technische universität
dortmund

Dissertation zur Erlangung des akademischen Grades
Dr. rer. nat. Physik

Alternative Approaches in the Event Reconstruction of LHCb

Christoph Hasse
geboren am 25.03.1992 in Essen

Lehrstuhl für Experimentelle Physik V
Fakultät Physik
Technische Universität Dortmund

Der Fakultät Physik der Technischen Universität Dortmund zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften vorgelegte Dissertation.

Gutachter:
Dr. Johannes Albrecht
Prof. Dr. Kevin Kröninger

Datum des Einreichens der Arbeit:
28. September 2019
Datum der mündlichen Prüfung:
12. Dezember 2019

Acknowledgments

Throughout my time as a doctoral student I have often relied on the help and guidance of many people whom I would like to thank in the following. First of all, I want to thank Johannes Albrecht, who has supervised my Bachelor's, Master's, and PhD thesis. Thanks for helping me chose interesting topics, always being available when help is needed, and providing me with the opportunity to be a CERN doctoral student. Next, I would like to thank my second PhD supervisor, Ben Couturier. Thank you for making it possible for me to be a doctoral student at CERN, always being available to discuss project ideas, and supporting me during my studies. I also want to thank Prof. Dr. Kevin Kröninger for agreeing to review this thesis.

Further, I would like to thank the German Federal Ministry of Education and Research who has sponsored this work through the Wolfgang Gentner Programme (grant no. 05E15CHA).

I want to thank the people who have greatly contributed to the presented work by answering my seemingly endless amount of questions and providing guidance. Thanks, Alex, Ben, Marco, Niklas, Olli, Paul, Roel, Rosen, Sascha, Sebastien for helping me make sense of LHCb's software stack until the end. Thanks, to Sascha and Michel for sharing your many insights on track reconstruction and beyond. A special thank you to Sascha for guiding this work on a daily basis and for spending the time to review this document. Overall I want to thank the entire LHCb CERN group for the pleasant work atmosphere I was able to be part of over the last years.

While I have spent most of my time as a PhD student at CERN, I have enjoyed my visits to Dortmund to catch up with the people that have been supportive since I first started my studies in Dortmund. The five years in Dortmund before I started my PhD surely would not have been as pleasant without all of you! A special thank you goes to Mirko, thanks for being a great friend (even though you chose condensed matter physics).

I want to thank my family for always being supportive and encouraging my curiosity from an early age! And last but not least, thank you, Molly for your love and unlimited support especially during the time of writing this document.

Abstract

Starting in 2021, the LHCb experiment will switch to a fully software-based trigger system. It is an ongoing challenge to ensure that this system will be able to process events at the required rate of 30 MHz. Two alternative approaches to improve and speed up the particle track reconstruction performed in the first stage of LHCb's software trigger are presented. An alternative method to the Kalman filter track fit inside the VELO reconstruction is presented. This method uses neural networks to perform the fit and additionally provides an estimation of a track's momentum based on the track's scattering. The momentum information enables improved uncertainty estimates for a trajectory's impact parameter, an important quantity to select tracks from secondary vertices. It is shown that the neural network's uncertainty prediction enables an equally efficient selection of secondary tracks while reducing the amount of falsely selected prompt tracks by 30 %.

Additionally, an alternative procedure to reconstruct tracks that traverse the entire LHCb detector is presented. These tracks are LHCb's main physics objects and their reconstruction previously required a third of the overall processing time of the first trigger stage's reconstruction sequence. A new algorithm design is proposed, which is shown to yield similar reconstruction efficiencies while providing an over six-fold speedup over the current algorithm. A comparison of trigger configurations, which are able to process data at the rate of 30 MHz, is presented. It shows that a trigger based on the new algorithm yields significantly higher selection efficiencies.

Kurzfassung

Das LHCb Experiment wird 2021 zu einem vollständig softwarebasiertem trigger system über gehen. Es ist eine andauernde Herausforderung sicherzustellen, dass dieses System in der Lage ist Ereignisse mit einer Rate von 30 MHz zu prozessieren. In dieser Arbeit werden alternative Ansätze zur Verbesserung und Beschleunigung der Rekonstruktion von Teilchenspuren in der ersten Phase des LHCb Software Triggers vorgestellt. Eine Alternative zu Kalman-Filter basierten Parameterschätzungen von Teilchenspuren im VELO wird aufgeführt. Diese verwendet neuronale Netze zur Parameterschätzung und bestimmt zusätzlich eine Impulsschätzungen abhängig von der beobachteten Mehrfachstreuung der Teilchenspur. Solche Impulsschätzungen ermöglichen genauere Fehlervorhersagen bei der Bestimmung des Stoßparameters einer Spur, eine wichtige Größe zur Auswahl von Teilchenspuren aus sekundären Zerfällen. Es wird gezeigt, dass die Fehlervorhersage des neuronalen Netzes es ermöglicht eine ebenso effiziente Auswahl von Sekundärspuren zu treffen, bei gleichzeitiger Reduzierung des selektierten Untergrundes von 30 %.

Zusätzlich wird eine alternative Prozedur zur Rekonstruktion von Spuren, die den gesamten LHCb Detektor durchqueren, vorgestellt. Diese Spuren sind die wichtigsten Objekte für Physikanalysen bei LHCb. Deren Rekonstruktion benötigt jedoch ein Drittel der gesamten Rechenzeit der ersten Triggerstufe benötigt. Es wird gezeigt, dass der vorgeschlagene Algorithmus zu ähnlichen Rekonstruktionseffizienzen führt, jedoch nur ungefähr ein sechstel der Rechenzeit des aktuellen Algorithmus benötigt. Ein Vergleich von zwei Triggerkonfigurationen, welche es erlauben Ergeignisse mit einer Rate von 30 MHz zu prozessieren, wird dargelegt. Dieser zeigt, dass ein auf dem neuen Algorithmus basierender Trigger deutlich höhere Selektionseffizienzen ermöglicht.

Contents

1 Introduction **1**

2 Large Hadron Collider and its Objectives **3**
- 2.1 Overview of the Standard Model 3
- 2.2 The LHC 4

3 The LHCb experiment **7**
- 3.1 The LHCb detector 8
- 3.2 Tracking System 10
  - 3.2.1 Dipole Magnet 10
  - 3.2.2 Vertex Locator 11
  - 3.2.3 Upstream Tracker 13
  - 3.2.4 Scintillating Fiber Tracker 14
- 3.3 Particle Identification System 14
  - 3.3.1 Ring Imaging Cherenkov Detectors 15
  - 3.3.2 Electromagnetic and Hadronic Calorimeters 16
  - 3.3.3 Muon Stations 16
- 3.4 LHCb Software Framework 17

4 Particle Trajectory Reconstruction at LHCb **19**
- 4.1 Basic Principles 19
  - 4.1.1 Particle Interactions with Matter 19
  - 4.1.2 Passage Through Magnetic Field 21
  - 4.1.3 Measuring Reconstruction Performance 22
- 4.2 Reconstruction Strategies 23

5 LHCb Trigger System **29**
- 5.1 Trigger Scheme 29
- 5.2 Open Computing Challenges 33

6 Improving Impact Parameter Selections of VELO Tracks **39**
- 6.1 Alternative VELO Track Fit 40
  - 6.1.1 Model Design 42

6.1.2 Network Training Procedure 47
6.2 Results 48
6.3 Conclusion and Outlook 52

7 Fast Forward Tracking for the LHCb Upgrade 55
7.1 Algorithm Requirements Specification 56
7.2 Algorithm Design 57
7.2.1 Algorithm Overview 59
7.2.2 Finding Compatible Hit Doublets inside Last Station . . . 60
7.2.3 Doublet Confirmation in Last Station 65
7.2.4 Trajectory Forwarding 72
7.2.5 Parameter Estimation and Selection 74
7.3 Performance Evaluation 81
7.4 Comparison of Trigger Selection Efficiencies 87
7.5 Summary 94

8 Summary and Outlook 97

A Supplementary Information 99
A.1 Data Samples 99

Bibliography i

1 Introduction

The Large Hadron Collider (LHC) is the world's largest machine built to date. It is a circular particle accelerator with a circumference of 27 km, which is operated by the European Organization for Nuclear Research (CERN) to collide particles at nearly the speed of light. These collisions occur at four distinct locations along the accelerator tunnel at a rate of up to 40 million times per second. Each of these locations is home to one of the four main experiments at the LHC: ALICE, ATLAS, CMS, and LHCb.

The most important discovery at the LHC to date is the confirmation of the existence of the Higgs particle in 2012. Its discovery marked the completion of the current best theory of elementary particles and their dynamics, the Standard Model of particle physics. It predicts a multitude of experimental results to unprecedented levels of precision. However, many observed phenomena in our universe are not described by the Standard Model, marking it as an incomplete theory. To name one of these observations, the Standard Model does not provide any description of gravitational interactions. Furthermore, observations in the field of cosmology have lead to the postulation of dark matter and dark energy which are predicted to account for over 90 % of the universe's matter. Yet, the Standard Model includes no description for any of these two forms of matter. In conclusion, physics processes beyond the standard model must exist.

Any observations not described by the Standard Model could help to develop new theoretical frameworks that may solve some shortcomings of our current theory. The search for these processes is a main goal of the LHCb experiment, which is specialized in precision measurements of heavy flavor hadron decays. A crucial part of the LHCb experiment, and that of all four main LHC experiments, is a trigger system. Saving the detector data for all of the occurring particle collisions would require incredible amounts of data storage, which the experiments can not afford. However, as most of the interesting events are rare, it is sufficient to only save these events for a later analysis. It is the task of the trigger system to decide whether an event should be saved or discarded.

Until the end of its second operational period in 2018, the LHCb trigger system consisted of a first stage, which was was implemented in hardware, and a second software based stage. This system enabled the LHCb experiment to collect large amounts of data based on which it probed the Standard Model's predictions to new levels of precision. Before starting its third operational period in 2021, the LHCb detector is undergoing a significant upgrade including the replacement of its trigger system by a software only solution, which will operate at the event crossing rate of 30 MHz. In addition, the LHCb experiment will start to operate at an increased instantaneous luminosity. These changes will allow a collection of even more data per year and increase the total amount of collected data by a factor of five until the end of 2026. Subsequently, reducing statistical uncertainties for many physics analyses and enabling new searches for decays to rare to find within the current data set.

However, the latest evaluation of the of upgrade trigger system shows that it is merely able to reach a third of the required processing rate. As the budget and thus the computing resources are fixed, the success of the software trigger relies on the reduction of the required processing time of the used reconstruction algorithms. The work presented in the following thesis investigates alternative approaches to these reconstruction procedures with the goal of enabling an efficient software trigger for LHCb's upgrade detector. Chapter 2 first gives a brief summary of the Standard Model and an overview of the Large Hadron Collider. An introduction of LHCb's detector, software framework, and trigger system is presented in Chapter 3. Chapter 4 presents the theoretical background of particle trajectory reconstruction and outlines the algorithms, which perform this task within LHCb's trigger system. The current status and performance of the first stage of the software trigger as well as the open challenges are presented in Chapter 5, before Chapters 6 and 7 present two new approaches to overcome some of the outlined remaining challenges.

2 Large Hadron Collider and its Objectives

The Large Hadron Collider [1] is a circular particle accelerator operated by the European Organization for Nuclear Research (CERN) in Geneva, Switzerland. Since CERN's foundation in 1954, it has grown to be the largest particle physics laboratory in the world employing over 2500 people as well as having more than 12000 visiting scientists from all over the world conducting their research there [2]. The following chapter provides an overview of the LHC and its physics objectives. As most of these objectives are related to the Standard Model [3–5], the chapter starts with a brief summary (based on [6]) of the Standard Model.

2.1 Overview of the Standard Model

Originally developed over 50 years ago, the Standard Model [3–5] of physics is still undisputedly the best description of fundamental particles and their dynamics. It is able to provide predictions for the electromagnetic, strong, and weak force but does not include a description of gravity. The fundamental particles of the Standard Model are depicted in Fig. 2.1.

As shown, they are divided into fermions with half integer spin and four bosons with integer spin as well as the Higgs boson with a spin of zero. Not shown are the antifermions which are identical in mass and lifetime but exhibit inverted quantum numbers. The twelve fermions are made up of six quarks (up, down, charm, strange, top, and bottom) and six leptons (electron, muon, tau, and their corresponding neutrinos), each further grouped into three generations. The six quarks are unique in that they additionally carry a so called color charge and thus are able to interact via the strong force. While neutrally charged neutrinos can only interact via the weak force, the electron, muon, and tau are charged leptons which can also interact via the electromagnetic force. It is for this reason that neutrinos are difficult to observe as they only interact very rarely with matter.

Only discovered in 2012 [8, 9] the Higgs Boson was the last missing elementary particle predicted by the Standard Model. It is the only fundamental particle in the Standard Model with a spin of zero and gives rise to the fermion masses through

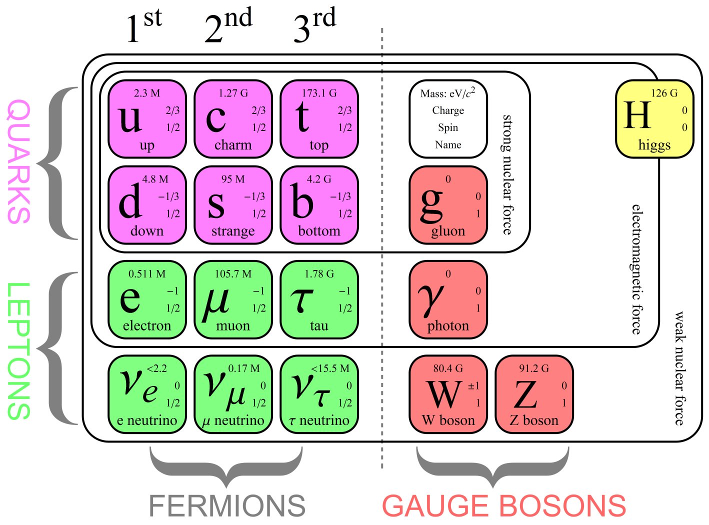


Figure 2.1: Fundamental particles of the Standard Model [7].

the Higgs mechanism. The remaining four bosons are responsible for mediating the three fundamental forces between particles. The gluon couples to the color charge and mediates the strong force. It is the only mediator boson which is able to couple to itself, as it also exhibits a color charge. The electromagnetic force is mediated by the photon and the weak force by the W and Z bosons.

2.2 The LHC

The 27 km long tunnel housing the LHC is located between 50 to 175 m underground and was previously used by the Large Electron-Positron Collider, which was dismantled in 2000 [10].

Experiments located at the LHC aim to test the Standard Model and look for hints of physics processes not described by the current theory. The LHC is designed to collide two proton beams which are accelerated in opposite direction yielding a center of mass energy of 14 TeV while achieving an instantaneous luminosity of $10^{34}/(\mathrm{cm}^2\,\mathrm{s})$. From the start of operations in 2009 to this day the LHC has completed two successful periods of data taking which are separated by a two-year-long shutdown (LS1) in 2013 and 2014. After the fist proton-proton collisions in 2009 the LHC reached a center of mass energy of 7 TeV in 2010 marking the start of the main research program and data taking operations. In 2012, the last year of Run 1 the energy was increased to 8 TeV. During LS1 many aspects of the LHC machine were upgraded enabling collisions at the design energy of 13 TeV during

Run 2 which lasted from 2015 to 2018. A second long shutdown (LS2) started in 2019 and will be used for further upgrades which aim to enable a third operational run at the design energy of 14 TeV starting in 2021.

Before protons are collided in the LHC they are produced through ionization of hydrogen and then pass through various preaccelerators, which are depicted in Fig. 2.2.

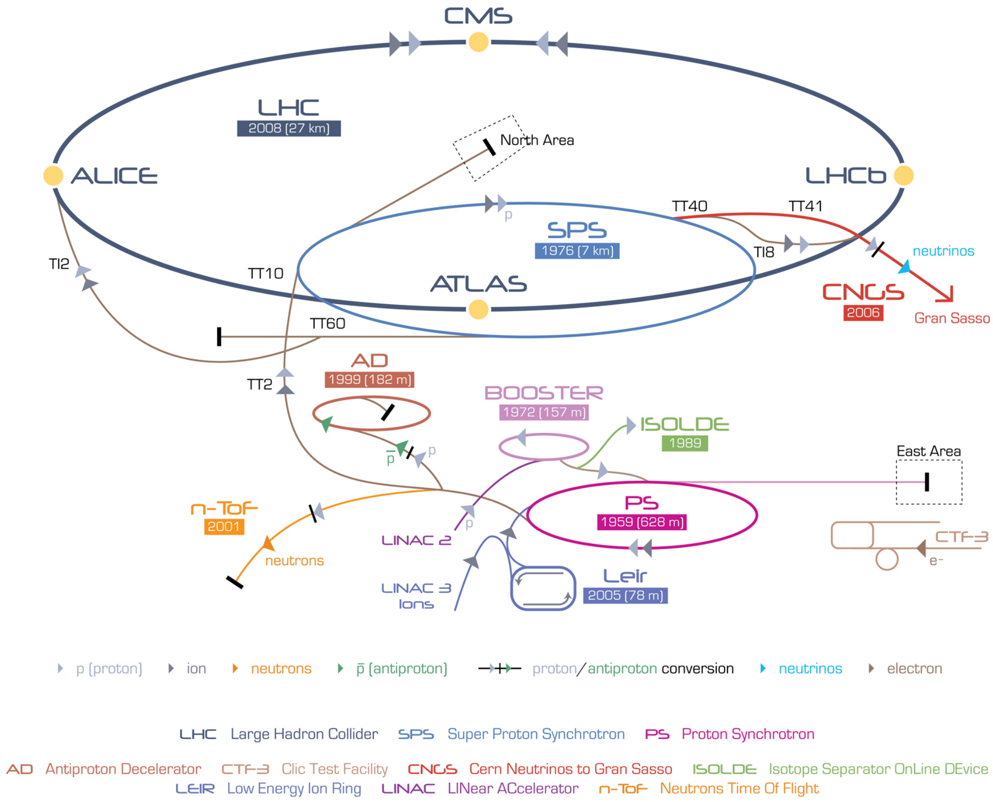


Figure 2.2: The Large Hadron Collider with its preaccelerators and the experiments [10]. The protons are accelerated by the LINAC2 and injected into the BOOSTER, where they reach an energy of 1.4 GeV. Next they pass through the PS, followed by the SPS, where they are boosted to an energy of 450 GeV before they enter the LHC.

The acceleration chain starts with the linear accelerator LINAC2 and is followed by the circular accelerator BOOSTER. The protons are then boosted to an energy of 450 GeV by the Proton Synchrotron (PS) and the Super Proton Synchrotron (SPS), before being injected into the LHC. Finally, the protons are boosted to their final energy of 4 TeV and brought to collision at four dedicated interaction points along the LHC.

These locations are home to the detectors of the four major experiments: ATLAS [11], ALICE [12], CMS [13], and LHCb [14], as shown in Fig. 2.2. ATLAS and CMS are the two largest experiments at the LHC. Both are general purpose detectors covering nearly the full solid angle. Their physics goals include the search for dark matter, supersymmetry, extra dimensions, and the Higgs Boson, which was first discovered by the two collaborations in 2012 [8, 9].

ALICE is mainly designed to study the physics of strongly interacting matter in heavy ion collisions. For this reason, the LHC is occasionally filled with lead ions.

The LHCb experiment will be introduced in more detail in the following section.

3 The LHCb experiment

The LHCb experiment is designed to enable high precision measurements of *CP* violation and study rare decays of B hadrons [14]. The main goal of these studies are the discovery of physics effects not described by the Standard Model.

In contrast to ATLAS and CMS, LHCb operated at a lower instantaneous luminosity during Run 1 and Run 2 such that on average only a single pp-interaction occurs per event. This in turn reduces the detector occupancy, leading to more precise measurements and a reduction of radiation damage. The data set LHCb collected in that period corresponds to an integrated luminosity of $9\,\mathrm{fb}^{-1}$ based on which close to 500 papers [15] were successfully published. LHCb has produced a multitude of important measurements. One such result is the first evidence of the very rare decay $\mathrm{B}_{\mathrm{s}}^{0} \to \mu^{+}\mu^{-}$ [16] followed by the first observation in 2015, by studying data collected at LHCb and CMS in combination [17]. The Standard Model makes precise predictions about the probability, called branching fraction, of the $\mathrm{B}_{\mathrm{s}}^{0}$ meson to decay into two muons. This well-defined prediction combined with the small value of the branching fraction make the decay an excellent candidate to search for contributions from processes not predicted by the Standard Model. The last and best measurement so far, published by LHCb in 2017 [18], is consistent with the Standard Model's prediction. While not showing any anticipated unpredicted contributions, this result provides crucial constraints for further development of theory extensions to the current model. Other major milestones include the first observation of pentaquarks [19] and *CP* violation in charm hadron decays [20]. Pentaquarks are states which contain five quarks and were first predicted in 1964 [21] but had not been observed prior to LHCb's discovery. This discovery shows that, while originally designed to study heavy flavor decays, LHCb has developed into a multipurpose detector enabling a broad spectrum of physics analysis.

The LHCb detector and its trigger system will undergo a significant upgrade in LS2 to enable LHCb to take data at higher luminosities and thus record more data than the current limit of about $1\,\mathrm{fb}^{-1}$ per year [22]. This significant increase will allow LHCb to collect up to $46\,\mathrm{fb}^{-1}$ [23] of data until the end of 2026. A data set of this size will enable further reduction of the statistical uncertainty on existing measurements as well as new measurements which are sensitive to beyond Standard Model processes. This is further outlined in detail in [22, 23].

This chapter will first provide an overview of the LHCb detector, focusing on the upgraded detector design for Run 3 in 2021, followed by an introduction of its software framework. An introduction of the trigger system is given in Chapter 5.

3.1 The LHCb detector

The LHCb detector is a single arm forward spectrometer designed to study decays of beauty- and charm-hadrons with high precision. Its angular coverage extends from 10 to 250 mrad vertically and from 10 to 300 mrad horizontally. This translates into a pseudorapidity range of $2 < \eta < 5$, with $\eta = -\log(\tan\theta/2)$.

The reason for this significantly different detector design compared to the other three major experiments at the LHC is a special characteristic of $b\bar{b}$ production in proton collisions. As shown in Fig. 3.1, a large fraction of the created $b\bar{b}$ pairs is produced with a large boost along the beam axis, thus allowing for the LHCb detector to study about 24 % of occurring $b\bar{b}$ pairs.

All components of LHCb's upgrade detector are shown in Fig. 3.2, and will be explained further in the following subsections. The Vertex Locator, Upstream Tracker, and SciFi Tracker can be categorized as tracking detectors whereas the RICH detectors, calorimeters, and muon stations are used to obtain information used for particle identification (PID). The data recorded by the means of these components is processed by a two stage software trigger system which either rejects an event or delegates it to be saved for later analysis. The following description uses LHCb's default global coordinate system which is defined such that the x-axis is horizontal, the y-axis vertical, and the z-axis along the beam line. It is positioned such that the origin is inside the VELO detector at the nominal position of the pp-interaction, and z-values increase towards the SciFi detector.

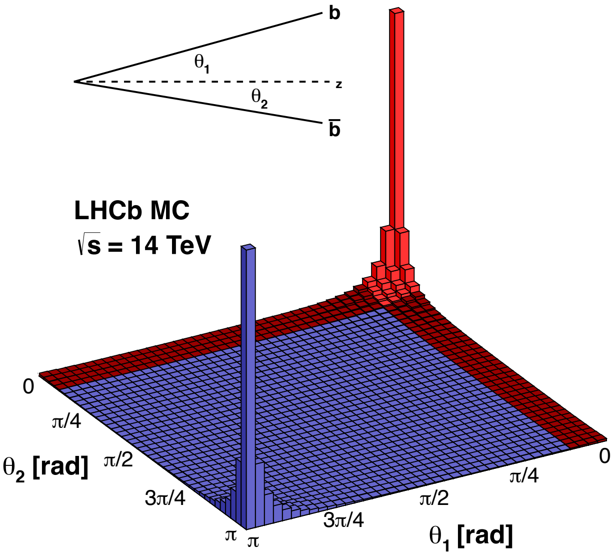


Figure 3.1: Angular distribution of $b\bar{b}$ pair production in proton collisions at a center of mass energy of 8 TeV. The region highlighted in bright red corresponds to the covered region by the LHCb detector [24].

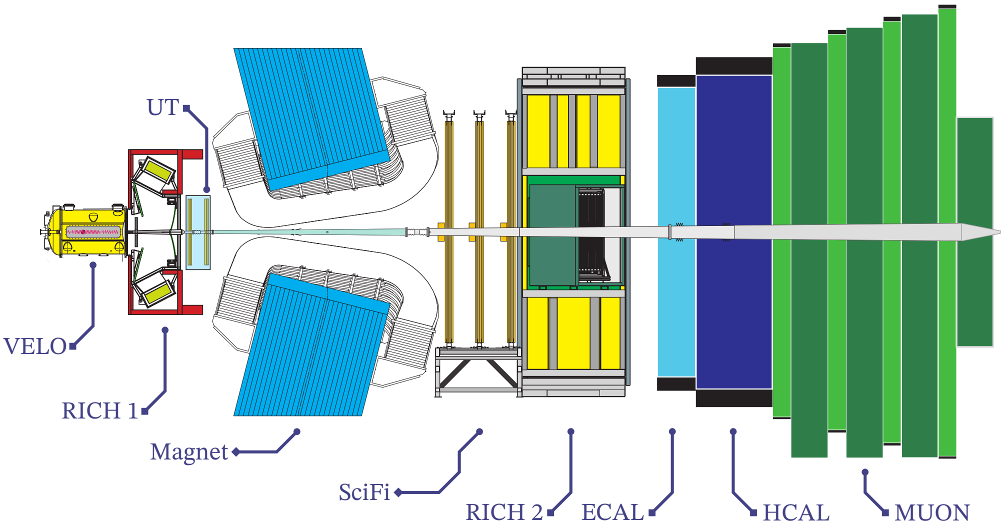


Figure 3.2: Schematic illustration of the LHCb detector, viewed from the side. From left to right: pp-interaction point, Vertex Locator (VELO), Ring Imaging Cherenkov detector one (RICH1), Upstream Tracker (UT), Magnet, SciFi Tracking stations, RICH2, electromagnetic and hadronic calorimeter (ECAL/HCAL), and muon stations [25].

3.2 Tracking System

LHCb's tracking system is used to reconstruct a charged particle's trajectory, called *track*, through the detector and determine its momentum. While the magnet is the same one as before the upgrade, the three tracking detectors detailed below are entirely new designs and replace their old counterparts.

3.2.1 Dipole Magnet

The dipole magnet is positioned between the Upstream Tracker and the Scintillating Fiber Tracker and produces an integrated magnetic field of 4 T m. The main component B_y is shown in Fig. 3.3 as a function of z-position along the beam line. Given that the magnet's main component is pointing in y-direction, the magnetic field deflects charged particles in the xz plane, further referred to as the *bending plane*.

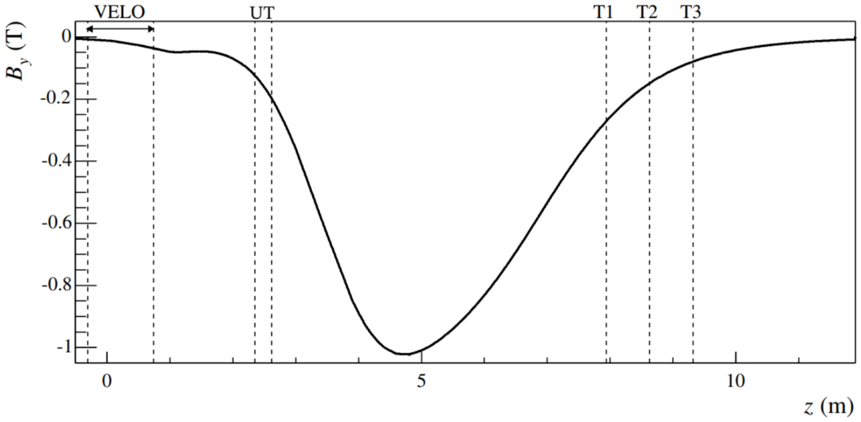


Figure 3.3: Magnitude of the magnetic field's main component B_y as function of z. The vertical dotted grey lines indicate the position of the VELO, Upstream Tracker, and the SciFi's tracking stations (T1, T2, and T3).

This deflection is a crucial part of the tracking system as it enables the measurement of a particle's momentum and charge. Given LHCb's detector design it is predicted that the achievable momentum resolution at LHCb is at the order of $\Delta p/p = 0.5\,\%$ [26]. Due to the magnetic deflection, oppositely charged particles are more likely to be detected on opposite sides of the detector. If the two sides exhibit different detection efficiencies, it could complicate measurements of charge asymmetries. These compare the rate of oppositely charged decays, for example $B^+ \to \phi K^+$ and

$B^- \rightarrow \phi K^-$. To better differentiate potentially observed measurements of charge asymmetries from differences in detection efficiency across the detector, the magnet is able to invert its magnetic field.

3.2.2 Vertex Locator

LHCb's Vertex Locator [27] is the first detector particles encounter. It is positioned around the interaction region with the first active sensor material positioned at a radius of only 5.1 mm away from the beam line. This is achieved by removing the beam pipe inside the VELO and creating a secondary vacuum which houses the sensor modules. To protect the sensors from electromagnetic induction from LHC's beam, this vacuum is separated from the beams vacuum by a 250 µm thin aluminum foil. A schematic overview of the design is shown in Fig. 3.4. The detector consists of 26 stations, which are positioned along the beam line such that tracks within LHCb's acceptance will at least traverse 4 stations. Each of these stations is made up of two L-shaped modules which can be retracted away from the beam line to prevent any damage to them during unstable beam conditions, e.g. during startup periods. The bottom two schematics of Fig. 3.4 depicts one such module pair in a closed (left) and open (right) configuration. Every module consists of 4 silicon pixel sensors, each with 768×256 pixels with a pitch of 55 µm. In total resulting in approximately 41 million active pixels inside the VELO detector.

