. As can be seen in the right plot in Figure 4.8, the requirement of at least four hits per track reduces the number of particles coming from material interactions. This is because particles with four hits need to have

hits in both arms of the LumiTracker and, therefore, the allowed slope and the scattering of the particle are strongly limited. The locations of the planes of the detector are optimized for particles from the luminous region, and particles coming from material interactions often traverse only a small subset of planes.

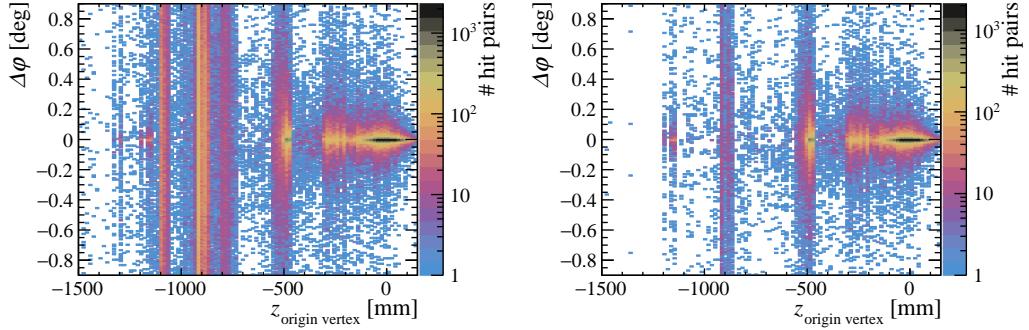


Figure 4.8: The hit pair distribution for all particles in the simulation (left) and only reconstructible particles (right) against the particle origin vertex position along the z axis.

In Figure 4.9 the distribution for the matched particles for $n_{\text{seeding}} = 1$ and $n_{\text{seeding}} = 3$ are shown as well as the deviation with respect to the distribution for all reconstructible particles. In the case of perfect reconstruction, these distributions would match. For $n_{\text{seeding}} = 1$, a clear bias toward the reconstruction of particles with a positive $\Delta\varphi$ between hits in neighboring planes can be seen. Particles due to material interactions at $z \approx 900$ mm are mostly not reconstructed. This does not change if the n_{seeding} is increased to 3. However, the bias toward positive $\Delta\varphi$ values is not visible anymore. Increasing the n_{seeding} parameter further only yields minimal improvements and is therefore here not shown. The bias is clearly present below $n_{\text{seeding}} = 3$.

4.2.3 Comparison between the two implementations

The physics performance is compared between the LumiTracker implementation in Table 4.1 and the VELO implementation in Table 4.3. It can be seen that the performance is compatible within uncertainties.

The computing performance eventually worsens for the VELO implementation compared to the LumiTracker one. The reason is that the sorting of hits in φ for each module can be removed in this case. The achieved throughput on CPU for the VELO implementation is of $(9.85 \pm 0.05) \times 10^6$ events/s while the LumiTracker implementation of $(1.691 \pm 0.011) \times 10^7$ events/s. The physics performance is, in

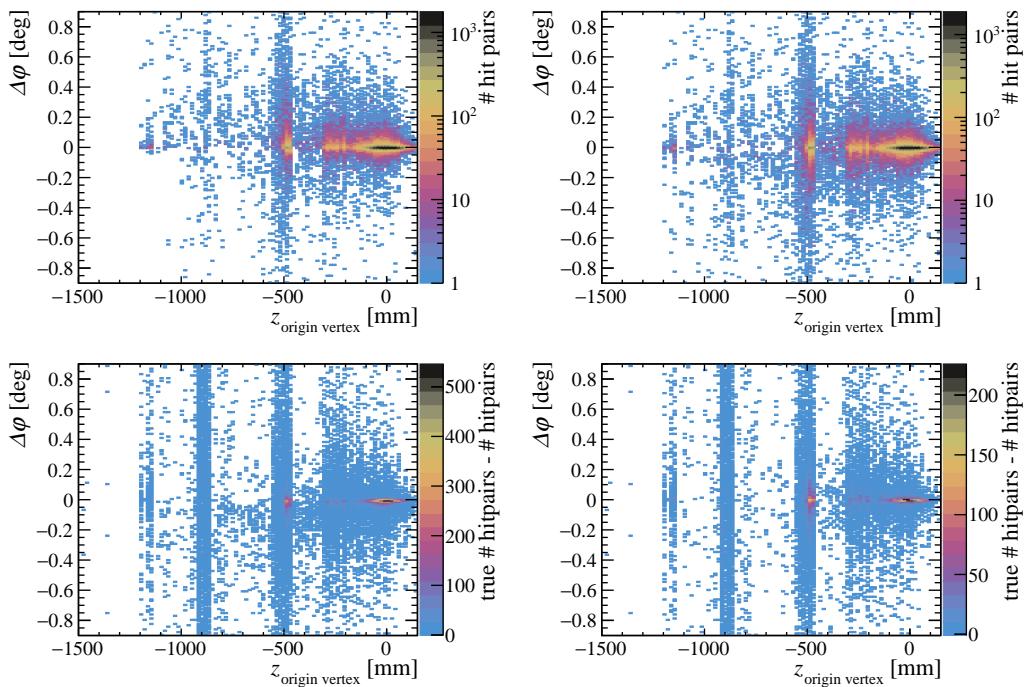


Figure 4.9: The values of $\Delta\varphi$ against the particle origin vertex z position for hit pairs in case of $n_{\text{seeding}} = 1$ (top left) and $n_{\text{seeding}} = 3$ (top right). The difference in each case to the distribution for all reconstructible particles in shown below.

both cases, nearly the same. In view of comparable physics performance, the LumiTracker implementation was chosen. The VELO implementation would benefit only from the better computing performance scaling for high occupancies.

4.3 Results

The results for the chosen LumiTracker implementation are presented here. First, the dependence on the momentum is discussed as a strong effect is expected. Because the reconstruction of the luminous region is one of the goals of the LumiTracker, the dependence of the reconstructed tracking efficiency on the origin vertex is investigated in section 4.3.2. The influence of the occupancy in a sensor when assigning a hit to the correct track is examined in section 4.3.3.

4.3.1 Dependence on momentum

The tracking efficiency strongly depends on the particle momentum. This is depicted in Figure 4.10 for all tracks and for tracks from the luminous region only. The tracking efficiency approaches in both cases 100 % for high momenta. For low momenta, the tracking efficiency drops. This behavior is expected as low momentum particles scatter much more and are therefore suppressed by the search window of the algorithm or leave the detector before traversing the required four planes. Nearly all high momentum particles are found, suiting the purpose of the LumiTracker since tracks from the luminous region carry higher momentum than particles generated in the interaction with the material, as can be seen by comparing the momentum distributions in Figure 5.3. This is because high momentum particles scatter less (see eq. 5.13).

4.3.2 Dependence on origin vertex position

Reconstructed tracks are extrapolated to the beamline since the distribution of the position of the closest approach to the beamline along the z axis represents the reconstructed luminous region distribution. Therefore ideally, the reconstruction efficiency must not strongly depend on the particle's origin vertex z position. To see if a strong correlation between the position and the tracking efficiency exists, they are plotted against each other, as shown in Figure 4.11.

It can be seen that particles coming from further downstream have higher reconstructed efficiency than particles coming from further upstream. This can be

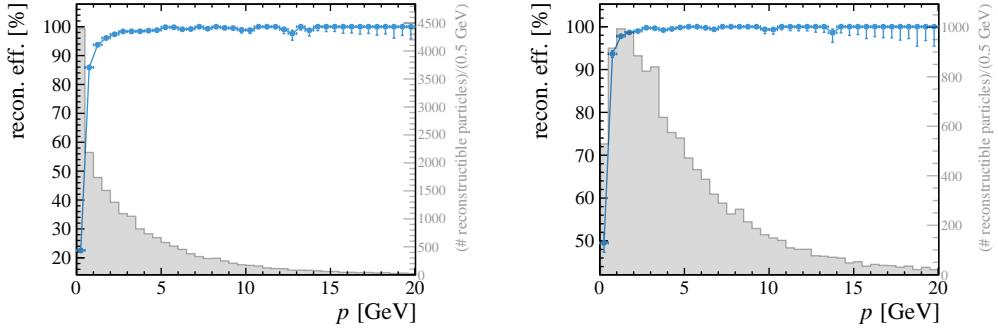


Figure 4.10: Tracking efficiency as function of the particle momentum for all reconstructible particles (left) and only reconstructible particles from the luminous region (right). The momentum distribution is overlaid.

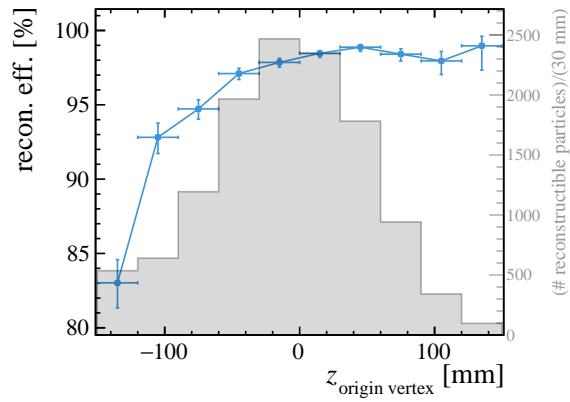


Figure 4.11: The reconstructed tracking efficiency against the origin vertex z position of the matched particle for particles from the luminous region.

explained by the location of the LumiTracker further downstream, as at lower $z_{\text{origin vertex}}$ fewer particles come directly from the PV and more particles from decays in flight and from interactions with the VELO material. This can lead to a bias of a few millimeters in the estimated z location of the luminous region if untreated, but this effect can be reduced by corrections based on the estimated $z_{\text{origin vertex}}$ location.

4.3.3 Dependence on occupancy

It is expected that with an increasing number of hits in a sensor, the number of correct assigned hits decreases. The number of correctly assigned hits, normalized to the number of hits from reconstructible particles, in a sensor is denoted as *sensor hit efficiency* and for all six sensors is plotted against the number of hits in the corresponding sensor in Figure 4.12.

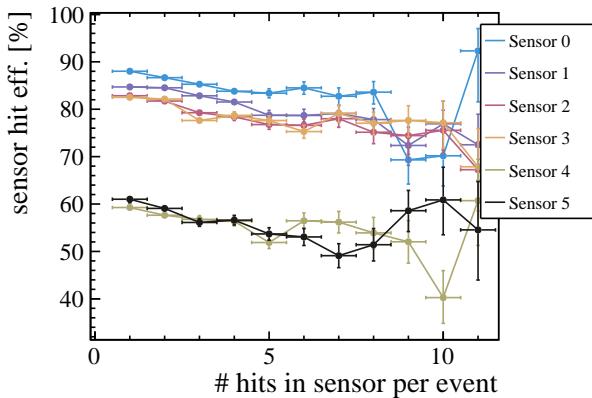


Figure 4.12: The number of correctly assigned hits divided by the number of assignable hits in the sensor for all hits against the overall number of hits in the sensor for all six sensors of the LumiTracker.

