

LHCb's Forward Tracking algorithm for the Run 3 CPU-based online track-reconstruction sequence

PAUL ANDRÉ GÜNTHER

*On behalf of the LHCb collaboration,
Physikalisches Institut
Heidelberg University, Germany*

ABSTRACT

In Run 3 of the LHC, the LHCb experiment faces very high data rates containing beauty and charm hadron decays. Thus the task of the trigger is not to select any beauty and charm events, but to select those containing decays interesting for the LHCb physics programme. LHCb has therefore implemented a real-time data processing strategy to trigger directly on reconstructed events. The first stage of the purely software-based trigger is implemented on GPUs performing a partial event reconstruction. In the second stage of the software trigger, the full, offline-quality event reconstruction is performed on CPUs, with a crucial part being track reconstruction, balancing track finding efficiency, fake track rate and event throughput. LHCb's CPU-based track reconstruction sequence for Run 3 is presented, highlighting the "Forward Tracking", which is the algorithm that reconstructs trajectories of charged particles traversing all of LHCb's tracking detectors. To meet event throughput requirements, the "Forward Tracking" uses SIMD instructions in several core parts of the algorithm, such as the Hough transform and the cluster search. These changes led to an improvement of the algorithm's event throughput by 60%.

PRESENTED AT

Connecting the Dots Workshop (CTD 2022)
May 31 - June 2, 2022

1 Introduction

During Run 3 of the LHC, LHCb will take data with new tracking detectors, new readout electronics and a new, purely software-based trigger [1, 2]. These upgrades to the LHCb experiment are necessary to efficiently process the data from 30 MHz of non-empty pp bunch crossings at the LHCb design luminosity of $\mathcal{L} = 2 \times 10^{33} \text{ cm}^{-2} \text{ s}^{-1}$ producing millions of beauty and charm hadrons per second. At these signal rates a hardware trigger is not anymore able to efficiently reduce the amount of data recorded as its bandwidth is already fully saturated by hadronic B decays [2, 3]. This problem is solved by a purely software-based trigger system that analyses and selects events in real time, *i.e.* performing the full, offline-quality event reconstruction. The most crucial part of the reconstruction sequence is the charged track reconstruction. The Run 3 LHCb detector employs three tracking detector technologies, a hybrid pixel detector (VELO, Vertex Locator) surrounding the pp interaction region [4], a silicon microstrip detector (UT, Upstream Tracker) [5] placed upstream of a dipole magnet and three stations of large-area scintillating fibre detectors (SciFi) [5] located downstream of the magnet. The whole LHCb detector is a single-arm forward spectrometer covering the pseudorapidity range $2 < \eta < 5$. The software trigger is split into two parts, the first stage is the *High Level Trigger 1* which performs a partial event reconstruction on GPUs [6–9], the second stage, called *High Level Trigger 2* (HLT2), reconstructs the full event using CPUs and selects decay candidates which are then sent to permanent storage for further analysis [10]. The full HLT2 event reconstruction includes charged track reconstruction, track parameter estimation using a Kalman filter, hadron, electron and muon identification and photon reconstruction. The HLT2 charged track reconstruction sequence is outlined in Sec. 2. One of the algorithms in this sequence is the *Forward Tracking*, highlighted in Sec. 3.

2 CPU-based charged track reconstruction in HLT2

LHCb defines five different track types described in Ref. [11]. Among these are VELO, T and Long tracks. VELO and T tracks have hits only in the VELO and SciFi detectors, respectively, while Long tracks traverse the whole tracking system and are required to have hits in both, the VELO and SciFi detectors, and can optionally have hits in the UT detector. Because they use all tracking detectors, Long tracks have the best momentum resolution and are therefore most used for physics analysis.

In HLT2, two independent algorithms reconstruct Long tracks, the *Forward Tracking* and the *Matching*, each of which employs a different track reconstruction method. The Forward Tracking and Matching both take VELO tracks as input, the reconstruction of which is done by the VELO Tracking described in Ref. [12]. The Matching further takes T tracks, reconstructed by the Hybrid Seeding algorithm described in Ref. [13], and matches them to VELO tracks which consequently become Long tracks. To find the correct combinations of VELO and T tracks, the Matching evaluates a multilayer perceptron trained on simulation and decides whether there is a match depending on the response of the neural network. The Forward Tracking is described in Sec. 3. Both algorithms are designed to find as many Long tracks as possible, which leads to a large overlap between the two sets of found Long tracks. In Run 2, this redundancy was accepted and resolved by selecting Long tracks according to their quality after applying a Kalman filter. This procedure increased the reconstruction efficiency [14]. In Run 3, however, the timing constraints posed by the real-time analysis trigger strategy favour avoiding redundancy. This is achieved by the reconstruction sequence shown in Fig. 1a. First the VELO tracking and Hybrid Seeding are run. The Matching then combines VELO tracks with T tracks. Residual VELO tracks, that could not be matched, are given to the Forward Tracking as input, which tries to create a Long Track using only SciFi hits that are not already taken into account by a Long Track found by the Matching.