This design enables the VELO to produce precise measurements of the tracks which in turn enables a precise estimation of the position of the pp-interaction, often called the primary vertex (PV), as well as the position of decay vertices of produced b- or c-hadrons which are often referred to as secondary vertex (SV). An accurate prediction of these vertex positions is crucial to achieve the necessary lifetime resolution needed for many of LHCb's analysis.

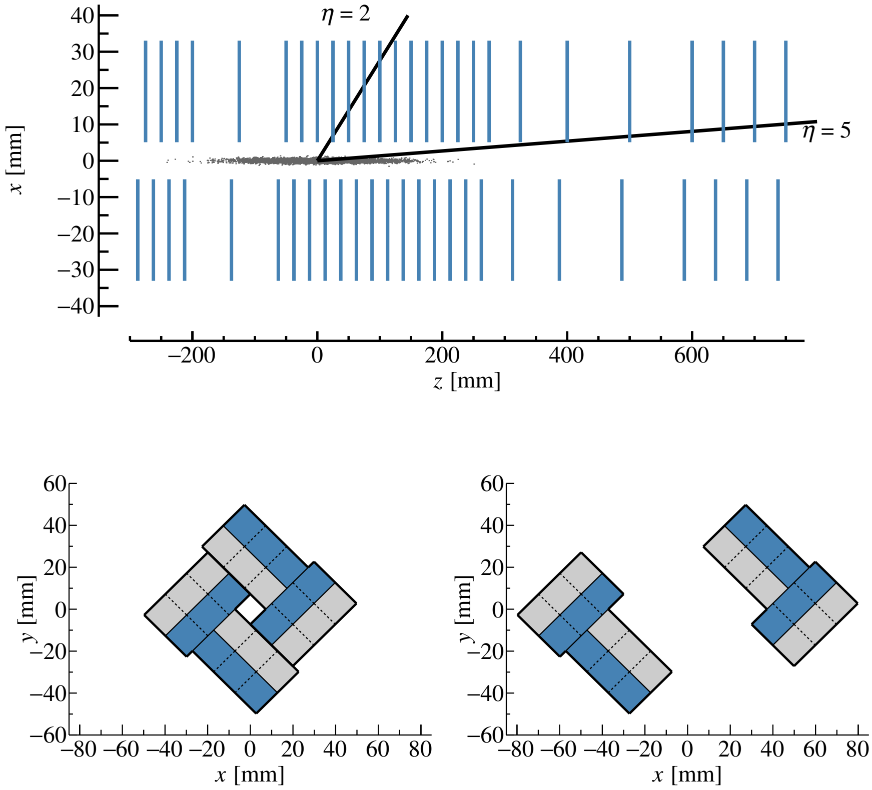


Figure 3.4: Sketch of the Vertex Locator Layout [27]. The cross-section at $y = 0$ on the top shows the module placement along the beam line as well as LHCb's pseudorapidity, η, acceptance. The bottom depicts a single module in the xy plane in a closed (left) and open (right) position.

3.2.3 Upstream Tracker

The Upstream Tracker [26] is LHCb's second tracking detector and is positioned just before the magnet as shown in Fig. 3.2. The detector consists of four planes of silicon micro-strip sensors which vary in size and granularity to account for the higher particle flux in the innermost region. A schematic overview of the planes and the placement of the three different sensor types is shown in Fig. 3.5.

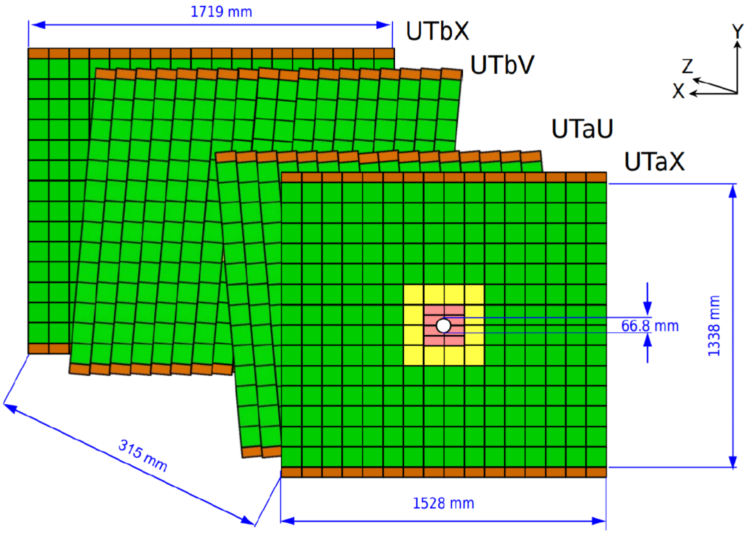


Figure 3.5: Schematic of the four detector plans of the Upstream Tracker. Red boxes correspond to 5 cm long sensors with a pitch of 95 µm, yellow ones corresponds to 10 cm long sensors with the same pitch, and the green boxes represent 10 cm long sensors with a pitch of 190 µm.

As shown, sensors with a length of 5 cm and a pitch of 95 µm are placed around the beam pipe. The neighboring sensors double in length to 10 cm while the outermost sensors double their pitch as well to 190 µm. Additionally, Fig. 3.5 shows that the inner two planes of the detector are rotated around the z-axis by $-5°$ and $+5°$, respectively. These layers are called stereo or uv-layers and provide information on the y-coordinate of a track when combined. The first and last layers, called x-layers, are aligned vertically resulting in their measurements only containing information on the x-coordinate. This configuration enables a precise measurement of a trajectory's parameters in the bending plane which is important to precisely estimate a particle's momentum. The later reconstruction of a trajectory passing through the magnet and into the last tracking detector can be significantly simplified due to this momentum and charge estimate.

3.2.4 Scintillating Fiber Tracker

The Scintillating Fiber Tracker [26] is the largest tracking system and is positioned between the magnet and the RICH2. As shown in Fig. 3.2, the SciFi consists of three tracking stations, each containing four layers which are arranged in an *x*-*u*-*v*-*x* configuration. Identical to the Upstream Tracker, the stereo layers are rotated by −5° and +5°, respectively. The last station is depicted in Fig. 3.6 which shows that a layer is composed of 12 modules, while only 10 modules are used for the smaller layers of station one and two.

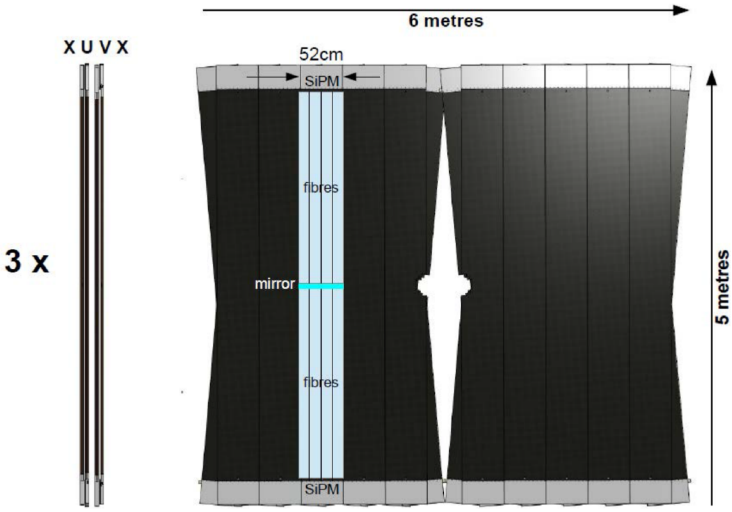


Figure 3.6: A *yz* (left) and *xy* (right) view of a tracking station of the SciFi detector.

The active sensor material of a module are scintillating fibers with a 250 μm diameter. Six layers of such fibers are stacked to form what is called a fiber mat, of which 8 are used to build a single module. Incident charged particles will create photons inside the scintillating fibers which travel along them via internal reflection. These photons are then read out by silicon photo multipliers positioned at the top and bottom of the module. The expected single hit efficiency of these scintillating fiber layers is estimated to be above 97.4 % at the end of the projected lifetime [26].

3.3 Particle Identification System

LHCb's particle identification System (PID) consists of two Ring Imaging Cherenkov detectors, an electromagnetic and hadronic calorimeter, and four muon stations.

These subsystems provide information which in turn is used to determine a likelihood of different particle hypotheses for each trajectory. Common *b*- and *c*-hadron decay products are pions, kaons, protons, and leptons. Therefore, the PID system is designed to distinguish between these particle species. In contrast to the tracking system, the PID system reuses many of the existing components from Run 2 as explained in detail below.

3.3.1 Ring Imaging Cherenkov Detectors

The first of the two Ring Imaging Cherenkov detectors (RICH1) is positioned just after the VELO while the RICH2 is placed between the SciFi and the ECAL as shown in Fig. 3.2. The particle identification in the RICH detectors is based on an analysis of the emitted Cherenkov light of a traversing particle. Cherenkov light is defined as the photons radiated of by a particle traversing a material at a higher speed than the speed of light in the material. The opening angle Θ_c under which these Cherenkov photons are emitted is given by

$$\cos(\Theta_c) = \frac{m}{np}, \tag{3.1}$$

where,m and p are the particle's mass and momentum and n is the refractive index of the traversed material. Followingly, the mass of a particle can be measured upon first determining its momentum by reconstructing the trajectory with LHCb's tracking system. The momentum range at which this procedure is able to differentiate between kaons and pions depends on the refractive index. This dependence is shown on the right in Fig. 3.7 for three different radiator materials.

The upgrade RICH detectors use the same overall structures as the ones in Run 2 with some modifications to RICH1 and an upgrade of all photon detectors to cope with the required readout rate of 40 MHz [28]. A schematic view of the previous RICH1 design is shown on the left of Fig. 3.7. RICH1 covers the full angular acceptance of LHCb and employs fluorocarbon gas C_4F_{10} as radiator material. The depicted aerogel radiators were already removed before the start of Run 2 [29], which results in a beneficial reduction of material for the tracking system. Furthermore, the newly available space in RICH1 is used to optimize its internal optics layout. After these modifications, the first RICH station provides π-K separation in the momentum range of 10 to 40 GeV. RICH2 uses the fluorocarbon gas CF_4 as its radiator and can thus provide the needed particle discrimination up to momenta of 100 GeV. Particles with these momenta are predominately produced in the forward region resulting in sufficient coverage with RICH2's limited angular acceptance of 15 to 120 mrad.

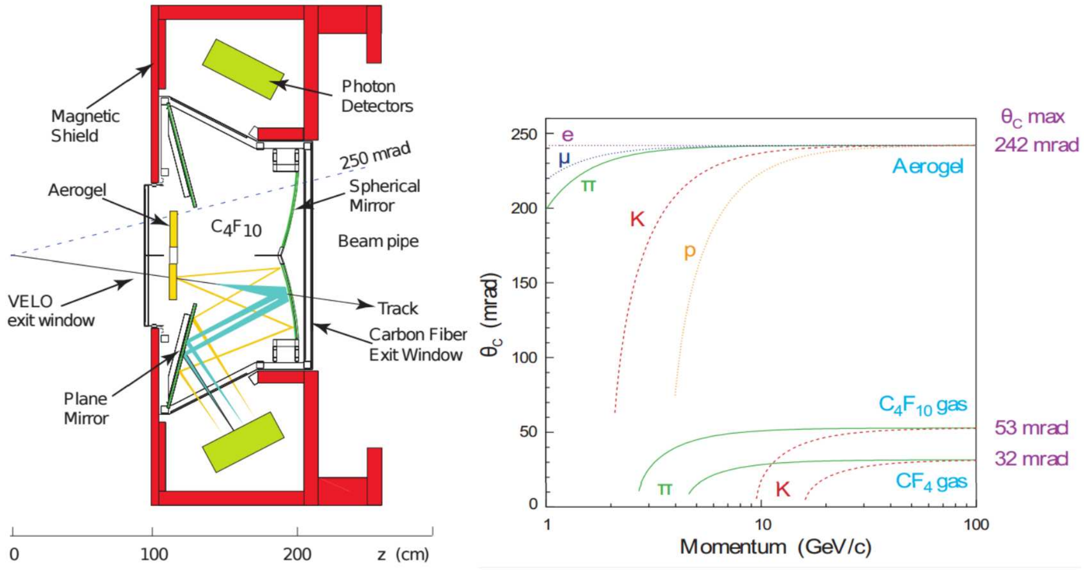


Figure 3.7: Schematic side view of the pre-upgrade RICH1 (left) and the momentum dependence of the Cherenkov light opening angle Θ_c for three different radiator materials (right). [14]

3.3.2 Electromagnetic and Hadronic Calorimeters

Additional PID information as well as an energy and position measurement are obtained through LHCb's calorimeter system [28]. It consists of an electromagnetic (ECAL) and hadronic (HCAL) calorimeter which are both built from alternating layers of scintillating material and absorber plates. This system is identical to the one from Run 2 and only its front end electronics are upgraded to handle the higher readout rate. The basic working principle of both detectors is to stop incident particles due to their traversal of the absorber plates and detect the resulting particle shower which produces photons in the scintillating material. By measuring the number of these resulting photons it is possible to determine the energy of the stopped particle. The ECAL uses lead as its absorber and is designed to identify photons and electrons while hadrons are detected in the iron based HCAL. Figure 3.2 depicts the position of the ECAL and HCAL inside the LHCb detector.

3.3.3 Muon Stations

LHCb's muon stations [28] are designed to detect whether a trajectory inside the detector was produced by a muon, which is essential to the selection of decays with a final state containing muons, for example $B_s^0 \rightarrow \mu^+\mu^-$. The four muon stations are located behind the HCAL as depicted in Fig. 3.2 and separated by 80 cm thick iron layers. Multi-wire proportional chambers (MWPC) are used in each station to detect charged particles. Muons are the only observable particles able to leave a signal in multiple muon stations as the significant amount of material in the calorimeters and inside the muon stations will stop other particles before. Similarly to the calorimeters, the overall structures of the muon stations are identical to the ones used in Run 2 and merely the readout electronics are upgraded. An additional difference to the previous muon system is the removal of the muon station which was located before the ECAL as it would not be able to deliver useful information at the foreseen upgrade luminosities.

3.4 LHCb Software Framework

Many different software packages are used inside of LHCb to simulate, reconstruct, and select particle decays. The following chapter will give a brief introduction into the most important ones which are relevant to the presented theses. All of the packages below are built upon the same underlying framework, the so called GAUDI framework [30]. This framework, originally developed by LHCb, is actively developed and used by the LHCb and ATLAS collaboration.

Especially important for the development of new event reconstruction algorithms but also for several aspects of physics analysis are simulated data sets. These are reconstructed with the same software used in real data taking scenarios, but provide additional information about the simulated particles allowing for a multitude of performance checks, e.g. the ones described in Section 4.1.3.

- **Gauss** The Gauss software package [31] is used to simulate the particle production in pp-collisions as well as the detector interactions of resulting particles. The former procedure is performed by employing third party generator libraries such as PYTHIA [32] or EVTGEN [33]. The propagation through the detector including its magnetic field are performed by the software package GEANT4 [34].
- **Boole** The output of the GAUSS simulation step is processed by the BOOLE package [35] in a step called digitization. This process simulates the detector response and outputs the results in the same format as the data which is taken with the full detector. At this point the simulated events should resemble real data as much as possible. However, uncertainties in detector resolutions,

unknown effects, as well as limited understanding of all underlying physics processes can result in small differences.

- **Brunel** The BRUNEL software package [36] performs the event reconstruction of real and simulated data in an offline environment. It contains the algorithms to reconstruct the trajectories of particles from the hits they left when traversing the tracking subsystem. Furthermore, particle identification algorithms are run on the reconstructed tracks and the combined PID and trajectory information is prepared as output for further processing.
- **DaVinci** The DAVINCI project [37] runs on the output of BRUNEL and aims to reconstruct decays of interest and apply selection criteria to reduce the background event rate. The result of this step is a data set containing decay candidates for the user specified decay topologies and represents the starting point for further physics analysis.
- **Moore** The software run inside the online trigger (see Chapter 5) is contained within the MOORE package [38]. Large parts of the reconstruction and selection algorithms are shared with BRUNEL and DAVINCI. However, since processing times are very limited inside the online environment some of these algorithms are differently configured or otherwise modified to reduce their processing time at the cost of small physics performance losses.

4 Particle Trajectory Reconstruction at LHCb

Particle trajectory reconstruction, for short tracking, is the process of determining a particle's flight path through a detector. Tracking systems rely on the effect of ionization which makes it impossible to determine the trajectory of neutral particles like the photon or neutral pion with these systems. A simplistic view of LHCb's tracking system consists of a set of xy-planes positioned along the beam line. Upon traversing one of these planes, a particle will ionize the detector material resulting in a measurable signal which determines the space point of traversal, often called hit. The first step of common tracking procedures is to identify the set of such hits belonging to a single particle, normally referred to as pattern recognition. Once all hits are collected, the second step is the track fit which is the mathematical procedure of finding the trajectory parameters which best describe the flight path.

First, the chapter below introduces important physics effects governing a particle's path of flight through a particle detector. Second, common quantities used to compare reconstruction algorithms are discussed. Last, an introduction of the algorithms used inside LHCb's reconstruction software is given.

4.1 Basic Principles

The main effects alternating a charged particle's straight path are its interaction with the detector material as well as the Lorentz force acting on the particle inside the magnetic field. The following sections will provide an overview of both phenomena.

4.1.1 Particle Interactions with Matter

While traversing the detector's material or simply flying through the air, particles can interact with matter in a multitude of ways. The ones significant to the tracking problem are the following electromagnetic and strong interactions [39]:

Ionization is the electromagnetic interaction of a charged particle with the outer shell electrons of an atom, leaving the atom either in an ionized or excited state. This process leaves the flight path of the incident particle unchanged while losing some of its energy to the atom. For heavy particles ($M \gg m_e$), the mean energy loss per distance is described by the Bethe-Bloch formula, detailed in [39]. For common particles — pion, kaon, muon — in the momentum range of 2 to 100 GeV the predicted mean energy loss $\langle -dE/dx \rangle$ is minimal, which is why these particles are often referred to as "minimum-ionizing particles". For these particles and common detector materials the energy loss can be approximated by

$$-\langle \frac{dE}{dx} \rangle \approx \rho \cdot 2\,\mathrm{MeV\,cm^2 g^{-1}}, \tag{4.1}$$

with the traversed material's density ρ.

Bremsstrahlung is the emission of photons when a charged particle is deflected by a charged particle, for example by an atom's nucleus. The magnitude of the energy loss is proportional to the rest mass of the particle as well as its velocity. At the particle energies commonly observed at LHCb, this effect is only relevant for electrons as its cross section is proportional to $(m_e/M)^2$, with M being the mass of the incident particle. At LHCb, electrons passing through the entire detector will require a momentum greater than about 2 GeV, otherwise they are deflected out of the detector by the magnetic field. At these momenta, Bremsstrahlung is the dominant effect by which electrons lose energy. The energy loss when traversing a material with radiation length X_0 for a distance of x cm can be described by

$$E_e(x) = E_0 e^{-\frac{x}{X_0}}, \tag{4.2}$$

with the initial Energy of the incident electron E_0.

Coulomb scattering is the scattering of an incident particle at the nucleus of an atom. While the energy of the particle remains unchanged, its trajectory is affected by Coulomb scattering. When a particle travels through material, multiple coulomb scatterings will occur which is referred to as **multiple scattering**. This process is visualized in Fig. 4.1.

For small angles the distribution of the multiple scattering angle θ_{plane} can be approximated by a Gaussian with a standard deviation:

$$\sigma(\theta_{plane}) = \frac{13.6\,\mathrm{MeV}}{\beta c p} z \sqrt{\frac{x}{X_0}} \left[1 + 0.038 \ln\left(\frac{x z^2}{X_0 \beta^2}\right)\right]. \tag{4.3}$$

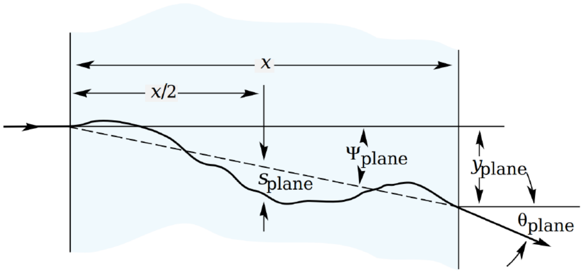


Figure 4.1: Multiple scattering of a particle traversing a material, and its final scattering angle θ_{plane} [39].

The momentum, velocity, incident particle charge, and radiation length are represented by p, βc, z, and x/X_0. As shown in Equation (4.3), the multiple scattering angle is inversely correlated with a particle's momentum. For momenta up to 80 GeV, multiple scattering is the limiting effect on the achievable momentum resolution at LHCb [26].

Hadronic Interactions take place when charged or neutral hadrons interact with nuclei via the strong force. These scattering processes can happen inelastically or elastically. During the former, the incident particle is lost, while the later results in the hadron's trajectory being altered at potentially large angles leaving the hadron unsuitable for further tracking. Hadronic interactions are quantified by the nuclear collision length λ_T, representing the mean free path a hadron is likely to fly between two scattering processes.

4.1.2 Passage Through Magnetic Field

Charged particles traversing a static magnetic field will experience a force

$$\vec{F} = q\vec{v} \times \vec{B} \tag{4.4}$$

acting upon them, called the Lorentz Force. The particle's velocity and charge are represented by $\vec{v}$ and q. $\vec{B}$ is the magnetic field vector. Neglecting material interactions, one can show that a particle's motion through the magnetic field can be described by [40]

$$t_x = \frac{dx}{dz}, \qquad t_y = \frac{dy}{dz}, \tag{4.5}$$

$$\frac{dt_x}{dz} = \frac{q}{p}\sqrt{1 + t_x^2 + t_y^2}\left[t_y\left(t_x B_x + B_z\right) - \left(1 + t_x^2\right) B_y\right], \tag{4.6}$$

$$\frac{dt_y}{dz} = \frac{q}{p}\sqrt{1 + t_x^2 + t_y^2}\left[-t_x\left(t_y B_y + B_z\right) + \left(1 + t_y^2\right) B_x\right], \tag{4.7}$$

with the particle's momentum p as well as t_x and t_y for the slope of the trajectory along the x- and y-direction. Additionally, Eqs. (4.6) and (4.7) show that the change of the slopes is inversely proportional to the momentum. This proportionality enables the estimation of a particle's momentum given the measured trajectory change between the VELO and SciFi. At LHCb the strongest component of the magnetic field is the y-component, thus making the change of the trajectory in x-direction most sensitive to the momentum.

Combining Eq. (4.6) with the difference between the measured x-slope in the SciFi tracker and the VELO detector gives

$$\begin{aligned} \Delta t_x &= t_x^{SciFi} - t_x^{VELO} \\ &= \frac{q}{P}\int_{VELO}^{SciFi} \sqrt{1 + t_x^2 + t_y^2}\left[t_y\left(t_x B_x + B_z\right) - \left(1 + t_x^2\right) B_y\right] dz. \end{aligned} \tag{4.8}$$

which yields the momentum upon solving the integral. As the magnetic field is inhomogeneous, the integration can only be solved by using numerical methods.

4.1.3 Measuring Reconstruction Performance

To compare different track reconstruction procedures a common set of metrics is needed. At LHCb these metrics are reconstruction efficiency, ghost rate, clone rate, and average execution time per event. While the first three quantify "physics performance", the execution time quantifies the processing speed and is especially relevant for algorithms running inside a trigger environment with a fixed per event time budget, as is the case at LHCb.

Reconstruction efficiency is a measure to quantify the likelihood that a particular algorithm will reconstruct a given track. It is determined by testing the algorithm on a simulated sample and counting the amount of correctly reconstructed tracks out of the set of so called "reconstructible tracks". At LHCb, a track is defined as reconstructible if it meets a set of criteria, e.g. having a minimum number of hits in each tracking detector. The specific requirements vary per track type and are checked based on the truth information available in simulated samples.

The **Clone rate** quantifies the amount of clone tracks in respect to all reconstructed tracks. A track is called a clone if there is more than one track correctly associated to a simulated particle trajectory.

The **ghost rate** represents the fraction of all created tracks which are ghost tracks. A ghost track is a track which cannot be associated to any simulated particle. These are normally created by combining a random combination of hits from multiple particles as well as noise hits.

4.2 Reconstruction Strategies

At LHCb, there are five distinct track types which are visualized in Fig. 4.2. A track is assigned a certain type based on the set of tracking detectors used in its reconstruction. That is, a track merely made of hits from the VELO is assigned the type of **VELO track**.

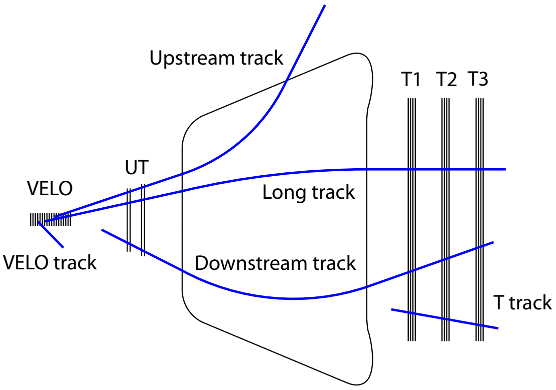


Figure 4.2: Visualization of the common track types in the LHCb's track reconstruction. A subset of VELO, Upstream, and Long tracks are reconstructed in HLT1, while Downstream and T-tracks are only available in HLT2.

Upon further adding hits from the UT to the track, it is upgraded to an **Upstream track** and can be extended to a **long track** if corresponding hits in the SciFi detector are found. A track only reconstructed from SciFi hits is referred to as **T track**, while combining hits from the SciFi and UT will yield a **downstream track**. All these types of tracks are reconstructed by various reconstruction algorithms of which some are not run until the full reconstruction stage in HTL2. The following will give an overview of the main set of reconstruction procedures.[1]

Vertex Locator Reconstruction: The VELO track reconstruction algorithm [42] is usually the first tracking algorithm inside the reconstruction sequence. As mentioned in Section 3.2.2, the detector is located outside the magnetic field. Thus, particles in the VELO follow an almost perfectly straight flight path, modulo some multiple scattering. This and the fact that the pixel sensors produce full 3D measurements lead to the relatively simple algorithm design of the VELO reconstruction. Starting with a hit inside the last module, the algorithm looks for a hit in the preceding module which has a similar ϕ-coordinate. At LHCb, ϕ is defined as the azimuthal angle in the xy-plane. Given the found hit satisfies certain criteria, this step is repeated until all hits of a particle are found. This track forwarding approach is first started for all hits inside the last module before starting with the remaining hits in the module before, and so on. Once a set of hits is found, a straight line fit produces a first approximation of the VELO track's parameters. Next, a simplified Kalman Filter is used to refit the track while accounting for each hit's slight deviation from the straight line due to the effects of multiple scattering in material. However, as no momentum information is available at this time, the Kalman Filter is not able to correctly account for multiple scattering on a per track basis. Rather, it assumes a momentum that works well on average for most tracks and therefore only leads to a small improvement over the straight line fit.

Primary Vertex Reconstruction: Primary vertices (PV) are defined as the position of the pp-interaction. On average an event will contain about five PVs which can be reconstructed from VELO tracks. The Primary vertex reconstruction is performed within LHCb's track reconstruction sequence as the resulting information is required by many selections in HLT1 and HLT2. The reconstruction itself is based on the fact that most reconstructed tracks in the VELO have their origin in one of the PVs. First, tracks are extrapolated to the position at which they are closest to the beamline, and then partitioned into clusters which exhibit a similar z-coordinate of that position. This step, often called seeding, is used to get an approximate set of tracks which are likely to have the same origin. An iterative procedure of fitting a common vertex for a cluster paired with a removal or inclusion of tracks is performed to find the best position of the vertex.

---

[1] Unfortunately, there is often no up-to-date documentation for the latest algorithm developments. The provided overview is based upon the state of the source code [41] at the time of this work. Additional references will be given if available.

Upstream Track Reconstruction: Extending VELO tracks into the UT detector and searching for matching hits is the task of the upstream track reconstruction [43]. An upstream track is only created if the algorithm is able to successfully match at least three UT hits to a VELO track. Particles which reach the UT will pass through an integrated magnetic field of about 0.15 T m which is sufficient to achieve a momentum resolution in the order of 15 % [26]. The algorithm is optimized to find high momentum tracks and provide a momentum and charge estimate for those, as this significantly reduces the combinatorics of finding matching hits in the SciFi tracker.

Standalone SciFi Track Reconstruction: The so called *Seeding* [44] algorithm tries to find tracks by only employing information available from the SciFi. These created T-tracks are used as input for the *Matching* and *Downstream* reconstruction described below. All of these algorithms are however only run in the full event reconstruction performed in HLT2.

Downstream Tracking Downstream tracks are reconstructed by searching for hits in the UT which match to T-tracks. Using the momentum estimate provided by the T-tracks, they are extrapolated into the UT where a search window is opened to look for corresponding hits. The resulting downstream tracks are commonly produced by daughters of long-lived particles like the K^0_S. The current algorithm is heavily based on the Run 2 implementation [45].

Reconstruction of Tracks traversing the entire Detector: Tracks that traverse the entire detector, the long tracks, provide a better momentum resolution than any other kind of track. They are used to perform all selections in HLT1 and are the basis of most physics analysis, making them the most important track category at LHCb. There are two common strategies which are employed to reconstruct these tracks. The *Matching* algorithm [46] produces long tracks based on VELO and T-tracks. First all input tracks are extrapolated to the middle of the magnet. If the algorithm finds a VELO and T-track combination which is close enough that it is able to match them, it produces a long track. Optionally, hits from the UT are added in a last step to reduce the ghost rate and improve the momentum resolution of the resulting long track.