A similar decrease in sensor hit efficiency is observed for higher occupancies for all sensors. In addition, the sensor hit efficiency is significantly lower for sensors 4 and 5. The reason is believed to be the larger distance between the sensors 3 and 4 of 100 mm instead of the 30 mm between planes of the same arm. The search window is for all planes the same. Hence the extrapolation to sensor 2 is less often accepted if the particle scatters. An improvement to the algorithm could be to add a different search window for sensor 2.

4.4 Conclusion

The pattern recognition algorithm implemented finds $(96.84 \pm 0.16)\%$ of particles from the luminous region, which are of increased importance for the LumiTracker. For particle momenta over 5 GeV near 100 % are reconstructed. The hit efficiency for tracks from the luminous region is with $(91.35 \pm 0.13)\%$ high enough to not deteriorate the track fit. Furthermore, the hit purity already is over 99.9 %. Both clone and ghost rate are under 1 % and therefore negligible. Dependence of the tracking efficiency on the origin vertex location was found. This needs to be accounted for in the estimation of the position of the luminous region. While the occupancy dependence of the pattern recognition was studied, besides the expected decrease of the efficiency with higher occupancy, a lower sensor hit efficiency for sensors 4 and 5 was found. This was traced back to the design of the algorithm. As a result, the algorithm could be modified with a different search window for sensor 2 for a possible better physics performance.

The LumiTracker implementation achieves with 14 threads a throughput of $(1.691 \pm 0.011) \times 10^7$ events/s on an Intel i7-8086k CPU. Hence two CPUs of this type would be enough to satisfy the required throughput of 30 MHz. Two or less of recent server CPUs would be therefore required. Besides the discussed execution on CPU, the sequence can also be run on GPU. The evaluated performance is in this case with a throughput of $(3.3525 \pm 0.0011) \times 10^6$ events/s on an NVIDIA GeForce GTX 1070 much worse. For an NVIDIA A10, a comparable performance was observed, even if it is much more powerful. The low computing performance for the execution on GPU comes from the very small event size for the LumiTracker. For the different algorithms in the sequence, an event is assigned to one thread block, and because the event is too small to utilize the threads fully in the block, the computing performance is impacted. This can be solved by assigning multiple events to one thread block. This requires some changes to the data structures used for the LumiTracker. It is then expected that one recent GPU would be enough to provide the necessary throughput.

5 Development of the track fit

In this chapter the track fit is introduced. The track fit aims at estimating the true track state vector with a corresponding covariance matrix based on the hits assigned to the track. This step is crucial for a good estimate of the primary vertex location.

This chapter focuses on the Kalman filter as this is the fitting algorithm implemented for the LumiTracker tracks but will also compare the results between different implementations of the Kalman filter and a straight-line fit.

5.1 Track state definition

Since at the LumiTracker position effectively no magnetic field is present, the particle trajectory is approximated as straight line segments between scattering planes. Each track segment is defined by a track state, identified by its z position in the LHCb coordinate system. A track state is defined by a state vector and an associated covariance matrix W , which gives the uncertainty of the components. The state vector for the LumiTracker is defined as:

$$\vec{x} = \begin{pmatrix} x \\ y \\ t_x \\ t_y \end{pmatrix}, \quad (5.1)$$

where $t_x = \frac{\partial x}{\partial z}$ and $t_y = \frac{\partial y}{\partial z}$ are the slopes in the x -direction and the y -direction. The combination of a track state with a measurement is in the following referred to as a node.

5.2 Kalman filter

The problem of linear prediction and filtering exists in many different fields such as space science, statistical communication theory, and high energy physics [33, 34, 35]. Kalman provided in 1960 the Kalman filter as a solution [36]. Since then, it

has been widely used, and many variations of it exist [37]. Today it is still used in high energy physics [38]. In the following, the original linear Kalman filter [36] is described since it is the one implemented. The notation used in the following is inspired by reference [17].

The Kalman filter is a linear estimator that minimizes the mean squared error between the estimated and true states [39]. It is recursive and uses two alternating steps:

- The *prediction step* predicts the next state and covariance based on the last state. A random process occurring between the two states can be accounted for and is reflected in the covariance of the predicted state. The added uncertainty from the random process is in the following referred to as *process noise*.
- The *filter step* updates the predicted state and covariance based on a measurement. The uncertainty of the measurement is referred to as *measuring noise*. The updating accounts to ‘filtering’ out the noise.

In addition, so-called smoother equations can obtain the best estimates for the previous states after adding all measurements. These equations are not needed for this work as only the state at the interaction point is of interest (see 2), which can be obtained in the last step. More details about them can be found in reference [33]. As the algorithm is recursive, an initial state is needed. This state can come from another fit. If this fit is performed on measurements used for the Kalman filter, the diagonal entries of the initial covariance matrix need to be set to large values to avoid a bias caused by multiple uses of the same measurements.

For good results, both the extrapolation and the noise need to be accurately defined. The Kalman filter minimizes the mean squared error optimally if, in addition, the process noise and measuring noise are Gaussian distributed [37, 40]. If the noise is not Gaussian, the performance degrades [40]. Of importance for the evaluation of the performance of the fit are the χ^2 distribution, the width of the residual distribution, and how well the pulls (see section 5.2.3) are Gaussian distributed [17].

5.2.1 Prediction step

The prediction step predicts a new state based on the last state. The new predicted state vector \vec{x}_k^{k-1} at node k is calculated by multiplying the previous state vector \vec{x}_{k-1} at node $k-1$ (after all previous measurements were added) with the transport matrix \mathbf{F}_k :

$$\vec{x}_k^{k-1} = \mathbf{F}_k \vec{x}_{k-1}. \quad (5.2)$$

The covariance matrix \mathbf{W}_k^{k-1} for the predicted state is calculated from the covariance matrix \mathbf{W}_{k-1} of the previous state and the noise matrix \mathbf{Q}_k :

$$\mathbf{W}_k^{k-1} = \mathbf{F}_k \mathbf{W}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k. \quad (5.3)$$

The noise matrix describes the additional uncertainty in the predicted state caused by a random process in the form of a covariance matrix.

5.2.2 Filter step

The filter step updates the predicted state. The updated state vector is calculated from the residual of the measurement \vec{r}_k^{k-1} to the predicted state, the gain matrix \mathbf{K}_k and the predicted state as following:

$$\vec{x}_k = \vec{x}_k^{k-1} + \mathbf{K}_k \vec{r}_k^{k-1}. \quad (5.4)$$

The residual is calculated as the difference of the measurement m_k to the projected predicted state:

$$\vec{r}_k^{k-1} = m_k - \mathbf{H}_k \vec{x}_k^{k-1}, \quad (5.5)$$

where \mathbf{H}_k is a linear projection of the state vector to the measurement space. The updated covariance matrix is given by:

$$\mathbf{W}_k = (\mathbf{I} + \mathbf{K}_k \mathbf{H}_k) \mathbf{W}_k^{k-1}. \quad (5.6)$$

The gain matrix \mathbf{K}_k is calculated as following:

$$\mathbf{K}_k = \mathbf{W}_k^{k-1} \mathbf{H}_k^T \underbrace{(\mathbf{V}_k + \mathbf{H}_k \mathbf{W}_k^{k-1} \mathbf{H}_k^T)^{-1}}_{=(\mathbf{R}_k^{k-1})^{-1}}, \quad (5.7)$$

with \mathbf{V}_k as the covariance matrix of the measurement and \mathbf{R}_k^{k-1} as the covariance of the residual. The residual of the measurement to the updated state and it covariance are then defined as:

$$\vec{r}_k = m_k - \mathbf{H}_k \vec{x}_k = (1 - \mathbf{K}_k \mathbf{H}_k) \vec{r}_k^{k-1} \text{ and} \quad (5.8)$$

$$\mathbf{R}_k = (1 - \mathbf{K}_k \mathbf{H}_k) \mathbf{V}_k = \mathbf{V}_k - \mathbf{H}_k \mathbf{W}_k \mathbf{H}_k^T. \quad (5.9)$$

The expected $(\chi^2_+)_k$ of the fit can then be updated as following:

$$(\chi^2_+)_k = \vec{r}_k^T \mathbf{R}_k^{-1} \vec{r}_k + (\chi^2_+)_k. \quad (5.10)$$

The final value of $\chi^2_+ = (\chi^2_+)_k$ describes how well the fit matches the measurements, given the expected uncertainties.

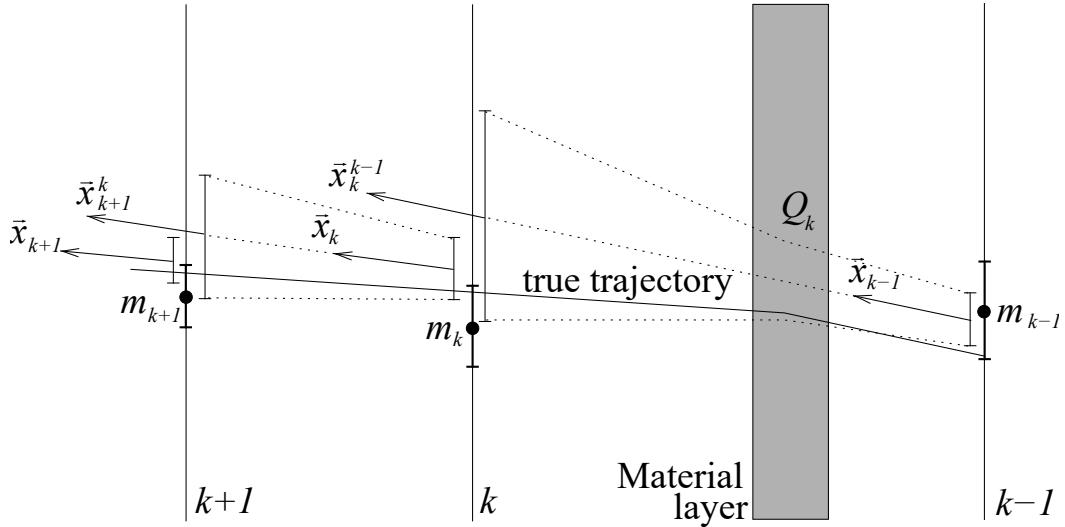


Figure 5.1: A schematic overview of the prediction and filter steps from node $k - 1$ to $k + 1$ for a Kalman filter for a track fit from reference [17]. The uncertainty of the position is indicated as error bars and the evolution with dotted lines.

In Figure 5.1 the evolution from a state at node $k - 1$ to $k + 1$ is shown for a track fit with straight-line extrapolation. The scattering of the particle in the material directly affects the uncertainty of the slope. By further extension of the track, the uncertainty of the position is also increased. In the filtering step, the predicted states are updated towards the measurements with an amount depending on the uncertainty of the measurement and the predicted state.

5.2.3 Performance evaluation

The performance of the Kalman filter is evaluated by comparing the track states to the true particle states known from the simulation. The residual of the track state with respect to the true state is defined as:

$$\vec{r}_{k;\text{true}} = \vec{x}_k - \vec{x}_{k;\text{true}}. \quad (5.11)$$

This residual is not to be confused with the residual of the measurement to the predicted state (see eq. 5.5). The width of the residual distribution should, for the best performance, be minimal. Half of the central 68% interval corresponding to a standard deviation for a Gaussian distribution is in the following regarded as resolution. The residual distributions show how well the track state vectors are estimated.