The event throughput of the HLT2 reconstruction sequence is shown in Fig. 1b, including contributions from the Kalman filter and particle identification. Avoiding the aforementioned redundancy in the Long Track reconstruction, the HLT2 reconstruction sequence reaches more than 500 events per second and computing node, with the Forward Tracking not being a throughput-dominating component. More performance metrics such as reconstruction efficiencies of the HLT2 charged track reconstruction can be found in Ref. [15].

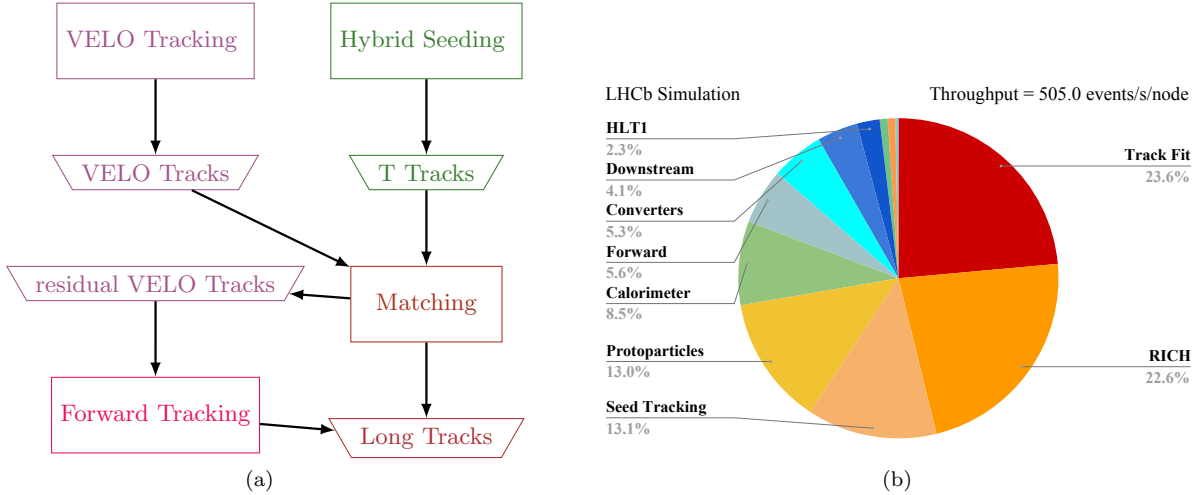


Figure 1: The Run 3 Long Track reconstruction sequence is shown in (a). Colours indicate that objects belong together. The throughput contribution of different components in the HLT2 reconstruction sequence is shown in (b) [16]. RICH, Calorimeter and Protoparticles refer to particle identification, the Track Fit is a Kalman filter. The Hybrid Seeding algorithm is called Seed Tracking here.

3 HLT2 Forward Tracking

The goal of the Forward Tracking is to find a forward extension of a given VELO track in the SciFi tracker and to estimate a preliminary* momentum of this Long Track. The SciFi detector consists of three stations, each with four layers of scintillating fibre modules. Therefore, the forward extension is a set of ten to twelve SciFi hits, with each hit belonging to a different layer of the SciFi detector. The hits the Forward Tracking algorithm searches for form a slightly curved trajectory that is compatible with originating from a VELO track. The line is curved in the xz - and yz -projection[†] because of fringe magnetic fields within the SciFi stations. The central problem to solve is finding the single correct combination of a VELO track with SciFi hits among the many possible extensions, or finding that no such combination is present, while keeping the reconstruction efficiency and event throughput high and the fake track fraction low. The complexity scales with the number of input tracks and the total number of hits recorded by the SciFi detector. Typical events with inelastic pp collisions lead to $\mathcal{O}(10^2)$ VELO tracks and $\mathcal{O}(10^3)$ SciFi hits. The Forward Tracking uses a method similar to a Hough transform [17] to efficiently and robustly spot hits forming the line pattern matching a VELO track. The following Sections 3.1-3.3 describe important components of the algorithm performed on individual input tracks.