Long tracks are additionally reconstructed by the so called *PrForwardTracking* [47–49]. The algorithm is based on approximating the magnetic field as a thin lens which kinks the flight path at a well-defined position similar to what a light ray experiences when traversing an optical lens. This is a crude approximation as the magnetic field has a non negligible width as well as fringe fields. Nevertheless, it is sufficient to determine a first momentum approximation. The algorithm is able to work with a VELO or upstream track as input which it extrapolates to the above mentioned fixed z-position inside the magnet. For each track, all hits on x layers, so called x-hits, are collected if they are within a defined search window. If the input is a VELO track, this search window is defined by a minimum momentum requirement. However, a charge and momentum estimate of an upstream track will enable an approximation of the flight path through the magnet which in turn allows for a reduced search window size. Next, a momentum estimate is calculated for each hit and the hit is projected along a hypothetical flight path onto a reference plane. This procedure is illustrated in Fig. 4.3. It shows that hits which do not correctly match with a given track will be distributed randomly on the projection plane. However, all hits which where produced by the same particle as the input track should form a dense cluster. A first fit of the hits inside a cluster is used to remove possible outliers before hits from the stereo layers are searched for. Once stereo hits are added and the total number of hits is above a given threshold the complete set of hits is again fitted to produce the final long track. Similar to the *Matching* algorithm, hits from the UT detector can be added in a post processing step if the input was a VELO track.

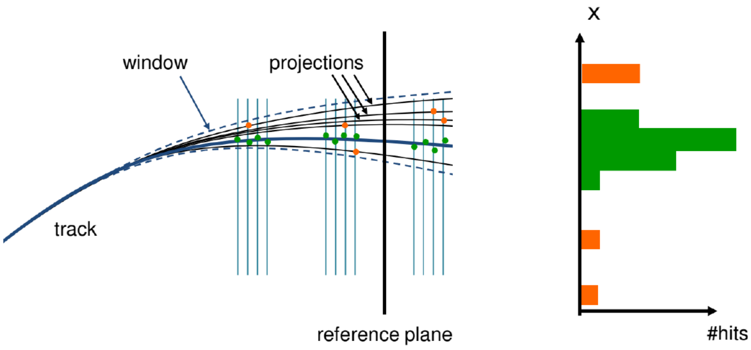


Figure 4.3: Sketch of the projection of x-hits. The true track (blue line) can be reconstructed from the true hits (green). Hits by other particles or noise hits are shown as orange dots. Taken from [49].

During the full event reconstruction performed in HLT2, the *Matching* and *PrForwardTracking* algorithm are both run to find the largest possible set of long tracks. In HLT1 this strategy would require too much processing time, which is why the *PrForwardTracking* tracking is the only algorithm run in HLT1 to produce long tracks.

Final Estimation of Trajectory Parameters: The pattern recognition algorithms usually employ relatively simple fit strategies and do not fit the entire track but merely the hits in the respective subdetector. Thus, a full fit of the track including all its hits is performed at the end of the reconstruction phase to obtain the most accurate estimate of a track's parameters as well as the corresponding covariances. At LHCb, this fit is performed using the *Kalman Filter* [50] method which minimizes the χ^2 of the measurements. An advantage of this method is that it does not require the inversion of large matrices that a global least-squares regression depends on. These inversions are usually difficult to compute and thus require significant amounts of processing time. A further advantage of the *Kalman Filter* is its ability to naturally include multiple scattering and energy loss in its regression. A detailed description of the formalism employed at LHCb is given in [51]

5 LHCb Trigger System

The LHCb detector observes events containing proton collisions about 30 million times per second and as a result produces data at a rate of 4 TB/s. Saving the entire amount of detector data to long term storage devices is practically impossible. For this reason, LHCb employs a trigger system which uses event information to determine whether an event is interesting and should be saved or if it should be discarded. During Run 2 the trigger system consisted of a hardware based trigger stage (L0) and a second High Level Trigger (HLT) stage which was implemented in software. The first hardware stage was used to reduce the event rate to 1 MHz at which the software trigger was able to process the data. This reduction was primarily based on the rejection of background events by using transverse momentum information of the calorimeters and muon stations. However, the planned luminosity increase for Run 3 will result in a fivefold increase of visible collisions for LHCb and a significantly larger fraction of events which contain some form of interesting signal, as shown in [52]. Thus requiring a paradigm shift from a background rejecting trigger to one that additionally has to efficiently categorize events according to determined physics requirements. To enable a trigger system flexible enough to achieve these goals, the L0 has been removed and LHCb will operate a fully software based trigger system [53] which will be required to process events at the rate of 30 MHz.

This chapter provides and overview of the planned upgrade trigger scheme, as detailed in the trigger's technical design report [53]. This is followed by a discussion of the outstanding computing challenges of implementing the described trigger system.

5.1 Trigger Scheme

A general overview of the upgrade trigger scheme is shown in Fig. 5.1. The data which is read out from the detector is processed by the first stage of the High Level Trigger (HLT1). At this stage, only a partial event reconstruction is performed which yields the trajectories of higher momentum particles that have traversed the entire spectrometer. Next, these tracks are used to perform a first set of selections to reduce the rate of events saved to the disk buffer. Accepted events are further used to perform an online calibration and alignment of the entire detector. This is necessary such that HLT2 can perform a full event reconstruction of offline quality. The following HLT2 selections are based upon that reconstruction and decide if an event should be saved to disk for later analysis.

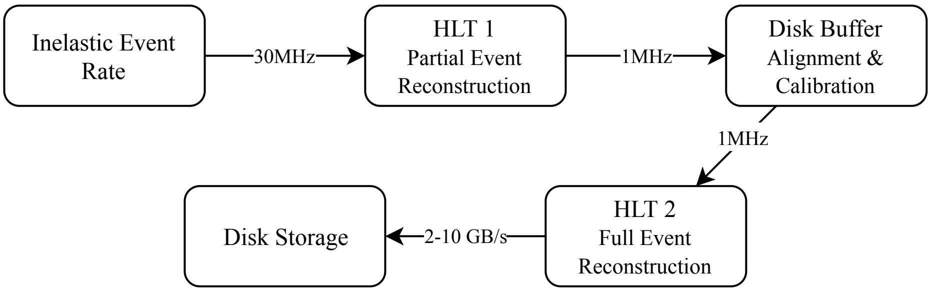


Figure 5.1: Illustration of the dataflow of the upgrade trigger system. Quoted values for event rate and bandwidth correspond to current best estimates as stated in [53].

The following subsections provide an overview of the inner workings of the mentioned trigger stages. As the work presented in this thesis focuses on the HLT1, the first stage is covered in more detail than the remaining ones.

The first trigger stage: The goal of HLT1 is to reduce the incoming event rate to the order of 1 MHz by selecting events which are likely to be of interest. The current selection strategy of HLT1 is based upon its predecessor from Run 1 and 2. It aims to select events which either contain a secondary decay of a heavy flavor hadron or a prompt decay producing at least one muon. Figure 5.2 visualizes the different topologies of these decays. On the left, an example of a heavy flavor hadron decay. The B^0 meson only decays after a short flight distance and thus the decay vertex is displaced from the primary vertex. As a result, the trajectories originating from such decays can commonly be identified by their large impact parameter (IP), as shown in Fig. 5.2. On the right, an example of a prompt decay. The Z^0 boson decays instantaneously, producing two high momentum muons in the process. Here, decay vertex and primary vertex are indistinguishable, thus the muons seem to originate from the primary vertex. A selection of this topology enables studies of many different kinds of prompt decays, ranging from the aforementioned electroweak decay of Z^0 to more exotic analyses like the search for dark photons [54]. Prompt decays which do not exhibit any muons in their final state are difficult to distinguish from the hadronic background which also originates from the primary vertex. For this reason these decays are only rarely searched for at LHCb, as they require very stringent selections to reject the mentioned background.

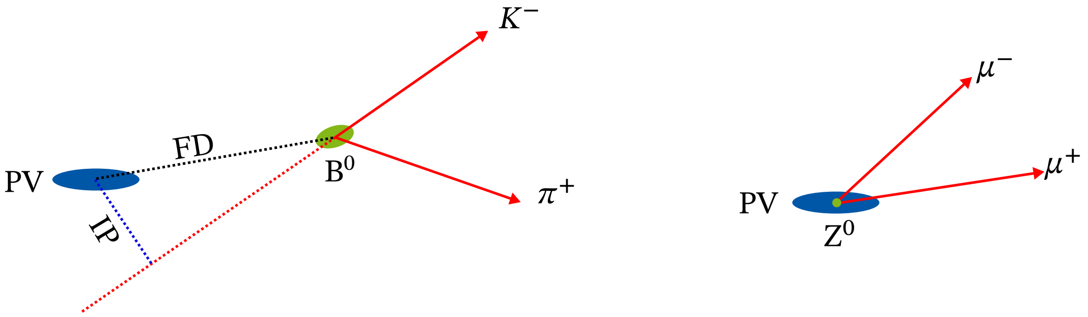


Figure 5.2: Two decay topologies of interest to LHCb, secondary decay (left) and a prompt decay with a final state that includes muons (right). On the left, a B^0 meson is created inside the primary vertex (PV) and decays after a short flight distance (FD) into a kaon and pion. The impact parameter (IP) of the kaon is visualized. On the right, a Z^0 is created in the pp-collision and decays promptly into two muons.

A selection of the two desired event topologies requires different strategies. Yet, they can usually be based upon the same partial event reconstruction sequence. Similar to Run 2, the current design proposal of the upgrade trigger [53] foresees a sequence that starts by first reconstructing VELO tracks. Next, the VELO tracks are used to reconstruct primary vertices as well as upstream tracks. Resulting upstream tracks are further used by the *PrForwardTracking* algorithm to reconstruct long tracks. After performing a fit of the found long track the upfront reconstruction is finalized and selections, potentially including additional reconstruction steps, are used to select the different topologies.

A combination of a reconstruction sequence and a specific HLT1 selection is commonly referred to as HLT1 line. These lines do not yet aim to completely reconstruct a specific event topology but instead search for partial decay signatures that are common in many interesting decay topologies. An event is accepted by the first trigger stage if any line yields a positive decision. Two important quantities for any line are its selection efficiency and its output rate. The former is the efficiency with which a signal is selected if one is present while the latter quantifies the rate of positive decisions when processing data as it is seen by the detector.

Selections of prompt decays are performed by multiple HLT1 lines. They differ in the number of muon tracks they require and in the requirements they impose on these tracks. All of these lines first perform an additional reconstruction step to combine the long tracks with the information of the muon stations to only select long tracks produced by muons. The single muon line requires a single muon track to satisfy certain momentum and quality criteria to accept an event. Lines which look for multiple muons loosen their momentum requirements but require these tracks to have a common vertex. Further transverse momentum requirements can be imposed on these vertices.

Heavy flavor decays are selected by two HLT1 lines. Both lines use the fact that decay products of heavy flavor hadrons commonly exhibit higher transverse momentum than tracks from light quark background and are displaced from the primary vertex. The one-track HLT1 line selects events that include at least one long track that satisfies its requirements on the significance of the impact parameter χ^2_{IP} and a minimum transverse momentum. The significance of the impact parameter is a quantity commonly used instead of the raw impact parameter (IP) as it additionally takes into account the resolution of the IP and primary vertex. The two-track line first uses similar but looser requirements to select a subset of long tracks. If a combination of any two of these yields a good vertex which satisfies some momentum and quality criteria a positive decision is returned.

All mentioned momentum, quality, and other requirements are adjustable parameters which are used to control the output rate of a HLT1 line. The limit on the combined output rate of all lines is determined by the processing rate of HLT2, which is estimated to process events at a rate of 1 MHz [53]. The requirements to keep the overall output rate within this limit leaves tracks below a transverse momentum of 500 GeV unused [53]. For this reason, it is common to only reconstruct long tracks with a transverse momentum above 500 MeV to reduce the processing time of the reconstruction sequence in HLT1. These long tracks provide a good basis for many selections including the prompt HLT1 lines. Nevertheless, it should be noted that decays exist which can not be selected by this strategy. A selection of special decay topologies can be performed by introducing HLT1 lines with a special reconstruction. Depending on their required processing time and output rate they can potentially only be executed on a subset of the processed events. In conclusion, the requirement on the transverse momentum is a working point which represents a trade off to provide a general upfront reconstruction for most of LHCb's physics program.

Alignment & Calibration: First developed in Run 2, the real-time alignment and calibration system [55] is a crucial part of the Run 3 trigger system. This system uses data collected by the first trigger stage to automatically determine the alignment and calibration constants needed to perform an offline quality reconstruction in HLT2. The alignment tasks include the alignment of the entire tracking system, the muon stations, and RICH mirrors, while the calibration tasks calibrate the gas refractive index and photon detectors of the RICH as well as the electronic calorimeter.

The second trigger stage: HLT2 aims to provide a full event reconstruction of offline quality such that no further offline processing is required to perform physics analyses. For this purpose the reconstruction previously performed in HLT1 is not sufficient as it is a compromise between efficiency and speed and thus aimed to only reconstruct higher momentum tracks. The reconstruction in HLT2 however focuses on finding as many tracks as possible to enable the full reconstruction of events. To achieve the highest efficiency of finding long tracks, HLT2 executes two mutually redundant long track reconstruction algorithms, the *PrForwardTracking* and *Matching* algorithm. In contrast to HLT1, the *PrForwardTracking* algorithm uses VELO tracks as input instead of upstream tracks to avoid small inefficiencies of the upstream reconstruction at the cost of higher processing times. In addition to the long tracks, HLT2 also reconstructs downstream tracks. Once all track reconstruction algorithms are finalized, the RICH and calorimeter reconstruction is performed, providing particle identification hypotheses for the reconstructed tracks. At this point HLT2 performs selections to decide whether to save or discard an event. An important constraint of HLT2 is the output bandwidth of events to be saved on disk, which can be approximated by

$$\text{Bandwidth} \propto \text{Trigger Rate} \times \text{Average Event Size}. \tag{5.1}$$

The available bandwidth during Run 3 is expected to be between 2 to 10 GB/s [53]. As a result the rate of events which HLT2 can accept depends on the average size of an event when saved to disk. To maximize the amount of events which can be used for analyses, most HLT2 selections in Run 3 will make use of a reduced event format [56, 57] which only stores the quantities needed in a later physics analysis, instead of the full raw data of an event. This procedure was successfully tested during Run 2 where it significantly increased the physics reach of many analyses [57].

5.2 Open Computing Challenges

The original technical design report of the High Level Trigger [58] outlined a detailed plan on how to implement the described software trigger. This allocates a budget of 2.8 million Swiss Francs to procure commodity x86 CPU servers to assemble a computing farm on which the trigger will be run. Nevertheless, a later reevaluation of the necessary computing resources in [59] revealed that the reconstruction algorithms were roughly a factor six too slow to enable a HLT1 processing rate of 30 MHz on the given computing farm. It is shown that this was mostly due to unmet assumptions regarding the expected CPU performance growth [59]. The processing speed of HLT2 was not part of the above reevaluation, yet it is likely that the actual performance falls short of the design performance by a similar factor. While the HLT2 is, of course, a substantial part of the overall trigger system, it is clear that the first step towards a functioning trigger system is an implementation of an HLT1 which is able to process all incoming events. For this reason, the most recent trigger evaluation [60] as well as this thesis focus on the HLT1.

The track reconstruction algorithms are responsible for the majority of the processing time of HLT1. Thus, a significant increase of the processing rate can only be achieved by a reduction of the processing time of these algorithms. The improved processing rate of most recent trigger performance evaluation [60] is based on different improvements which fall within one of two categories, as listed below.

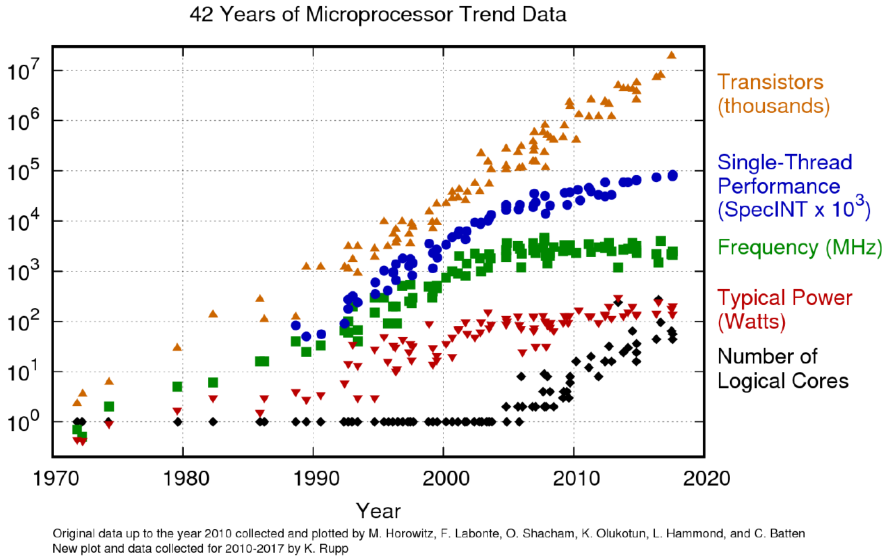
Algorithm Requirements & Selections: Improvements of this category reduce the processing time by reducing the required computations a reconstruction algorithm needs to perform. However, they come at the cost of fewer reconstructed tracks and thus reduce the overall physics output of the trigger. The two used methods in [60] include:

First, more stringent requirements on the minimum transverse momentum in the upstream and long track reconstruction are used to reduce the necessary combinatorics and avoid the reconstruction of tracks with a lower transverse momentum.

Second, a selection after the VELO tracking based upon a track's impact parameter (IP) is introduced. The IP cut at this stage is beneficial to the overall processing performance of the reconstruction sequence as it greatly reduces the amount of tracks which need to be processed by the following algorithms. However, this cut removes most of the prompt tracks making the selection of prompt decay topologies based upon this reconstruction sequence not possible. While a similar cut is commonly performed on long tracks to select secondary decays, the IP cut on VELO tracks also comes with disadvantages for this decay topology. Due to the missing momentum information for VELO tracks a cut on the raw IP has to be used. However, as the impact parameter resolution is better for tracks with a high transverse momentum, the cut on the IP is tighter for them than for tracks with lower transverse momentum. A similar cut was necessary in the trigger system of Run 1 [61, 62] while an increase of the computing farm and algorithm optimizations later enabled a removal of the IP cut for the Run 2 trigger [55].

Software Optimizations: The software optimizations can be split into optimizations of the underlying GAUDI framework and improvements to the reconstruction algorithms. Both require detailed knowledge of the x86 CPU architecture and, in case of the latter, also of the reconstruction procedures and their underlying physics processes (see Section 4.1). For this reason, developing these improvements can take a long time, yet the advantage in respect to the previous category is that the resulting tracking efficiency and thus the physics output of the trigger remain mostly unchanged.

LHCb invested a significant effort into ensuring that the GAUDI framework would utilize the available resources of modern CPUs most efficiently. Figure 5.3 shows how the performance of modern CPUs has developed since the 1970s. The significant drop of single thread performance growth is in accordance with the mentioned unmet growth predictions in [59]. Alongside this growth decrease, it is visible that the amount of cores on a single CPU has grown significantly. Therefore, to fully utilize the processing power of a modern CPU it is necessary to perform multiple things in parallel to make the best use of all available cores. This kind of parallelism is often referred to as *task parallelism*. For Run 3, the GAUDI framework was overhauled to enable this form of parallelism [58]. Each reconstruction algorithm is now represented as an independent task which can run simultaneously to other tasks, given its inputs are available. For example consider the reconstruction of a single event. The upstream tracking can not be executed in parallel with the VELO tracking as it requires the output of the VELO tracking, the VELO tracks, as input. Yet once the VELO tracking is finalized, the upstream tracking as well as the pri-



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Figure 5.3: Development of microprocessor performance characteristics during the last 42 years. Taken from [63].

mary vertex reconstruction can be executed in parallel. Given that most algorithms in the HLT1 reconstruction sequence have a dependency on the previous one, the GAUDI framework additionally enables the reconstruction of multiple events at the same time to fully utilize all available CPU cores. The author has made significant contributions to the effort of adapting the existing algorithms to work within the new GAUDI framework and validating their correctness. However, as this work is of very technical nature it will not be further discussed within the scope of this thesis.

With task parallelism fully utilized, the further optimization of individual reconstruction algorithms is left focusing on other areas. Modern CPUs additionally provide special *Single Instruction Multiple Data* (SIMD) [64] instructions which enable an *instruction level parallelism*. These instructions enable the programmer to perform the same instruction on multiple input data at the same time as is visualized in Fig. 5.4. Adopting existing algorithms to operate on a set (vector) of data instead of scalar values is often referred to as vectorization. This can be complicated and potentially require a rewrite of significant portions of an algorithm's source code. At the time of the performance evaluation in [60], only few parts of the overall software were adapted to increase the use of these instructions.

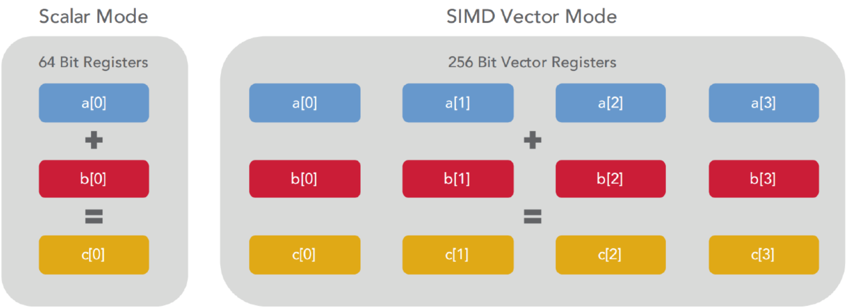


Figure 5.4: Visualization of a scalar and vectorized operation. Taken from [65].

Overall, the combination of the listed improvements yielded an estimated HLT1 throughput of 12.4 MHz. The majority of this speedup compared to the first reevaluation is due to the used IP cut and transverse momentum requirements [58, 60]. It results in the combination of *Upstream tracking* and *PrForwardTracking* only accounting for a bit over 5 % of the measured computing time [60], while the *PrForwardTracking* usually requires the largest fraction of processing time if no IP cut is used. The VELO reconstruction requires roughly a third of the processing time, with a large part of the remaining time being spent in steps which decode and prepare the incoming detector data for further processing.

Following this last evaluation, the implementation of an HLT1 that reaches the necessary processing rate of 30 MHz, while maintaining high selection efficiencies for all decay channels of LHCb's broad physics program, remains to be an open challenge. Introducing even tighter selections within the reconstruction sequence to increase the throughput would further lower the overall physics efficiency and is therefore considered to be the last resort. Therefore, the only alternative is to even better leverage the processing performance of the available computing resources. There is an ongoing effort at LHCb to further improve the use of SIMD instructions within existing algorithms as well as to develop new algorithmic approaches to enable the best possible throughput for HLT1 [58]. It should be noted that there is also an investigation into the viability of using other computing hardware to accelerate the processing of HLT1 [58]. The considered alternatives are mainly graphic processing units (GPU) or field programmable gate arrays (FPGA). Yet, the baseline solution is the usage of x86 CPUs and will be the architecture considered in this thesis.

The final trigger strategy which will be employed during Run 3 will depend heavily on the success of the ongoing optimization efforts. A simplified analysis yields three different scenarios. The first scenario, the most pessimistic, is one in which the improvements will not enable the above outlined reconstruction sequence to run at the required rate of 30 MHz. This will in turn require further tightening of selections to reduce processing time and will inevitable come at the cost of reducing the physics output of the trigger.

If a significant improvement can be achieved, an important question will be whether it is possible to remove the IP cut. For the scenario in which the cut is needed, Chapter 6 presents an approach to replace the IP cut with a more efficient alternative.

A removal of the IP cut significantly increases the amount of tracks which need to be processed by the following reconstruction algorithms. In this case, most of the additional time is used by the *PrForwardTracking*, while the *Upstream Tracking* only exhibits a small increase in processing time. For this reason, Chapter 7 studies the possibility of an alternative algorithm for the reconstruction of long tracks in HLT1. The goal of the new algorithm is to significantly reduce the required processing time and thus enable the removal of the IP cut.

6 Improving Impact Parameter Selections of VELO Tracks

Selecting tracks with a high impact parameter is a commonly used procedure to determine which tracks likely originate from secondary vertices. It is an important selection for many of LHCb's analysis. Ideally, it is possible to postpone this selection until after the reconstruction of long tracks, at which point it is possible to perform them with better efficiency. Yet, as mentioned before, the stringent timing requirements of the trigger reconstruction might force LHCb to perform an impact parameter selection after the VELO reconstruction.

A simplified sketch of the VELO, including a typical signal decay, is shown in Fig. 6.1. It visualizes the impact parameter as the distance between the primary vertex (PV) and the point on a track at which it is closest to the PV (CTPV). As the reconstruction of primary vertices is based on the VELO tracks, the CTPV position can not be determined during the VELO track reconstruction. Subsequently, the Kalman Filter in the VELO reconstruction extrapolates the track to the closest to beam position (CTB). This position is usually close to the CTPV position and contained within the VELO's vacuum. For this reason, the CTPV position can be obtained by a straight line extrapolation from the CTB point, as no multiple scattering between these points is expected.

This procedure corresponds to the IP selection which was done in LHCb's trigger during Run 1. In Run 2 these selections are delayed until after the reconstruction of long tracks. The momentum information available for long tracks is crucial to improve the quality of an IP cut. Based on this momentum information, the Kalman Filter is able to correctly account for the momentum dependent effects of multiple scattering and therefore estimate correct uncertainties on its predictions. This is beneficial as it allows to determine the significance of the impact parameter χ^2_{IP} which includes a reconstruction quality of the CTPV and PV position, enabling a better separation of prompt and secondary tracks.

As detailed in Section 4.2, the Kalman Filter at the VELO tracking stage assumes a constant momentum for all tracks. This subsequently leads to incorrect uncertainty estimates for the determined CTB position. The following section introduces an

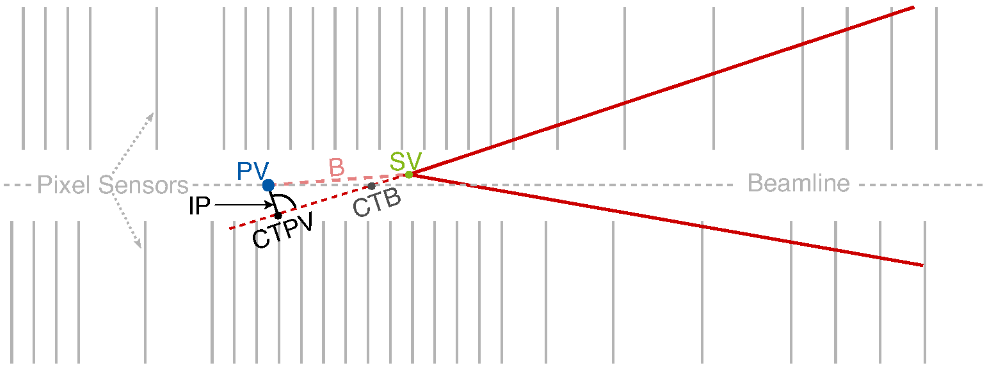


Figure 6.1: Sketch of B meson coming from the primary vertex (PV) and decaying inside the LHCb Vertex Locator into two daughter particles at the secondary vertex (SV). Additionally, the closest to beam position (CTB), closest to PV position (CTPV), and the impact parameter (IP) are shown.

alternative method which aims to tackle this shortcoming of the default VELO track fit to provide an improved uncertainty estimate. Subsequently, Section 6.2 evaluates and compares the performance of an IP-based selection using the new method to one that uses the Kalman Filter. A large part of the following work has been previously presented at the *Computing in High Energy Physics* conference and was later published as part of the conference proceedings [66].

6.1 Alternative VELO Track Fit

The following subsections introduce an alternative track fit to the existing Kalman Filter inside the VELO reconstruction. Its goal is to provide an estimate of the CTB state (Eqs. (6.1) to (6.5)) that is equally precise to that of the Kalman Filter while providing an improved estimate of the uncertainty on the CTB prediction. To achieve this, a neural network based method is designed to learn to deduce a track's momentum and use this information to provide the above estimates. The general approach is explained in more detailed in the following subsection. Section 6.1.1 provides a brief introduction of neural networks and the used architectures before outlining the method's implementation and design choices. Finally, Section 6.1.2 will detail the used data set and training procedure.

As mentioned in Section 4.2, the VELO reconstruction performs a straight line least squares fit of each track before the Kalman filter is used to improve the fit results.

The results of the first fit yield an approximation of the track's parameters which can be used to determine a first estimate of the CTB state:

$$x = x_0 + t_x \cdot z \tag{6.1}$$

$$y = y_0 + t_y \cdot z \tag{6.2}$$

$$z = -\frac{x_0 \cdot t_x + y_0 \cdot t_y}{t_x^2 + t_y^2} \tag{6.3}$$

$$\frac{\delta x}{\delta z} = t_x \tag{6.4}$$

$$\frac{\delta y}{\delta z} = t_y, \tag{6.5}$$

with the track's x- and y- slope t_x, t_y and intercept x_0, y_0. This approximation is relatively close to what the current Kalman Filter based method achieves as depicted in Fig. 6.2. Additionally, Fig. 6.2 shows that the difference in resolution is more significant in the case of the position while the comparison of the slope's resolution only shows minor differences. Taking into account that the slopes are only used for the extrapolation from the CTB to the CTPV state, which are usually small, the differences in the slope will not significantly impact the estimation of the IP. Therefore, the focus of the following method is to improve the position of the CTB state while ignoring the slight discrepancies in the slopes. This choice is made in an attempt to keep the neural network's size and complexity as low as possible. If a resolution of the position is achieved at which the slopes' resolution becomes the limiting factor, this decision can be reevaluated.