With the residuals, the pull for one component a of the state vector can be defined as:

$$\Delta_a = \frac{a_k - a_{k;\text{true}}}{\sigma_k}. \quad (5.12)$$

In this case, σ_k is the square root of the corresponding covariance matrix element of the filtered state (see eq. 5.6). The pull distributions give information on how well the uncertainties for the states are estimated. If the pull is standard normal distributed (i.e., the mean is zero and the standard deviation one), the state uncertainties are correctly predicted. The width of the pull distribution is in the following calculated as the sigma value of a Gaussian fit to the distribution and should be close to 1. Errors in the description of the measuring uncertainty, the process noise, and propagation lead to a distortion of the pull distribution.

Another way to evaluate the performance uses the χ^2 values of the track fit. In this thesis, the χ^2_+ values defined in eq. (5.10) are used. The distribution of χ^2_+ divided by the number of degrees of freedom should have a mean value of one. The number of degrees of freedom, also shortened as $ndof$, is given through the number of hits n_{hits} by $ndof = 2n_{\text{hits}} - 4$, where the factor and subtrahend come from the dimensionality of the measurements and state vector respectively. In addition, the distribution of the probability of having a χ^2 value larger than χ^2_+ for the corresponding number of degrees of freedom should be flat. This probability is in the following denoted as $\text{Prob}(\chi^2)$.

5.3 Application to the LumiTracker

For the LumiTracker, measurements of a track are fitted and then extrapolated to the interaction point to estimate the primary vertex location of the corresponding particle. The large distance from the interaction point, particle scattering in material layers, and secondary vertices make an accurate estimate of the origin vertex position difficult, with the distance being the main contributing factor. The possible path of a particle is schematically depicted in Figure 5.2 to give an overview of the situation for the LumiTracker. The material layers between the LumiTracker and IP are modeled as one unique block of material. Important to note is that the VELO elliptical head visible in Figure 3.1 was not in the simulation, and therefore part of the material is missing between the LumiTracker and IP.

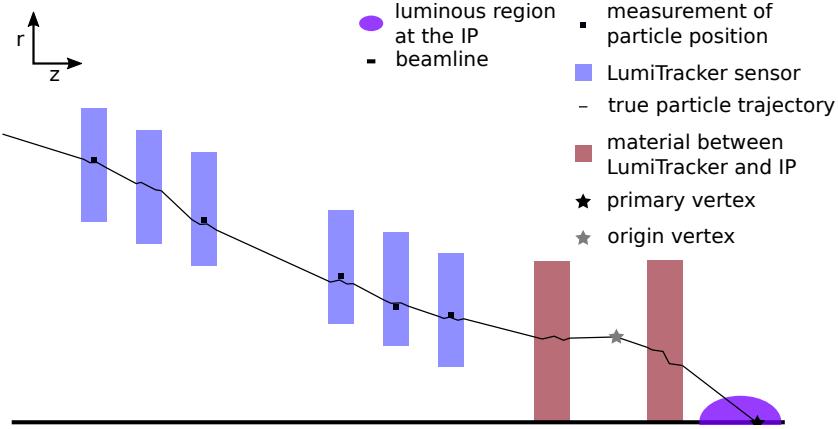


Figure 5.2: Schematic display of a particle from material which traversed the LumiTracker. The radial distance is denoted as r . Many other scenarios are also possible. The schematic is not to scale.

5.3.1 Scattering

A charged relativistic particle traversing a material is scattered multiple times by small angles. Most scattering is due to Coulomb scattering on nuclei and hence it is also called multiple Coulomb scattering. Through the central limit theorem, many small scattering angles lead to an overall Gaussian scattering distribution. The occurrence of less frequent larger scattering angles results in non-Gaussian tails. The scattering is well described by the theory of Molière [41]. For many applications a Gaussian approximation is sufficient. For the Gaussian, a width θ_0 is used given by the root squared error of central 98 % of the projected scattering angle distribution. The value is often obtained from a parametrization suggested by Highland [42]. This parametrization has been improved by Lynch Dahl [43] and is given by:

$$\theta_0 = \frac{13.6 \text{ MeV}}{\beta cp} z \sqrt{\frac{x}{X_0}} \left(1 + 0.038 \ln \left(\frac{x}{X_0} \right) \right), \quad (5.13)$$

where βc is the velocity, p the momentum, and z the charge of the particle. The thickness of the scattering material in radiation lengths is denoted as x/X_0 .

One of the main properties of a Kalman filter is that it can account for Gaussian noise introduced between the measurements. As the multiple Coulomb scattering for particles with the same momentum results in nearly Gaussian distributed scattering angles, it can be taken into account correctly if the momentum of a particle is known. Because the momentum of particles is not known for the LumiTracker, a reasonable estimate is needed for the particles of interest. In this case, particles

from the luminous region. These often have higher momentum than particles from material and hence scatter less, according to equation (5.13). The particle momentum distribution can be seen for particles from material and the luminous region in the Figure 5.3. For thin detectors like the LumiTracker, the energy loss is

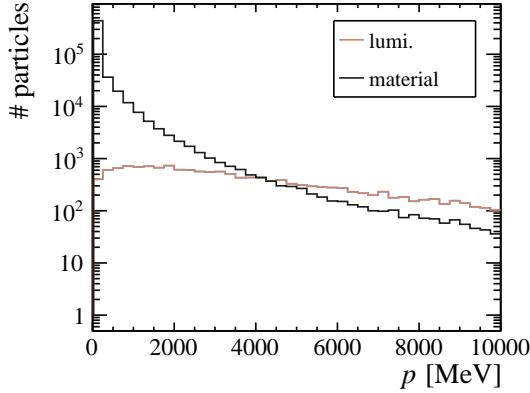


Figure 5.3: The momentum distribution for particles from the luminous region and from material.

negligible compared to the traversing particle's energy. Hence it is assumed that its momentum does not vary with the interaction with LumiTracker planes. The overall amount of scattering occurring in every plane is assumed to be the same, as they are identical. For the material between the LumiTracker and IP, much more scattering is expected as much more material is present. The fit is therefore divided into two parts: The fit inside the LumiTracker and the following extrapolation to the IP, which is tuned separately.

The parametrization for the width θ_0 of the projected scattering angle distribution from (5.13) is further simplified for the LumiTracker and then used to determine the noise matrix. The approximate θ_0 is calculated as:

$$\theta_0 \approx C \frac{\sqrt{x'}}{\beta p} \Leftrightarrow \theta_0 \frac{\beta p}{\sqrt{x'}} \approx C, \quad (5.14)$$

where C is a constant factor for all particles and $x' = \sqrt{1 + t_x^2 + t_y^2}$ with the slopes t_x and t_y . To see how well this approximation holds inside the LumiTracker for the simulated particles the distribution of $\theta \beta p / \sqrt{x'}$ is plotted in Figure 5.4a. The projected scattering angle is calculated for each sensor as:

$$\theta = \arctan \left(\tan \left(\frac{\arctan(t_{x,\text{after}}) - \arctan(t_{x,\text{before}})}{\sqrt{1 + t_{y,\text{before}}^2}} \right) \right), \quad (5.15)$$

where $t_{x,\text{after}}$ denotes the “true” slope in x direction after the sensor and $t_{x,\text{before}}$ the “true” slope before the sensor in x and y direction respectively. The “true” slopes are calculated from the true hit positions in two neighboring planes as is depicted in Figure 5.6. As there are no true hits before the 0th sensor or after the 5th sensor the “true” slope before and after the sensor respectively can not be calculated. Therefore the scattering inside sensors 0 and 5 is can not be determined, but is assumed to be the same as for the other planes.

It can be seen that the overall distribution without protons is close to a Gaussian distribution, except for larger tails. The value of $C \approx 0.76$ MeV is obtained from the Gaussian fit to the distribution. The tails come mostly from electrons, as can be seen in Figure 5.4b. The distribution for all other particles is relatively close except for protons. For them, the distribution is very narrow. This unexpected low scattering for protons is not understood and might be related to the simulation. Hence they were ignored for any tuning of the Kalman filter.

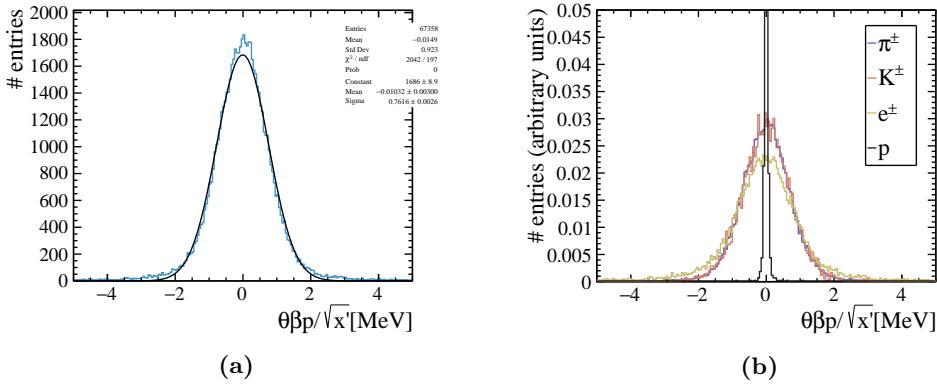


Figure 5.4: The distribution of $\theta\beta p/\sqrt{x'}$ for all planes and for all particles which are not protons with a Gaussian fit (a) and for the most common particle types (b).

The distribution of $\theta\beta p/\sqrt{x'}$ can also be used to determine if the assumption of the same scattering behavior in different sensor planes is sensible. Therefore the distribution is plotted for the central four sensors in Figure 5.5a. No notable differences between the four distributions are visible and therefore a similar scattering is assumed for all sensors. The distribution of $t_{x,\text{after}} - t_{x,\text{before}}$ shown in Figure 5.5b. It visualizes how much the slope for particles changes while traversing one plane. As expected the distribution is highly non-Gaussian.