3.1 SciFi hit selection

Starting from a state vector $(x, y, \frac{\partial x}{\partial z}, \frac{\partial y}{\partial z}, \frac{q}{p})$ containing the position, slopes and so far unknown charge q and momentum p at the end of the VELO, *i.e.* a VELO track, the Forward Tracking defines a polynomial $P(\frac{\partial x}{\partial z}, \frac{\partial y}{\partial z}, p)$ parameterising the propagation of the track in the xz -plane through the magnetic field down to the SciFi layers. As the momentum and charge are not known yet, this parameterisation is used to calculate x_{\min} and x_{\max} positions for each layer assuming a minimum reconstructible track momentum p_{\min} , *e.g.* $p_{\min} = 1.5$ GeV, and both possible charges. Only SciFi hits with $x \in [x_{\min}, x_{\max}]$ are selected for further processing, reducing the complexity of the problem. The determination of this hit search window by a polynomial is computationally cheap and therefore preferred over numerically solving the equations of motion

*The best momentum estimate is obtained by applying a Kalman filter to the track found bound by the Forward Tracking, using the preliminary momentum estimate as input.

[†]The LHCb coordinate system is a right-handed system with positive z running along the beamline from the interaction point into the detector and positive y pointing upward.

in the magnetic field in a real-time application.

3.2 Simplified particle trajectory

Similarly to the parameterisation used to define the hit search window (Sec. 3.1), the Forward Tracking defines a simplified particle trajectory by treating the magnet as an optical lens as described in Ref. [18]. Just like the model of light rays refracted by a thin lens, the particle’s movement in the xz -plane through the magnetic field is modelled by a straight line that gets a kick at the centre of the magnetic field and propagates further as a straight line with a different slope. Once a single hit (x, z) downstream of the magnet is taken into account, predicting the x coordinate at a given z position is a simple linear extrapolation within the model. Deviations from this model occur because of fringe magnetic fields that reach into the SciFi detector and are corrected for by parameterising the effect using polynomials as described in Ref. [19].

3.3 Hough-like transform

The main component of the Forward Tracking applies a map-reduce pattern inspired by the Hough transform to sieve out sets of SciFi hits that do not form a matching extension to the VELO track.

The x positions of SciFi hits selected by the search window described in Sec. 3.1 are mapped to a reference plane at a fixed z position. All hits’ x positions at the reference plane are calculated using the simplified trajectory introduced in Sec. 3.2 and filled into a histogram. The histogram counts the number of unique SciFi detector layers that are present among the hits in one bin. This way, hits that do not qualify as an extension to the VELO track form a flat distribution, while hits that match the VELO track accumulate in a few bins as depicted in Fig. 2. Subsequently, the histogram is scanned for small groups of neighbouring bins exceeding a layer-count threshold, thus reducing the large set of hits from within the search window to none, one or several small sets of hits, which become candidates for the VELO track extension. The found hit sets are then cleaned from outliers, fitted using a third-order polynomial function and further selected according to the fit result. The remaining candidates are promoted to Long tracks and their charge and momentum are estimated.