Focusing again on the left plot in Fig. 6.2, the worse resolution of the linear regression is mainly due to the fact that it has no information of the process of multiple scattering and thus treats the scattering deviations from a straight trajectory incorrectly.

The goal is to correct for this effect by defining a machine learning model which is able to use the approximated CTB state and improve upon it based on the observed hit-residuals $\vec{r_i}$

$$\vec{r_i} = \begin{pmatrix} x_{hit} - (x_0 + (z_{hit} - z) \cdot t_x) \\ y_{hit} - (y_0 + (z_{hit} - z) \cdot t_y) \\ z_{hit} \end{pmatrix}, \tag{6.6}$$

between the straight line estimation and the hits. Assuming these residuals are mainly due to multiple scattering, they will be directly correlated with the scat-

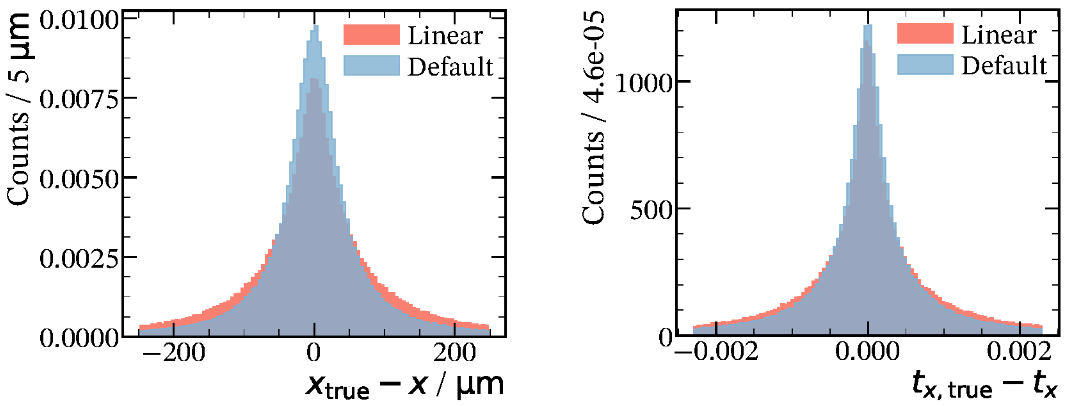


Figure 6.2: Resolution of the x-position (left) and x-slope (right) for the linear least-squares regression (red) and that of the default simplified Kalman Filter (blue).

tering angle θ_{plane} depicted in Fig. 4.1 and defined in Eq. (4.3), here repeated for convenience.

$$\sigma(\theta_{plane}) = \frac{13.6\,\text{MeV}}{\beta c p} z \sqrt{\frac{x}{X_0}} \left[1 + 0.038 \ln\left(\frac{xz^2}{X_0 \beta^2}\right)\right].$$

Equation (4.3) shows that the multiple scattering angle is directly correlated with a particle's momentum. A neural network should therefore be able to learn to deduce an approximated momentum from the provided residuals and use it to describe the multiple scattering magnitude. Given that the above assumptions hold, the neural network should furthermore be able to use the gained knowledge to improve the estimation of the CTB state's position and predict its uncertainty.

6.1.1 Model Design

The method explained above defines the inputs of the model to be the approximated CTB state as well as the hit-residuals $\vec{r_i}$. The output is the improved prediction of the position (x, y, z) as well as an uncertainty estimate for these values, which determines the model's output dimension to be fixed in size. The input dimension is more complicated as it can vary for every track since the number of hits on a track can range from as low as three up to more than 15. In addition to these two constraints, any proposed model design should ensure that its architecture or size will not limit its later use in a trigger environment due to too high inference time.

Given these constraints, it is decided to build a model based on two commonly used neural network architectures. The first being a standard dense neural network layer as well as a Long-Short-Term Memory (LSTM) [67] neural network unit. A brief overview of these methods is given below.

A neural network is a common machine learning algorithm used to solve classification and regression tasks. Its design is inspired by biological neural networks. A visualization of a feed forward neural network, or Multilayer Perceptron, is shown in Fig. 6.3. For simplicity the depicted network only consists of an input, hidden, and output layer with a few neurons. Though, it is not uncommon for networks to have many layers each made up of hundreds of neurons. Neural networks with multiple hidden layers are classified as deep neural networks.

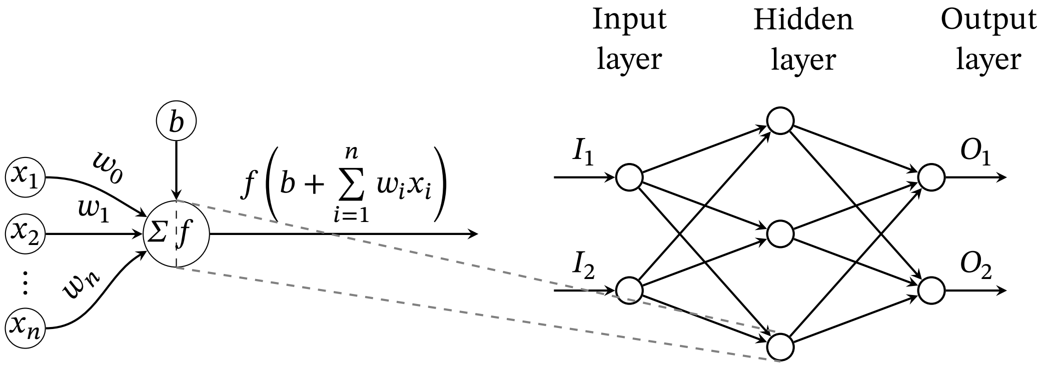


Figure 6.3: Visualization of a feed forward neural network including a close up view of a single neuron. The function f represents the neurons activation function. Figure adapted from [68].

The calculations performed inside a neural network can be explained by looking at the inner workings of a neuron. As shown in Fig. 6.3, a neuron's output is determined by applying an activation function f to the sum of the neuron's bias and weighted inputs. Many different choices for f exist, common ones are the hyperbolic tangent or sigmoid function. To use a neural network to solve a specific task, one first has to train it. This only requires a training data sample which includes the correct output for every input sample it provides. The fact that no task specific knowledge is required for this learning procedure is a major advantage over other methods. Training the network is performed by iteratively evaluating each sample of the data set and determining the observed error between neural network's output and the correct solution as given by the training sample. Based on the observed errors, an overall loss is determined by the means of a loss function, e.g. mean squared error. Next, the weights and biases are updated by using a method called backpropagation [69]. It does so by computing the gradient of the observed loss with respect to each weight or bias. An important variable for this process is the learning rate which quantifies the size of the update step along the respective gradient. By repeating this learning step many times the backpropagation method will eventually lead to the set of weights and biases which minimize the given loss function. During the learning iterations, the neural network's loss is additionally evaluated on a so called validation data sample. Comparing the evolution of the observed loss on both samples ensures that the training can be stopped once the loss of the validation sample starts to increase thus indicating the neural network is overfitting to the training sample.

While Multilayer Perceptrons explained above have many applications in todays world, one of its downsides is the fact that its dimensions are fixed. From this follows that the neural network is not able to process sequential data with no predetermined size. This is problematic for the given use case as a track can be seen as a sequence of 3 to 15 hits.

A possible solution is the usage of a recurrent neural network (RNN). Their architecture is visualized in Fig. 6.4, which shows an unrolled RNN processing a sequence $x_0, \dots, x_t$. At each iteration the RNN cell, here denoted A, uses its internal state from the previous iteration as input to the next iteration and outputs a hidden state h_i. The last hidden state h_t contains information about the entire processed sequence. It can either directly be used to for example predict the next character of a word or be processed by a simple neural network to derive the desired output. Different kinds of recurrent neural networks have been developed which mostly differ by their inner workings of the RNN cell. One of the simplest versions of an RNN is the Elman network [71] which can be described by a single formula,

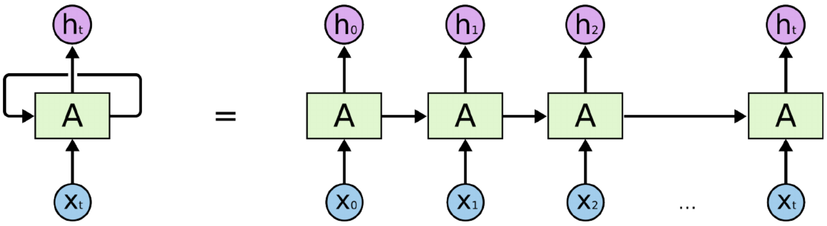


Figure 6.4: Visualization of an unrolled recurrent neural network [70]. The input sequence is given by $x_0, ldots, x_t$, which is processed by the RNN cell A to produce the hidden states h_i.

$$h_t = \tanh\left(W_i x_t + b_i + W_h h_{t-1} + b_h\right), \tag{6.7}$$

where W represents a matrix of weights and b a vector of biases. The suffix i or h indicate whether the weights and biases are used to process the input vector or hidden state vector.

It is shown in [72] that this architecture has problems to retain the information of early parts of a sequence when processing long sequences. A solution to this is provided by the LSTM [67] neural network which use a more complex procedure to memorize information. This involves four neural network layers inside their cell, as shown in Fig. 6.5. Their behavior is governed by the following equations,

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{6.8}$$

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{6.9}$$

$$\tilde{c}_t = \tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right) \tag{6.10}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t \tag{6.11}$$

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \tag{6.12}$$

$$h_t = o_t \otimes \tanh\left(c_t\right). \tag{6.13}$$

One of the main innovations of the LSTM is the introduction of a cell state c_t which acts as memory and is passed to the subsequent cell. The layer f_t holds values between zero and one and is often called *forget gate* as its multiplication with the cell state leads to the erasure of some parts of the LSTM's memory. The layers i_t (*input gate*) and $\tilde{c}_t$ can be interpreted as the new memory which is added to the previous memory in this cell's processing. At last, the LSTM will output its hidden state h_t which is determined based upon the current cell state c_t and the layer o_t, referred to as *output gate*.

LSTMs have shown great success capturing longer term correlations in sequential data. Considering that a track is essentially a sequential data set of hits, the LSTM is an excellent choice for our studies, especially, due to its capability to process input data of arbitrary length.

A sketch of the model's architecture is shown in Fig. 6.6. In the first stage of the model the hit-residuals $\vec{r_i}$ are processed by an LSTM in reversed order. This is done to ensure that the hits which are more sensitive to the CTB state's position have a stronger correlation to the LSTM's output. The second stage is based on two independent single layer feed forward networks, which each take as input a concatenation of the LSTM's output and the approximated CTB state $(x, y, z, \frac{\delta x}{\delta z}, \frac{\delta y}{\delta z})$. One of them will be trained to estimate the position of the CTB state while the other one will predict the uncertainty of that prediction.

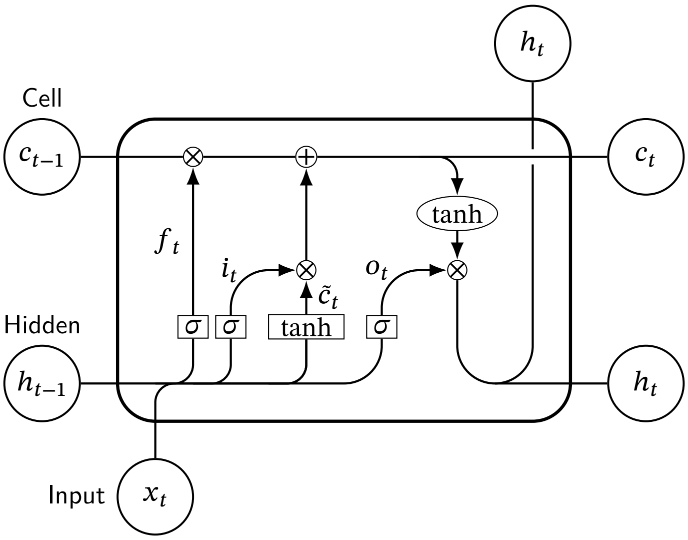


Figure 6.5: Visualization of an LSTM cell [70]. The input sequence is given by x_t, the hidden state by h_t, and the cell state by c_t. Elementwise multiplication and addition are represented by $\bigotimes$ and $\bigoplus$ respectively. The ellipse labeled by tanh represents an elementwise application of the hyperbolic tangent function. Rectangular boxes are neural network layers, thus they include weights, biases and an activation function of either the sigmoid function σ or the hyperbolic tangent. Merging lines imply a concatenation of the incoming vectors. Figure adapted from [73].

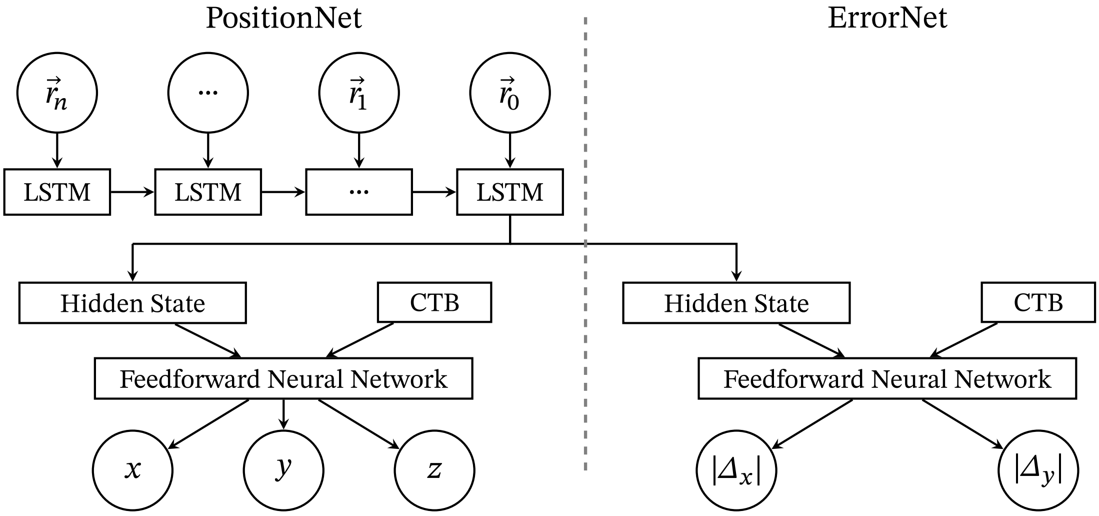


Figure 6.6: Visualization of the chosen model architecture to predict the x,y, and z position and its uncertainty of the CTB state.

6.1.2 Network Training Procedure

The data used during training and validation is extracted from reconstructed LHCb Monte Carlo events. To avoid biasing the network towards specific signals, it is important to define a training data set which sufficiently covers the phase space of possible decay channels. Simulated minimum bias data samples which aim to mimic the data produced in p - p collisions before any selections would be good candidates. However, these samples normally contain only a few signal events. For this reason, a sample of simulated signal events which contains a wide range of possible b-decays is chosen (Appendix A.1, Sample 1). These should sufficiently sample the phase space of possible VELO tracks while also yielding a large enough data set to achieve a good generalization during training.

The training of the described model is performed in two distinct stages. At first the training focuses on the part of the model that predicts the position (left half of the sketch in Fig. 6.6), further referred to as the PositionNet. The true CTB state's position, the networks learning target, is determined based on the Monte Carlo truth information and used to evaluate the performance of a prediction. To calculate the error of a prediction, the predicted position is translated to the z-position of the simulated CTB state to calculate the differences in x and y at the correct z-position:

$$CTB_p = (x_p, y_p, z_p) \tag{6.14}$$

$$CTB_t = (x_t, y_t, z_t) \tag{6.15}$$

$$\Delta_x = x_t - \left[x_p + (z_t - z_p) \cdot t_x\right] \tag{6.16}$$

$$\Delta_y = y_t - \left[y_p + (z_t - z_p) \cdot t_y\right] \tag{6.17}$$

The subscript p and t indicate predicted and Monte Carlo truth information respectively. The slopes t_x and t_y are taken from the approximated CTB state. For the training of the network we use the following linear loss function,

$$\mathbb{L} = 100 \cdot (|\Delta_x| + |\Delta_y|) + |z_t - z_p|, \tag{6.18}$$

using Eq. (6.16), Eq. (6.17), and the difference of the predicted and true z−position. This loss allows the network to focus on predicting a CTB position which is on the right trajectory but not necessarily at the perfect z-position. It was observed that this loss leads to slower convergence than the often used Mean-Squared-Error (MSE) loss function, yet the final results are observed to be better. Additionally, it is observed that the training improves significantly by using a cyclic learning rate, similar to what is introduced in [74].

For the second stage of the training, the weights and biases of the PositionNet are fixed. The training and validation data set are fed into the network to calculate an observed absolute error $|\varDelta_{x,y}|$ which will be used as the training target for the dense network depicted on the right half of Fig. 6.6 (ErrorNet). A standard MSE loss is used to train the ErrorNet, which will result in the network being optimized to, on average, correctly estimate the absolute error. Assuming the prediction will approximately represent the mean absolute deviation, it is possible to relate this to the standard deviation

$$\sigma_{x,y} = \sqrt{\pi/2}\langle|\varDelta_{x,y}|\rangle. \tag{6.19}$$

6.2 Results

To compare the performance of the presented method to that of the Kalman Filter based method multiple cross checks are performed. The first comparison focuses on the central values of the predicted CTB position and the achieved precision on the impact parameter. As the impact parameter is a non-negative distance, the residual distribution between predicted and true value does not exhibit a Gaussian shape and makes it difficult to define a meaningful resolution. For this is reason it is common to study the resolution of the predicted x- or y-position of the CTPV position instead. The true CTPV position is determined based on the available truth information in simulated data. This position is then compared to the result of extrapolating the predicted CTB state to the CTPV's true z-position.

Figure 6.7 shows the observed residuals between the predicted x- and y-position and their true value for the current method in comparison to the new LSTM based method. As shown, the presented approach is able to achieve a resolution that is slightly better than that of the default method. This validates the prior assumption that the estimated values of the least-squares fit for t_x and t_y are sufficiently precise. As the predicted CTB position is slightly more precise, it is fit to be used in subsequent reconstruction steps. This is important as the CTB state is also an input to the primary vertex reconstruction.

Additionally, to the momentum integrated view above, Fig. 6.8 shows the observed x-resolution as a function of the inverse transverse momentum of the simulated particle. It shows that the observed error is linearly correlated to the inverse transverse momentum. This is in accordance with the predicted $1/p_T$ dependence of the error,

$$\sigma^2_{\mathrm{IP}x} = \frac{r_1^2}{p_\mathrm{T}^2}\left(0.013\,6\,\mathrm{GeV}\sqrt{\frac{x}{X_0}}\left(1 + 0.038\ln\left(\frac{x}{X_0}\right)\right)\right)^2 + \sigma^2_{\mathrm{extrap}}, \tag{6.20}$$

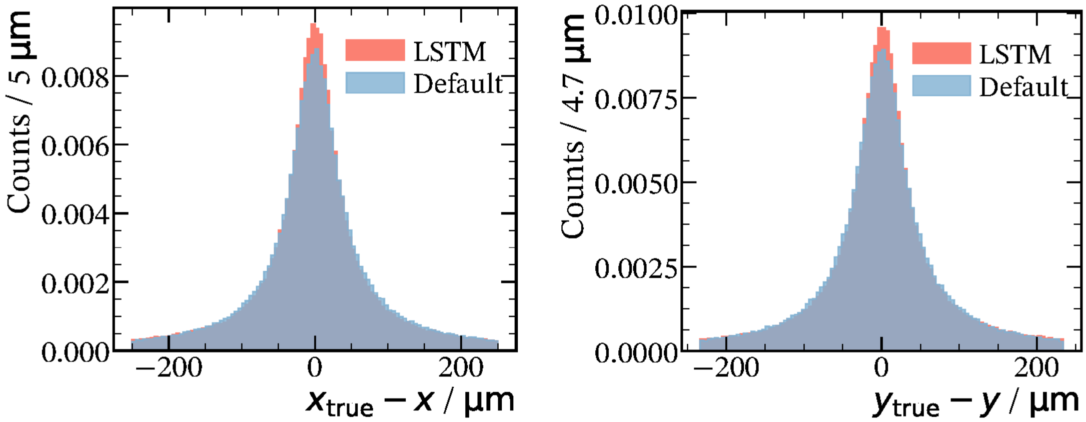


Figure 6.7: Comparison of the CTPV's x- and y-resolution between the ML based method and the default simplified Kalman Filter.

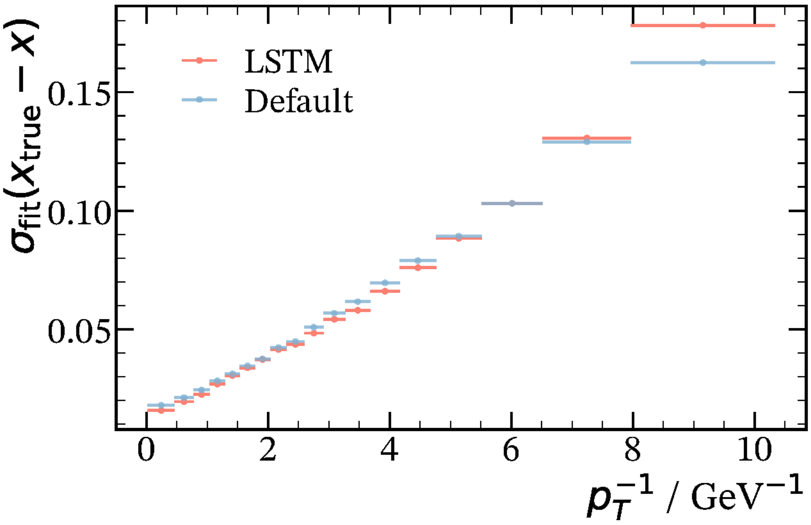


Figure 6.8: Observed resolution of the x-coordinate of the CTPV position for the default and LSTM based method. The value of $\sigma_{\text{fit}}(x_{\text{true}} - x)$ is determined for each bin by fitting a Gaussian distribution to the core of the observed residual distribution. Each bin contains 5 % of the tracks in the studied data set. Errors on σ_{fit} are omitted as they are below 1 % and thus not visible on this scale.

formulated in [27]. Here, r_1 represents the radius of the first measured point in the VELO and x/X_0 quantifies the fractional radiation length the track encounters before the second hit. The last term σ^2_{extrap} is a momentum independent offset that mostly depends on the VELO's hit resolution.

To evaluate the accuracy of the predicted uncertainties, a study of a so called pull distribution is performed. A pull distribution shows the observed residual between predicted and true value divided by the predicted uncertainty. In the scenario of correctly estimated uncertainties, one should observe a Gaussian distribution with $\mu = 0$ and $\sigma = 1$. Similarly to Fig. 6.8, the pull distribution is determined in bins of inverse transverse momentum. For each bin, a Gaussian model is fit to the observed pull distribution to determine the observed standard deviation. The result is shown in Fig. 6.9. Looking at the observed standard deviations of the default method (blue curve), it clearly shows a linear $1/p_T$ dependence. This is to be expected as the current method does not incorporate any knowledge of a track's momentum, and thus predicts an uncertainty that is constant across the momentum range. Subsequently, the observed $1/p_T$ dependence is in fact very similar to the that of the resolution in Fig. 6.8 divided by a constant factor across all bins. In the case that

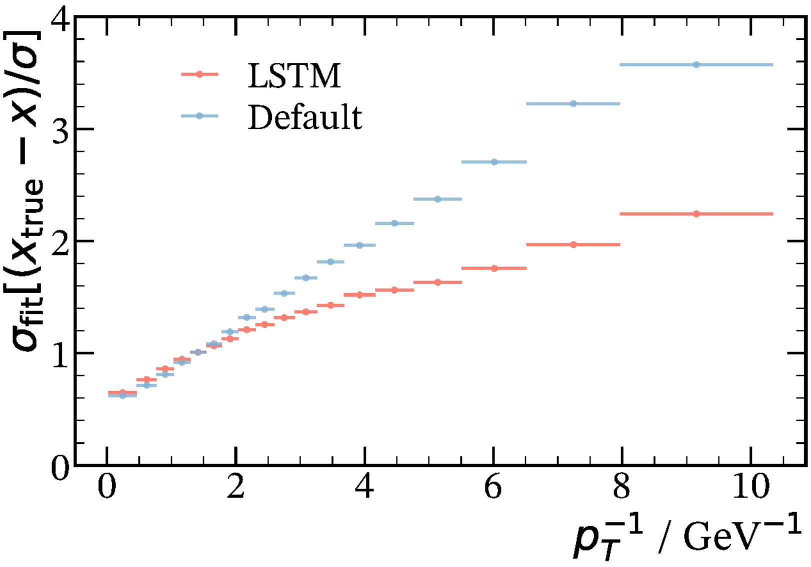


Figure 6.9: Observed values of σ for a pull distribution of the CTPV's x-position, shown for the default and LSTM based method. The value of $\sigma_{\text{fit}}[(x_{\text{true}} - x)/\sigma]$ is determined for each bin by fitting a Gaussian distribution to the core of the observed pull distribution. Each bin contains 5 % of the tracks in the studied data set. Errors on σ_{fit} are omitted as they are below 1 % and thus not visible on this scale.

an exact momentum information is available, one expects the observed standard deviation of the pull distribution to be flat across the momentum range.

It is shown in Fig. 6.9 that the new method (red curve) does not yield a flat distribution of σ_{fit}. However, it is clearly visible that the observed values of the standard deviation are closer to one in the lower momentum range and no longer follow the same linear $1/p_T$ dependence as the residuals in Fig. 6.8. In the high momentum range for $p_T > 0.5\,\text{GeV}$ the behavior is identical to the default method. This is understandable, as these high momentum tracks are unlikely to experience any significant scattering and thus don't provide any input for the Neural Network to extract momentum information from. Overall the observed standard deviations of the LSTM method (red curve) shows that the presented method is able to extract valuable momentum information in the lower momentum range and adjust its uncertainty prediction accordingly.

Further, it is determined if the improved uncertainty prediction is sufficient to provide a better IP cut selection. This is best done by studying the efficiency of separating secondary tracks from prompt ones. For this study, secondary tracks are defined as tracks from a B meson that exhibits a minimum transverse momentum of 2 GeV and a lifetime above 0.2 ps. These requirements are common requirements in analyses of B meson decays and ensure a reasonable sample of secondary tracks which are commonly selected with IP cuts. Figure 6.10 shows the determined receiver operating characteristic (ROC) curve for four different methods. It shows the signal efficiency on the y-axis as function of the fraction of falsely as signal identified prompt tracks on the x-axis. In red and blue are the achieved performances for using a simple cut on the calculated impact parameter, for the default and new method respectively. As shown, both methods yield identical efficiencies within the shown precision. Given the observed similar resolution of the CTPV position for both methods, this is to be expected.

When taking into account each method's predicted uncertainties, the selection efficiency improves in both cases. Here, the selection cut is performed on a weighted IP defined as,

$$\text{IP}/\sigma = \sqrt{\frac{(x_{\text{PV}} - x_{\text{CTPV}})^2}{\sigma_x^2} + \frac{(y_{\text{PV}} - y_{\text{CTPV}})^2}{\sigma_y^2} + (z_{\text{PV}} - z_{\text{CTPV}})^2}. \tag{6.21}$$

The default procedure is able to reject more prompt tracks in the lower signal efficiency region. While the Kalman Filter does not incorporate any momentum information, it does take into account the slope in x- and y-direction of the trajectory. Thus, the slight increase in efficiency is likely do to the higher errors predicted for tracks which exhibit high values of η. The best observed ROC curve is produced

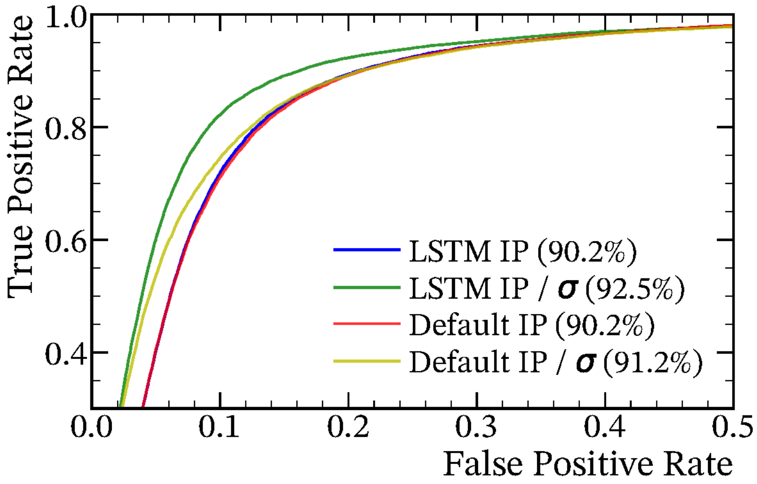


Figure 6.10: Receiver operating characteristic (ROC) curve for selecting secondary tracks with an IP cut based method. The background consists of random prompt tracks. Blue (new method) and red (default) curves represent the ROC curve for a simple cut on the calculated IP based on the predicted CTPV position. Using the quantity of IP/σ, defined in Eq. (6.21), as selection criterion yields the green (new method) and yellow (default) curve.

when using the predicted IP weighted by its uncertainty from the LSTM based method. Depending on the exact cut that is chosen, the background rate is up to 30 % lower at equal signal efficiency compared to the next best solution. In conclusion, the proposed neural network is able to predict an uncertainty which is able to improve the efficiency and purity of an impact parameter based selection.

All of the above presented comparisons were performed using a data sample of simulated $B^0 \to K\pi$ events (Appendix A.1, Sample 2), instead of the data used during the training and validation. No differences, modulo statistical uncertainties, were observed between the presented results and the observed results on the validation data set used during training. This suggests that the observed performances do not significantly depend on the specific topology of the signal decay.

6.3 Conclusion and Outlook

A novel machine learning based method has been introduced as an alternative to the simplified Kalman Filter in the VELO track reconstruction. It is able to provide an estimate of a track's closest to beam state including a momentum dependent uncertainty. It is shown, that the method is able to successfully predict the CTB position to an accuracy at which it is competitive with the currently employed Kalman Filter based method.