The noise matrix for a particle traversing a homogeneous material of the thickness l_z can be calculated from the projected scattering angle θ_0 and the track slopes t_x

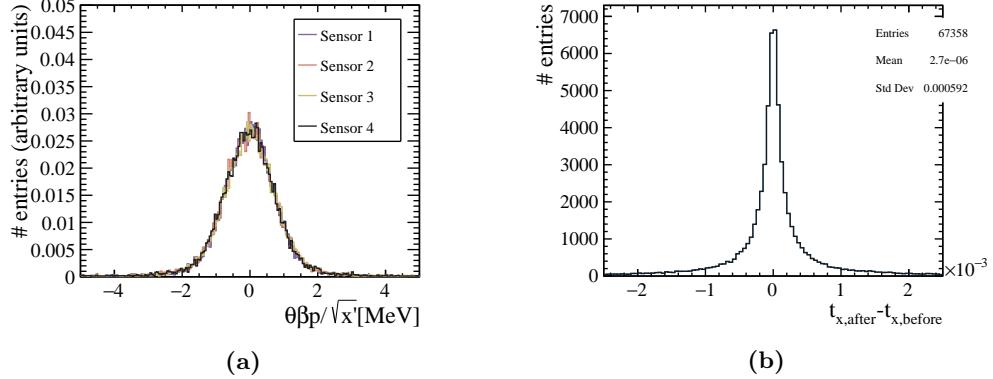


Figure 5.5: The distribution of slope changes in $\theta\beta p/\sqrt{x'}$ inside sensor planes for all particles which are not protons for each sensor (a) and the distribution of the changes in t_x (b).

and t_y . According to reference [44], it is given by:

$$\mathbf{Q} = \begin{pmatrix} \langle P_3, P_3 \rangle \frac{l_z^2}{3} & \langle P_3, P_4 \rangle \frac{l_z^2}{3} & \langle P_3, P_3 \rangle \frac{l_z}{2} & \langle P_3, P_4 \rangle \frac{l_z}{2} \\ \dots & \langle P_4, P_4 \rangle \frac{l_z^2}{3} & \langle P_3, P_4 \rangle \frac{l_z}{2} & \langle P_3, P_4 \rangle \frac{l_z}{2} \\ \dots & \dots & \langle P_3, P_3 \rangle & \langle P_3, P_4 \rangle \\ \dots & \dots & \dots & \langle P_4, P_4 \rangle \end{pmatrix}, \quad (5.16)$$

with the matrix entries:

$$\langle P_3, P_3 \rangle = \theta_0^2(1 + t_x^2)(1 + t_x^2 + t_y^2), \quad (5.17)$$

$$\langle P_3, P_4 \rangle = \theta_0^2 t_x t_y (1 + t_x^2 + t_y^2), \text{ and} \quad (5.18)$$

$$\langle P_4, P_4 \rangle = \theta_0^2 (1 + t_y^2)(1 + t_x^2 + t_y^2). \quad (5.19)$$

Because the thickness of the LumiTracker planes is with 0.4 mm small compared to the distance between planes, the scattering is approximated as occurring directly in the measurement plane. The resulting noise matrix then only has the four entries for the slopes. How the Kalman filter is modeled and how the true trajectory of the particle inside the LumiTracker is expected to look is schematically depicted for two sensor planes in Figure 5.6.

5.3.2 Measurements

Measurements are only done at the LumiTracker sensors. A hit is simulated with a resolution of 0.015 mm. Hence the corresponding covariance matrix for the

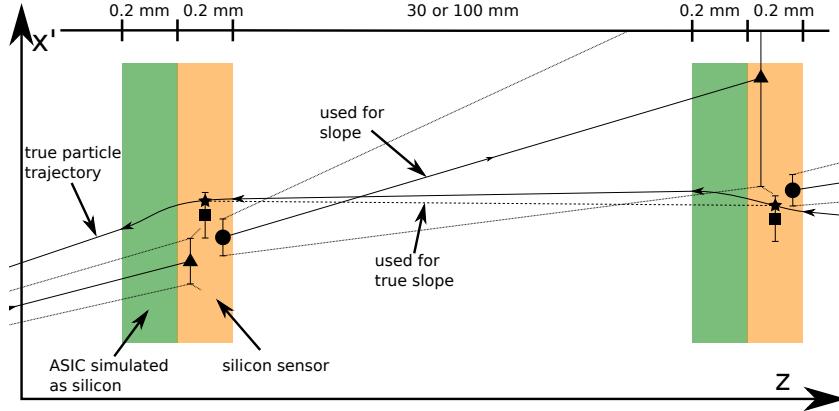


Figure 5.6: A more detailed schematic description of the model inside the Lumi-Tracker between two different sensor planes with measurements. The “true” track slope used is obtained from two true hit positions and is slightly different from the slope of the true particle trajectory. The arrow on the true particle trajectory indicates the direction of the particle, and on the line segment, the direction of the Kalman filter. True hit position (star), measured hit position (square), predicted state (triangle), and filtered state (circle) are indicated.

measurement used in the filter step is:

$$\mathbf{V}_k = \begin{pmatrix} 0.015^2 & 0 & 0 & 0 \\ 0 & 0.015^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (5.20)$$

As the measurements lie in the same x - y plane as the states the projection of the state into the measuring space is easily done with the 2×4 projection matrix:

$$\mathbf{H}_k = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad (5.21)$$

which is also used in the filter step.

5.3.3 Extrapolation to the IP

As the position of the primary vertices is of interest, the Kalman filter begins the fit at the measurement furthest away from the interaction point. It then progresses towards it as the estimate improves with the number of used measurements. The way to estimate the z position of the primary vertex starting from the state within the LumiTracker nearest to the interaction point is partly depicted in Figure 5.2.

A straight-line extrapolation is again appropriate as effectively no magnetic field is present. As the primary vertex is expected to be close to the beamline, the extrapolation is done to the closest position to the beamline.

As more material is traversed, more scattering is expected. Other than inside the LumiTracker, the scattering takes place in the material of VELO modules, VELO tank, and beam pipe. The various material between the IP and the LumiTracker is treated as a single scattering plane for the purpose of the scattering estimation. Two additional parameters η_{xx} and η_{xtx} are introduced to describe the influence of the distribution of material on the noise matrix. The noise matrix with these parameters is given as follows:

$$\mathbf{Q} = \begin{pmatrix} \langle P_3, P_3 \rangle d_z^2 \eta_{xx}^2 & \langle P_3, P_4 \rangle d_z^2 \eta_{xx}^2 & \langle P_3, P_3 \rangle d_z \eta_{xtx} & \langle P_3, P_4 \rangle d_z \eta_{xtx} \\ \dots & \langle P_4, P_4 \rangle d_z^2 \eta_{xx}^2 & \langle P_3, P_4 \rangle d_z \eta_{xtx} & \langle P_3, P_4 \rangle d_z \eta_{xtx} \\ \dots & \dots & \langle P_3, P_3 \rangle & \langle P_3, P_4 \rangle \\ \dots & \dots & \dots & \langle P_4, P_4 \rangle \end{pmatrix}. \quad (5.22)$$

The parameter η_{xx} and η_{xtx} have a limited range of $[0, 1]$. In the limit $\eta_{xx} \rightarrow 0$ and $\eta_{xtx} \rightarrow 0$, it is the same form as the one used inside the LumiTracker and describes a thin scattering medium directly before the predicted state. The other limit of $\eta_{xx} \rightarrow 1$ and $\eta_{xtx} \rightarrow 1$ describes a thin scattering layer directly after the last state. In general if $\eta_{xx} = \eta_{xtx}$ holds a thin scattering layer $\eta_{xx} \times 100\%$ of the way before the predicted state is described.

5.4 Tuning the fit inside the LumiTracker

The tuning is performed for reconstructible particles from the luminous region. Different cases are considered and compared. First, the Kalman filter is tuned using the momentum βp from the simulation for each particle in section 5.4.1. This allows determining the best value for the constant C in equation (5.14). This case should also yield the best possible results and is therefore used for comparison. As the momentum information is not available, different approaches for this case were considered in section 5.4.2. The presented approaches include the high momentum limit of a straight-line fit, using an optimized constant momentum βp , and estimating the momentum based on the χ^2 -values obtained from the straight-line limit.

5.4.1 Case with momentum

In this case, the true momentum and mass of the particle are used to estimate the value of βp . This is the ideal case as it is only possible with information from the

simulation but represents the best benchmark for tuning the scattering parameters. The case without true momentum only consists of finding a good value for βp . The Kalman filter was performed for different values of C , and the performance was compared. In this case, the width of the pulls (see eq. (5.12)) for the four different state variables was determined for each C value by a Gaussian fit. The width of each fit σ_{fit} is then plotted against the value of C . This is depicted in Figure 5.7. The true state must be known to calculate the residuals. This is not the case for the last state of the fit inside the LumiTracker. Therefore the second to last state was used. As stated in section 5.2.3 the width of all pulls should be one. This is the

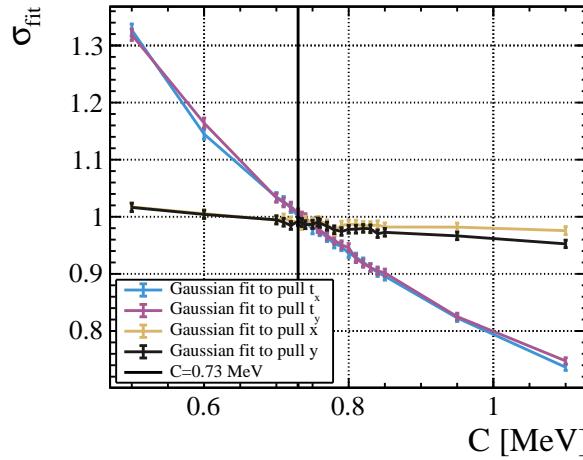


Figure 5.7: The width of the pulls for the four state variables against the different values of C for reconstructible particles from the luminous region.

case for $C = 0.73 \text{ MeV}$ within the uncertainties. It is, therefore, the chosen value. That the graphs do not cross at this value is caused by a slight increase of the σ_{fit} value due to the larger tails for the scattering distribution of electrons (see 5.4b). The corresponding χ^2 distribution is shown in the Figure 5.8 together with the χ^2 probability distribution. The χ^2 distribution has a mean of 1.08 and is therefore slightly above the expected value of one. This deviation is again related to the tails of the scattering angle distribution, as for them, the Kalman filter underestimates the scattering, which leads to increased χ^2 values. For the probability distribution, it can be seen that the distribution is mostly flat. The spike in the probability distribution for low probability values is caused by the fact that the scattering angle distribution in Figure 5.4a has non-Gaussian tails. This spike was also observed in Figure 14b in [44] and is missing for a Gaussian scattering angle distribution as shown in Figure 14a in the same reference.

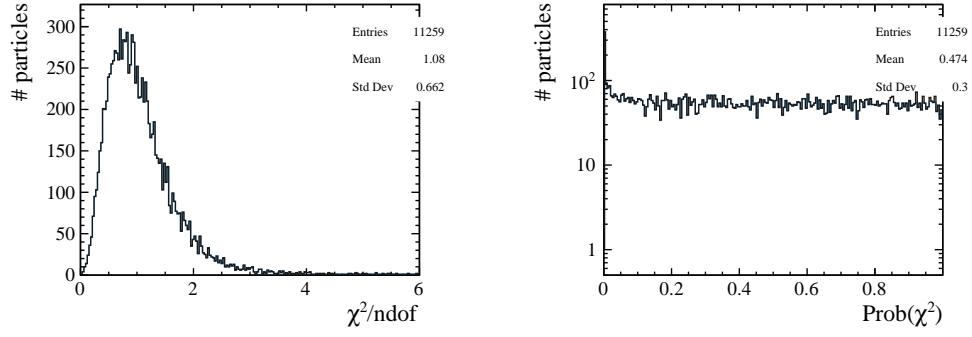


Figure 5.8: The χ^2 distribution (left) and the corresponding probability distribution of having a larger χ^2 as the given for the given number of degrees of freedom (right) for fits of reconstructible particles from the luminous region for $C = 0.73 \text{ MeV}$.