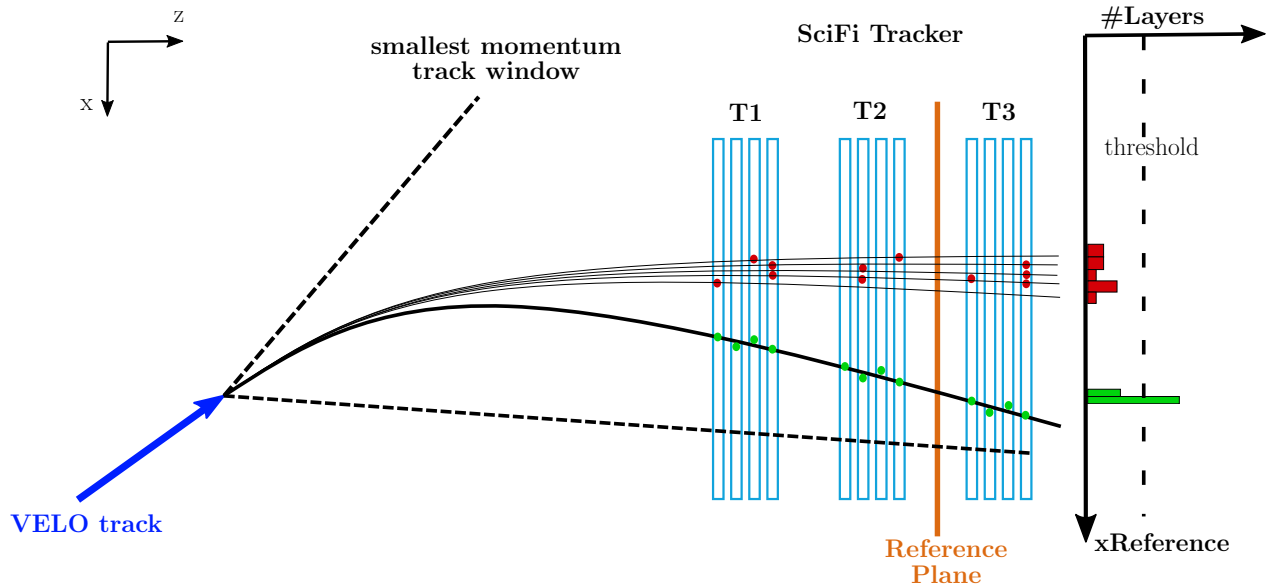


Figure 2: Sketch of the key components of the Forward Tracking. Starting from a VELO track (blue), a smallest-momentum hit search window is calculated (black dashed line). The x positions of hits in the twelve SciFi detector layers (three stations T1, T2 and T3 with four layers each in light blue) are projected to the reference plane (orange) using a simplified track model (not shown here). The rightmost part of the figure shows the histogram counting the number of unique SciFi detector layers depending on the projected x positions. Hits belonging to the VELO track are shown in green, and other hits are in red.

3.4 Event throughput optimisation

Optimisation of tracking algorithms usually deals with a trade-off between event throughput, track reconstruction efficiency and fake track fraction. However, the capabilities of modern CPUs offer new opportunities to improve the throughput of an algorithm without compromising the other two metrics.

The throughput performance of the Forward Tracking is optimised by exploiting data-level parallelism using the vector registers of the CPU. These registers contain the operands for a single CPU instruction that acts on multiple data points at once (SIMD, single instruction multiple data). For this to work efficiently, an appropriate choice of data structures is crucial. Therefore, information about SciFi detector hits is stored in a structure-of-arrays layout, *i.e.* all x positions are arranged successively in memory. This enables fast data loading to SIMD registers as well as fast storing of the computed result. The Forward Tracking mostly applies this form of parallelism in the computationally expensive map-reduce part explained in Sec. 3.3. The mapping of hits' x positions to the reference plane is done for multiple hits at once using single instructions and likewise the threshold scan reducing the data is performed on several histogram bins in parallel. Using the Advanced Vector Extensions 2 (AVX2), SIMD registers hold 256 bits equal to eight single precision floating point or integer numbers that then can be processed in parallel. Adapting the Forward Tracking algorithm to efficiently use AVX2 improves its event throughput by 60% without losses regarding reconstruction efficiency or fake track fraction.

3.5 Reconstruction performance

The Long track reconstruction efficiency achieved by the Forward Tracking is shown in Fig. 3. A high efficiency is particularly important for studies of charm and beauty decays with many final state particles in which track reconstruction inefficiencies have a large impact on the total decay reconstruction efficiency. The integrated efficiency for reconstructing hadrons and muons originating from a B meson is around 90%, while tracks with a momentum higher than 10 GeV reach more than 95% efficiency. Electrons are in general harder to reconstruct with high efficiency as they lose energy predominantly via bremsstrahlung when interacting with the detector material which unpredictably alters their trajectory. This effect is particularly visible in Fig. 3b in the high pseudorapidity region in which particles traverse more material. It is also this region that

