LHCb-PUB-2017-007
November 6, 2017

Optimization of the muon reconstruction algorithms for LHCb Run 2

R. Aaij[1], J. Albrecht[2], F. Dettori[3], K. Dungs[1,2], H. Lopes[4], D. Martinez Santos[5], J. Prisciandaro[5], B. Sciascia[6], V. Syropoulos[7], S. Stahl[1], R. Vazquez[1].

[1] *European Organization for Nuclear Research (CERN), Geneva, Switzerland*
[2] *Fakultät Physik, Technische Universität Dortmund, Dortmund, Germany*
[3] *Oliver Lodge Laboratory, University of Liverpool, Liverpool, United Kingdom*
[4] *Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil*
[5] *Universidade de Santiago de Compostela, Santiago de Compostela, Spain*
[6] *Laboratori Nazionali dell'INFN di Frascati, Frascati, Italy*
[7] *Nikhef National Institute for Subatomic Physics and VU University Amsterdam, Amsterdam, The Netherlands*

Abstract

The muon identification algorithm in the LHCb HLT software trigger and offline reconstruction has been revisited in view of the LHC Run 2. This software has undergone a significant refactorisation, resulting in a modularized common code base between the HLT and offline event processing. Because of the latter, the muon identification is now identical in HLT and offline. The HLT1 algorithm sequence has been updated given the new rate and timing constraints. Also, information from the TT subdetector is used in order to reduce ghost tracks and optimize for low p_{T} muons. The current software is presented here together with performance studies showing improved efficiencies and reduced timing.

Contents

1 Introduction

A large fraction of the LHCb experiment physics programme is based on the identification of muons in the final state: for example the measurement of ϕ_s through $B_s^0 \to J/\psi(\to \mu^+\mu^-)\phi$ decays, the study of CP violation through semileptonic decays, and the branching fraction measurement of several rare decays such as $B^0 \to K^{*0}\mu^+\mu^-$ and $B_s^0 \to \mu^+\mu^-$. It is therefore of major importance to maintain and improve the efficiency and purity of identified muons in the Run 2 data-taking of LHCb.

During the shutdown period between Run 1 and Run 2, effort has been made to re-optimize the muon identification algorithm [1, 2] (Muon-ID), with the main goal of obtaining a unified code, that can be run both online in the LHCb software trigger [3] (High Level Trigger, HLT) and in the offline reconstruction. An increase in performance both in CPU and memory usage has also been achieved, as well as in muon identification efficiency. Moreover, the readability and ease of maintenance of the algorithm have been improved.

Specifically for the first stage of the software trigger (HLT1), due to looser timing constrains in Run 2, all the tracks above 500 MeV/c are passed directly to the forward tracking and track fitting algorithms without any preselection. This was not the case in Run 1. In addition, new possibilities for muons below 500 MeV/c were introduced.

The above mentioned changes in the Muon-ID and in the HLT1 are presented in sections 3 and 4, respectively.

2 Muon identification

The identification of muons in LHCb is mostly based on the Muon detector, which is composed of five detecting stations interleaved by four filtering iron walls. Only muons are able to penetrate the calorimeters and muon filters and thus leave a signal on the dedicated muon chambers. A scheme of the Muon detector is shown in Fig. 1, and a detailed description can be found in Ref. [4].

In the following we describe which variables are constructed exploiting the information coming only from the Muon detector. Additional discrimination with respect to other particles is achieved through the use of combined information using the Calorimeters and RICH detectors, the details of which are beyond the scope of this note; the reader can refer to Ref. [5] Chapter 4, and references therein, for further details.

2.1 Boolean Muon-ID variables

Different identification variables can be constructed exploiting the information of the Muon detector [6]. The first identification variable is a boolean decision, called IsMuon, obtained from the extrapolation of a long or a downstream track[1] through the muon

[1]Particle tracks in LHCb can be reconstructed as *long* if they have hits at least in the Vertex Locator, VELO [7], and in the downstream tracking stations, *downstream* if they are reconstructed from hits