The resulting pull and residual distributions for $C = 0.73 \text{ MeV}$ are shown in Figure 5.9. The Gaussian fits for the pull distributions match the distributions. The Gaussian shape can, for all pulls, be seen with only slight deviations. Most notable for the slope pulls. The residual distributions are not required to have a Gaussian shape. Nevertheless, the position residuals resemble a Gaussian shape. Noteworthy is that the standard deviation is with around 0.0125 mm significantly under the used smearing of 0.015 mm . This means that the fit is as expected, improving the estimate of the true hit position compared to just using the single measurements. The determined resolution in x and y direction are with $(81.0 \pm 1.1)\%$ and $(81.8 \pm 1.1)\%$ of the hit resolution respectively also compatible with this statement. The residual for the slopes resembles the distribution of the slope changes in a sensor plane (see 5.5). This is expected because even if the Kalman filter updates every state position to the true position, the scattering would not be accounted for (see 5.6). Altogether the performance of the Kalman filter inside the LumiTracker with $C = 0.73 \text{ MeV}$ is as expected. The Kalman filter with true momentum is later also referred to as *kalman true momentum*.

5.4.2 Case without momentum

No magnetic field is present which would allow for an estimation of momentum based on the radius of curvature of the track. Different approaches for a momentum estimate have been considered, and a selected subset is in the following presented. For all of them $C = 0.73 \text{ MeV}$ is used as determined in the previous section.

5.4 Tuning the fit inside the LumiTracker

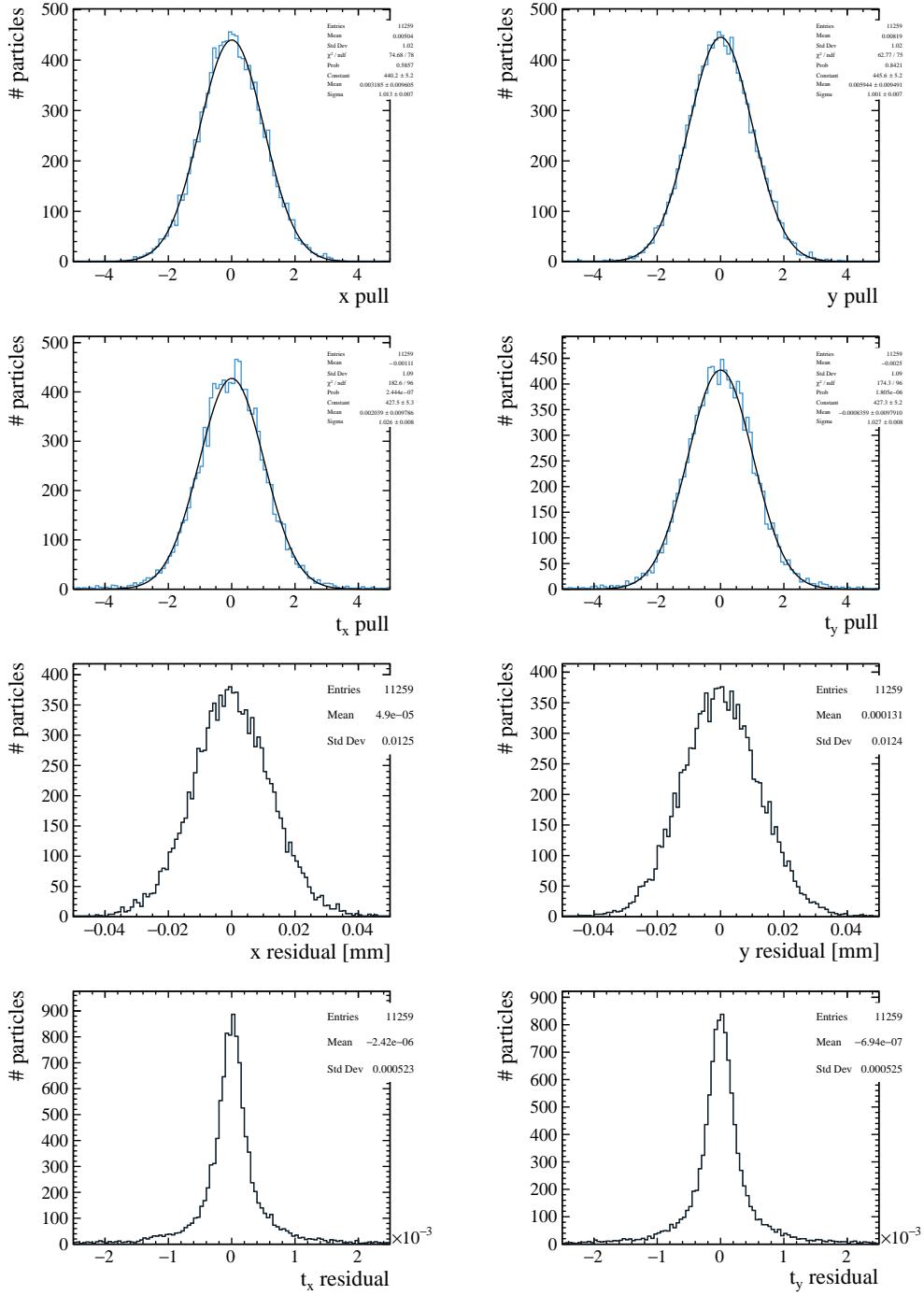


Figure 5.9: All four pull and residual distribution for the chosen $C = 0.73$ MeV for reconstructible particles from the luminous region. The true momentum is used for the Kalman filter.

Straight-line limit

The Kalman filter is compared to the even simpler approach of a straight-line fit. The process noise is removed to achieve a straight-line fit. This is the case for $\beta p \rightarrow \infty$. The Kalman filter assumes, in this limit, straight-line trajectories and becomes equivalent to a straight-line fit. This limit is used to simulate the results of a straight-line fit without a separate implementation. This approach is also referred to as *linear* and is by construction most accurate for very high momentum tracks. As particles from the luminous region have a finite momentum, scattering does occur, and therefore a finite value for βp is expected to yield better results.

Constant momentum estimate

One obvious approach is to replace the unknown factor βp with a constant factor for all particles. A similar approach as in section 5.4.1 is used to choose the parameter βp . Therefore all reconstructible particles from the luminous region were fitted with different values of βp . The width of the Gaussian fits of the pulls against the used value of βp is depicted in Figure 5.10. That a constant momentum estimate is not

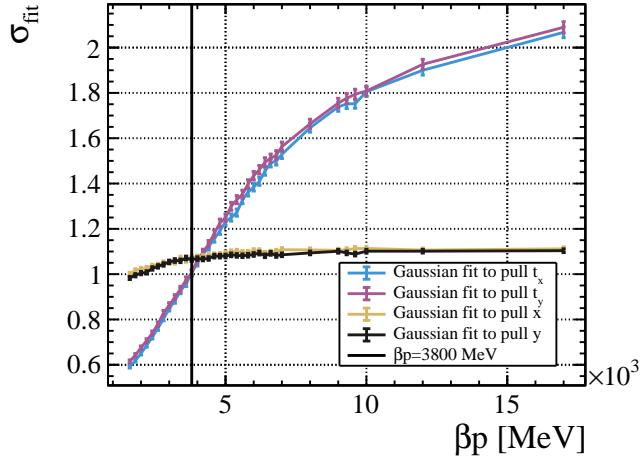


Figure 5.10: The width of the pulls for the four state variables against the different values of βp for a constant value of $C = 0.73$ MeV. The chosen value of $\beta p = 3800$ MeV is marked with a vertical line.

optimal can already be seen, as the width of the positions and slope do not approach one at the same time. The chosen value of $\beta p = 3800$ MeV is a compromise and

minimizes the overall deviation from one. This approach is in the following denoted as *kalman 3800*.

Dynamic momentum estimate

As an estimate of the momentum for each particle based on the measurements should allow for a smaller overall deviation from the true momentum this is also attempted. Firstly the dependency between the momentum and the slope of a straight line between the first and last measurements is evaluated. The mean true momentum βp is depicted in Figure 5.11 as a function of these slopes t_x and t_y . It can be seen that the acceptance of the detector is directly visible. As no dependence besides that is visible, it was concluded that no easy dynamic estimate could be derived based on the track slope. In addition, a bias for different slopes would be introduced, which would probably lead to an unwanted dependence on the origin vertex location along the z axis.

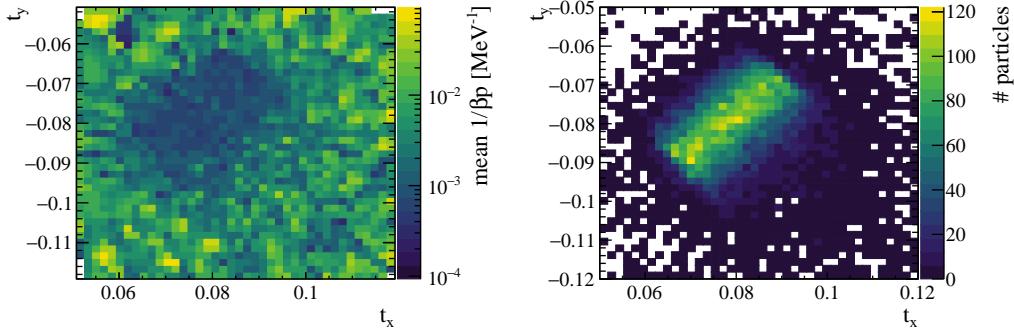


Figure 5.11: The dependence of the mean of $\frac{1}{\beta p}$ on the track slope (left) and the distribution of track slopes (right) for reconstructible particles.

As the slope is not a suitable choice, it was instead considered to estimate the momentum based on the overall scattering of the particle. The χ^2 values of straight-line fits do increase on average with the amount of scattering. An additional straight-line fit needs to be performed to get the χ^2 value. The smearing of the hits also contributes to the value of χ^2 . Hence it is expected that for particles with high momentum, the momentum can not be estimated well, as the deviation from a straight line is dominated by the smearing. In Figure 5.12 the mean of βp and $1/(\beta p)$ are depicted over the full χ^2 range. A clear dependence between χ^2 and the mean momentum is visible. Using the mean of βp tends to favor accurate estimates for small scattering particles and is also referred to as *kalman improved*

linear. The other approach of using the mean of $1/(\beta p)$ favors particles with low momenta instead and is referred to as *kalman improved linear inv*. The depicted splines are used to estimate the momentum βp from the χ^2 value of the straight-line fit. This estimated momentum is then used for the Kalman filter. It can be seen that the splines also pick up the statistical fluctuations. As this only concerns a small percentage of particles, it is tolerated in this first attempt.

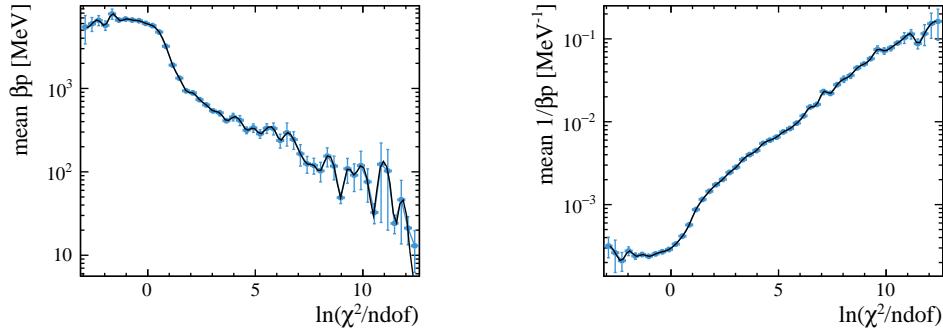


Figure 5.12: The dependence of the mean of βp (left) and of $\frac{1}{\beta p}$ (right) on the χ^2 value of the linear fit. The spline later used to get the momentum estimate is in both cases indicated through a black line.