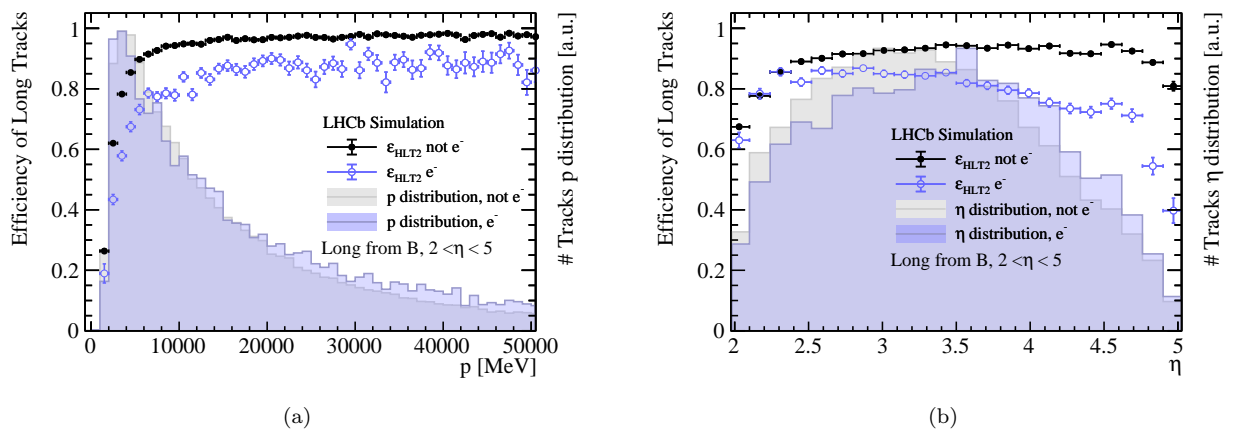


Figure 3: Long track reconstruction efficiency of tracks reconstructed by the Forward Tracking algorithm versus momentum (a) and pseudorapidity (b) for reconstructible electrons and non-electron particles from B decays (empty blue and filled black circles, respectively) within $2 < \eta < 5$ [16]. The underlying histograms show the corresponding distributions of the reconstructible particles. Being reconstructible requires a minimum number of hits in different tracking detectors as described in Ref. [11].

exhibits the highest fake track fraction, as shown in Fig. 4b, partly because of increased multiple scattering and hadronic interactions due to the material, but even more because of the higher SciFi hit density in

the very forward direction leading to more possible random hit combinations. Fig. 4a shows the fake track fraction in dependence of the fake track momentum. The integrated fake track fraction coming from the Forward Tracking amounts to 15%. It is afterwards reduced by applying a Kalman Filter and evaluation of a fake track classifier.

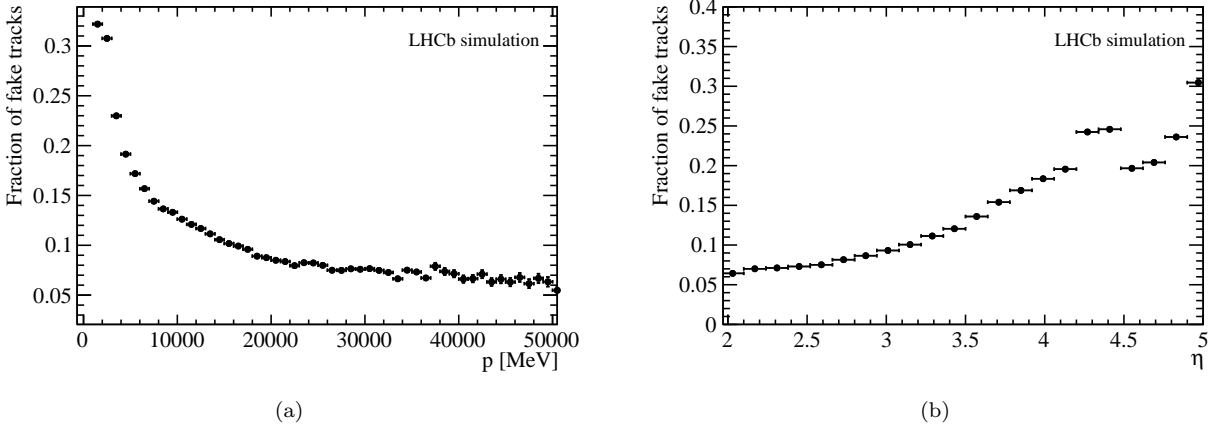


Figure 4: Fake track fraction of Long tracks reconstructed by the Forward Tracking algorithm as a function of fake track momentum (a) and pseudorapidity (b) [16].