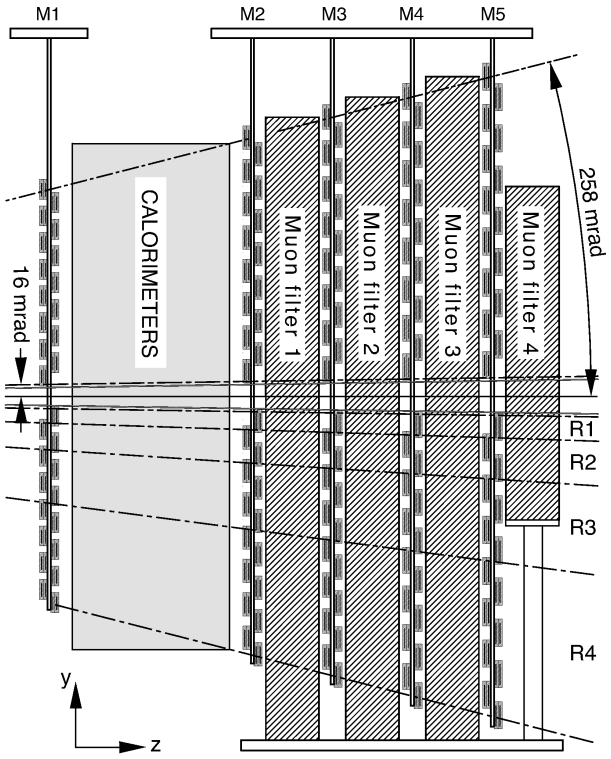

Figure 1: Side view of the LHCb Muon detector.

stations, making a statement about whether a track is consistent with a muon hypothesis. The extrapolation yields the expected track coordinates in the muon stations M2, M3, M4, and M5. Only extrapolations within the acceptance of the muon stations are considered. For each of the stations, a search for hits within an elliptic, momentum dependent, field of interest (FoI) around the extrapolated track is performed. Three parameters for each dimension $\rho_{\{x,y\}\times\{1,2,3\}}$ define a FoI,

$$\mathrm{FoI}_a(p) = \rho_{a,1} + \rho_{a,2} \cdot \exp\left(\frac{-\rho_{a,3} \cdot p}{\mathrm{GeV}/c}\right) , \tag{1}$$

with $a \in \{x, y\}$. A hit (h_x, h_y) with corresponding pad dimensions $\mathrm{pad}_{\{x,y\}}$ is considered to be within the field of interest around an extrapolation (e_x, e_y) if

$$\|h_x - e_x\| < \mathrm{pad}_x \cdot \mathrm{FoI}_x \quad \text{and} \quad \|h_y - e_y\| < \mathrm{pad}_y \cdot \mathrm{FoI}_y \, . \tag{2}$$

Depending on the momentum, the track is given the attribute `IsMuon` if a minimum number of stations, always including M3, are found to have at least one hit within the FoI. Table 1 gives an overview of the required stations depending on the momentum of

everywhere but in the VELO, *Velo-TT* or *Upstream* if they are reconstructed from hits in the VELO and in the TT [8] and simply *VELO* if they are reconstructed only from hits in this subdetector.

the track. The minimum momentum of $p > 3$ GeV/c is needed by a muon to reach the M3 station without being absorbed in the upstream material; this threshold is therefore required as minimum momentum to be classified as muon.

Table 1: Required stations with hits within FoI for `IsMuon` and `IsMuonLoose` with respect to track momentum [1]. `IsMuonTight` has the same requirements as `IsMuon` but exploiting only *crossed* hits (see text for details).

p [GeV/c]	Required stations	
	`IsMuon`	`IsMuonLoose`
$p < 3$	*Always false*	*Always false*
$p < 6$	M2 & M3	At least two of M2–M4
$6 < p < 10$	M2 & M3 & (M4 ‖ M5)	At least three of M2–M5
$p > 10$	M2 & M3 & M4 & M5	At least three of M2–M5

The readout of the muon detector is given by the OR of horizontal and vertical *physical pads*, and the crossing of the two defines a *logical pad* whose dimensions give the *x, y* pad size associated to the hit. If there is no simultaneous readout, the *x, y* pad size is given by the whole physical dimensions of the hitted *physical pad*. This design guarantees high efficiency for the typical low occupancies of the Muon system while reducing significantly the number of readout channels. In the following we will refer to the single hits given by the *physical pads* as *uncrossed hits*, and to the *logical pads* as *crossed hits* (see Fig. 2).

Similar to `IsMuon`, other two boolean variables are constructed testing the hypothesis of the track being consistent with coming from a muon: `IsMuonLoose` and `IsMuonTight`; as the name says, the requirements are respectively looser and tighter with respect to `IsMuon`. `IsMuonLoose` requires a fewer amount of hits with respect to `IsMuon` (see Table 1), while `IsMuonTight` requires the same amount of hits but using only crossed hits.