5.4.3 Comparison

The performance for the fit inside the LumiTracker is compared between all different approaches in the following. As the distributions for the x and y direction are nearly the same here, this is only shown for the x direction. The χ^2 probability distribution for the given number of degrees of freedom is depicted in Figure 5.13 for all introduced approaches. The probability distribution for the *kalman 3800* shows a larger spike for low probabilities than the *kalman true momentum*. The distribution is nearly flat, apart from the spike and a slight slope. The same is the case for the *linear* approach, but with a larger spike and slope. The *kalman improved linear inv* seems to deviate the most from a flat distribution, as low probabilities are underrepresented. The number of particles increases in the direction of higher probabilities until it reaches around 0.35 its maximum and then declines slowly. In the range of 0.55 to 1, it is closest to the distribution of the *kalman true momentum*. The *kalman improved linear* is located for probabilities over 0.3 between the *linear* and *kalman 3800*. For low probabilities, only a small spike exists, but the probabilities between 0.05 and 2 are instead overrepresented.

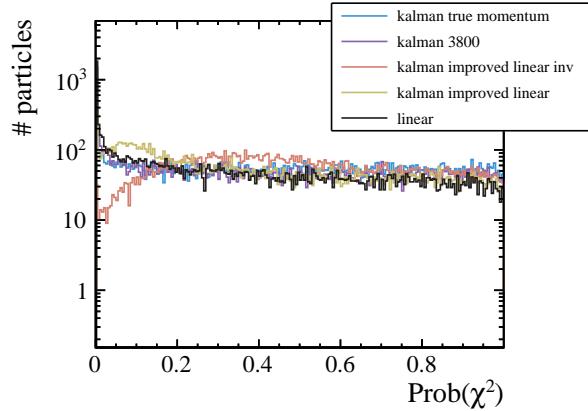


Figure 5.13: The χ^2 probability distribution for all introduced approaches for reconstructible particles from the luminous region.

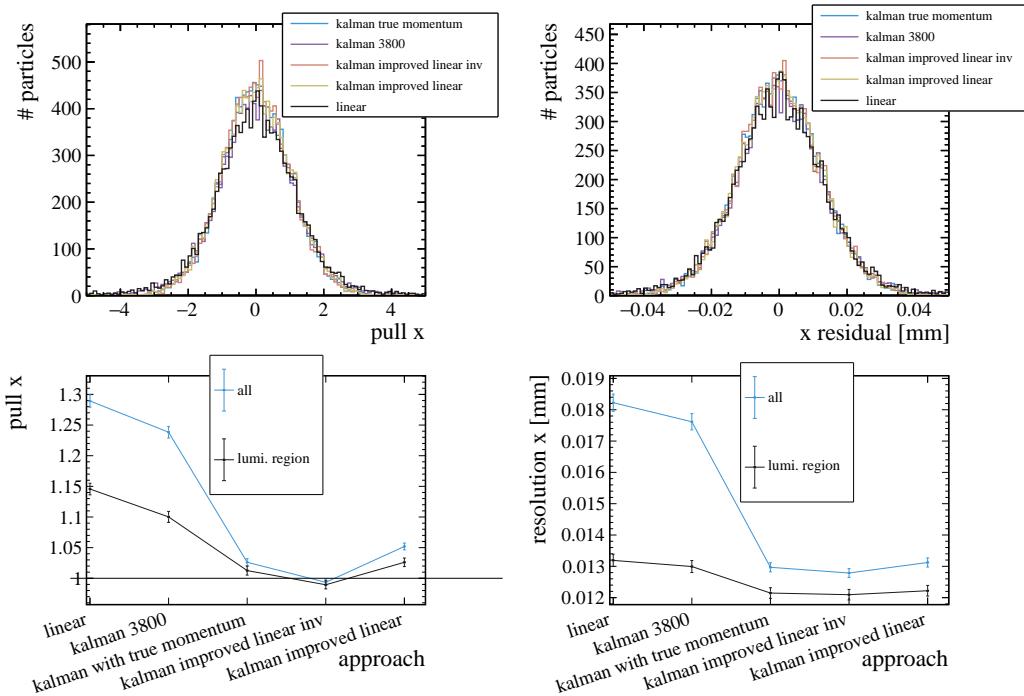


Figure 5.14: The pull x distribution (top left) and x residual distribution (top right) for all approaches for reconstructible particles from the luminous region. As all distributions are close together, the sigma of a Gaussian fit to the pull distributions is also shown for all approaches for all particles and for particles from the luminous region in the bottom left. The same is done for the resolution (see 5.2.3) of the residual distributions in the bottom right.

5 Development of the track fit

The pull and residual distributions for the position x are depicted in Figure 5.14 for all approaches. It can be seen that pull and residual distributions are close together for all approaches. That the *kalman 3800* and *linear* approach perform worse is only barely visible. The resolution in x and the x pull distribution width are also shown for all approaches to make the difference more visible. It can be seen that the approaches with a variable momentum estimate outperform the approaches with the same momentum for all particles. The best overall performance is achieved by the *kalman improved linear inv* with only a negligible difference from the other approaches with dynamic momentum estimate.

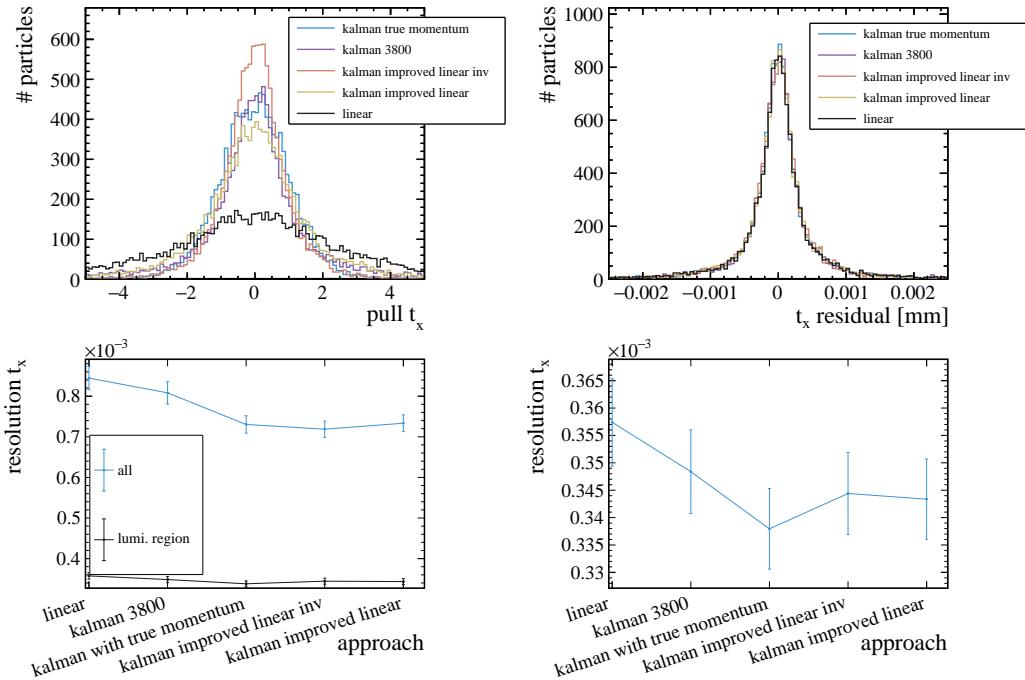


Figure 5.15: The pull t_x distribution (top left) and t_x residual distribution (top right) for all approaches for reconstructible particles from the luminous region. As all distributions for the residuals are close together, the resolution (see 5.2.3) of the residual distributions is also shown for all approaches in the bottom left for all particles and particles from the luminous region. As the difference in performance for particles from the luminous region is too small to clearly visible only the resolution for particles from the luminous region is shown again in the bottom right.

The pull and residual distributions for the slope t_x are depicted in Figure 5.14 for all approaches. This time, the differences between the pull distributions are clearly visible. The *linear* approach underestimates, as expected, the uncertainty on the

slope as it assumes all measurements come from a straight line. The approaches *kalman 3800* and *kalman improved linear* are comparable. Both have too heavy tails and are relatively close to each other, with the latter having a slightly more Gaussian shape. The *kalman improved linear inv* does not have those heavy tails but also suffers from overestimation, which results in the central distribution being too narrow. The residual distributions are again close together, and therefore, the resolution in t_x is separately depicted for all approaches. As the resolution for all and only particles from the luminous region are much more different than the differences between the approaches for particles from the luminous region, the evaluation of the performance for those is still difficult. Hence the resolution for particles from the luminous region is also shown separately. It can be seen that the approaches with a dynamic momentum estimate perform again better than the other two for all and also for particles from the luminous region. The second-best approach is *kalman improved linear* closely followed by *kalman improved linear inv*.

Overall the Kalman filter with the true momentum yields the best performance, followed by the approaches which use a momentum estimate based on the χ^2 of a linear fit. It was decided to use the *kalman improved linear inv* approach, but the *kalman improved linear inv* approach would yield similar results.

5.5 Tuning the extrapolation to the IP

Here the tuning of the extrapolation of the last state inside the LumiTracker to the beamline is presented. Because in addition to the previous parameter C for this case denoted as C_{IP} , now also the parameters η_{xx} and η_{xtx} need to be tuned (see 5.3.3), doing the manual tuning is complex. A cost function is minimized instead. A minimization like in reference [38] was attempted, but with a 4D-Gaussian instead of a 2D-Gaussian. This attempt had numeric precision problems and did not converge. A cost function inspired by Ref. [35] was finally used. It was then minimized using the Migrad algorithm from Minuit2 [45]. The cost function used is:

$$y(\vec{b}) = \sum_{p \in P} \log \left(\frac{\vec{r}_{\text{true}}^T \mathbf{W}^{-1} \vec{r}_{\text{true}}}{d} \right), \quad (5.23)$$

where $\vec{b} = (C_{\text{IP}}, \eta_{xx}, \eta_{xtx})$ are the parameters to tune, $d = 4$ the dimension of the state vector, \vec{r}_{true} the residual of the predicted state to the true state and \mathbf{W}^{-1} the inverse of the expected covariance matrix. P denotes the subset of reconstructible particles from the luminous region. For the tuning the true states nearest to the IP are extrapolated to the beamline. These are replaced for the final track fit by the states obtained from the fit inside the LumiTracker. The residuals are determined

by comparing the predicted state nearest to the beamline to the true state at the origin vertex. The formula used to predict z position of the nearest state to the beamline is:

$$z_{\text{pred}} = z + \frac{t_x(x_{\text{beamline}} - x) + t_y(y_{\text{beamline}} - y)}{t_x^2 + t_y^2}, \quad (5.24)$$

where x_{beamline} and y_{beamline} describe the position of the beamline. The extrapolation is then performed as previously described. From the predicted state and its covariance matrix, the error is then propagated by using the relation (5.24). The tuned values for the parameters are listed in Table 5.1. The pull z and z resolution distribution

Table 5.1: The tuned parameters for the estimation of the noise for the extrapolation to the IP.