4 Conclusion

The Forward Tracking is one of the algorithms finding Long tracks in LHCb's Run 3 HLT2 reconstruction sequence. To avoid the redundancy in Long track finding with the Matching algorithm and hence to improve the throughput of HLT2, the Forward Tracking is only run on residual VELO tracks and SciFi hits. Including the Kalman filter and particle identification algorithms, the HLT2 reconstruction sequence reaches its goal of an event throughput of more than 500 events per second and computing node. The Forward Tracking uses a method similar to the Hough transform to recognise the line patterns left by particles traversing the SciFi detector. With this, it reaches more than 90% reconstruction efficiency for tracks originating from a B meson decay and a fake track fraction of 15%. Hence, the HLT2 reconstruction sequence and the Forward Tracking are well prepared for data taking during Run 3 of the LHC.

ACKNOWLEDGEMENTS

The author would like to acknowledge the support by the LHCb collaboration and in particular thank the LHCb Computing, RTA and Simulation teams for their work that makes developing and benchmarking our algorithms possible.

References

- [1] I. Bediaga *et al.* [LHCb], “Framework TDR for the LHCb Upgrade: Technical Design Report,” CERN-LHCC-2012-007.
- [2] LHCb Collaboration, “LHCb Trigger and Online Upgrade Technical Design Report,” CERN-LHCC-2014-016.
- [3] A. Piucci, “The LHCb Upgrade,” J. Phys. Conf. Ser. **878**, no.1, 012012 (2017) doi:10.1088/1742-6596/878/1/012012
- [4] LHCb Collaboration, “LHCb VELO Upgrade Technical Design Report,” CERN-LHCC-2013-021.
- [5] LHCb Collaboration, “LHCb Tracker Upgrade Technical Design Report,” CERN-LHCC-2014-001.
- [6] R. Aaij *et al.* “Allen: A high level trigger on GPUs for LHCb,” Comput. Softw. Big Sci. **4**, no.1, 7 (2020) doi:10.1007/s41781-020-00039-7 [arXiv:1912.09161 [physics.ins-det]].
- [7] R. Aaij *et al.* [LHCb], “A Comparison of CPU and GPU Implementations for the LHCb Experiment Run 3 Trigger,” Comput. Softw. Big Sci. **6**, no.1, 1 (2022) doi:10.1007/s41781-021-00070-2 [arXiv:2105.04031 [physics.ins-det]].
- [8] T. Boettcher, “Allen in the first days of Run 3,” PROC-CTD2022-33.
- [9] A. Scarabotto, “Tracking on GPU at LHCb’s fully software trigger,” PROC-CTD2022-28.
- [10] LHCb Collaboration, “LHCb Upgrade Computing Model Technical Design Report,” CERN-LHCC-2018-014.
- [11] P. Li, E. Rodrigues and S. Stahl, “Tracking Definitions and Conventions for Run 3 and Beyond,” LHCb-PUB-2021-005.
- [12] A. Hennequin, B. Couturier, V. Gligorov, S. Ponce, R. Quagliani and L. Lacassagne, “A fast and efficient SIMD track reconstruction algorithm for the LHCb Upgrade 1 VELO-PIX detector,” JINST **15**, no.06, P06018 (2020) doi:10.1088/1748-0221/15/06/P06018 [arXiv:1912.09901 [physics.ins-det]].
- [13] R. Quagliani, Y. S. Amhis, P. Billoir and F. Polci, “The Hybrid Seeding algorithm for a scintillating fibre tracker at LHCb upgrade: description and performance,” LHCb-PUB-2017-018.
- [14] R. Aaij *et al.* [LHCb], “Design and performance of the LHCb trigger and full real-time reconstruction in Run 2 of the LHC,” JINST **14**, no.04, P04013 (2019) doi:10.1088/1748-0221/14/04/P04013 [arXiv:1812.10790 [hep-ex]].
- [15] LHCb Collaboration, “Selected HLT2 reconstruction performance for the LHCb upgrade,” LHCb-Figure-2021-003.
- [16] LHCb Collaboration, “HLT2 reconstruction throughput and Forward Tracking performance for Run 3 of LHCb,” LHCb-Figure-2022-005.
- [17] P. V. C. Hough, “Machine Analysis of Bubble Chamber Pictures,” Conf. Proc. C **590914**, 554-558 (1959)
- [18] M. Benayoun and O. Callot, “The forward tracking, an optical model method,” LHCb-2002-008.
- [19] Y. Amhis, O. Callot, M. De Cian and T. Nikodem, “Description and performance studies of the Forward Tracking for a scintillating fibre detector at LHCb,” LHCb-PUB-2014-001.