2.2 Additional variables

The average squared distance in units of pad size between the track extrapolation into the muon stations and the corresponding closest hit for each station yields a good separation between muons and non-muons [1]. This distance is defined as

$$D^2 = \frac{1}{N}\sum_{i=0}^{N}\left(\left(\frac{x_{\text{closest},i} - x_{\text{track},i}}{\text{pad}_{x,i}}\right)^2 + \left(\frac{y_{\text{closest},i} - y_{\text{track},i}}{\text{pad}_{y,i}}\right)^2\right), \tag{3}$$

where $\{x, y\}_{\text{closest},i}$ denotes the $\{x, y\}$ coordinate of the closest hit in station i. Likewise, $\{x, y\}_{\text{track},i}$ is the $\{x, y\}$ coordinate of the extrapolated track in station i. The value of $\text{pad}_{\{x,y\},i}$ is the $\{x, y\}$ dimension of the pad associated to the hit. Due to multiple scattering the D^2 distance depends on the momentum of the muon candidate, the traversed material, and the *logical pad* size, chosen such that its contribution to the transverse momentum

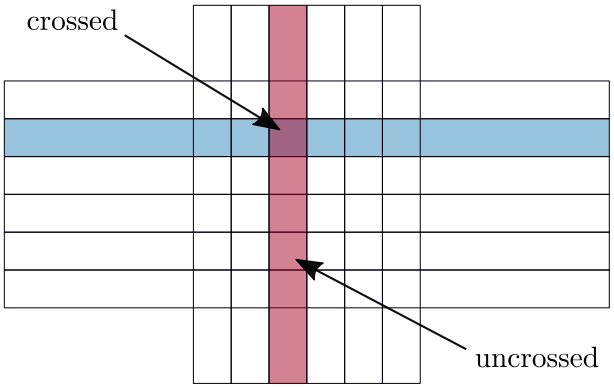

Figure 2: Visualisation of the difference between crossed and uncrossed hits. A hit in a muon station is considered a crossed hit if it is registered both by a horizontal and a corresponding vertical strip. If a hit is only seen by one of them, it is considered uncrossed. Uncrossed hits have, by construction, much larger pad sizes in one of their dimensions.

resolution is about equal to the multiple scattering contribution. Each muon detector station is subdivided in four regions with fixed dimensions of the *logical pads*, which scales a factor of two from one region to the next. Due to those dependencies, the hypothesis tests are performed in bins of detector region and momentum. For a given bin, two tests are performed that yield $P(\mu)$, the probability of the candidate being a muon, and $P(\text{not}\,\mu)$, the probability of the candidate not being a muon. From those quantities the delta log likelihood, DLL, is calculated as

$$\text{DLL} = \log\left(\frac{P(\mu)}{P(\text{not}\,\mu)}\right) = \log\left(P(\mu)\right) - \log\left(P(\text{not}\,\mu)\right)\,. \tag{4}$$

Details on the procedure and the binning can be found in Ref. [1].

It should be noted here, that due to the two-dimensional binning, many calibration constants are needed. D^2 and the DLL are saved in the muon track object. In the offline reconstruction, the quantities $\log(P(\mu))$ and $\log(P(\text{not}\,\mu))$ are stored in the muon PID object. Additionally, a track fit is performed on the extrapolation using only the closest hits. The resulting χ^2/ndof is also stored in the muon track object.

The variable defined in Eq. 4 is used in the full LHCb particle identification procedure to be combined with the information from the other detectors to evaluate the combined DLL variable. Furthermore, $\log(P(\mu))$ and $\log(P(\text{not}\,\mu))$ are used as input to a Neural Network based particle identification called `ProbNN`, which we will not discuss further. Details on the combined particle identification can be found in Ref. [5].

2.2.1 NShared

Each muon track has also a property called `nShared`. It is an unsigned integer that helps distinguish between actual tracks and potential ghost tracks. For each hit within the FoI of a given extrapolation into the muon chambers, the algorithm checks whether any of the

other long or downstream tracks passing `IsMuonLoose` has already used that hit. If so, the value of `nShared` for the track with the larger sum of squared distances between the extrapolation and the corresponding closest hit is incremented. This principle is illustrated in Fig. 3. It follows that a track with a value of `nShared` equal to zero is likely to be an isolated while one with a high value of `nShared` is likely to be accompanied by other tracks. Tracks sharing numerous hits are rarely two muons very close to each other, and most likely one true muon and a second track associated to random and genuine hits in the muon station by mistake. The `nShared` variable has thus discriminating power against background from noise and other effects, reducing the particle misidentification.