C_{IP}	η_{xx}	η_{xtx}
6.6077	0.323139	0.286861

obtained from the tuned extrapolation combined with the fit inside the LumiTracker for the *kalman improved linear inv* and *kalman true momentum* approach are depicted in Figure 5.16. As can be seen, the approach *kalman true momentum* does perform much better when the estimated uncertainty is used to improve the distribution. For the *kalman improved linear inv* approach, only a slight improvement is visible. This is caused by what is already discussed in section 5.4.2, that the momentum estimate based on the χ^2 approach can not differentiate between high momentum particles well, resulting in similar estimates for the uncertainty of the z position. The Figure 5.16 also gives an indication of the per-track resolution achievable for the luminous region, which is of the order of a few millimeters. The depicted distributions include many strongly scattering low momentum particles from the luminous region (see Figure 5.3), which increase the overall width of the residual distribution. Selecting only particles with high momentum is expected to make the distribution more narrow. Applying a χ^2 cut on the initial straight-line fit has a similar effect for tracks as it is the same as selecting tracks based on their estimated momentum. It can also be required that the estimated location is close to the beamline to improve the resolution further. It is therefore expected that the final resolution is significantly lower than shown in this thesis.

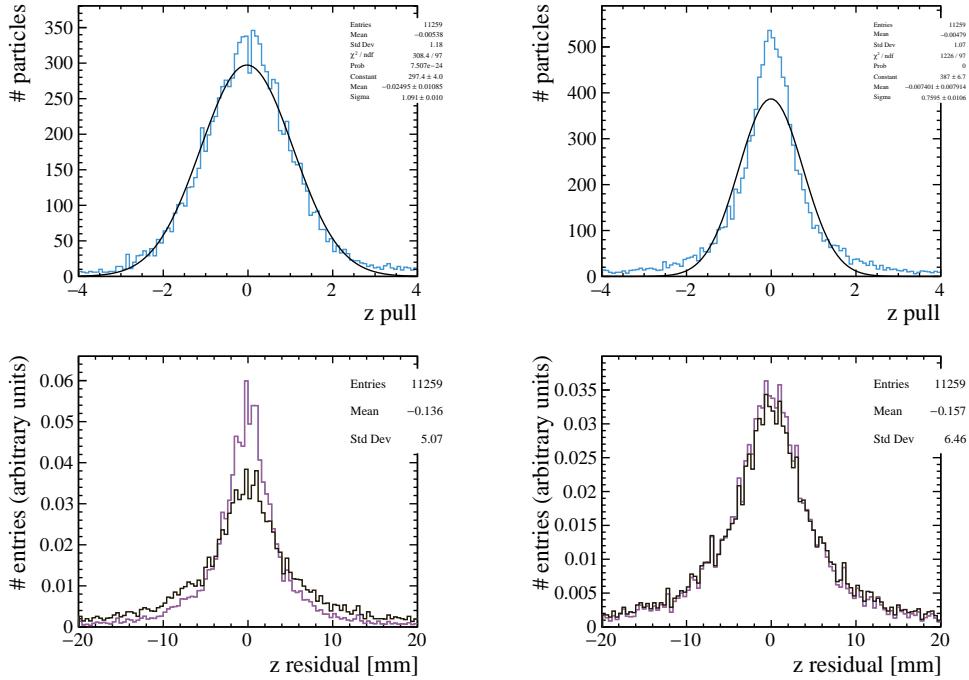


Figure 5.16: The estimated pull z distribution (top) and z residual distribution (bottom) for the *kalman true momentum* (left) and *kalman improved linear inv* (right) approach. Indicated with the purple line is the distribution weighted with $1/\sigma_z$. The statistic box in the resolution plots is for the weighted distribution.

5.6 Final results

The previous results were achieved by the Kalman filter described sections 5.2 and 5.3 and performed in an independent offline analysis. As this implementation uses the full 4×4 matrices for calculations, it is relatively computing expensive. As the off-diagonal entries in the covariance matrix for states inside the LumiTracker are around two magnitudes smaller than the diagonal entries, they are deemed negligible. Therefore the *simplified Kalman filter* filters separately in x and y . This simplified Kalman filter is only implemented in Allen and also uses instead of double precision only single precision. As such, the distributions for the Kalman filter used in the offline analysis are also depicted as a comparison. In both cases, the reconstructed tracks from the luminous region are used for the fit. The pull z and z residual distributions for the fit with Allen and offline are depicted in Figure 5.17. It can be seen that even though the fit was simplified and also only performed with single-precision, the distribution is relatively close to the offline fit. Therefore these simplifications are deemed sensible.

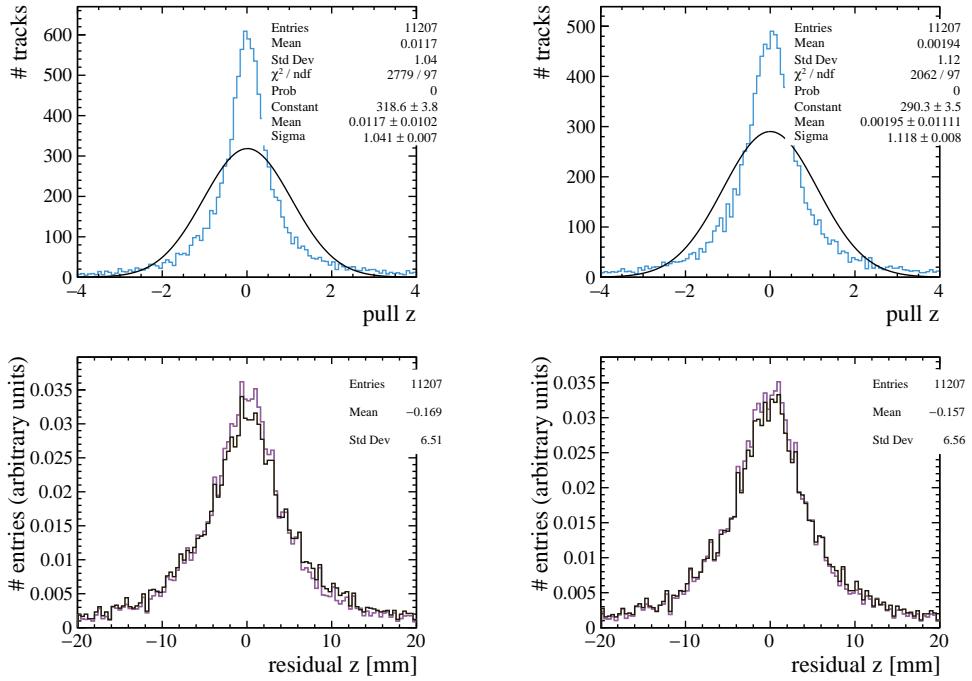


Figure 5.17: The estimated pull z distributions (top) and z residual distributions (bottom) within Allen(left) and offline (right) for reconstructed tracks from the luminous region. Indicated with the purple line is the distribution weighted with $1/\sigma_z$. The statistic box in the resolution plots is for the weighted distribution.

6 Summary and Outlook

A real-time track reconstruction for the proposed LumiTracker detector was implemented in Allen. The reconstruction can be performed on two Intel i7-8086k CPUs in real-time as the combined throughput is over 30 MHz. Execution on GPU is currently not recommended as the current throughput is low compared to execution on CPU. The physics performance is independent of the device used for the execution. The LumiTracker implementation for the Search by Triplet algorithm was chosen for the pattern recognition because of its better computing performance. The pattern recognition achieves a tracking efficiency of $(96.84 \pm 0.16)\%$ for particles from the luminous region and of $(80.69 \pm 0.28)\%$ for all reconstructible particles. The hit purity is over 99.9 % and the clone and ghost rate under 1 %. The fraction of reconstructed tracks from material interaction is $(28.93 \pm 0.35)\%$. It was observed that the tracking efficiency strongly depends on the particle momentum. Dependence of the tracking efficiency on the origin vertex z location was also observed. When studying the performance for different occupancies in the sensors of the LumiTracker, it was concluded that a different search window size for the second sensor could result in a better physics performance. The integration of the Kalman filter with a constant momentum in the Search by Triplet algorithm is expected to yield also slightly better results than the current implementation, as the predicted position in the next plane would be more accurate.

For the track fit, a Kalman filter was implemented. Different approaches have been considered to estimate the momentum. Estimating the momentum based on a previous straight-line fit yields the best results. The extrapolation to the IP is tuned separately and uses a simplified material description. The per-track resolution for the luminous region is expected to be of the order of a few millimeters. The resolution could be improved by applying additional requirements for tracks used in the extrapolation to the IP. A possibility is only to use tracks where the estimated location in the x - y plane or the χ^2 value of the straight-line fit is under a certain threshold.

The presented implementations can be further developed. The performance of the implementation on GPU can be improved for a possibly even greater throughput than on a CPU. For a standalone sequence on GPU, the following changes are promising for a significant improvement of the throughput: Assigning more events

6 Summary and Outlook

to a thread block for LumiTracker algorithms and distributing the workload equally among threads. For this, the used data structures for the LumiTracker need to be modified. In addition, more events should be processed per slice to keep the overhead in copying data to the GPU low. The studies should also be repeated with the entire updated detector geometry. Reconstructed tracks and other output is currently written to JSON-like files for use in the offline analysis performed in this work. The determination and propagation of the luminosity counters information have been recently implemented in the LHCb trigger. The output of the LumiTracker sequence can be integrated within the same framework for offline luminosity determination.

Bibliography

- [1] P. Kopciewicz et al. “The upgrade I of LHCb VELO—towards an intelligent monitoring platform.” In: *Journal of Instrumentation* 15.06 (June 2020), pp. C06009–C06009. DOI: 10.1088/1748-0221/15/06/c06009. URL: <https://doi.org/10.1088%2F1748-0221%2F15%2F06%2Fc06009>.
- [2] S Akar et al. *Review Document: Low Level Trigger (LLT)*. Tech. rep. -. Geneva: CERN, May 2014. URL: <https://cds.cern.ch/record/1700272>.
- [3] LHCb Collaboration. *LHCb Upgrade GPU High Level Trigger Technical Design Report*. Tech. rep. Geneva: CERN, May 2020. DOI: 10.17181/CERN.QDVA.5PIR. URL: <https://cds.cern.ch/record/2717938>.
- [4] Esma Mobs. “The CERN accelerator complex - August 2018. Complexe des accélérateurs du CERN - Août 2018.” In: (Aug. 2018). General Photo. URL: <http://cds.cern.ch/record/2636343>.
- [5] Lyndon Evans and Philip Bryant. “LHC Machine.” In: *Journal of Instrumentation* 3.08 (Aug. 2008), S08001–S08001. DOI: 10.1088/1748-0221/3/08/s08001. URL: <https://doi.org/10.1088/1748-0221/3/08/s08001>.
- [6] Burkhard Schmidt. “The High-Luminosity upgrade of the LHC: Physics and Technology Challenges for the Accelerator and the Experiments.” In: *Journal of Physics: Conference Series* 706 (Apr. 2016), p. 022002. DOI: 10.1088/1742-6596/706/2/022002. URL: <https://doi.org/10.1088/1742-6596/706/2/022002>.
- [7] *ATLAS: technical proposal for a general-purpose pp experiment at the Large Hadron Collider at CERN*. LHC technical proposal. Geneva: CERN, 1994. DOI: 10.17181/CERN.NR4P.BG9K. URL: <https://cds.cern.ch/record/290968>.
- [8] *Technical proposal*. LHC technical proposal. Cover title : CMS, the Compact Muon Solenoid : technical proposal. Geneva: CERN, 1994. URL: <https://cds.cern.ch/record/290969>.
- [9] G. Aad et al. “Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC.” In: *Physics Letters B* 716.1 (2012), pp. 1–29. ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2012.08.020>. URL: <https://www.sciencedirect.com/science/article/pii/S037026931200857X>.