Furthermore, the neural network is shown to be able to extract sufficient momentum information from the hit residuals to adjust its uncertainty estimate for tracks in the medium to low momentum range. A study of the signal efficiency and purity of a selection aiming to select displaced trajectories shows that the improved uncertainty estimates enable a reduction of up to 30 % of the rate of incorrectly selected prompt tracks while providing equal signal efficiency.

A detailed comparison will have to be performed to validate that the processing times of both methods are similar. Yet, as this requires a C++ implementation of the presented method, it is left for future work. It is expected that the presented method introduces no significant timing overhead due to the similarity in the amount of necessary computations and the fact that the Kalman Filter currently requires less than one percent of the overall processing time. Based on this assumption and given that the achieved overall throughput of the reconstruction sequence depends on the amount of tracks which pass the IP cut, the above mentioned background reduction will enable a looser IP cut at identical overall throughput.

An additional concern which requires further investigation is the similarity of scattering and detector resolution effects in simulation and data. As the presented method so far solely relies on simulation for its training, it is important that the simulated data matches what will later be observed in real data. One strategy could be to first fall back onto using a simple IP cut selection and using early data to retrain the described model. Given the small size of the network, training times are not very high, thus retraining the network on real data is not a costly process. This retraining process could be repeated in certain intervals as the hit resolution of the detector is expected to slowly change due to radiation damage. Furthermore, before an adoption of this method it is recommended to validate the shown results on additional decay signals. The training and validation was performed on a large sample including many decay topologies and the results match those obtained on the $B^0 \rightarrow K\pi$ sample. Thus, no differences for specific decay topologies are expected. A verification of this statement, for example by studying three-body, four-body, and charm decays, is left for future work.

In summary, the presented LSTM based procedure is producing promising results which indicate that a method of this kind can be used to improve the quality of an IP based selection of VELO tracks. Its final application will however require additional studies.

7 Fast Forward Tracking for the LHCb Upgrade

The reconstruction of trajectories from particles that traversed the entire detector — long tracks, see Section 4.2 — is one of the most important steps in LHCb's reconstruction as they are the basis of HLT1 selections. However, the algorithm responsible for their reconstruction requires almost half the overall processing time of the reconstruction sequence run in HLT1. Because of this, as discussed in Chapter 5, an impact parameter cut on VELO tracks is employed to achieve a higher HLT1 throughput in the current best scenario detailed in [60]. A possible removal of this cut would significantly improve the physics efficiency of the trigger, especially for prompt signals. This removal would however require a major reduction of the processing time required to reconstruct long tracks.

As previously discussed in Chapter 5, the reconstruction of long tracks in HLT1 is focused on finding tracks with a transverse momentum above a certain threshold. In the original design of the HLT [53], this value was chosen to be 500 MeV. This significantly reduces the amount of tracks which will be reconstructed compared to the HLT2 scenario in which the goal is to reconstruct as many long tracks as possible. Additionally, it focuses the search on tracks with higher momentum, as shown in Fig. 7.1. These tracks are easier to reconstruct as their trajectories more closely resemble that of a straight line. This is because higher momentum tracks experience less bending in the magnetic field and have a smaller likelihood to undergo significant multiple scattering. Subsequently, the reconstruction algorithms are able to focus their search on a smaller subset of possible hits which greatly reduces the combinatorics, as will be explained in more detail later.

During Run 2 the reconstruction of long tracks in HLT1 and HLT2 was performed by an algorithm which is based upon a Hough clustering approach [48]. Thanks to its various options, this algorithm was flexible enough to perform well in both trigger stages. The currently best algorithm for the long track reconstruction following LHCb's upgrade, the *PrForwardTracking* is heavily based on this algorithm. Nevertheless, even the fastest configurations are found to be too slow to enable a future HLT1 to process events at rate of 30 MHz while also maintaining high reconstruction efficiencies.

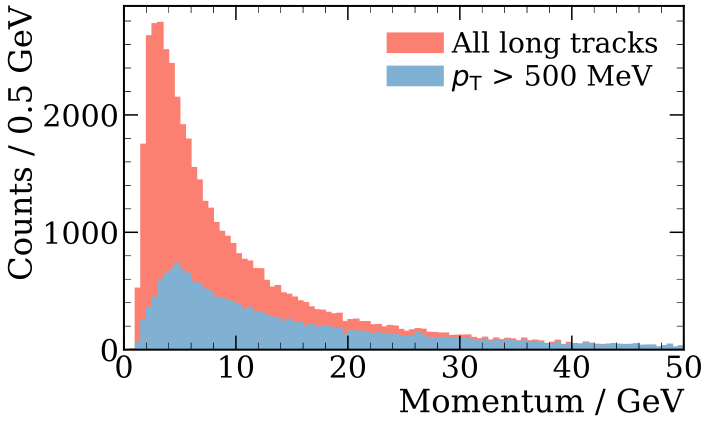


Figure 7.1: Momentum Spectrum of all long tracks (red) compared to the same sample after requiring a minimum transverse momentum of 500 MeV.

The following work presents a new algorithmic approach to reconstruct long tracks from upstream tracks, further referred to as the *TrackForwarding* algorithm. It is designed to focus on the reconstruction of tracks with a higher transverse momentum than 500 MeV, and thus replace the current solution in the HLT1 scenario. At first the algorithm's requirements specification is outlined before the following subsections detail the design and implementation. At last, a thorough CPU and physics performance comparison of the current and proposed solution is presented.

7.1 Algorithm Requirements Specification

Qualitatively, the long track reconstruction is meant to extend upstream tracks into the SciFi and find corresponding hits. The correct set of hits together with the upstream track will form a long track and enable a precise momentum estimate of the particle.

The available information for this procedure is a single trajectory state as defined inside the UT reconstruction,

$$\left(x, y, t_x, t_y, \frac{q}{p}\right), \qquad \text{at } z = 770\,\text{mm}. \tag{7.1}$$

With x, y, z defining the state's position and q, p, t_x, t_y describing the particle's charge, momentum, and slope in x- and y-direction. In theory, a precise estimation of the particle's trajectory through the magnet and into the SciFi is computable given the charge, momentum, and the equations of motion in Eqs. (4.6) and (4.7).

However, as shown in [26], an upstream track's momentum resolution is of the order of 15 %. This has implications on the possible usage of the momentum estimate as will be outlined in Section 7.2.2.

To specify the requirements, it is helpful to differentiate between two different kinds of upstream tracks. First, an upstream track which is created by a particle that is reconstructable in the SciFi detector and has a momentum greater than 500 MeV, where reconstructable in the SciFi detector implies the particle produces hits in at least one x- and one stereo layer per station. Second, an upstream track created by a particle which doesn't satisfy the above requirement.

An algorithm should find the correct set of hits for as many tracks of the former category as possible. The fraction of tracks out of that category for which it is able to do so defines the algorithm's reconstruction efficiency (see Section 4.1.3). The efficiency achieved by a new solution should be competitive to that of the *PrForwardTracking* which achieves a reconstruction efficiency of about 86 % for long tracks which exhibit a momentum larger than 3 GeV and a transverse momentum larger than 500 MeV.

For tracks belonging to the latter category, the algorithm should reject the input without creation of a long track. In reality, it is possible that an upstream track from the second category can be paired with a random set of hits and resemble a correctly reconstructed long track. This scenario leads to the creation of ghost tracks and is usually not completely avoidable. Yet the amount of ghost tracks should be minimized as much as possible, as they can cause the later HLT1 selection to incorrectly accept an event that does not contain any interesting physics. The *PrForwardTracking* exhibits a ghost rate of about 3 %.

At last, the algorithm should be able to provide a precise momentum estimate for a reconstructed long track. The current algorithm is able to achieve a resolution at the order of 1 %, which is sufficiently precise for HLT1 selections to be implemented based upon these estimates.

7.2 Algorithm Design

The new algorithm design investigates the usability of a track forwarding approach, which is a method that finds a track's hits by stepwise extrapolation of the track from one detector plane to the next to one-by-one find the remaining hits. Similar solutions are implemented inside of the VELO reconstruction or an early predecessor of the *PrForwardTracking* [47].

The design choice is motivated by the foreseen advantages of this method which are a reduction of combinatorics and an early possibility to reject an input track. These should in turn lead to a reduction in processing time compared to the current solution. As mentioned in Chapter 4, the *PrForwardTracking* directly processes x-hits inside a certain search window of all x-layers. In comparison, the *TrackForwarding* only searches for hits inside a comparably large search window in the third station where it can potentially already stop and reject an input track if no compatible hits are found. If hits are found, they enable the calculation of an improved momentum estimate thus enabling precise extrapolations into other stations. The later search in station one and two will then simplify to a selection of a single hit which is closest to the given extrapolation. Starting the search in the last station is motivated by the fact that it is the only station which is positioned almost outside of LHCb's magnetic field as shown in Fig. 7.2. Subsequently, higher momentum tracks will not exhibit a significant curvature inside the last tracking station. This lack of curvature should allow for a focus on finding combinations of hits which form a straight line inside the last station, making the search simpler than one which has to account for curved trajectories.

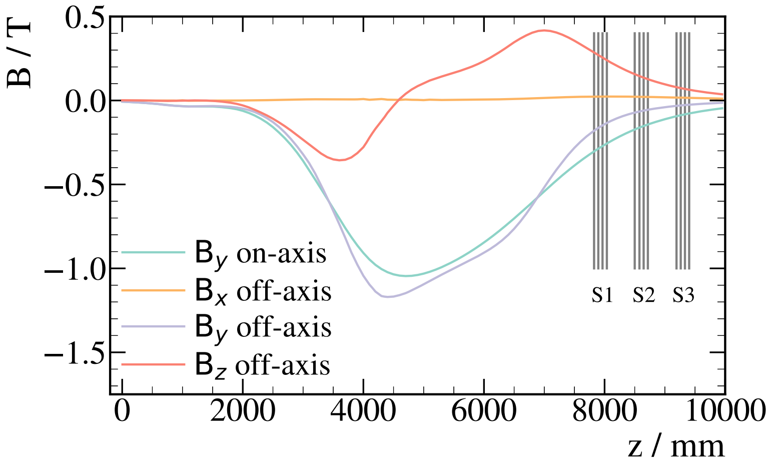


Figure 7.2: Visualization of the magnetic field component B_y along the x-axis as well as the encountered magnetic field components further away from the x-axis. The off-axis components B_x, B_y, and B_z are determined along the trajectory of an exemplary particle with positive slopes in x and y and a momentum of 10 GeV. The vertical gray lines indicate the position of the four detection layers per SciFi Station (S1, S2, and S3).

7.2.1 Algorithm Overview

A brief overview of the algorithm's individual steps is given below, while a detailed description of each step will be provided in the following subsections.

Figure 7.3 shows a visualization of a fully completed reconstruction procedure of the proposed algorithm.

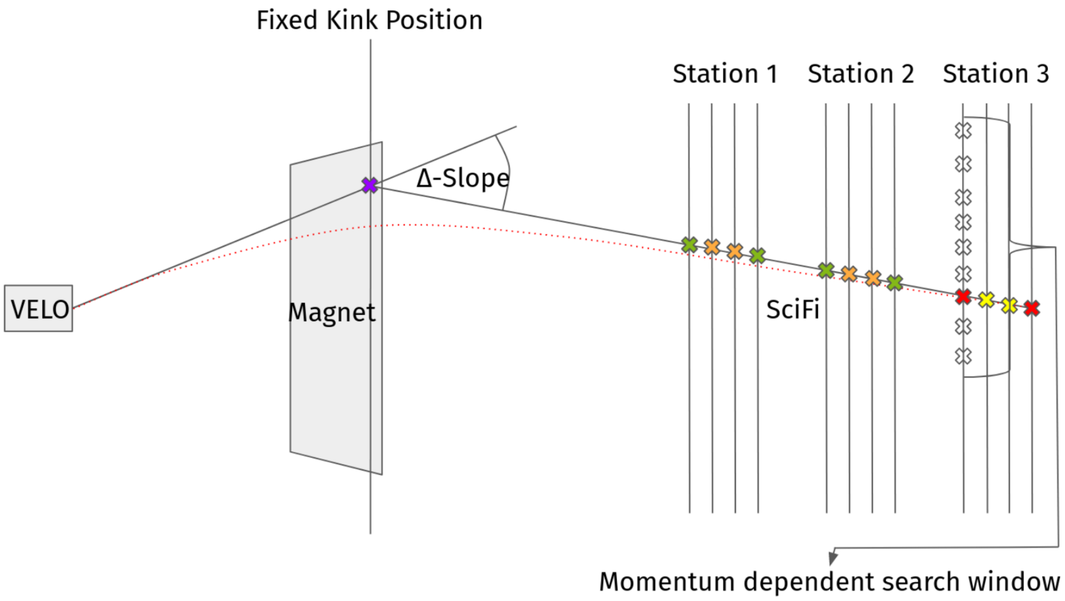


Figure 7.3: Visualization of the internal pattern recognition procedure in the xz-plane. For an explanation of the shown content see the text section.

Hit doublet search in last station At first, the position and the momentum estimate of the upstream track are used to open a search window and select hits which potentially belong to the trajectory (white crosses). Next, the upstream track is extrapolated along a straight line to determine its x-position at a fixed z-position inside the magnet (purple cross). The selected hits are iteratively processed by first defining a straight line between the hit and the previously defined x-position (purple cross). Given the correct hit is picked from the search window (red cross, first layer) it is possible to find the corresponding hit in the last layer of station 3 (red cross) by extrapolating along the defined straight line. This step is explained in detail in Section 7.2.2.

Hit doublet confirmation Upon successfully finding a hit doublet, hits in the stereo layers (yellow crosses) are added to form a full station 3 candidate. The details of this subroutine are discussed in Section 7.2.3.

Search in Station one and two Based on the found hits inside the last station a precise extrapolation into the second and first station is possible. Section 7.2.4 will outline the details of said extrapolation as well as the search for the remaining hits (green and orange crosses).

Parameter Estimation Once all hits belonging to a trajectory are collected, the last subroutine performs a track fit and estimates the track's momentum. Last, various characteristics of the final track candidate are used to estimate a likelihood of it being a ghost track. This likelihood is used to pick the best candidate, if multiple candidates are created or to reject those tracks which exhibit a high likelihood of being a ghost track. Details of the last subroutine are presented in Section 7.2.5.

7.2.2 Finding Compatible Hit Doublets inside Last Station

The first step of the new algorithm is the search for two matching x-hits, a hit doublet, in the x-layers of the last tracking station. In the beginning, hits of interest in the first layer of the third station are selected by defining a search window and collecting all hits within. A large search window and thus a selection of multiple hits is necessary as a precise extrapolation of the upstream track into the given layer is not possible. This is due to the relatively large resolution of the upstream track's momentum estimate. Depending on the upstream track's momentum, resulting search window sizes can range from 20 cm up to 1.2 m, as is detailed below.

The center of the search window is calculated by extrapolating the input track through the magnet and up to the mentioned layer. Given the expected error of the extrapolation, the only requirements for the extrapolation procedure are to yield an unbiased estimate of the expected position with a precision such that it does not significantly increase upon the expected error due to the poor momentum resolution. This leads to the decision to use a simple and fast to evaluate formula over more complicated procedures such as the Runge-Kutta method [75]. The Runge-Kutta method is a commonly used iterative procedure to approximate a solution to ordinary differential equations like the equations of motion inside a magnetic field. At LHCb, this method is one of the default methods to extrapolate trajectories with high precision, yet their iterative nature tend to require more computing resources to arrive at a solution compared to a simple polynomial. The determined formula for this extrapolation,

$$\begin{aligned} x_{ext} =& x + t_x(z_{layer} - 770) + x_{curve} \\ x_{curve} =& \frac{q}{p}\Bigg[c_0 + t_x(c_1 s + c_2 t_x) + t_y^2(c_3 + t_x(c_4 s + c_5 t_x)) \\ &+ \frac{q}{p}\left(c_6 t_x + c_7 \frac{q}{p}\right)\Bigg], \end{aligned} \tag{7.2}$$

is based on linear term and a polynomial curvature correction. Here, charge and momentum of the particle are q and p, while the slopes of the input track are t_x and t_y. The starting value x is given by the input track as defined in Eq. (7.1), while z_{layer} is the z-position of the first layer of the last SciFi station. The polynomial's coefficients c_i are listed in Table 7.1. The set of used monomials for the curvature term is determined on simulated data. First a general polynomial is fitted and then monomials which do not contribute significantly to the final result are removed. The residual between the result of the extrapolation and the true position of the hit is shown in Fig. 7.4, for the Runge-Kutta method and the above formula. As shown, the proposed formula and the Runge-Kutta method exhibit equally large residuals, ensuring that the proposed extrapolation formula's precision is sufficient given the upstream track's momentum resolution. No significant bias is observed thus leading to the validation of the proposed formula as a sufficient solution which satisfies the stated requirements.

Table 7.1: Coefficients tuned on simulated data by performing a regression using the polynomial defined in Eq. (7.2).

c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7
4824.3	−426.3	7071.1	12080.4	14077.8	13909.3	9315.3	3209.5

The shown residuals of the polynomial extrapolation in Fig. 7.4 are used to determine the size of the search window. While the search window should be as small as possible to not unnecessarily include random hits, it should be big enough to contain the correct hit in as many cases as possible. Figure 7.5 shows a two-dimensional histogram of the extrapolation residuals as a function of particle momentum multiplied by its charge. It is visible that the observed residuals significantly reduce with higher momentum. Additionally, a strong asymmetry of the residuals is observed which differs in direction between the two possible particle charges. This is an artifact of the fact that the momentum is estimated based on the inverse of a track's

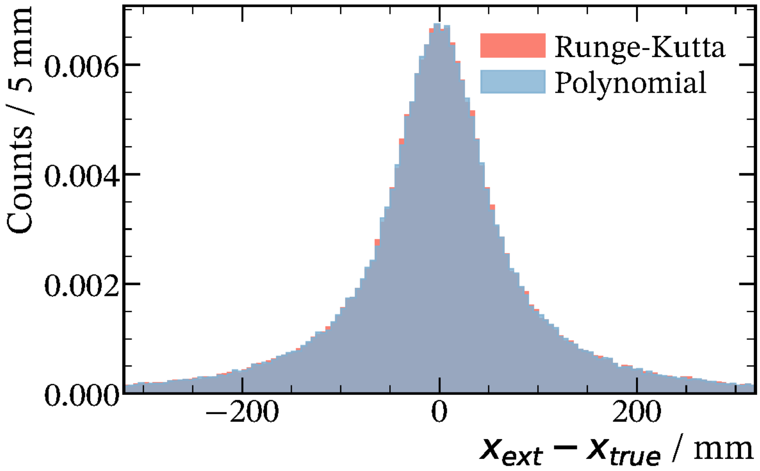


Figure 7.4: Comparison of the extrapolation residuals of the Runge-Kutta method (red distribution) and a polynomial (blue distribution) when using momentum estimate provided by the upstream track. Residuals are calculated as difference between the true x-position of the simulated hit in the first layer of the last SciFi station and the extrapolation of the upstream track's x-coordinate to the target hit's z-coordinate.

observed curvature. As the observed curvature exhibits a mostly Gaussian error caused by detector resolution and multiple scattering, the error of its inverse, of the momentum, no longer follows a Gaussian distribution.

To minimize the necessary size of the search window, this correlation is taken into account in the formulation of the search window limits. Equation (7.3) defines an asymmetric bound such that the correct hit will lie within the range $[x_{\text{min}}, x_{\text{max}}]$ in about 99 % of the cases.

$$\begin{aligned}
\Delta_{\text{min}} &= \begin{cases} 50\,\text{mm} + \frac{1\,400\,\text{mm GeV}}{p} & \text{if } p < 2.54\,\text{GeV} \\ 600\,\text{mm} & \text{otherwise} \end{cases} \\
\Delta_{\text{max}} &= \begin{cases} 150\,\text{mm} & \text{if } p > 56\,\text{GeV} \\ 100\,\text{mm} + \frac{2\,800\,\text{mm GeV}}{p} & \text{otherwise} \\ 600\,\text{mm} & \text{if } p < 5.6\,\text{GeV} \end{cases} \\
x_{\text{min}} &= \begin{cases} x_{ext} - \Delta_{\text{min}} & \text{if } q_{\text{mag}} > 0 \\ x_{ext} - \Delta_{\text{max}} & \text{if } q_{\text{mag}} < 0 \end{cases} \\
x_{\text{max}} &= \begin{cases} x_{ext} + \Delta_{\text{max}} & \text{if } q_{\text{mag}} > 0 \\ x_{ext} + \Delta_{\text{min}} & \text{if } q_{\text{mag}} < 0 \end{cases}
\end{aligned} \tag{7.3}$$

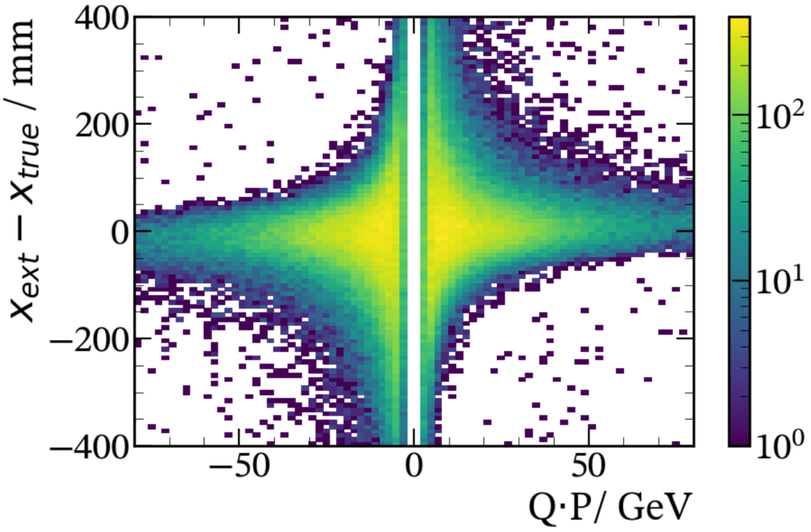


Figure 7.5: Two-dimensional histogram of the extrapolation residuals in x, as function of the product of the particle's charge and momentum.

The value of q_{mag} represents the charge multiplied by the magnet polarity and determines the direction in which the magnet bends the track. The hits excluded by this range are the outliers visible in Fig. 7.5. They are most commonly produced by tracks that have undergone significant scattering and are as a result almost impossible to reconstruct. A large enough search window to collect these hits would only result in the unnecessary selection of more random hits increasing combinatorics and ghost rate. Overall, defining the above search window limits by including 99 % of the hits represents a relatively loose cut. As a result almost all hits from tracks that have not scattered to a degree that leaves them impossible to reconstruct are included.

Following the selection of a set of hits using the above search window, the algorithm continues to iteratively process them one by one. For each hit it is possible to search for a matching hit in the last station by making use of the magnetic lens approximation, similar to the one used in the *PrForwardTracking* (see Section 4.2). This procedure is visualized in Fig. 7.3. It is shown that the particle trajectory before the magnet is approximated as a straight line which eventually exhibits a kink in the x-plane at a fixed z-position inside the magnet, called kink position. After the change of direction, the trajectory is again modeled by a straight line.

The straight line before the magnet is defined by the input track. The x-coordinate of the kink position is determined by extrapolating the input track to the respective

z-position along a straight line

$$x_{\text{kink}} = x + (z_{\text{kink}} - 770) \cdot t_x, \tag{7.4}$$

with z_{kink} being the z-coordinate of the kink position. The value of z_{kink} is determined on simulated data to be 5 282.5 mm.

The above position combined with the x-coordinate of a hit in the last station enables an estimation of the x-slope for the straight line after the magnet,

$$t_x^{\text{SciFi}} = \frac{x_{\text{hit}} - x_{\text{kink}}}{z_{\text{hit}} - z_{\text{kink}}}. \tag{7.5}$$

This slope enables a straight line extrapolation into the last station which is used to predict a x-position to search for a matching hit. The closest hit to said position is picked as a potential match, which corresponds to the second red cross in Fig. 7.3. At this point, the x-slope in the last station is recalculated based on the two selected hits. The two hits form a valid hit doublet if the residual of the second hit and the x-prediction r_{hit} satisfies

$$r_{\text{hit}} < \begin{cases} 0.48\,\text{mm} + 4.25\,\text{mm} \cdot \varDelta_{\text{slope}} & \text{if } \varDelta_{\text{slope}} < 0.59 \\ 3.0\,\text{mm} & \text{otherwise} \end{cases}. \tag{7.6}$$

This threshold is defined such that a correctly matched hit will be accepted in about 99 % of the cases, while aiming to reject unphysical combinations. The value of $\varDelta_{\text{slope}}$,

$$\varDelta_{\text{slope}} = t_x^{\text{SciFi}} - t_x, \tag{7.7}$$

is the change in x-slope at the kink position. It represents a measure of a particle trajectory's curvature inside the magnetic field which is inversely proportional to the particle's momentum. Thus, $\varDelta_{slope}$ is used to define a momentum dependent upper bound for r_{hit} to account for larger residuals due to scattering effects which lower momentum tracks are more likely to experience.

At this point, using a fitted polynomial, it is possible to calculate a more precise momentum estimate based on $\varDelta_{slope}$. The details of this method will be further outlined in Section 7.2.5 as the algorithm does not calculate a momentum estimate at this point. Any momentum dependent effects are rather formulated as function of the directly measured quantity of $\varDelta_{slope}$. This has the benefit of avoiding the costly transformation to a momentum value until the end of the algorithm. However, Figure 7.6 shows that the momentum resolution which is achievable at this point is within about 2 %. This fact validates the stated assumption in the motivation of the algorithm that a hit doublet would allow for a relatively precise momentum

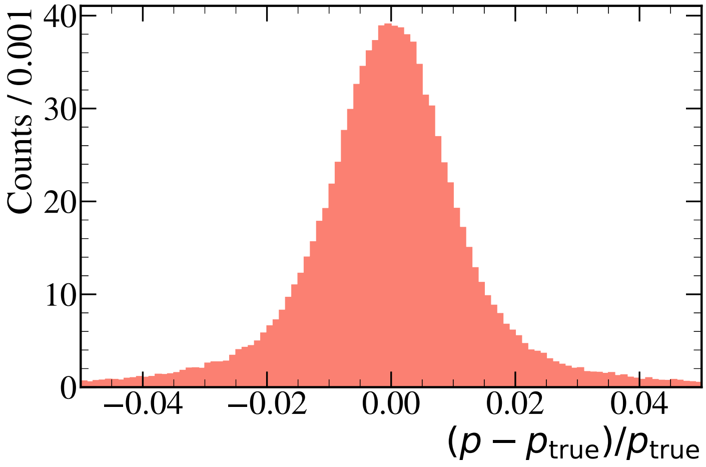


Figure 7.6: Achievable momentum resolution when combining an upstream track with a correct hit doublet from the last SciFi station. The momentum calculation is based on a polynomial similar to the method in Section 7.2.5.

estimate and thus enables precise extrapolations into the other stations. These extrapolations and their precision are later shown to again validate this assumption in Section 7.2.4.

Overall, only about one in ten long track candidates, an upstream track matched to a hit doublet, will result in a full trajectory at the end of the algorithm. This is mainly due to the high probability of an upstream track to be matched to more than the correct hit doublet, resulting in multiple wrong matches. Consequently, most of the candidates at this point are ghost tracks. The following two subroutines are focused on searching for additional hits and will impose hit requirements. That is, only those candidates which are able to be matched to a minimum number of hits per station are processed further. These requirements will lead to the rejection of a large fraction of the above mentioned ghost tracks.

7.2.3 Doublet Confirmation in Last Station

Given the two x-hits inside the first and last layer, the trajectory's x-prediction inside the second and third layer is calculated by a straight line interpolation. However, searching for the two remaining hits requires additional information next to the given x-predication as the hits are inside stereo layers. During the previous steps, the algorithm only considers the trajectory in a xz-plane as sketched in Fig. 7.3. To use the measurements in the stereo layers it is necessary to reconstruct the trajectory's y-coordinate at the given layer. This is further illustrated in Fig. 7.7, which shows an example trajectory crossing the four layers of a single station. As depicted, the x-coordinate of a hit in a vertical module (red line) is independent of the track's y-position. In contrast, due to the 5° rotation of stereo layers (yellow line), hits in these layers require the y-coordinate at which the track traversed the module to reconstruct the correct x-coordinate or vice versa.

The prediction of an upstream track's y-coordinate can be approximated by a simple straight line extrapolation into the SciFi. Taking Eq. (4.7) into account as well as the magnitude of the fringe fields shown in Fig. 7.2, it is clear that this approximation does not properly describe the trajectory. The previous algorithms in [47, 49] first employ a simple straight line extrapolation to find a set of hits and during later processing introduce a correction on y. This correction is meant to account for the non-negligible impact of the fringe fields on the trajectory's path in the yz-plane. In the mentioned algorithms, these corrections only depend on the momentum of the track.

However, this is shown to be insufficient by studying simulated data and visualizing the error on the y-prediction of an upstream track. For this purpose an upstream track is extrapolated along a straight line to calculate a y-prediction in the second layer of the last station. Given that the detector hits in the SciFi only provide 2D measurements, the true 3D position of the corresponding hit is determined from the additional information available in simulated data. The residual

$$r_y = y_{\text{true}} - \left[y + t_y \left(z_{\text{layer}} - 770\right)\right], \tag{7.8}$$

is defined as the difference between the simulated hit position and the straight line extrapolation. All upstream tracks in the data set are divided into bins depending on their slopes (t_x, t_y). Each bin's value is defined as the mean of the observed residuals,

$$\bar{r}_y = \frac{1}{n-1} \sum_{i=0}^{n} r_{y,i}. \tag{7.9}$$

The result is the two-dimensional histogram shown in Fig. 7.8. Only tracks produced by particles with a negative charge were used as the observed effect is opposite in direction for positively charged particles and thus is only visible in an average when considering a single charge. If a y-prediction method correctly accounts for the magnetic field, the observed residuals should primarily be caused by multiple scattering and resolution effects. As both of these effects lead to Gaussian deviations, the resulting mean of the observed residuals should be zero. However, the observed biases of the mean reach values of up to 50 mm in y, which translates into up to 4.4 mm shifts of the resulting x-coordinate. This validates the need for a correction on the straight line extrapolation.