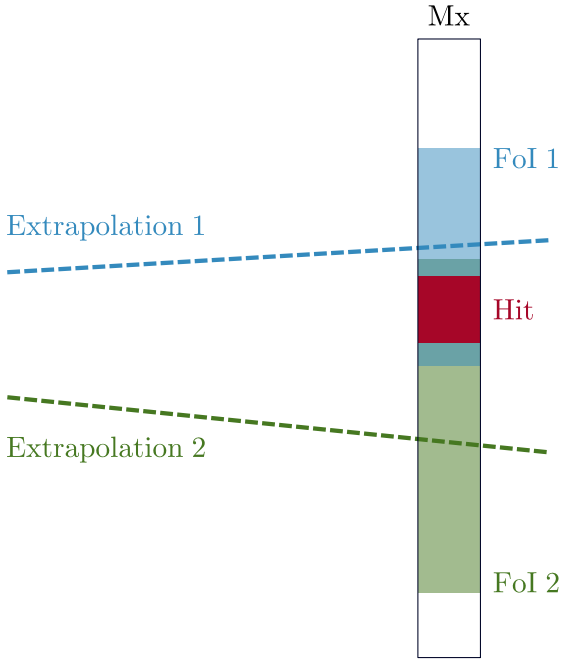

Figure 3: Example scenario for nShared. Two extrapolations into the muon station Mx (where $x = 2, \ldots, 5$) share the same hit because it is contained in both fields of interest around the extrapolated track. For the track which has a larger sum of squared distances between the extrapolations and the corresponding closest hits, the value of `nShared` is incremented. Note that this sketch is not to scale.

3 Muon-ID software re-optimisation

The muon identification code runs in the online reconstruction for the first and second level of the LHCb software trigger (HLT1 and HLT2), and in the offline reconstruction.

During Run 1 two separate code bases for the online and the offline reconstruction of muons existed. The offline muon identification was too time consuming to be run online in the trigger. At the trigger level only the `IsMuon` variable was evaluated and with a more time-optimized code, while offline the full chain of variables was computed.

For the Run 2 data-taking a larger computing farm has been deployed for the trigger. The upgrade corresponds to a 90% increase of the available computing power for Run 2 with respect to Run 1. In addition, the two software trigger levels, which in Run 1 where run sequentially, have been made independent from each other. These changes allow for larger time budgets giving the possibility to improve the Muon-ID software at the trigger level and to unify it with the offline one.

A unique Muon-ID code removes possible discrepancies between online and offline performance, and allows an easier code maintainability. Additionally, this provides also for the offline usage a software optimized for CPU time consumption reducing the needed resources.

3.1 Common muon identification

Based on a previous work [3], a code was written (`CommonMuonTool`) that can been used both at the trigger (online) level and at the offline reconstruction level. Figure 4 visualizes the intended use of the tool and additional tools for offline-only properties. This removes duplication and thus greatly improves maintainability. Additionally, an effort was made to split the tool into small functions with clearly defined duties.

The `CommonMuonTool` is now used both at the HLT2 and offline reconstruction level to calculate the aforementioned variables, and is used at the HLT1 level to calculate `IsMuon`.

It was verified that for the variables which are unchanged with respect to Run 1, the new tools produce identical results. The CPU time consumption in the online and offline reconstruction has been found to have reduced by more than a factor of three. More details can be found in subsection 5.5. The introduced tool is currently in use in the Run 2 data-taking of LHCb.

CommonMuonTool - Code structure

The tool makes use of the following constants: a scale factor (of 1.2) which multiply the x and y dimension of the FoI given in Eq. 1, the value of the pre-selection momentum cut, the two edges of the momentum bins (see Table 1), the z-coordinates of the muon stations, the dimensions of the muon stations, and three FoI parameters for both x and y-dimension. Additionally, it makes use of two tools called CommonMuonHitManger and DeMuonDetector, which extract the hit information from the muon raw detector data.

The `CommonMuonTool` offers a dedicated method for each logical step in the `IsMuon` algorithm as depicted in Fig. 5. In addition, functions offering functionality to calculate `IsMuonLoose` and `IsMuonTight` are implemented. What follows is an overview of the methods which are used both in the offline reconstruction by the `MuonIDAlgLite` code and in the HLT1 trigger by the `IsMuonTool` code. More details on the algorithm are given below.

- The `initialize` method sets up the tool. It loads additional tools and fetches the constants from the database.