Bibliography

- [10] S. Chatrchyan et al. “Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC.” In: *Physics Letters B* 716.1 (2012), pp. 30–61. ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2012.08.021>. URL: <https://www.sciencedirect.com/science/article/pii/S0370269312008581>.
- [11] *LHCb : Technical Proposal*. Geneva: CERN, 1998. URL: <https://cds.cern.ch/record/622031>.
- [12] *ALICE: Technical proposal for a Large Ion collider Experiment at the CERN LHC*. LHC technical proposal. Geneva: CERN, 1995. URL: <https://cds.cern.ch/record/293391>.
- [13] Thomas Sven Pettersson and P Lefèvre. *The Large Hadron Collider: conceptual design*. Tech. rep. Oct. 1995. URL: <https://cds.cern.ch/record/291782>.
- [14] LHCb Collaboration. *LHCb VELO Upgrade Technical Design Report*. Tech. rep. Nov. 2013. URL: <https://cds.cern.ch/record/1624070>.
- [15] LHCb Collaboration. *LHCb Tracker Upgrade Technical Design Report*. Tech. rep. Feb. 2014. URL: <https://cds.cern.ch/record/1647400>.
- [16] LHCb Collaboration. *LHCb PID Upgrade Technical Design Report*. Tech. rep. Nov. 2013. URL: <https://cds.cern.ch/record/1624074>.
- [17] J Van Tilburg. “Track simulation and reconstruction in LHCb.” Presented on 01 Sep 2005. 2005. URL: <https://cds.cern.ch/record/885750>.
- [18] C. Møller and N. Bohr. *General Properties of the Characteristic Matrix in the Theory of Elementary Particles I*. Matematisk-Fysiske meddelelser / Kongelige Danske Videnskabernes Selskab. Munksgaard Komm., 1945. URL: <https://books.google.de/books?id=hEDFzgEACAAJ>.
- [19] and R. Aaij et al. “Measurement of the inelastic pp cross-section at a centre-of-mass energy of 13 TeV.” In: *Journal of High Energy Physics* 2018.6 (June 2018). DOI: 10.1007/jhep06(2018)100. URL: <https://doi.org/10.1007/2Fjhep06%282018%29100>.
- [20] T Pieloni and B Muratori. *Luminosity levelling techniques for the LHC*. en. 2014. DOI: 10.5170/CERN-2014-004.177. URL: <http://cds.cern.ch/record/1957033>.
- [21] CERN (Meyrin) LHCb Collaboration. *LHCb PLUME: Probe for LUminosity MEasurement*. Tech. rep. Geneva: CERN, Jan. 2021. DOI: 10.17181/CERN.WLU0.M37F. URL: <https://cds.cern.ch/record/2750034>.
- [22] George Coombs, Massimiliano Ferro-Luzzi, and Rosen Matev. “Beam-Gas Imaging Measurements at LHCb.” In: (2019), WEPB13. 5 p. DOI: 10.18429/JACoW-IBIC2018-WEPB13. URL: <https://cds.cern.ch/record/2716029>.

- [23] Vladislav Balagura. “Van der Meer scan luminosity measurement and beam–beam correction.” In: *The European Physical Journal C* 81.1 (Jan. 2021). DOI: 10.1140/epjc/s10052-021-08837-y. URL: <https://doi.org/10.1140/2Fepjc%2Fs10052-021-08837-y>.
- [24] Kazuyoshi Carvalho Akiba et al. *The LumiTracker detector*. Tech. rep. Geneva: CERN, Dec. 2020. URL: <https://cds.cern.ch/record/2746596>.
- [25] E. Buchanan et al. “Spatial resolution and efficiency of prototype sensors for the LHCb VELO Upgrade.” In: (Jan. 2022). arXiv: 2201.12130 [physics.ins-det].
- [26] Jason Sanders and Edward Kandrot. *CUDA by example: an introduction to general-purpose GPU programming*. Addison-Wesley Professional, 2010.
- [27] Pradeep Gupta. *CUDA Refresher. The CUDA Programming Model*. 2020. URL: <https://developer.nvidia.com/blog/cuda-refresher-cuda-programming-model/#:~:text=Figure%201%20shows%20that%20the,function%20executed%20on%20the%20GPU..>
- [28] Torbjörn Sjöstrand. “The Pythia event generator: Past, present and future.” In: *Computer Physics Communications* 246 (Jan. 2020), p. 106910. DOI: 10.1016/j.cpc.2019.106910. URL: <https://doi.org/10.1016%2Fj.cpc.2019.106910>.
- [29] I Belyaev et al. “Handling of the generation of primary events in Gauss, the LHCb simulation framework.” In: *Journal of Physics: Conference Series* 331.3 (Dec. 2011), p. 032047. DOI: 10.1088/1742-6596/331/3/032047. URL: <https://doi.org/10.1088/1742-6596/331/3/032047>.
- [30] S. Agostinelli et al. “Geant4—a simulation toolkit.” In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506.3 (2003), pp. 250–303. ISSN: 0168-9002. DOI: [https://doi.org/10.1016/S0168-9002\(03\)01368-8](https://doi.org/10.1016/S0168-9002(03)01368-8). URL: <https://www.sciencedirect.com/science/article/pii/S0168900203013688>.
- [31] G Barrand. “Panoramix.” In: (2005). DOI: 10.5170/CERN-2005-002.365. URL: <https://cds.cern.ch/record/865604>.
- [32] Daniel Hugo Cámpora Pérez, Niko Neufeld, and Agustín Riscos Nuñez. “A Fast Local Algorithm for Track Reconstruction on Parallel Architectures.” In: *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. May 2019, pp. 698–707. DOI: 10.1109/IPDPSW.2019.00118.
- [33] H. E. Rauch, F. Tung, and C. T. Striebel. “Maximum likelihood estimates of linear dynamic systems.” In: *AIAA Journal* 3.8 (1965), pp. 1445–1450. DOI: 10.2514/3.3166. eprint: <https://doi.org/10.2514/3.3166>. URL: <https://doi.org/10.2514/3.3166>.

Bibliography

- [34] E.A. Wan and R. Van Der Merwe. “The unscented Kalman filter for nonlinear estimation.” In: *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. Oct. 2000, pp. 153–158. DOI: 10.1109/ASSPCC.2000.882463.
- [35] Zhaozhong Chen et al. *Weak in the NEES?: Auto-tuning Kalman Filters with Bayesian Optimization*. 2018. DOI: 10.48550/ARXIV.1807.08855. URL: <https://arxiv.org/abs/1807.08855>.
- [36] Rudolph Emil Kalman. “A New Approach to Linear Filtering and Prediction Problems.” In: *Transactions of the ASME-Journal of Basic Engineering* 82.Series D (1960), pp. 35–45.
- [37] Matti Raitoharju, Robert Piché, and Henri Nurminen. “A systematic approach for Kalman-type filtering with non-Gaussian noises.” In: *2016 19th International Conference on Information Fusion (FUSION)*. July 2016, pp. 1853–1858.
- [38] P. Billoir et al. “A parametrized Kalman filter for fast track fitting at LHCb.” In: *Computer Physics Communications* 265 (Aug. 2021), p. 108026. DOI: 10.1016/j.cpc.2021.108026. URL: <https://doi.org/10.1016%2Fj.cpc.2021.108026>.
- [39] Aziz Kaba and Emre Kiyak. “Optimizing a Kalman filter with an evolutionary algorithm for nonlinear quadrotor attitude dynamics.” In: *Journal of Computational Science* 39 (2020), p. 101051. ISSN: 1877-7503. DOI: <https://doi.org/10.1016/j.jocs.2019.101051>. URL: <https://www.sciencedirect.com/science/article/pii/S1877750319302376>.
- [40] Seyed Fakoorian et al. “Robust Kalman-Type Filter for Non-Gaussian Noise: Performance Analysis With Unknown Noise Covariances.” In: *Journal of Dynamic Systems, Measurement, and Control* 141 (Mar. 2019). DOI: 10.1115/1.4043054.
- [41] H. A. Bethe. “Molière’s Theory of Multiple Scattering.” In: *Phys. Rev.* 89 (6 Mar. 1953), pp. 1256–1266. DOI: 10.1103/PhysRev.89.1256. URL: <https://link.aps.org/doi/10.1103/PhysRev.89.1256>.
- [42] Virgil L. Highland. “Some practical remarks on multiple scattering.” In: *Nuclear Instruments and Methods* 129.2 (1975), pp. 497–499. ISSN: 0029-554X. DOI: [https://doi.org/10.1016/0029-554X\(75\)90743-0](https://doi.org/10.1016/0029-554X(75)90743-0). URL: <https://www.sciencedirect.com/science/article/pii/0029554X75907430>.

- [43] Gerald R. Lynch and Orin I. Dahl. “Approximations to multiple Coulomb scattering.” In: *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms* 58.1 (1991), pp. 6–10. ISSN: 0168-583X. DOI: [https://doi.org/10.1016/0168-583X\(91\)95671-Y](https://doi.org/10.1016/0168-583X(91)95671-Y). URL: <https://www.sciencedirect.com/science/article/pii/0168583X9195671Y>.
- [44] R. Mankel. “ranger - a Pattern Recognition algorithm for the HERA-B Main Tracking System. Part IV: The Object-Oriented Track Fit.” In: (May 1998). URL: http://www-hera-b.desy.de/general/publications/pub_notes/98-079.ps.gz.
- [45] Fons Rademakers et al. *root-project/root: v6.24/06*. Version v6-24-06. Sept. 2021. DOI: [10.5281/zenodo.848818](https://doi.org/10.5281/zenodo.848818). URL: <https://doi.org/10.5281/zenodo.848818>.

Acknowledgement

I want to thank everyone who supported me during this thesis. First, I like to thank Prof. Dr. Johannes for providing me with the opportunity to write this thesis and for welcoming me to the working group. Furthermore, I like to thank him for leading the working group with a lot of dedication, even through difficult times. I also like to thank Prof. Dr. Kevin Kröninger for agreeing to be the second corrector of the thesis.

My special thanks go to Elena, as she always took time out of her busy schedule for discussions. As an expert on the LumiTracker and luminosity measurements, her knowledge was invaluable for this work. She was always very patient with me and provided me with precise feedback and suggestions for every aspect of the thesis. The thesis in this form would not have been possible without her.

I also like to thank Holger and Lukas for their patience and support. They were always open to questions and guided me in the right direction when I was unsure how to proceed. I am also grateful to the entire working group, as everyone was always ready to help, and the atmosphere was great. I regret not being more often present in the offices.

Finally, I would like to thank my family and friends for supporting me, not only during this work but also throughout my studies.