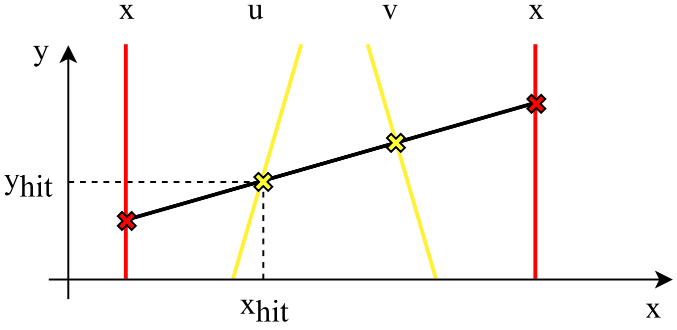


Figure 7.7: A xy-view of the last SciFi station of the same event which is shown in a xz-view in Fig. 7.3. Shown are the track as a black line crossing the four SciFi detector layers ($x - u - v - x$) leaving hits (crosses). The rotation of the stereo layers is exaggerated for illustration purposes.

Yet, the observed correlation between the values of $\bar{r}_y$ and a track's slopes clearly show that a correction on a straight line prediction can not sufficiently account for these effects as long as it is only based on momentum information. The observed shape of the mean residuals suggests that a possible solution could be to define a corrective term based on a track's momentum and slopes. Thus, defining the y-prediction of an input track at a given layer as

$$y_{\text{layer}} = y + t_y \cdot (z_{\text{layer}} - 770) + \Delta_y(t_x, t_y, \Delta_{\text{slope}}), \tag{7.10}$$

where the corrective term is determined by a function Δ_y. Its functional form is yet to be defined. This equation is similar to the approach used in Eq. (7.3), where the x-trajectory through the magnetic field is modeled by a combination of a linear term and a curvature term. Here, in contrast to the x-extrapolation, the momentum is known with a significantly better resolution, consequently the extrapolation should be as precise as possible.

Given these requirements, it is decided to determine the ideal functional form of Δ_y on a simple data set, called toy data set, instead of one from a full simulation. The full simulation aims to generate events which are as close as possible to the events which will later be observed in the detector. While this is of utmost importance for many studies, the task at hand is better served by a much simpler data set, as is explained below. A toy datum is generated by creating a track state,

$$(x, y, z, t_x, t_y, q/p) = (0, 0, 0, t_x, t_y, q/p), \tag{7.11}$$

which is then extrapolated through LHCb's magnetic field up to the z-coordinate corresponding to the second layer of the last SciFi station. Additionally, an identical

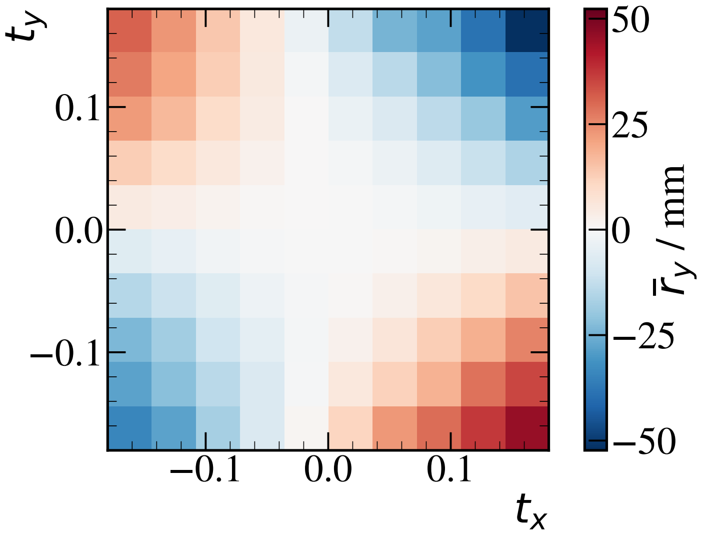


Figure 7.8: 2D Histogram of the mean residual, $\bar{r}_y$, between a straight line extrapolation of an upstream track into the second layer of the last SciFi station and the y-coordinate of the corresponding simulated hit. Tracks are binned depending on their slopes t_x, t_y while the color indicates the observed value of $\bar{r}_y$ in a given bin. Only tracks produced by particles with a momentum below 10 GeV and negative charge are used.

state is extrapolated to the same final position but along a simple straight line. The difference between the two state's y-coordinate yields the simulated residual r_y for the given track state. This procedure is repeated for various values of t_x, t_y, and q/p to sufficiently cover the entire phase space. In conclusion, the final data set is a list of mappings from (t_x, t_y, qp)-triplets to r_y residuals.

The main advantage of the toy data set is that the observed residual is only caused by the magnetic field. There are no detector resolution, material, or multiple scattering effects involved. This is a benefit, as common regression methods (e.g. least-squares minimization) only perform well as long as the observable's error follows a Gaussian distribution. Especially the long non-Gaussian tails of the multiple scattering or hadronic scattering angle's distribution would lead to non-Gaussian errors on r_y. Subsequently, a fit based on these residuals will potentially converge slower and possibly lead to biased solutions. Additionally, the toy data provides a method to verify that the systematic differences observed in Fig. 7.8 are solely caused by the magnetic field.

An exemplary toy data set, which was generated with a value of $qp = -5$ GeV, is visualized in Fig. 7.9. Each dot represents a single datum that maps a (t_x, t_y) value pair to a value of r_y. A comparison of this visualization to the one presented in Fig. 7.8 shows an almost identical structure suggesting that the observed effects are indeed only caused by the magnetic field. Therefore, it is plausible that a correction based on the generated toy data will be sufficient to correct for the observed effects.

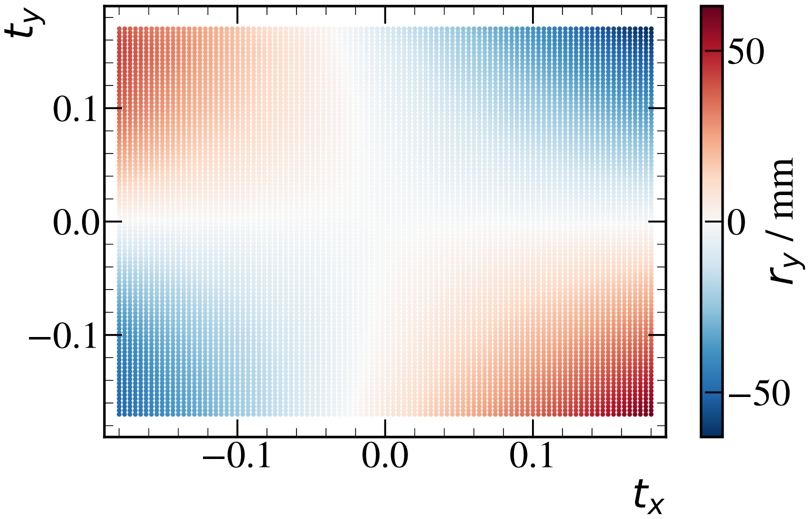


Figure 7.9: Visualization of a toy data set for a fixed value of $q \cdot p = -5\,\mathrm{GeV}$. Each dot represents a single datum in the set, with its color indicating the value of r_y.

At this point, the functional form of the curvature term Δ_y needs to be defined. It is decided to again opt for the use of a polynomial due to their fast evaluation times compared to iterative procedures. Given that the polynomial solution provides sufficient precision, as will be shown below, more complicated methods are not required. The polynomial is determined following the same iterative procedure described in Section 7.2.2, which yields

$$\begin{aligned}\Delta_y = |\Delta_{\text{slope}}| \Big[& c_0 q + c_1 t_y + c_2 q t_x t_y + c_3 t_x^2 t_y + c_4 q t_x^3 t_y \\ & + c_5 t_y^3 + c_6 q t_x t_y^3 + c_7 t_x^2 t_y^3 \Big] \end{aligned} . \tag{7.12}$$

The coefficients c_i are listed in Table 7.2. The introduced polynomial in Eq. (7.12) and Eq. (7.10) are used to determine an improved y-prediction to study the observed mean residuals. The mean residual between the improved prediction and the true y-coordinate of the simulated hit is visualized in Fig. 7.10, using the same underlying data as was used in Fig. 7.8. While some small biases remain visible, especially in the higher t_x region, their magnitude is mostly below 2 mm. At this magnitude the remaining biases are not expected to have a significant impact on the pattern

Table 7.2: Polynomial coefficients of Δ_y, defined in Eq. (7.12).

c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7
3.7884	73.164	7353.9	−6347.7	20270	3721.0	−46038	230943

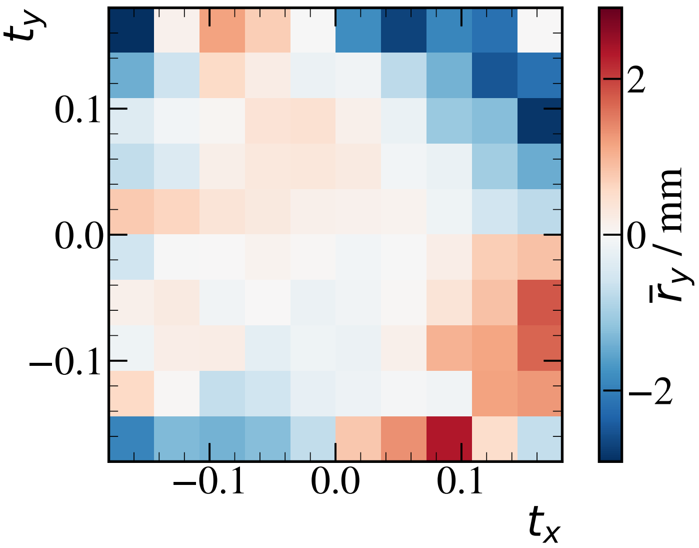


Figure 7.10: 2D Histogram of the mean residual, $\bar{r}_y$, between a prediction based on Eq. (7.12) and the true y-coordinate of the simulated hit. See Fig. 7.8 for further details.

recognition. Subsequently, this significant reduction of the biases, verifies that the proposed polynomial is able to sufficiently correct for the effects observed in Fig. 7.8.

It should be noted that the defined polynomial is tuned such that the predicted corrective term is only correct for the specific z-location of the second layer in the last station. Repeating the above procedure for stereo layers in station one and two, it is found that the resulting correction can alternatively be obtained by a constant scaling of the already defined formula for Δ_y, where the constant's value depends on the position of the given layer. This can be explained by the fact that the necessary correction for a given track solely depends on the integrated magnetic field along its path. Thus, the y-correction for a track in an earlier layer should approximately be reduced by the fraction of the total integrated magnetic field the track has encountered.

As shown, using Eqs. (7.10) and (7.12), it is possible to obtain a good prediction of the y-coordinate of a track in each stereo layer. The x-predictions in the second and third layer are provided by a straight line interpolation between the two selected hits. A simple search procedure would combine each hit in the stereo layer with the predicted y-coordinate to obtain a x-position which one can compare to the x-prediction to select the best hit. But, this procedure involves a significant amount of computations which can be avoided by the following strategy. During the readout of the detector, the x position at $y = 0$ is calculated for all stereo hits, and they are sorted accordingly. The predicted x- and y-position of the trajectory are used to determine a predicted x-coordinate at $y = 0$,

$$x_{y=0} = x_{\mathrm{pred}} - \tan(\pm 5^\circ) \cdot y_{\mathrm{pred}}. \tag{7.13}$$

Here $\pm 5^\circ$ represent the rotation of the first and second stereo layers.

Next, the hit in each layer which is closest to the above prediction is selected if its x-residual, r_{hit}, satisfies,

$$r_{\mathrm{hit}} < 0.5\,\mathrm{mm} + 10\,\mathrm{mm} \cdot \mathit{\Delta}_{\mathrm{slope}}. \tag{7.14}$$

Similar to the previous thresholds in Section 7.2.2, this threshold aims to include the correct hit in about 99 % of all cases. Given that one of the primary goals of searching for additional hits is the ghost rate reduction, it is necessary to introduce these thresholds to reject unphysical hit combinations. The lower the possible value of the threshold can be, the more random combinations can be eliminated. Ignoring the momentum dependence, Fig. 7.11 visualizes the observed x-residuals of correctly matched hits, between using a straight line to predict the y-coordinate (red) and using the polynomial method introduced above (blue). Both distributions are cut such that they include the innermost 99 % of the distribution. This highlights the impact of the improved y-prediction which enables the thresholds to be significantly smaller. Consequently, the algorithm is able to find the correct hits while rejecting more random combinations due to the tighter threshold. If the algorithm is not able to find any hits in the stereo layer which match with a candidate, that candidate is rejected. Candidates with one or two found hits in the uv-layers are considered good candidates. For these the algorithms continues with a search of for matching hits in the remaining stations of the SciFi.

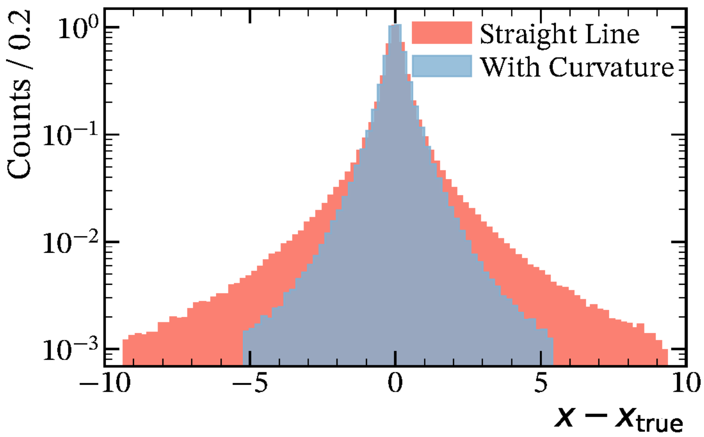


Figure 7.11: Comparison of the distribution of x-residuals in the second layer between two different methods of determining the y-coordinate. In red for the method which predicts y based on a simple straight line and in blue for the method based on Eq. (7.12). Both distributions are cut such that they show the innermost 99 %.

7.2.4 Trajectory Forwarding

The search for additional hits inside the first and second station is conceptually identical and is thus discussed as one in the following chapter. Given a candidate from the previous step, the following procedure extrapolates the trajectory into the remaining layers to search for potentially matching hits. Looking at Fig. 7.2, it is clear that the extrapolation to the other stations will pass through a non-negligible magnetic field.

The algorithm first searches for matching hits inside the x-layer and in a second step uses the found hits to search for matching hits inside the stereo layers. Thus, the first step requires the track candidate's x-prediction in the x-layers of station one and two. These are calculated using the polynomial,

$$\begin{aligned} x_{\mathrm{S,L}} &= x_{3,0} + t_x^{\mathrm{SciFi}}(z_{\mathrm{S,L}} - z_{3,0}) + \Delta_{\mathrm{S,L}} \\ \Delta_{\mathrm{S,L}} &= (\alpha + \gamma t_y^2)\Delta_{\mathrm{slope}}(z_{\mathrm{S,L}} - z_{3,0})^2 + \beta\Delta_{\mathrm{slope}}(z_{\mathrm{S,L}} - z_{3,0})^3. \end{aligned} \tag{7.15}$$

Similar to previously defined polynomials, the trajectory is described by a linear term and a curvature term $\Delta_{\mathrm{S,L}}$ correcting for the effects of the magnetic field. Here, the suffix $\mathrm{S} \in \{1, 2\}$ indicates the station while $\mathrm{L} \in \{0, 3\}$ represents the index of the layer. Note, the numbering follows the, somewhat confusing, convention which is used in LHCb's geometry description. A station's indexing starts at one, while a layer's starts at zero. The polynomial is fitted on simulated data to obtain the

Table 7.3: Coefficients of $\varDelta_{\mathrm{S,L}}$, defined in Eq. (7.15).

α	β	γ
1.470682×10^{-5}	$-3.152273942 \times 10^{-9}$	-0.000335

coefficients listed in Table 7.3. Note that the curvature term $\varDelta_{\mathrm{S,L}}$ is significantly simpler compared to $\varDelta_y$ in Eq. (7.10). This is the result of empirically determining the polynomial in the same way it was done in the previous sections. During this procedure it was found that the additional terms yielded no improvement in precision and were thus removed. This is plausible, given that the extrapolations modeled by Eq. (7.15) are shorter and within an area where the magnetic field is much weaker. Furthermore, a special toy data set was not necessary as the fit of the simple polynomial yielded good enough precision on the data from full simulation.

After calculating the extrapolations into the x-layers of station two, the algorithm selects the closest hit in each layer given the residual is below the calculated threshold for that layer,

$$r_{\mathrm{S,L}} < b_{\mathrm{S,L}} + m_{\mathrm{S,L}} \varDelta_{\mathrm{slope}}. \tag{7.16}$$

The constants to determine the thresholds for the remaining layers are given in Table 7.4. As before, they are chosen such that they include the correct hit at a likelihood of about 99 %.

Table 7.4: The slope and offset used to calculate the hit residual threshold depending on the station and layer (S,L) with Eq. (7.16).

S,L	1, 0	1, (1, 2)	1, 3	2, 0	2, (1, 2)	2, 3
$b_{\mathrm{S,L}}$	2.6	1.0	2.6	2.0	0.5	1.8
$m_{\mathrm{S,L}}$	3.6	7.0	3.5	2.0	10.0	1.0

After the search for x-hits, candidates are rejected if the total amount of added SciFi hits is less than five. This requirement is used to reduce the amount of ghost tracks as real particles will only rarely miss to produce a hit in the detector.

Next, the algorithm searches for matching hits inside the stereo layers if a hit was found in both x-layers. While it would be possible to perform this search in cases where only one hit was found, requiring both hits simplifies the search and is an optimization to keep processing times low. Given the short distances within a station the predictions inside the *uv*-layers are calculated by a straight line interpolation between the two hits of the first and last layer. The threshold for accepting the closest hits is given by Eq. (7.16). If at least one stereo hit is found, the algorithm proceeds by performing the same search for hits in the first station. During this procedure, the algorithm will apply similar requirements on the number of found hits as in station two. At the end of the search a final track candidate will be required to have matched to at least five out of the six possible x-hits and a minimum number of three stereo hits.

Having collected all matching hits for an upstream track, this marks the end of the pattern recognition step. For some input tracks it is possible that multiple candidates reach this step. However, as it does not represent reality to have multiple long tracks formed from a single upstream track this usually implies that at most one of them is a real particle trajectory. The employed procedure to select the best candidate as well as the final estimation of a long track's parameters are detailed in the following subsection.

7.2.5 Parameter Estimation and Selection

The final step of the algorithm is the parameter estimation and selection of candidates. At first, each candidate is fitted to improve the existing estimates of the trajectory's parameters. These are further used to calculate the candidate's momentum estimate. To keep the processing time low, the performed fit is a simple one-dimensional fit of the x-hits. A general track model for these hits is given by Eq. (7.15), assuming $\Delta_{\mathrm{S,L}}$ is zero for the last hit in the last station. The free parameters of the fit are the position of $x_{3,0}$ and the slope t_x^{SciFi} while the curvature term is kept fixed. Thus, the solution is given by solving the least squares minimization problem of a straight line. The required residuals to calculate this solution have already been determined during the pattern recognition step as they correspond to the residuals used to determine if a hit is accepted.

Once the fit is performed, the improved estimate of t_x^{fit} is used to calculate a momentum estimate for the track candidate using the polynomial

$$\begin{aligned} qp =& \frac{C}{t_x^{\mathrm{fit}} - t_x} + c_7 \mathrm{sgn}(\Delta_{\mathrm{slope}}) \\ C =& c_0 + (t_x^{\mathrm{fit}})^2 \left(c_1 + c_2 (t_x^{\mathrm{fit}})^2\right) + c_3 t_x t_x^{\mathrm{fit}} + t_y^2 (c_4 + c_5 t_y^2) + c_6 t_x^2. \end{aligned} \tag{7.17}$$

This is an empirically determined solution inspired by previously existing approaches in [47–49]. The formally correct procedure to determine the charge and momentum is by using Eq. (4.8). However, determining the integral in Eq. (4.8), is only possible by performing a costly numerical integration, which is why the above formula approximates the value of that integral by the polynomial C. The coefficients c_i of Eq. (7.17) are determined on simulated data and are listed in Table 7.5.

Table 7.5: Coefficients for the polynomial defined in Eq. (7.2).

c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7
1239.41	486.057	6.71587	632.728	2358.58	−9256.28	241.460	42.0486

The precision of the polynomial is evaluated on simulated data by studying the relative difference between the estimated value and true momentum of the simulated particle.

$$\mathit{\Delta}_p/p = (p_{\text{reco}} - p_{\text{true}})/p_{\text{true}} \tag{7.18}$$

Figure 7.12 shows the root meant square (top) and median (bottom) of the observed σ_p/p distributions as function of momentum. In comparison to the *PrForwardTracking*, the observed momentum resolution is better at momenta below 10 GeV. This is due to the improved calculation of C in Eq. (7.17), which includes the additional term $c_6 t_x^2$. The addition is motivated by the fact that C approximates the integrated magnetic field along the particle's trajectory, thus a track's x-slope before the magnet contains valuable information. Furthermore, the improved momentum formula lead to the elimination of the observable biases of the old formula as visible from the comparison of the medians.

With greater momentum, the observed resolution rises for both algorithms while the slope is larger for the *TrackForwarding*. Similar results are obtained for the *TrackForwarding* when using the same procedure to determine the momentum as in the *PrForwardTracking*. This points to the fact, that it is not a shortcoming of the formula by which the momentum is determined but of the input to it. Subsequently, the only possible source of the differences is the track's x-slope inside the SciFi as it is the only quantity which is determined by the algorithm. At higher momenta, the observed difference between the x-slopes before and after the magnet becomes smaller and is thus more sensitive to the errors on t_x^{fit}. Consequently, the observable effect is likely a result of the simpler track fit performed in the proposed algorithm.

The momentum estimates main use inside HLT1 is the selection of tracks with a larger transverse momentum. The largest transverse momentum requirements are usually those of the one-track HLT1 lines which require values of 1 to 2 GeV. As shown in Fig. 7.13, the resolution differences in that region are relatively small and unlikely to significantly impact a later selection in HLT1. For this reason and the fact that the majority of tracks exhibit a momentum resolution of $\mathcal{O}(1\,\%)$, the fit quality is sufficient for the long track reconstruction in HLT1.

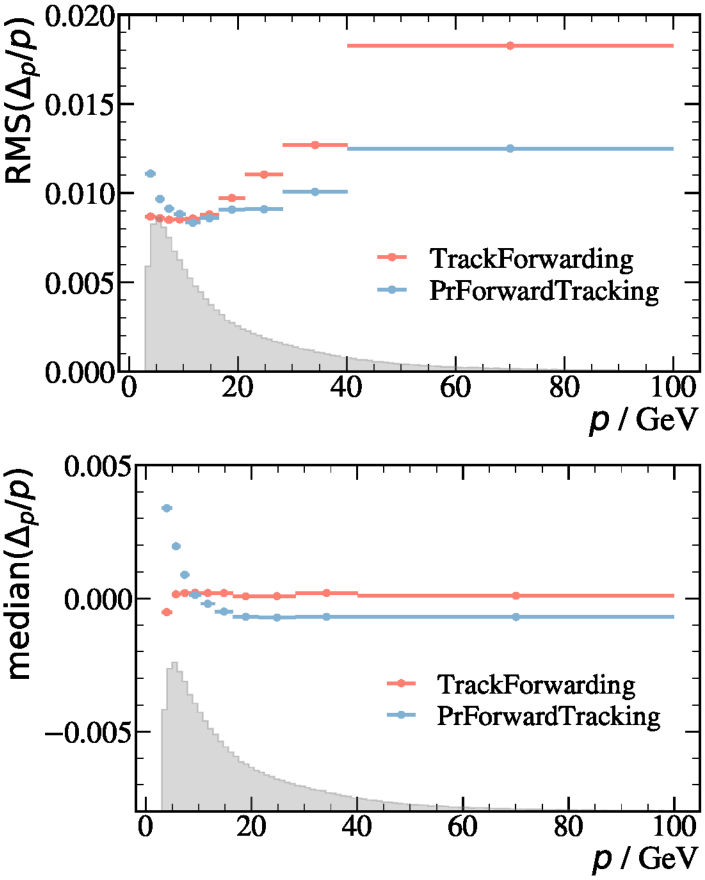


Figure 7.12: Comparison of the momentum resolution and bias between the *TrackForwarding* and the *PrForwardTracking*. Shown is the root mean square (top) and median (bottom) of the relative momentum difference between the reconstructed and simulated momentum of a particle track as a function of the simulated momentum. The width of each bin is chosen such that a bin contains 10 % of the tracks in the studied sample. The gray histogram shows the momentum distribution of these tracks.

Following the above calculation of a momentum estimate, the reconstruction of a long track candidate is complete. At last, the algorithm decides whether to reject or keep a candidate based upon a quality criterion. This selection is necessary for two reasons. First, as previously mentioned, it is possible that the algorithm successfully produces multiple candidates for a single input track. For this reason it is necessary

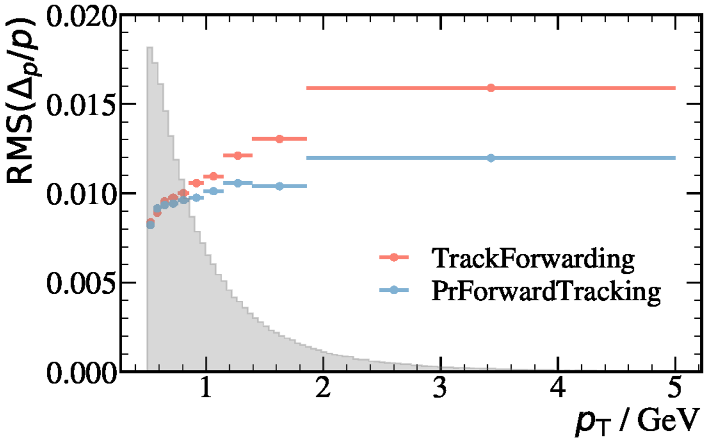


Figure 7.13: Continuation from Fig. 7.12. Shown is the root mean square of the relative momentum difference between the reconstructed and simulated momentum of a particle track as a function of the simulated transverse momentum.

to select which candidate is most likely to represent a real trajectory and should be kept. Second, it can be the case that the input upstream track is produced by a particle which does not reach the SciFi as it is bend out of the detector acceptance or decays. As a result, there cannot be a valid candidate for such an input and any created candidates are ghost tracks which should be rejected.

Both selections require a quality criterion which indicates whether a given candidate is a real track or a ghost track. Tasks of this nature are referred to as *Binary Classification* and are a very common problem in the field of machine learning. Accordingly, there are many possible solutions including but not limited to the commonly used boosted decision trees [76, 77] or neural networks. Nevertheless, the *TrackForwarding* uses the linear model,

$$Q = \sum_{i=0}^{n} \beta_i x_i, \tag{7.19}$$

to determine the quality score Q of a track. Here, x_i are the input features and β_i coefficients which are determined by performing a logistic regression. In contrast to more complicated methods like a neural network, this linear model requires very little processing time due to its simple structure. Additionally, as there are only eight different input features the dimensionality of the problem is low. In such scenarios, logistic regression is expected to yield sufficiently good results, not so far off from the more evolved methods.

The set of input variables is determined by selecting those which provide a good separation between real and ghost tracks. Each input variable's distributions is shown in Figs. 7.14 and 7.15. Tracks which are produced by a simulated particle are referred to as true tracks and make up the red distribution while those qualified as ghost tracks produce the blue distribution. In most cases, the natural logarithm of one plus a variable's value is used to reduce the variable's range and constrain it to positive values. The following explanation follows the order of the presented figures but focuses on only the variables, leaving out the natural logarithm.

$|\varDelta(qp)|/p_{\mathrm{UT}}$ measures the relative difference between the upstream track's momentum estimate and that of the final long track. As shown, random combinations are more likely to exhibit large differences than true trajectories.

χ^2_{fit} is the output of the track fit of the final candidate. Its value is expected to be higher for random combinations.

p_T is the track's transverse momentum. The density of hits is higher in the inner regions of the detector making it more likely for ghost tracks to be produced in this forward pointing inner region which is correlated with a low transverse momentum.

$\varDelta(\mathrm{UV}_1)$ is defined as the sum of the observed x-residuals of the stereo hit search. In case of a deviation of the y-coordinate, correctly selected hits should exhibit similar residuals in opposite direction. This is an effect of the rotation in opposite direction of the stereo layer. In cases where only a single stereo hit is collected this quantity is set to zero. The quantities $\varDelta(\mathrm{UV}_2)$ and $\varDelta(\mathrm{UV}_3)$ are defined similarly for the second and third layer.

#UV, #X Hits represents the number of stereo and x-layer hits a track contains. The more hits are on a track the more likely it is that the given track is not a ghost track.

As shown in the bottom plot of Fig. 7.15, the output of the linear model is able to differentiate well between the two classes. Based on this calculated track quality it is now possible to perform the previously mentioned selections. First, tracks which are likely to be ghost tracks are rejected by requiring all candidates to exhibit a quality score above 4.5. If multiple candidates are created, only the one with the highest quality score is kept. While it is possible for a ghost track to exhibit a higher score than the correct candidate, it is found that this is rare, $\mathcal{O}(1\,\%)$. To avoid rejecting these tracks would require allowing the creation of multiple long tracks per input track. As a result, more ghost tracks are created which is found to outweigh the small gain in efficiency. Thus, the single candidate selection based on a logistic regression remains the baseline choice for the *TrackForwarding*.