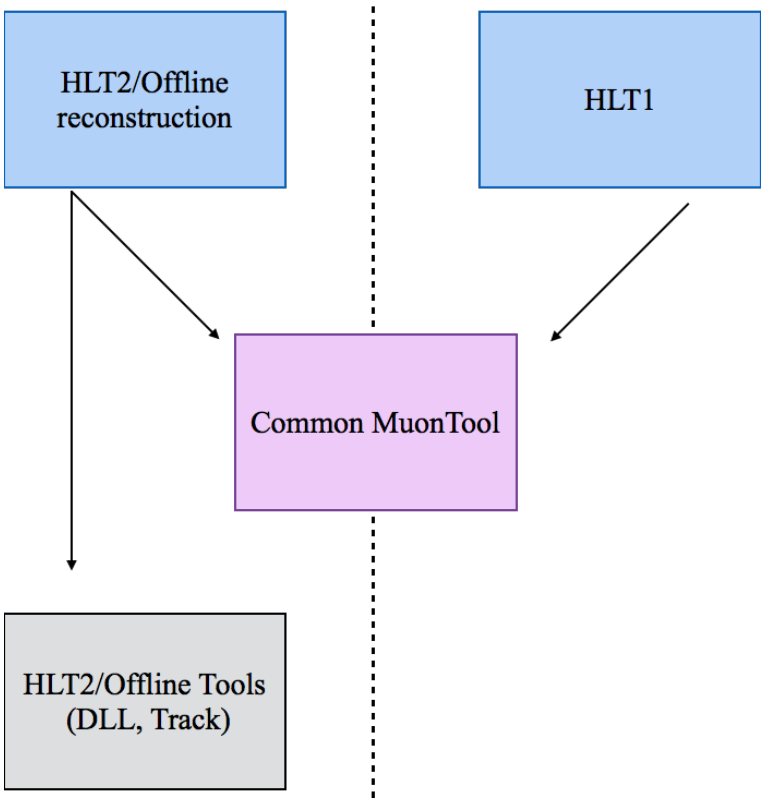

Figure 4: Use of the `CommonMuonTool` from online and offline code.

- `preSelection` takes a track object and returns whether the track passes the pre-selection criteria. In this case it just checks if the track momentum is larger than the cut value ($p > 3$ GeV/c).
- `extrapolateTrack` takes a track object and extrapolates it through the muon stations. It returns a point (x, y) for each station (at a fixed z) except M1 which is not used.
- `inAcceptance` uses the output of `extrapolateTrack` in order to check whether the coordinates of the extrapolated hits are within the acceptance of the muon stations.
- `hitsAndOccupancies` takes both a track and a `MuonTrackExtrapolation` container as input and returns two containers: the first holds the hits that are found in the

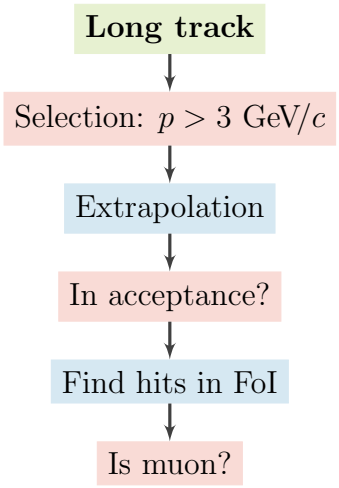

Figure 5: Simplified flow of the `IsMuon` sequence. A long track with $p > 3$ GeV/c is extrapolated into the muon stations. If the extrapolation is within the acceptance, hits within the FoI are searched for. If enough stations have hits, the attribute `IsMuon` is assigned.

muon stations within the FoI around the extrapolations, the second container holds the total number of hits in each station, which is called the occupancy of the station. The latter is also used to check whether a station has hits within a FoI.

- `extractCrossed` takes the hits in the muon stations as input and selects only those that are crossed. Additionally, it also calculates the new occupancies considering only the crossed hits.
- `isMuon` uses occupancies and the track momentum to classify it according to Table 1. If it obtains a container of occupancies from crossed hits, it calculates `IsMuonTight` by definition.
- `isMuonLoose` also takes a container of occupancies and a track momentum and calculates `IsMuonLoose` according to Table 1.
- `foi` for a given station, region, and momentum returns the edge of the field of interest.

Usage in HLT1 In HLT1, muon lines make use of the `IsMuonTool`, which has been adjusted in order to use the functionalities offered by the `CommonMuonTool`. Like every tool that is used by a trigger line, it exploits a method, `tracksFromTrack`, which takes the current HLT1 reconstructed track (trigger track in the following) as input and writes to an output container if the `IsMuon` criterion is met. It uses the functions `preSelection`, `extrapolateTrack`, `inAcceptance`, `hitsAndOccupancies`, and `isMuon` in sequence as depicted in Fig. 5.

Usage in the HLT2 and in the offline reconstruction. At the HLT2 and offline reconstruction levels the algorithm receives a collection of reconstructed tracks as input, and the calculation of `IsMuon` is embedded inside a loop over the tracks. For the purpose of backwards compatibility and as fall-back solution, the former `MuonIDAlg` is kept and maintained as working scenario. In addition a new algorithm called `MuonIDAlgLite` is added and is now used by default. A specific effort has been made to optimize the procedure in terms of performance. Furthermore the algorithm accepts both long and downstream tracks.

HLT2 and offline tools Two additional tools are introduced in order to provide additional information that is not used in HLT1. Those are called `DLLMuonTool` and `MakeMuonTool`. The `DLLMuonTool` is responsible for calculating the delta log likelihood (DLL) of the muon hypothesis. It loads all the parameters for the hypothesis tests in different bins in momentum and region as described in Ref. [1] in order to calculate $P(\mu)$ and $P(\text{not}\,\mu)$. Two different implementations can be used via a flag: `calcMuonLL_tanhist` and `calcMuonLL_tanhist_landau`. In the first case $P(\mu)$ and $P(\text{not}\,\mu)$ are extracted using the reference histograms for signal and background, without analytical description. In the second case $P(\mu)$ is computed as in the `calcMuonLL_tanhist` implementation, while a Landau description is used for $P(\text{not}\,\mu)$. The default method is the second one [1].

Both return the likelihood for the muon hypothesis $P(\mu)$ as well as for the background hypothesis $P(\text{not}\,\mu)$. It also contains the squared distance of the muon track D^2. The DLL is then calculated according to Eq. 4. Additionally, the tool allows to calculate the `nShared` variable via the `calcNShared` method. This variable relies on relationships between the tracks in an event. The `MakeMuonTool` is intended to create a muon track once all the necessary information is there, through a function called `makeMuonTrack`. If a corresponding flag is set, the tool also performs a track fit in order to obtain the χ^2 of the track.

4 HLT1 muon identification

The role of HLT1 is to select interesting events based on reduced event information. The software is divided in lines selecting different physics signatures. The HLT1 muon lines are mainly organized as single muon and dimuon; both exploiting the same muon identification procedure which will be described in the following. The algorithm sequence of the HLT1 muon lines during Run 1 was tuned in order to comply with the output rate limitations given by the existing computing infrastructure, namely the Event Filter Farm (EFF). Since then the EFF has undergone a significant upgrade resulting in an increased processing power along with increased storage capabilities. In view of this infrastructure upgrade, as well as the Run 2 LHC running conditions, the total output rate of the entire trigger system was increased from 5 kHz to 12.5 kHz. The impact on the HLT1 is that its output rate is roughly doubled and additional processing time is available. This offered chances for optimization of the HLT1 muon identification, for revisiting the existing muon trigger

lines and adding new ones. The change in the lines will not be described in the present document, where only the differences due to the Muon-ID will be underlined.

4.1 HLT1 muon algorithm sequence in Run 1

Muon identification during Run 1 started directly from VELO tracks which were identified as muons with a simple extrapolation through the `MatchVeloMuon` algorithm [3], which will be detailed in subsection 4.3, before being upgraded to long tracks by the `LooseForward` tracking algorithm. After this, the `IsMuon` algorithm was applied. In addition, quality cuts are applied at each algorithm step, such as requirements on the number of VELO hits of the track or the fitted track χ^2_{dof}, as well as momentum cuts applied after `LooseForward`. The minimum required momentum and transverse momentum were 6 GeV/c and 0.5 GeV/c, respectively. A diagram of the HLT1 muon trigger lines sequence during Run 1 is shown in Fig. 7, while more details can be found in Ref. [3].

The efficiency of the muon lines as a function of the track p_{T} during Run 1 is shown in Fig. 6, where a large efficiency loss in the dimuon lines is observed. The detailed inefficiency breakdown of the Run 1 HLT1 muon lines is shown in Fig. 11 and Table 2, where it can be seen that the main source of efficiency loss is due to the quality and momentum cuts. In addition, the `MatchVeloMuon` algorithm reduces the efficiency by roughly 4%.

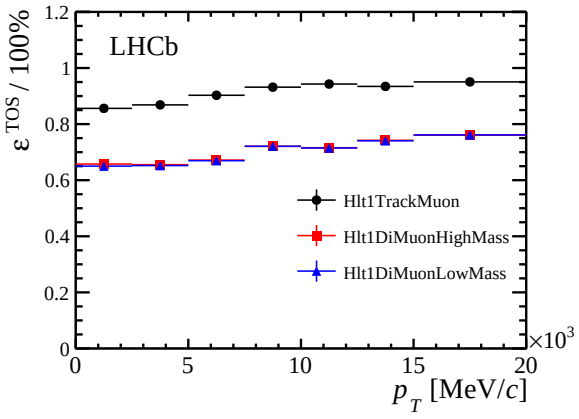

Figure 6: Single and dimuon HLT1 lines efficiency in Run 1 [9].

4.2 HLT1 muon algorithm sequence in Run 2

An improved sequence for the muon HLT1 lines has been defined for Run 2. In addition to the sequence update, described hereafter, the `MatchVeloMuon` algorithm for softer muons is also upgraded (see Sect. 4.3).