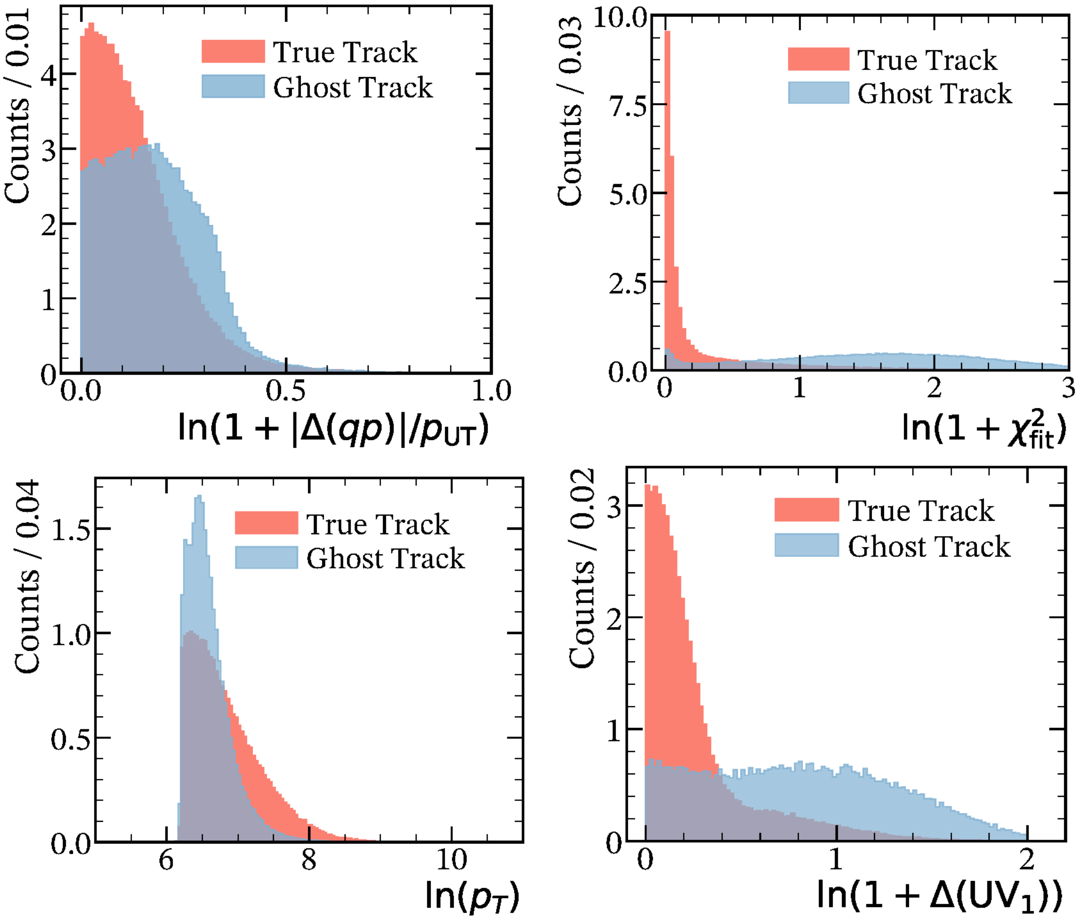


Figure 7.14: Distributions of the input variables to the multivariate classifier used to distinguish real tracks from ghost tracks. The red distributions show the reconstructed quantity of true tracks, while the blue distributions are from ghost tracks. The top left plot shows the relative difference of the momentum estimate from the long track candidate in respect to the upstream track's estimate. Next to it, the logarithm of one plus the χ^2 of the long track's fit, while the logarithm of the transverse momentum is shown below it. The bottom right shows the logarithm of one plus the sum of the x-residuals in the first station's stereo layers. In the case that both hits cannot be found, the value of Δ_{UV_1} is set to zero. For visibility reasons, these cases are omitted from the above figure.

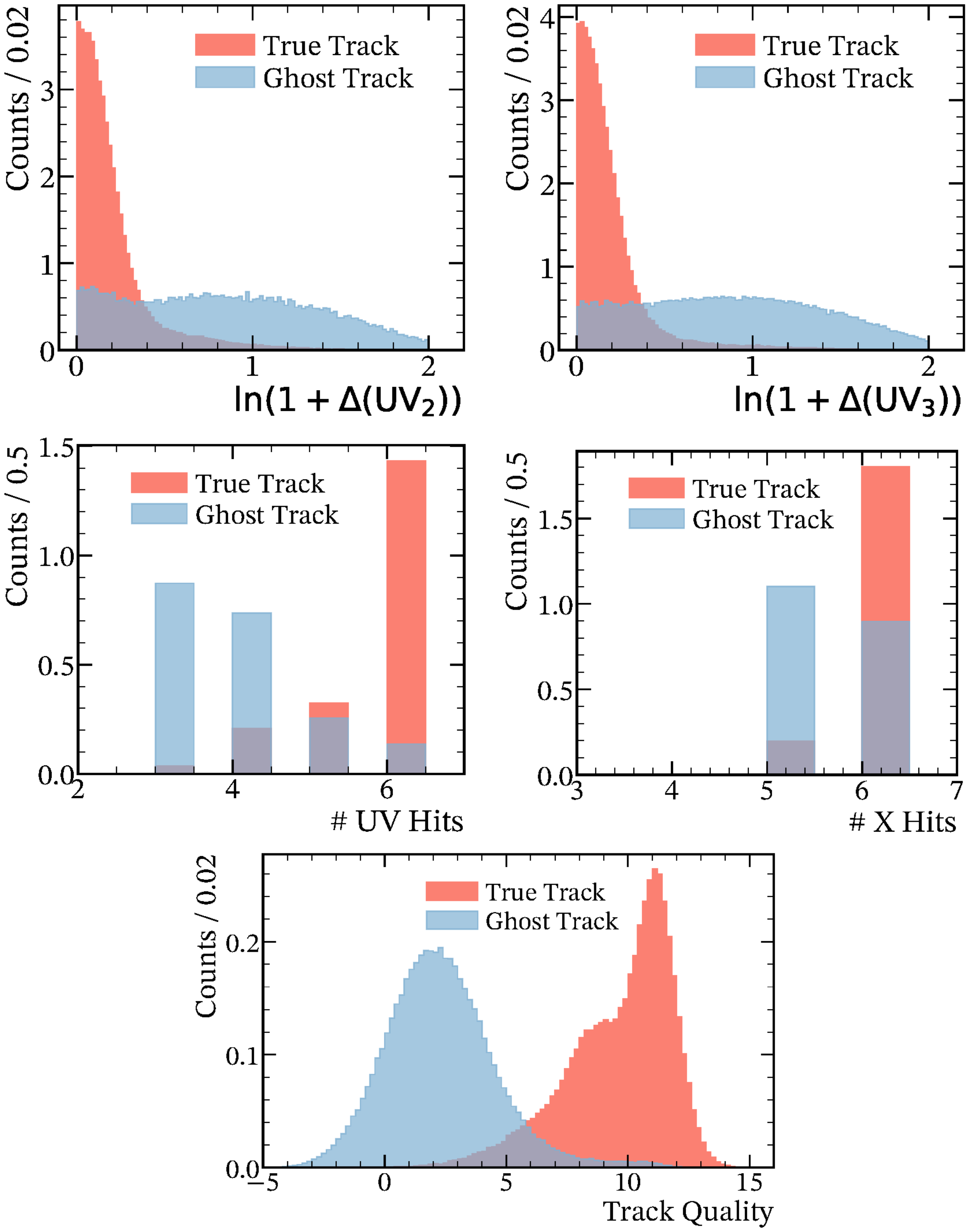


Figure 7.15: Continued from Fig. 7.14. The top two figures show the same quantity as $\ln(1+\Delta(\mathrm{UV}_1))$ but for the second and third station respectively. A long track candidate's number of found x-layer (left) and uv-layer (right) hits is shown the second row, while the last figure shows the response of the multivariate classifier.

7.3 Performance Evaluation

The following chapter presents a detailed performance comparison of the *TrackForwarding* to the current reference, the *PrForwardTracking*. At first, the physics performance, as it was described in Section 4.1.3, is presented for both algorithms. This is followed by a comparison of the required processing times of the two algorithms.

The tracking efficiencies in Table 7.6 were determined on a data sample of $B_s^0 \to \phi\phi$ signal events. While other signals are equally interesting, this sample represents a common choice to determine reconstruction efficiencies at LHCb. For this study, the *PrForwardTracking* is configured to use the recommended settings for an HLT1 scenario. It is worth reiterating, that in this configuration the goals are identical to the design requirements of the new algorithm. That is, to reconstruct only tracks with a transverse momentum greater than 500 MeV while keeping processing time as low as possible.

Table 7.6: Comparison of tracking efficiencies between the *TrackForwarding* and the *PrForwardTracking*. The efficiencies are normalized with respect to the upstream reconstruction. The last row shows the average ghost rate per event.

Track Type	TrackForwarding ϵ / %	PrForwardTracking ϵ / %
long, $p > 3\,\text{GeV}$, $p_T > 0.5\,\text{GeV}$	85.74 ± 0.00	85.74 ± 0.00
long, from B, $p > 3\,\text{GeV}$, $p_T > 0.5\,\text{GeV}$	86.67 ± 0.01	88.51 ± 0.01
ghost rate	$(4.74 \pm 0.00)\,\%$	$(2.54 \pm 0.00)\,\%$

The presented efficiencies are normalized to the reconstruction efficiencies of the upstream tracking. Thus, enabling a simpler interpretation as no knowledge of the performances of prior reconstruction steps is required. Table 7.6 lists the observed efficiencies for reconstructing specific subsets of long tracks. The first row lists the efficiency for the subset of tracks which are produced by particles with a momentum above 3 GeV and a transverse momentum above 500 MeV. It is precisely this subset of tracks that both algorithms are configured/designed to reconstruct. Between the two, no significant difference in reconstruction efficiency is observed for this category.

The efficiency given in the last row is determined for tracks from particles that meet both momentum requirements and are a direct or indirect daughter of a B meson decay. In contrast to most long tracks, tracks of this category originate from a displaced vertex instead of directly from a primary vertex. All analysis at LHCb which study the decays of beauty, charm, or strange mesons rely on an efficient reconstruction of these tracks. For this reason, it is important to check that the reconstruction does not penalize tracks which do not come from a position along the beam line. As shown in Table 7.6, both algorithms obtain higher efficiencies for tracks in this category than for the one listed on the first line. This is expected if the reconstruction does not penalize displaced tracks, as the particles coming from B meson decays exhibit a higher average momentum than those coming from the primary vertex. Combined with the momentum dependence of the reconstruction efficiency, shown in Fig. 7.16, this leads to the observed higher efficiency. The observed ghost rate is higher for the new algorithm due to its simpler ghost rejection procedure. Based on previous experience with common ghost rates of about 5 to 6 % in Run 1 and 2 [78], this ghost rate is considered not problematic. If a further ghost rejection is necessary, a more evolved rejection procedure can be introduced as discussed in Section 7.5. A more detailed comparison of the reconstruction efficiencies is depicted in Fig. 7.16. It shows the efficiencies of the first category as a function of momentum p, transverse momentum p_T, pseudorapidity η, and azimuthal angle ϕ. Shown in the first plot is the momentum dependence of the reconstruction efficiency for both algorithms. While the proposed algorithm shows better efficiencies in the low momentum region, the *PrForwardTracking* is more efficient for tracks above 7 GeV. The observed efficiency for the *TrackForwarding* remains almost constant for momenta higher than 10 GeV. In contrast, the *PrForwardTracking* does not reach its maximum efficiency until about 30 GeV. At that point the difference between the two algorithms is at the order of $\mathcal{O}(5\,\%)$. Furthermore, the *PrForwardTracking's* reconstruction efficiency decreases for higher momenta, which is a shortcoming of the used formula to define the initial momentum dependent hit search window. A fix for this behavior is likely possible but was not yet available at the time of this work.

The next two plots show the efficiency as function of transverse momentum and pseudorapidity. The observed relative differences are smaller in the low p_T and low η region as these are correlated to lower momentum. For higher values of η and p_t the efficiency differences reach the above mentioned levels. The efficiency as function of ϕ, shown in the last plot, exhibits a periodic shape. Observed reconstruction efficiencies are highest for the regions which correspond to the x- and y-axis with lower efficiencies between them. This is due to a higher material budget in the lower efficiency regions which increase the likelihood of tracks in this region to undergo scattering and thus become harder or impossible to reconstruct.

The observable drop of the *TrackForwarding's* reconstruction efficiency at $\phi =$

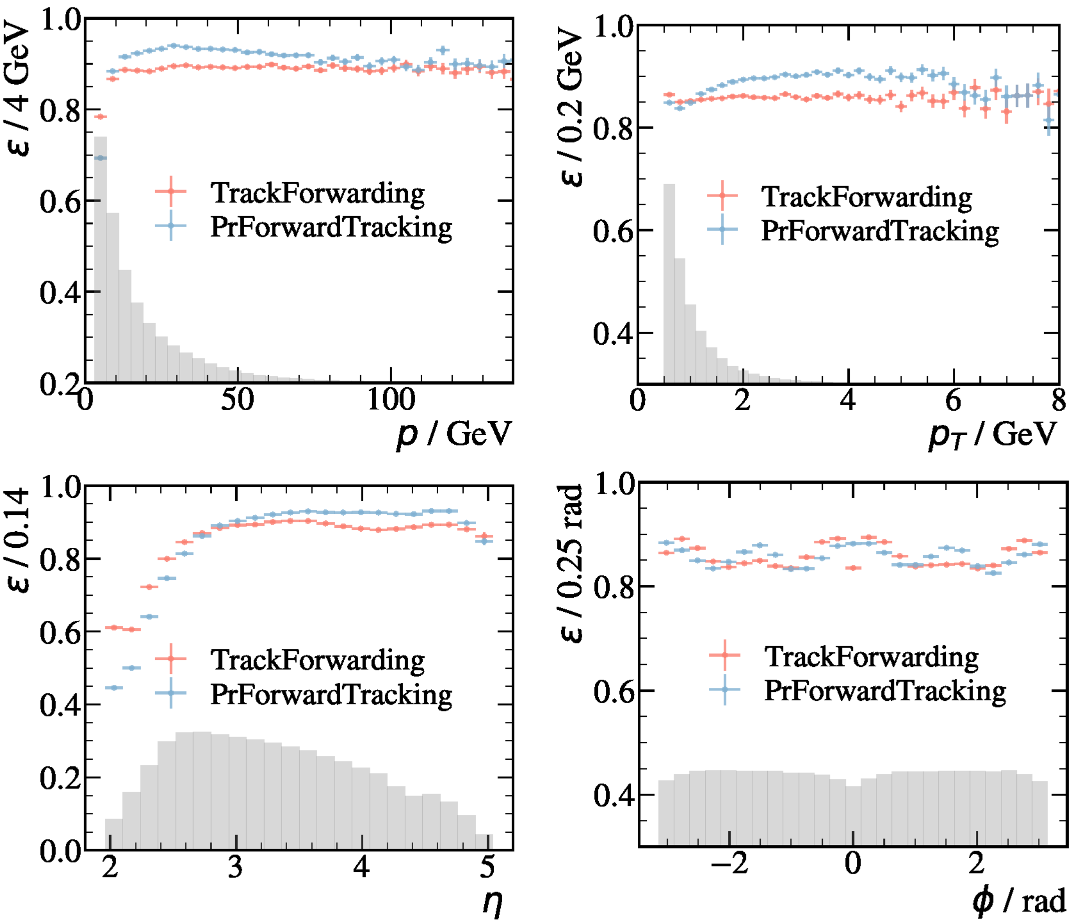


Figure 7.16: Track reconstruction efficiencies as function of the kinematic variables, momentum p, transverse momentum p_T, pseudorapidity η, and azimuthal angle ϕ. Efficiencies are compared between the *TrackForwarding* (red) and the *PrForwardTracking* (blue). The gray histogram shows the distribution of all tracks in the respective variable.

$[\pm\pi, 0]$ is an effect of the SciFi's geometry. Both algorithms only consider hits from the upper or lower half of the detector depending on the predicted flight path of the input upstream track. However, it is possible for tracks with the given values of the azimuthal angle to leave hits in detector modules of the opposite detector half they fly through. This is due to the rotation of the stereo modules, which is visualized in Fig. 7.17. As shown, it is possible for a track which traverses the detector above the x-axis to leave a hit in the lower detector half of the stereo layers. The *PrForwardTracking* accounts for this effect by performing an additional search for hits if tracks are within these regions. So far this has not been implemented in the new algorithm but is foreseen to be included in future work. The overall effect of this is however negligible, which is why the implementation of other features took precedence thus far.

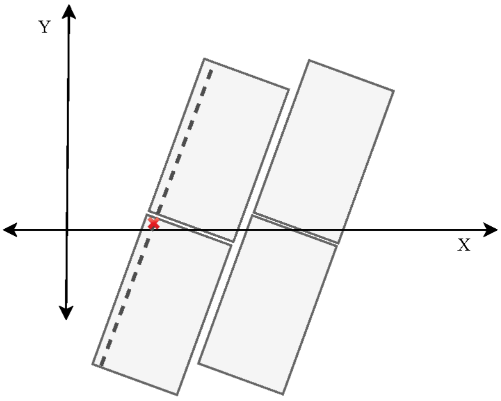


Figure 7.17: Schematic view of two rotated SciFi modules. Due to their rotation, a hit in the upper half (red cross) can leave a hit in a module of the lower detector half.

Summarizing these findings, the better relative performance in the low momentum range is likely due to the improved handling of magnetic field effects as described in Section 7.2.3. Handling this region well, it is surprising that the *TrackForwarding* is outperformed by the *PrForwardTracking* for the higher momentum tracks which are less complex to reconstruct. Using the additional information available in simulated data it is possible to determine the cause of this effect.

It is found that the different requirements on which hits are needed to reconstruct a track explain the effect. Overall, both algorithms require a minimum of five x-layer hits, yet the new algorithm requires two of them to be in the last station. The efficiencies of these hit requirements are determined using the same sample as above. The *PrForwardTracking* requires any combination of five out of the six

x-layers which is possible for 98 % of the reconstructible tracks in the sample. In contrast, only 92 % of the reconstructible tracks leave a hit in each of the two x-layers of the last station. Thus, it is the requirement of finding the x-layer doublet in the last station that imposes an upper bound on the achievable reconstruction efficiency. On the other hand, the required doublet provides a good estimate of the track's slope and in turn enables precise extrapolations into the remaining stations. Loosening this requirement is likely to require a more complex procedure as well as larger thresholds to find the remaining hits. For this reason, the hit requirements are kept as is at this point. At last, whether the observed loss in efficiency is deemed acceptable depends on the comparison of the processing time of the two algorithms.

To ensure a realistic estimation of an algorithm's processing time in the trigger, the timing measurements are performed using so called minimum bias data. These data samples contain simulated data that aims to resemble the data which will be read out from the detector. The remaining procedure is similar to the one described in Chapter 5 and [60]. That is, a HLT1 reconstruction sequence is configured and used to reconstruct a large amount of simulated data. At the start of the sequence a global event cut (GEC) is applied to filter out events which exhibit a large multiplicity. The cut is implemented as a requirement on the sum of the number of hits in the upstream and SciFi detector. Very busy events cost significantly more time to reconstruct and usually exhibit lower track reconstruction efficiencies, thus it is beneficial to remove them early on.

Events which pass the GEC first undergo the VELO track reconstruction which is followed by the determination of the primary vertices. Next, upstream tracks are reconstructed if they exhibit a transverse momentum larger than 400 MeV. Based on these, the long track reconstruction aims to find all long tracks with a transverse momentum larger than 500 MeV. In contrast to the used sequence in [60], this reconstruction sequence does not include a cut on the impact parameter and uses looser requirements on the transverse momentum. Similar to [60], a Kalman Filter fit of the long tracks is not included in the benchmarked sequence. In its current implementation it introduces a significant overhead and a better alternative for HLT1 is in the process of being implemented. As this version was not available at the time of this work, it is decided to omit the fit of the long tracks from the benchmarked sequence. The fraction of the overall processing time taken up by the final fit is expected to be of the order of 1 to 2 %. Consequently, the omission is not expected to have any impact on the results or conclusions of the following measurement.

The timing measurements are performed on one of LHCb's available reference server nodes. This is useful, as LHCb's upgrade event filter farm is expected to contain the computing power of about 1000 of these machines. Subsequently, the necessary processing rate per machine is required to reach at least 30 kHz. The results of the timing measurements are presented in Table 7.7.

Table 7.7: Comparison of the processing time between the new and current long track reconstruction. Shown in the first row is the throughput of the entire HLT1 reconstruction including either the *TrackForwarding* or the *PrForwardTracking*. The fraction of processing time spend inside the long track reconstruction algorithm is shown in the second row with the resulting theoretical throughput of the algorithm itself in the last row.

	PrForwardTracking	TrackForwarding
HLT1 Throughput	21.09 kHz	29.76 kHz
Fraction in Algorithm	32.92 %	7.22 %
Algorithm Throughput	62.19 kHz	412.16 kHz

It is worth noting, that there have been significant advances in LHCb's reconstruction software since the last performance evaluation in [60]. As shown, the per machine throughput of the HLT1 reconstruction without an IP cut is at 21.09 kHz compared to the previously measured 12.4 kHz for a reconstruction including an IP cut in [60]. This tremendous speedup is due to many improvements of which a majority are within the VELO and primary vertex reconstruction as well as inside the underlying software framework (GAUDI) itself.

As listed, the *PrForwardTracking* takes almost one third of the entire processing time of the presented sequence. This is the largest fraction of processing time required by a single algorithm, followed by VELO tracking at 24 %. In contrast, it is shown that a reconstruction sequence which uses the *TrackForwarding* is able to process events at a rate of nearly 30 kHz. Therefore, for the first time enabling an HLT1 reconstruction sequence without an IP cut which fits within the predicted computing budget. Furthermore, the new algorithm only accounts for a bit over 7 % of the total processing time, which translates into a throughput of the algorithm of 412 kHz. An over six-fold relative speedup compared to the *PrForwardTracking*.

Given the significant speedup, the *TrackForwarding* might enable better selection efficiencies in HLT1 compared to the default solution. For this reason the following section studies HLT1 selection efficiencies for the two different long track reconstruction strategies.

7.4 Comparison of Trigger Selection Efficiencies

While the proposed reconstruction procedure for long tracks provides a significant speedup, its lower efficiency raises the question which solution is better suited for the reconstruction of long tracks in HLT1. To answer that question, HLT1 lines, similar to the ones described in Chapter 5, are implemented and studied for both algorithms. At the time of this work, the muon reconstruction for the upgrade has yet to be finalized. For this reason, the following study does not include any HLT1 lines aiming to select prompt decays.

Signal samples from LHCb's official Monte Carlo simulation are used as data for these studies. These signal samples ensure the presence of the given signal decay in each event, thus providing large enough statistics even for rare decays. Yet, not all decay products are ensured to be in LHCb's acceptance. For this reason, a loose selection is applied to extract the signal events which can be fully reconstructed. The selection additionally includes loose cuts on the transverse momentum of all particles and a lifetime requirement for the parent hadron. A summary of the selection requirements is shown in Table 7.8. In a later physics analysis of any of the studied decays, a similar selection is usually used to remove background events. Thus, the additional cuts reject events that would later not be selected in a physics analysis. Subsequently, not selecting them in HLT1 should not be counted as a loss in efficiency. When determining the selection efficiency of an HLT1 line, the number of signal events which pass this loose selection represents the efficiency denominator while the enumerator is given by the amount of events which are selected by the HLT1 line when processing the filtered signal sample.

Table 7.8: Loose selection criteria applied to signal samples to determine a set of fully reconstructible events.

Particle	Selection cut
Parent Hadron	$p_T > 2\,\text{GeV}$ $2 < \eta < 5$ $\tau < 0.2\,\text{ps}$
Daughters	$p_T > 200\,\text{MeV}$ $2 < \eta < 5$

For this comparison, three different scenarios are studied. Each consists of an upfront reconstruction upon which a one-track and two-track HLT1 selection is implemented. The reconstruction sequences only differ by their employed long track reconstruction. The remaining sequence is identical to the one used in the timing studies outlined in the previous section. However, the previously mentioned Kalman Filter is appended to the end of the sequence. The fit of the long tracks is necessary to achieve a better IP resolution and the resulting χ^2 of the track's fit is used in the selections. Earlier mentioned overheads of the Kalman Filter's implementation can be disregarded as the processing speed is irrelevant to this study.

The first scenario uses the *PrForwardTracking* in the same configuration as was used in the previous section. This scenario is further referred to as *loose* scenario. As shown in Table 7.7, the throughput of this sequence would not suffice to actually enable an HLT1 processing rate of 30 MHz. Yet it is educational to study this scenario to see the impact of the previously shown differences in long track reconstruction efficiencies on the efficiencies of an HLT1 line. The *realistic* scenario is based on a reconstruction sequence which also uses the *PrForwardTracking* but raises its minimum transverse momentum requirement to a value of 1.4 GeV. The value is chosen such that the overall throughput of the sequence is identical to that of the one using the *TrackForwarding*. This requirement represents one of many possible configurations to enable a processing rate of 30 MHz for reconstruction sequences based on the default long track reconstruction. Other options include the use of an IP cut as in [60] or a combination of them. The transverse momentum cut based configuration is chosen for its advantage that it enables the reconstruction and selection of prompt signal decays. The third scenario uses a reconstruction sequence that performs the long track reconstruction by the means of the here *TrackForwarding*.

The following selections for the one- and two-track lines are based upon their predecessors from Run 2. Slight modifications are made to adapt them to the upgrade data taking conditions. These changes are identical to the ones made in previous upgrade studies [60, 79]. Both selections only consider tracks which exhibit a good fit quality, that is the fit's χ^2 per degree of freedom is below a value of 2.5.

The selection of the one-track HLT1 line is based on the following requirements,

$$p_T > 1 + \alpha \tag{7.20}$$

$$\ln(\chi^2_{\mathrm{IP}}) > \begin{cases} \frac{1}{(p_T-\alpha)^2} + \frac{1.2}{25}[25 - (p_T - \alpha)] + \ln(7.4) & \text{if } p_T < 25\,\mathrm{GeV} \\ \ln(7.4) & \text{otherwise} \end{cases}, \tag{7.21}$$

on the transverse momentum p_T and the impact parameter significance χ^2_{IP}. The value of the variable α is chosen such that the output rate of the HLT1 line is 500 kHz when processing minimum bias data.

Table 7.9 lists the selection criteria for the two-track line. First, the momentum, transverse momentum, and IP requirements are applied to all long tracks. Second, a vertex fit is performed for all two-track combinations. If the quality of the vertex fit χ^2_{vertex} is good, the remaining vertex requirements are checked. The corrected mass is defined as

Table 7.9: Loose selection criteria applied to signal samples to determine a set of fully reconstructible events.

Object	Selection cut
Tracks	$p_T > 0.5\,\mathrm{GeV}$ $p > 5\,\mathrm{GeV}$ $\chi^2_{\mathrm{IP}} > 6$
Vertex	$\chi^2_{\mathrm{vertex}} < 10$ $p_T > 2\,\mathrm{GeV}$ $m_{\mathrm{corrected}} > 1\,\mathrm{GeV}$ $2 < \eta_{\mathrm{PVtoSV}} < 5$ MVA Response $> \beta$

$$m_{\mathrm{corrected}} = \sqrt{m^2 + |p_{T,\mathrm{miss}}|^2} + |p_{T,\mathrm{miss}}|, \tag{7.22}$$

where $p_{T,\mathrm{miss}}$ is the missing momentum transverse to the parent hadron's direction of flight as defined by the primary vertex and reconstructed two-body vertex. The pseudorapidity of the direction of flight vector η_{PVtoSV} is required to be within LHCb's acceptance. At last a MatrixNet [80] multivariate classifier is used to predict the likelihood that the two-track combination is produced by the decay of a heavy flavor hadron. This classifier is identical to the one used in Run 2 [81], as a retrained version for Run 3 was not yet available at the time of this work. Inputs to the classifier are the sum of the transverse momentum of the two tracks, the fit quality of the two-track vertex, the number of tracks which satisfy $\chi^2_{\mathrm{IP}} < 16$, and the significance of the flight distance. The minimum required response value to accept a two-track combination is given by β. Its value is determined in the same way as α to ensure an output rate of 500 kHz.

The chosen values for α and β for the three different scenarios are given in Table 7.10. It shows a lower value of α and β for the one- and two-track line based upon the

Table 7.10: Used values for α and β in the one- and two-track HLT1 lines. They are chosen such that the respective HLT line delivers an output rate of 0.5 MHz.

Variable	Loose	Realistic	TrackForwarding
α	1.0	1.0	0.74
β	0.9537	0.0	0.9439

TrackForwarding. A result of their lower rate on minimum bias data compared to the lines based on the *PrForwardTracking*. As the one-track line based on the *loose PrForwardTracking* only uses tracks with a transverse momentum above 2 GeV, increasing the respective cut in the tracking algorithm to 1.4 GeV has no effect on the rate of the line. Subsequently, the value for α remains the same for the *realistic* scenario. However, this scenario's more stringent requirements have a high impact on the two-track line which was originally using tracks with a higher transverse momentum than 0.5 GeV. The resulting rate of this line is low enough in the *realistic* scenario that the MatrixNet classifier is not required to further lower the rate.

Table 7.11 shows the resulting selection efficiencies for the one- and two-track lines as well as for their combination. The total selection efficiency for a given signal decay is obtained by multiplying the listed efficiency of the global event cut and that of a given HLT1 line. Comparing the third and last column, it is shown that the combined selection efficiencies of the *loose* HLT1 lines are on average slightly higher than that of the lines using the *TrackForwarding*. Yet, observed differences are smaller than one might expect based upon the listed tracking efficiencies in Table 7.6. This can be seen in the comparison of the selection efficiency for the decay of $B_s^0 \rightarrow \phi\phi$, which is the same decay used to determine the track reconstruction efficiencies. While Table 7.6 states an up to 3 % higher tracking efficiency of the *PrForwardTracking*, the combined selection efficiency of the HLT1 lines based on this algorithm are only 0.2 % higher. Furthermore, the one-track line alone is 2.5 % less efficient than the one using the *TrackForwarding*.