During the course of this work it became apparent that the HLT1 `IsMuon` calculation in Run 1 did not use the correct pad sizes for uncrossed hits. The tool used an internal

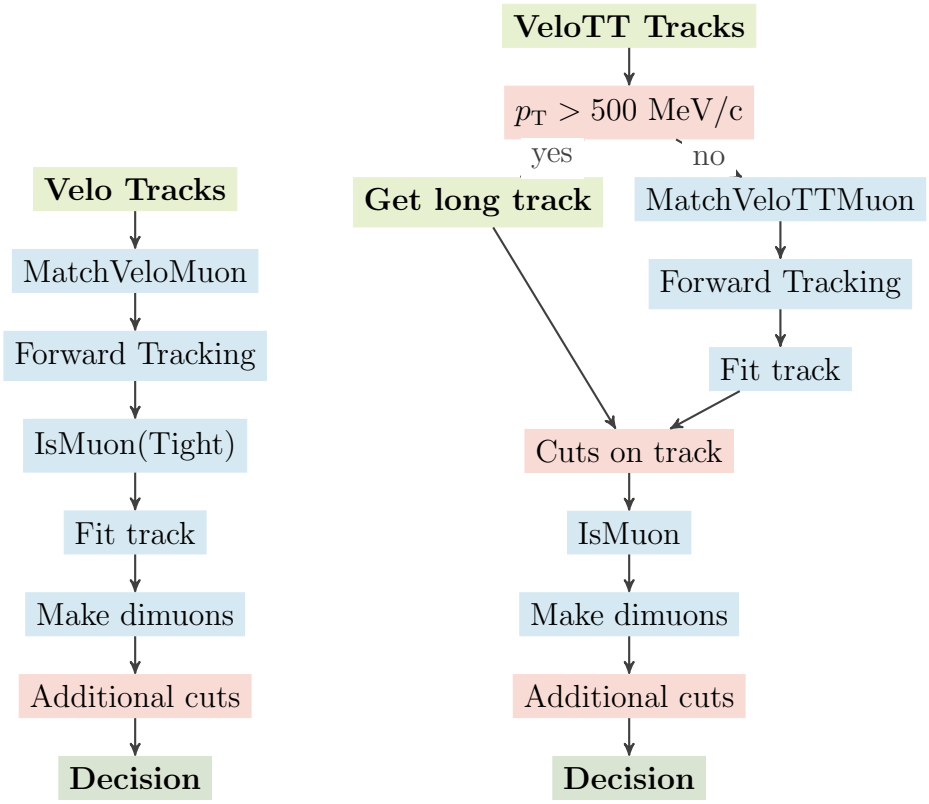

Figure 7: Run 1 (left) and Run 2 (right) HLT1 algorithms compared.

definition of the pad sizes for each station and region instead of the information from the hit object itself. The use of these incorrect pad sizes resulted in uncrossed hits being wrongly rejected, transforming *de facto* the `IsMuon` request into the `IsMuonTight` one. This results in a lower efficiency. However, since the trigger efficiency is always measured directly on data, public LHCb results are not affected. The updated HLT1 algorithm sequence for Run 2 is shown in Fig. 7 (right). In the Run 2 sequence, all the VELO tracks with p_{T} larger than 500 MeV/c are directly passed to the forward tracking algorithm, and therefore upgraded to long tracks. Soft quality and kinematic cuts are then applied, followed by the `IsMuon` requirement. The selected muon tracks are finally combined into dimuon pairs following the same strategy as in Run 1. This strategy assures a much higher efficiency than in Run 1, and guarantees the same reconstruction as the offline one.

Tracks with transverse momentum smaller than 500 MeV/c cannot be immediately passed to the forward tracking because of timing consumption. A preliminary muon track reconstruction, detailed in Sec. 4.3, is therefore applied, in order to upgrade only the muon candidates.

4.3 Matching VELO and VELO-TT tracks to muon hits

In Run 1 all VELO tracks in HLT1 were filtered by the `MatchVeloMuon` tool as documented in Refs. [3] and [10]. The tool matched a VELO track to hits in the muon stations before running the `LooseForward` tracking algorithm. In this way, only muon candidates were upgraded to long tracks (see Fig. 7(left)), and further processed by the HLT1 muon lines.

The `MatchVeloMuon` algorithm takes a VELO track as input and look first for hits inside a search window in the M3 muon station. The choice of M3 as first station follows from the fact that hits in M3 are always required by `IsMuon`, and from its lower occupancy compared to the M2 station.