To provide a simplified explanation for this effect, tracking efficiencies are assumed to be momentum independent. Consequently, the *PrForwardTracking* has a 11 % chance to miss the reconstruction of a track compared to 14 % for the new algorithm. The given decay produces four kaon tracks from the two instantly decaying phi mesons. While one to four out of these can satisfy the one-track line's requirements, only one of them is necessary to correctly accept the event. From this follows that the difference in selection efficiency of an event is given by the difference of the likelihoods to miss the reconstruction of all final state particles which satisfy the lines requirements. Let n_d denote the number of final state particles which satisfy the *loose* one-track line's requirements and n_n the corresponding number for the line of the *TrackForwarding* scenario, then

$$\mathcal{L}_{\text{default}} = 0.11^{n_d}, \tag{7.23}$$

$$\mathcal{L}_{\text{new}} = 0.14^{n_n}, \tag{7.24}$$

represent the likelihood of missing the track reconstruction of all these particles, for the *PrForwardTracking* and *TrackForwarding* respectively. From this follows that the difference of the likelihoods becomes less relevant the larger the values of n_d and n_n. In addition, it is possible that n_n is larger in some cases than n_d due to the one-track line's lower transverse momentum requirement in the new scenario. Subsequently, the *loose* scenario line will exhibit a lower probability to correctly select the event in these cases. Looking at Table 7.11, it is clear that this mostly occurs for decays with many final states as they are more likely to exhibit lower transverse momenta in the 1.74 to 2 GeV range. In conclusion, the one-track line of the *loose* scenario only exhibits superior performances for two-body decays with high momentum daughters, see for example $B_s^0 \to \mu^-\mu^+$.

A similar logic as above applies to the two-track lines' selection efficiencies. Here, the larger differences between the two scenarios are due to the greater difference between the likelihoods of missing the reconstruction of a two-track combination. The estimated chance to miss a combination is 20 % and 26 % the *PrForwardTracking* and *TrackForwarding* respectively. While the two-track line using the new algorithm requires a looser MatrixNet response, this is not sufficient to entirely remedy the effects of the higher tracking inefficiency of the *TrackForwarding*. Resulting differences in selection efficiency are observed to be higher than the previously mentioned gains in efficiency of the new scenario's one-track line. Consequently, the combined efficiencies of the *loose* scenario's HLT1 lines provide a marginally better selection compared to those using the new algorithm.

At last follows the comparison of the fourth column of Table 7.11, the *realistic* scenario, to the efficiencies of the HLT1 lines based on the *TrackForwarding*. Given that both of these scenarios are able to process events at the necessary rate of 30 MHz, it is this comparison that determines which algorithm is the better choice for HLT1. For decays which are simpler to select, for example $B_s^0 \to \mu^-\mu^+$ or $B^0 \to K^-\pi^+$, the observed relative efficiency gain of the *TrackForwarding* based trigger selection is merely about 3 %. Yet, for harder to select decay topologies the gain in selection efficiency can increase up to a two-fold improvement. Looking at the selection efficiency of $B^0 \to K^*e^-e^+$, The relative gain of the HLT1 lines using the *TrackForwarding* over the *realistic* scenario is about 34 %. A significant improvement, especially considering that the selection of this decay represents a crucial part of LHCb's efforts to improve its previous tests of lepton universality [82]. Other key decay topologies for LHCb's upgrade physics program are the decay of $D^0 \to K^0_S\pi^-\pi^+$ and $\tau^- \to \mu^-\mu^-\mu^+$, for which the *TrackForwarding* enables a two-fold increase of the selection efficiency. In conclusion, when considering scenarios that process events at the required rate of 30 MHz, the *TrackForwarding* is shown to currently provide the best solution to configuring one- and two-track HLT1 lines for the selection of events including displaced hadron decays. An additional improvement of the *TrackForwarding* and *realistic* scenario is the possibility for the selection of prompt signal decays in contrast to the presented scenario in [60].

It should be noted, that the presented results are obtained using a snapshot of LHCb's software framework at the time of this work. Due to the ongoing software developments, the quoted performances are likely to change before data taking resumes in 2021.

Table 7.11: Trigger selection efficiencies of various decay modes for three different track reconstruction scenarios. The total selection efficiency of a line is given by the multiplied efficiencies of the global event cut and that of the chosen line. All quantities are given in percent.

		PrForwardTracking		**TrackForwarding**
Decay Mode	**GEC**	**Loose, 21 MHz**	**Realistic, 30 MHz**	**30 MHz**
		1-Track / 2-Track (Combined)		
$B_s^0 \to \mu^-\mu^+$	85.5	75.8 / 66.2 (82.7)	75.8 / 43.3 (78.5)	74.3 / 62.4 (80.6)
$B^0 \to K^*\mu^-\mu^+$	83.9	62.2 / 80.4 (84.1)	62.2 / 46.8 (70.0)	63.3 / 79.0 (83.1)
$B^0 \to K^*e^-e^+$	83.5	48.3 / 69.3 (73.9)	48.3 / 32.7 (54.8)	49.8 / 67.9 (73.3)
$B^+ \to D^0(K_S^0\pi^+\pi^-)K^+$	83.6	62.3 / 65.2 (79.0)	62.3 / 25.9 (65.9)	63.3 / 63.2 (78.2)
$B^0 \to D^+D^-$	83.7	56.3 / 85.2 (87.6)	56.3 / 42.0 (62.9)	58.6 / 84.1 (87.0)
$B^0 \to K^-\pi^+$	84.4	74.8 / 64.8 (81.8)	74.8 / 41.4 (77.5)	73.4 / 61.1 (79.8)
$B^0 \to K_S^0\pi^-\pi^+$	85.0	62.6 / 56.4 (74.2)	62.6 / 26.1 (66.1)	62.6 / 54.7 (73.7)
$B_s^0 \to \phi\phi$	85.3	48.0 / 78.7 (81.1)	48.0 / 50.0 (60.9)	50.5 / 78.2 (80.9)
$B_s^0 \to \phi\gamma$	85.2	29.8 / 47.0 (54.4)	29.8 / 25.3 (37.4)	31.6 / 45.4 (54.1)
$B^0 \to D^{*+}\tau^-(\pi^-\pi^-\pi^+\nu_\tau)\bar{\nu}_\tau$	83.4	62.7 / 85.2 (89.1)	62.7 / 42.1 (69.7)	65.1 / 84.6 (88.8)
$B^0 \to D^{*+}\tau^-(\mu^-\nu_\tau\bar{\nu}_\mu)\bar{\nu}_\tau$	83.9	56.7 / 73.7 (82.0)	56.7 / 32.1 (63.0)	59.1 / 72.1 (81.5)
$D^0 \to K^+K^-$	84.1	18.9 / 41.8 (46.1)	18.9 / 13.7 (24.7)	19.9 / 40.3 (45.4)
$D^0 \to K_S^0\pi^-\pi^+$	84.3	11.2 / 23.8 (27.8)	11.2 / 5.3 (13.6)	12.5 / 23.4 (28.4)
$\tau^- \to \mu^-\mu^-\mu^+$	82.7	26.1 / 59.6 (62.9)	26.1 / 16.8 (31.4)	28.1 / 59.2 (63.0)

7.5 Summary

A new algorithm to reconstruct long tracks from upstream tracks, the *TrackForwarding*, has been presented. This approach exhibits better to similar track reconstruction efficiencies for particle momenta up to 7 GeV. For tracks with higher momentum, the presented algorithm suffers from its stricter hit requirements and thus exhibits lower efficiencies than the *PrForwardTracking*. Studying the processing times shows that the *TrackForwarding* is a factor of six faster than the *PrForwardTracking*.

A comparison of the selection efficiencies of HLT1 lines shows that an HLT1 based on the *TrackForwarding* yields a only slightly lower performance than HLT1 lines using the *PrForwardTracking* with its recommended requirements. However, while the *TrackForwarding* based HLT1 is able to process events at about 30 MHz, the above mentioned HLT1 based on the *PrForwardTracking* is only able to reach an estimated 21 MHz. For this reason, a more realistic comparison is presented using two HLT1 scenarios that both process data at 30 MHz, one based on the *TrackForwarding* and one on a *PrForwardTracking* with more stringent requirements. This comparison shows that the *TrackForwarding* based HLT1 is able to provide higher efficiencies for all of the studied decay topologies. Including a 34 % relative gain for the $B^0 \rightarrow K^* e^- e^+$ decay and up to a two-fold improvement for the decays of $D^0 \rightarrow K^0_S \pi^- \pi^+$ and $\tau^- \rightarrow \mu^- \mu^- \mu^+$. These are significant improvements for decays which represent a substantial part of LHCB's upgrade physics program.

In conclusion, the new algorithm enables, for the first time, the configuration of a HLT1 selection that is able to process events at 30 MHz, ensure an efficient selection of secondary decays, and enable the selection of prompt signals. To determine the selection efficiencies of prompt decays, a similar study is planned once the muon reconstruction software is finalized. It is expected that the results will be similar to those observed for the selection of heavy flavor signals.

Current developments of the *TrackForwarding* are focused on improving the use of the CPU's SIMD instruction within the presented algorithm. First results are promising and hint at the possibility of a reduction in processing time of 20 % or more. The additionally available processing time can then be employed to improve upon previously discussed shortcomings of the algorithm. Primarily three possibilities come to mind.

First and likely most beneficial is revisiting the requirement of two hits in the last station. The requirement exists, as the error of the estimated slope from a straight line between a single hit and the point in the magnet is too large. Thus, it is not possible to calculate precise extrapolations into the other stations solely based on a single x-layer hit. However, these errors are smaller for higher momentum tracks, for which it might be sufficient to only require a single hit inside the last station. Other options include the use of a more evolved iterative procedure which can determine the slope with higher precision. Yet, this calculation is within the high combinatorics region of the algorithm which executes about 25 times per input track. Therefore, it is unclear if such procedures will not be too time intensive. An additional alternative is to search for a x-hit doublet within the second station if no accepted candidate is found for an input track. As a track is required to be reconstructed from at least five x-layer hits, one of these stations must contain a valid hit doublet for a given track.

In the case that more processing time can be spared it should be considered to improve the currently used fit and ghost rejection procedure. The former should be changed to a two-dimensional fit which considers all available hits. As a result, the observed momentum resolution is likely to be improved for higher momentum tracks. At last, it should be studied whether it is worth to replace the logistic regression for a more evolved ghost rejection procedure. Given the problem falls within a standard binary classification task, a likely improvement would be the adoption of a gradient boosted decision tree like LightGBM [83]. Nevertheless, an advantage of the current logistic regression is the simple implementation and training procedure. For this reason, it is likely beneficial to keep this procedure as long as other parts of the algorithm undergo changes and frequent retuning of the classifier is needed.

8 Summary and Outlook

At the time of the presented work, LHCb is performing a significant upgrade of its detector and trigger system, before restarting to take data in 2021. The focus of this thesis lies on the first stage of the upgrade trigger system, the HLT1, which will be required to process events at a rate of 30 MHz. Yet, the latest evaluation of the processing rate of the HLT1 software [60] shows that the fastest configuration is merely able to reach a processing rate of 12.4 MHz. In addition, the reconstruction sequence used to reach this throughput relies on an impact parameter selection after the VELO tracking. As a result, most trajectories originating from primary vertices are removed making a later selection of prompt signal decays impossible.

Based upon these findings, it is clear that significant improvements to LHCb's reconstruction software are necessary. This thesis presents two alternative approaches to improve said reconstruction software.

The alternative approach presented in Chapter 6 is a novel method which aims to improve the selection efficiencies of the above mentioned impact parameter cut on VELO tracks. The method is based on a new machine learning procedure that performs the fit of a VELO track while additionally providing a momentum estimate from the track's observed multiple scattering. It is shown that the used machine learning procedure successfully estimates a track's momentum, which enables improved uncertainty estimates for a track's origin position, especially for lower momentum trajectories. These results are further shown to enable a selection of displaced tracks that is superior to the one based on the Kalman Filter track fit. At equal signal efficiency, the new method is able to reduce the rate of false positives by 30 %. Thus, the new method represents a good alternative to the currently used IP selection as it can enable a looser IP requirement in HLT1 at equal throughput.

In a reconstruction scenario without an IP cut on VELO tracks, the current long track reconstruction is responsible for about one third of the overall processing time. For this reason, a significant speedup of this reconstruction procedure is a crucial step towards enabling the removal of the IP cut. Chapter 7 presents an alternative algorithm for the reconstruction of long tracks from upstream tracks, the *TrackForwarding*. The new algorithmic approach is shown to provide a six-fold speedup while achieving reconstruction efficiencies which are only marginally lower than that of the current solution, the *PrForwardTracking*. A following study of trigger selection efficiencies shows that an HLT1 configuration that is able to process events at 30 MHz, exhibits higher selection efficiencies when based on the *TrackForwarding* over the *PrForwardTracking*. The gain in selection efficiencies is observable for all studied decay topologies. It ranges from slight improvements for decays like $B_s^0 \to \mu^-\mu^+$, to 34 % relative gain for $B^0 \to K^*e^-e^+$, and up to a two-fold improvement for the decays of $D^0 \to K_S^0\pi^-\pi^+$ and $\tau^- \to \mu^-\mu^-\mu^+$. These improvements represent a significant gain for decay topologies that are of great interest to LHCb. Overall, the *TrackForwarding* enables a HLT1 configuration that is, for the first time, able to process events at the required rate and ensure high selection efficiencies of secondary decays while still enabling the selection of prompt decays. This represents a significant step forward in the efforts of providing a purely software based trigger system for Run III.

The *TrackForwarding* is in the process of becoming the default long track reconstruction in HLT1. Nevertheless, more work on the first trigger stage and the *TrackForwarding* is needed to improve the physics efficiencies as well as the throughput. This is necessary to increase the available time for the reconstruction and selection sequence of HLT2, which is estimated to require a similar speedup to HLT1. Due to the success of the presented algorithm, an adaption of a similar strategy will also be investigated as solution for the second trigger stage. Parts of the gained insights in Chapter 7 have already been successfully used to improve the long track reconstruction used in HLT2.

In conclusion, a team of physicists and computer scientists, including the author, has modernized LHCb's software framework and adapted existing algorithms to work within the new framework. Building on these improvements, the author developed and presented two new alternative approaches for the HLT1 reconstruction sequence. One of them followingly enabled an implementation of the first trigger stage that is for the first time reaching sufficient performance levels in processing time and physics efficiencies. The future implementation of the second stage is an outstanding and interesting problem, that is to benefit from the lessons learned during the optimization of HLT1.

A Supplementary Information

A.1 Data Samples

The following chapter includes additional information about the used data samples within this thesis. Most of it is LHCb specific information that is aimed to ease a possible reproduction of the presented results. Listed Event Types can be used to obtain additional information at̃citedecfiles. With sufficient access rights all of the used samples can be retrieved from LHCb's bookkeeping (BK) via the DIRAC portal [84].

Sample 1 A sample of inclusive b decays. Event Type: 10000000.
BK path: `Beam7000GeV-Upgrade-MagDown-Nu7.6-25ns-Pythia8/Sim09c-Up02/10000000`

Sample 2 A sample of $\mathrm{B}^0 \rightarrow K\pi$ decays. Event Type 11102005.
BK path: `Beam7000GeV-Upgrade-MagDown-Nu7.6-25ns-Pythia8/Sim09c-Up02/Reco-Up01/11102005`

Sample 3 A sample of inclusive $\mathrm{B}^0_\mathrm{s} \rightarrow \phi\phi$ decays. Event Type 13104012.
BK path: `Beam7000GeV-Upgrade-MagDown-Nu7.6-25ns-Pythia8/Sim09c-Up02/13104012`

Sample 4 A sample of minimum bias events. Event Type 30000000.
BK path: `Beam7000GeV-Upgrade-MagDown-Nu7.6-25ns-Pythia8/Sim09c-Up02/30000000`

Samples 5-18 Samples of the decays listed in Table 7.11, Event types are listed in the same order. Event Types: 13112001, 11114001, 11124001, 12165106, 11296013, 11102005, 11104124, 13104012, 13102202, 11160001, 11574001, 27163002, 27165100, 31113001.
BK path: `Beam7000GeV-Upgrade-MagDown-Nu7.6-25ns-Pythia8/Sim09c-Up02/Reco-Up01/EventType`

Bibliography

[1] L. Evans and P. Bryant. “LHC Machine”. *JINST* 3 (2008).

[2] “Annual Report Full text - English version”. *CERN Annual Reports* 0.0 (2016).

[3] S. L. Glashow. “Partial-symmetries of weak interactions”. *Nucl. Phys.* 22 (1961), pp. 579–588.

[4] A. Salam and J. Ward. “Electromagnetic and weak interactions”. *Phys. Lett.* 13 (1964), pp. 168–171.

[5] S. Weinberg. “A Model of Leptons”. *Phys. Rev. Lett.* 19 (21 Nov. 1967), pp. 1264–1266.

[6] D. Griffiths. *Introduction to Elementary Particles*. Physics textbook. Wiley, 2008.

[7] *The Standard Model of particle physics*. URL: `http://www.physik.uzh.ch/groups/serra/StandardModel.html` (visited on 09/07/2016).

[8] ATLAS Collaboration. “Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC”. *Phys. Lett.* B716 (2012), pp. 1–29.

[9] CMS Collaboration. “Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC”. *Phys. Lett.* B716 (2012), pp. 30–61.

[10] C. Lefevre. “LHC: the guide (English version). Guide du LHC (version anglaise)”. Feb. 2009. URL: `https://cds.cern.ch/record/1165534`.

[11] ATLAS Collaboration. “The ATLAS Experiment at the CERN Large Hadron Collider”. *JINST* 3.08 (Aug. 2008), S08003–S08003.

[12] ALICE Collaboration. “The ALICE experiment at the CERN LHC”. *JINST* 3.08 (Aug. 2008), S08002–S08002.

[13] CMS Collaboration. “The CMS experiment at the CERN LHC”. *JINST* 3.08 (Aug. 2008), S08004–S08004.

[14] LHCb Collaboration. “The LHCb Detector at the LHC”. *JINST* 3 (2008).

[15] *LHCb Publications*. URL: `https://lhcbproject.web.cern.ch/lhcbproject/Publications/LHCbProjectPublic/Summary_all.html`.

[16] LHCb Collaboration. "First Evidence for the Decay $B_s^0 \to \mu^+\mu^-$". *Phys. Rev. Lett.* 110 (2 Jan. 2013), p. 021801.

[17] CMS & LHCb Collaboration. "Observation of the rare $B_s^0 \to \mu^+\mu^-$ decay from the combined analysis of CMS and LHCb data". *Nature* 522 (2015), pp. 68–72.

[18] LHCb Collaboration. "Measurement of the $B_s^0 \to \mu^+\mu^-$ branching fraction and effective lifetime and search for $B^0 \to \mu^+\mu^-$ decays". *Phys. Rev. Lett.* 118.19 (2017), p. 191801.

[19] LHCb Collaboration. "Observation of $J/\psi p$ Resonances Consistent with Pentaquark States in $\Lambda_b^0 \to J/\psi K^- p$ Decays". *Phys. Rev. Lett.* 115 (2015), p. 072001.

[20] LHCb Collaboration. "Observation of CP violation in charm decays". *Phys. Rev. Lett.* 122 (Mar. 2019), 211803. 12 p.

[21] M. Gell-Mann. "A Schematic Model of Baryons and Mesons". *Phys. Lett.* 8 (1964), pp. 214–215.

[22] LHCb Collaboration. *Letter of Intent for the LHCb Upgrade*. Tech. rep. CERN-LHCC-2011-001. Mar. 2011.

[23] T. Gerson. *Updated sensitivity projections for the LHCb Upgrade*. Tech. rep. CERN-LHCb-PUB-2013-015. Sept. 2013.

[24] *$b\bar{b}$ production angle plots*. URL: `https://lhcb.web.cern.ch/lhcb/speakersbureau/html/bb_ProductionAngles.html` (visited on 08/08/2016).

[25] LHCb Collaboration. *LHCb reoptimized detector design and performance: Technical Design Report*. Tech. rep. CERN-LHCC-2003-030. 2003.

[26] LHCb Collaboration. *LHCb Tracker Upgrade Technical Design Report*. Tech. rep. CERN-LHCC-2014-001. Feb. 2014.

[27] LHCb Collaboration. *LHCb VELO Upgrade Technical Design Report*. Tech. rep. CERN-LHCC-2013-021. Nov. 2013.

[28] LHCb Collaboration. *LHCb PID Upgrade Technical Design Report*. Tech. rep. CERN-LHCC-2013-022. Nov. 2013.

[29] A. Papanestis and C. D'Ambrosio. "Performance of the LHCb RICH detectors during the LHC Run II". *Nucl. Instrum. Meth.* A876 (2017), pp. 221–224.

[30] *The Gaudi project*. URL: `http://proj-gaudi.web.cern.ch/proj-gaudi/`.

[31] *The Gauss project*. URL: `https://lhcbdoc.web.cern.ch/lhcbdoc/gauss/`.

[32] T. Sjöstrand, S. Mrenna, and P. Skands. "PYTHIA 6.4 physics and manual". *Journal of High Energy Physics* 2006.05 (May 2006), pp. 026–026.

[33] D. J. Lange. “The EvtGen particle decay simulation package”. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 462.1 (2001). BEAUTY2000, Proceedings of the 7th Int. Conf. on B-Physics at Hadron Machines, pp. 152–155.

[34] J. Allison et al. “Geant4 developments and applications”. *IEEE Transactions on Nuclear Science* 53.1 (Feb. 2006), pp. 270–278.

[35] *The Boole project*. URL: `https://lhcbdoc.web.cern.ch/lhcbdoc/boole/`.

[36] *The Gaudi project*. URL: `http://proj-gaudi.web.cern.ch/proj-gaudi/`.

[37] *The DaVinci project*. URL: `https://lhcbdoc.web.cern.ch/lhcbdoc/davinci/`.

[38] *The Moore project*. URL: `https://lhcbdoc.web.cern.ch/lhcbdoc/moore/`.

[39] Particle Data Group. “Review of particle physics”. *Phys. Rev.* D98 (2018), p. 030001.

[40] R. Frühwirth and R. K. Bock. “Data analysis techniques for high-energy physics experiments”. *Camb. Monogr. Part. Phys. Nucl. Phys. Cosmol.* 11 (2000). Ed. by H. Grote, D. Notz, and M. Regler, pp. 1–434.

[41] *The Rec project*. URL: `https://lhcbdoc.web.cern.ch/lhcbdoc/rec/`.

[42] T. Bird et al. *VP Simulation and Track Reconstruction*. Tech. rep. CERN-LHCb-PUB-2013-018. Oct. 2013.

[43] E. Bowen and B. Storaci. *VeloUT tracking for the LHCb Upgrade*. Tech. rep. CERN-LHCb-PUB-2013-023. Apr. 2014.

[44] R. Quagliani. “Study of double charm B decays with the LHCb experiment at CERN and track reconstruction for the LHCb upgrade”. PhD thesis. Oct. 2017.

[45] A. Davis et al. *PatLongLivedTracking: a tracking algorithm for the reconstruction of the daughters of long-lived particles in LHCb*. Tech. rep. CERN-LHCb-PUB-2017-001. Jan. 2017.

[46] S. Esen and M. De Cian. *A Track Matching Algorithm for the LHCb upgrade*. Tech. rep. CERN-LHCb-PUB-2016-027. Dec. 2016.

[47] M. Benayoun and O. Callot. *The forward tracking, an optical model method*. Tech. rep. LHCb-2002-008. Feb. 2002.

[48] O. Callot and S. Hansmann-Menzemer. *The Forward Tracking: Algorithm and Performance Studies*. Tech. rep. CERN-LHCb-2007-015. May 2007.

[49] Y. Amhis et al. *Description and performance studies of the Forward Tracking for a scintilating fibre detector at LHCb*. Tech. rep. CERN-LHCb-PUB-2014-001. Apr. 2014.

[50] R. Kalman. “A New Approach To Linear Filtering and Prediction Problems”. *Journal of Basic Engineering (ASME)* 82D (Jan. 1960), pp. 35–45.

[51] J. Van Tilburg. “Track simulation and reconstruction in LHCb”. PhD Thesis. 2005.

[52] C. Fitzpatrick and V. V. Gligorov. *Anatomy of an upgrade event in the upgrade era, and implications for the LHCb trigger*. Tech. rep. CERN-LHCb-PUB-2014-027. May 2014.

[53] LHCb Collaboration. *LHCb Trigger and Online Upgrade Technical Design Report*. Tech. rep. CERN-LHCC-2014-016. May 2014.

[54] LHCb Collaboration. “Search for Dark Photons Produced in 13 TeV pp Collisions”. *Phys. Rev. Lett.* 120.6 (2018), p. 061801.

[55] LHCb Collaboration. “Design and performance of the LHCb trigger and full real-time reconstruction in Run 2 of the LHC”. *JINST* 14.04 (2019), P04013.

[56] LHCb Collaboration. “Tesla : an application for real-time data analysis in High Energy Physics”. *Comput. Phys. Commun.* 208 (2016), pp. 35–42.

[57] LHCb Collaboration. “A comprehensive real-time analysis model at the LHCb experiment”. *JINST* 14.04 (2019), P04006.

[58] LHCb Collaboration. *Upgrade Software and Computing*. Tech. rep. CERN-LHCC-2018-007. Mar. 2018.

[59] R. Aaij et al. *Upgrade trigger: Biannual performance update*. Tech. rep. CERN-LHCb-PUB-2017-005. Feb. 2017.

[60] M. De Cian et al. *Status of HLT1 sequence and path towards 30 MHz*. Tech. rep. CERN-LHCb-PUB-2018-003. Mar. 2018.

[61] LHCb Collaboration. “The LHCb Trigger and its Performance in 2011”. *JINST* 8 (2013), P04022.

[62] J. Albrecht et al. “Performance of the LHCb High Level Trigger in 2012”. *J. Phys. Conf. Ser.* 513 (2014), p. 012001.

[63] K. Rupp. *42 Years of Microprocessor Trend Data*. URL: `https://github.com/karlrupp/microprocessor-trend-data/tree/master/42yrs` (visited on 06/13/2019).

[64] M. J. Flynn. “Some Computer Organizations and Their Effectiveness”. *IEEE Transactions on Computers* C-21.9 (Sept. 1972), pp. 948–960.

[65] *Vectorisation Performance with QUANTIFI*. URL: `https://software.intel.com/sites/default/files/managed/98/cd/Vectorization_Performance_Quantifi.pdf` (visited on 06/14/2019).

[66] J. Albrecht, B. Couturier, C. Hasse, and S. Ponce. "New approaches for track reconstruction in LHCb´s Vertex Locator". *EPJ Web Conf.* 214 (2019), p. 01042.

[67] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780.

[68] *Multilayer Perceptron*. URL: `https://github.com/PetarV-/TikZ/tree/master/Multilayer%20perceptron` (visited on 08/05/2019).

[69] H. J. Kelley. "Gradient theory of optimal flight paths". *Ars Journal* 30.10 (1960), pp. 947–954.

[70] C. Olah. *Understanding LSTM Networks*. URL: `https://colah.github.io/posts/2015-08-Understanding-LSTMs` (visited on 08/05/2019).

[71] J. L. Elman. "Finding Structure in Time". *Cognitive Science* 14.2 (1990), pp. 179–211.

[72] J. Schmidhuber. "Untersuchungen zu dynamischen neuronalen Netzen". PhD thesis. Technical University Munich, 1991.

[73] *Tex StackExchange*. URL: `https://tex.stackexchange.com/questions/432312/how-do-i-draw-an-lstm-cell-in-tikz` (visited on 08/05/2019).

[74] L. N. Smith. "No More Pesky Learning Rate Guessing Games". *CoRR* abs/1506.01186 (2015).

[75] W. Kutta. "Beitrag zur Näherungsweisen Integration Totaler Differentialgleichungen". *Z. Math. Phys.* 46 (1901), pp. 435–453.

[76] L. Breiman et al. *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group, 1984.

[77] Y. Freund and R. E. Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". *Journal of Computer and System Sciences* 55.1 (1997), pp. 119–139.

[78] M. Stahl. "Machine learning and parallelism in the reconstruction of LHCb and its upgrade". *J. Phys. Conf. Ser.* 898.4 (2017), p. 042042.

[79] C. Fitzpatrick et al. *Upgrade trigger: Bandwidth strategy proposal*. Tech. rep. CERN-LHCb-PUB-2017-006. Feb. 2017.

[80] A. Gulin, I. Kuralenok, and D. Pavlov. “Winning The Transfer Learning Track of Yahoo!’s Learning To Rank Challenge with YetiRank”. *Proceedings of the Learning to Rank Challenge*. Ed. by O. Chapelle, Y. Chang, and T.-Y. Liu. Vol. 14. Proceedings of Machine Learning Research. Haifa, Israel: PMLR, June 2011, pp. 63–76.

[81] T. Likhomanenko et al. “LHCb Topological Trigger Reoptimization”. *J. Phys.: Conf. Ser.* 664 (Oct. 2015). 21st International Conference on Computing in High Energy Physics (CHEP2015).

[82] LHCb Collaboration. “Test of lepton universality with $B^0 \to K^{*0}\ell^+\ell^-$ decays”. *JHEP* 08 (2017), p. 055.

[83] G. Ke et al. “LightGBM: A Highly Efficient Gradient Boosting Decision Tree”. *NIPS*. 2017.

[84] *The LHCb Dirac Portal*. URL: `https://lhcb-portal-dirac.cern.ch/DIRAC/`.