The vertical centre of the search window is defined by a straight line extrapolation of the VELO track to M3, and its vertical size is twice the height of a muon pad in the outermost region of M3. Since no momentum or charge information are available at this stage of the reconstruction, the horizontal size of the search window (FoI) is defined based on the maximum deflection that the magnetic field would give to a track passing trough the LHCb magnet. In order to compute the maximum deflection, a track momentum of 6 GeV/c is assumed, and both the positive and negative charge hypotheses are taken in account.

Once the M3 hits inside the FoI are found, an estimate of the track momentum is computed for each hit using the kick method [11]. Based on this momentum estimate, the candidate track is extrapolated to the stations M2, M4 and M5, searching for hits in a window of 20×20 cm.

At least 3 hits in the muon stations M2-M5 are required by the algorithm in order to accept the initial VELO track.

As a last step, a linear χ^2 fit in the horizontal plane is performed, and a quality cut is set at $\chi^2/\mathrm{DoF} < 25$ to remove fake muon candidates. Fake muon tracks are tracks that originate from random combinations of hits in the muon stations.

The surviving tracks are upgraded to long track with the forward tracking algorithm and passed to the HLT1 line algorithm, where further selection criteria and the standard `IsMuon` algorithm are applied.

As described in Ref. [12] in the Run 2 data-taking all the tracks are upgraded to VELO-TT segments, exploiting information from the TT tracking stations which, being closer to the magnet, guarantees a first momentum estimation with a resolution of 15%. An upgraded version of the `MatchVeloMuon` algorithm, named `MatchVeloTTMuon`, is used to identify muon candidates with $p_{\mathrm{T}} < 500\,\mathrm{MeV}/c$. The new `MatchVeloTTMuon` algorithm is improved in the following key points:

- First, the charge information of the `VeloTTCandidates` is used, allowing to halve the FoI size. In addition, the number of required hits in the muon stations is increased to 4. These two changes reduce the number of potential muon tracks that are created during the seeding procedure in M3. As a result, computation time is saved and the number of fake muon tracks is suppressed.

- Second, the minimum momentum to calculate the maximum deflection is reduced

to 3 GeV, allowing for softer muons to be processed by the next algorithms of the muon triggers. In addition, the empirical parametrization of the magnet focal plane z coordinate, already used in the old `MatchVeloMuon`, is updated.

- Third, the uncertainties on the focal plane intersection are empirically estimated using simulated data. Specifically, since the true muon track is known, it is possible to tune these uncertainties such that they yield the same momentum estimation as the full computation of the muon trajectory would do. Note that a momentum estimate is available as soon as a seed hit in M3 is found, using the kick method.

- Lastly, the possibility for including the vertical plane in the χ^2 fit performed by the `MatchVeloTTMuon` algorithm is added. In more detail, the χ^2 fit is a straight line fit performed to all the possible muon candidate tracks that originate from the seed hits of a given initial VELO track.

An important caveat arises from the fact that there can be tracks that are labelled as `VeloTTCandidates` but they actually do not have TT hits. These tracks are normal high p_{T} tracks that pass through the inner hole of the TT stations where there is no active detector material. Thus, they are just copied to the list of output tracks when the `VeloTTCandidates` are reconstructed. `MatchVeloTTMuon` does not discard this type of tracks either but deals with them properly by invoking the old `MatchVeloMuon` algorithm.

5 Performances

5.1 Efficiencies

In order to assess the efficiencies of the Muon-ID, two different kinds of studies are performed. To get a very fine-grained insight into the structure of the efficiencies in Run 1, the trigger sequence was run on a sample of simulated $B^0 \to K^{*0}\mu^+\mu^-$ signal events. For a second, comparative study, different versions of the trigger sequence are run independently on a sample of simulated $B^+ \to J/\psi K^+$ decays. Both studies use only events that have already been processed by the whole offline reconstruction chain and thus all measured efficiencies are referred to offline selected events, as is typically done in LHCb analyses.

5.2 Analysis of Run 1 inefficiencies

A sample of simulated $B^0 \to K^{*0}\mu^+\mu^-$ signal events was used in order to estimate Run 1 efficiencies. All the trigger lines mentioned in the following are described in Ref. [13]. Signal candidates are selected by an inclusive offline selection described in Ref. [14]. For each event that passes the hardware trigger lines `L0Muon` or `L0DiMuon`, the trigger tracks associated with the `Hlt1TrackMuon` line were considered as well as B^0 candidates that were selected by the offline selection. The B^0 candidates were required to have $p_T > 0.5$ GeV/c. In Fig. 8, the distributions of mass, momentum, and transverse momentum of the candidates after this selection are shown. The sample contains 19 249 candidates. The daughter

muons are then matched to the trigger tracks by requiring at least 70 % overlapping track hits. The cut at 70 % is motivated in Fig. 9.

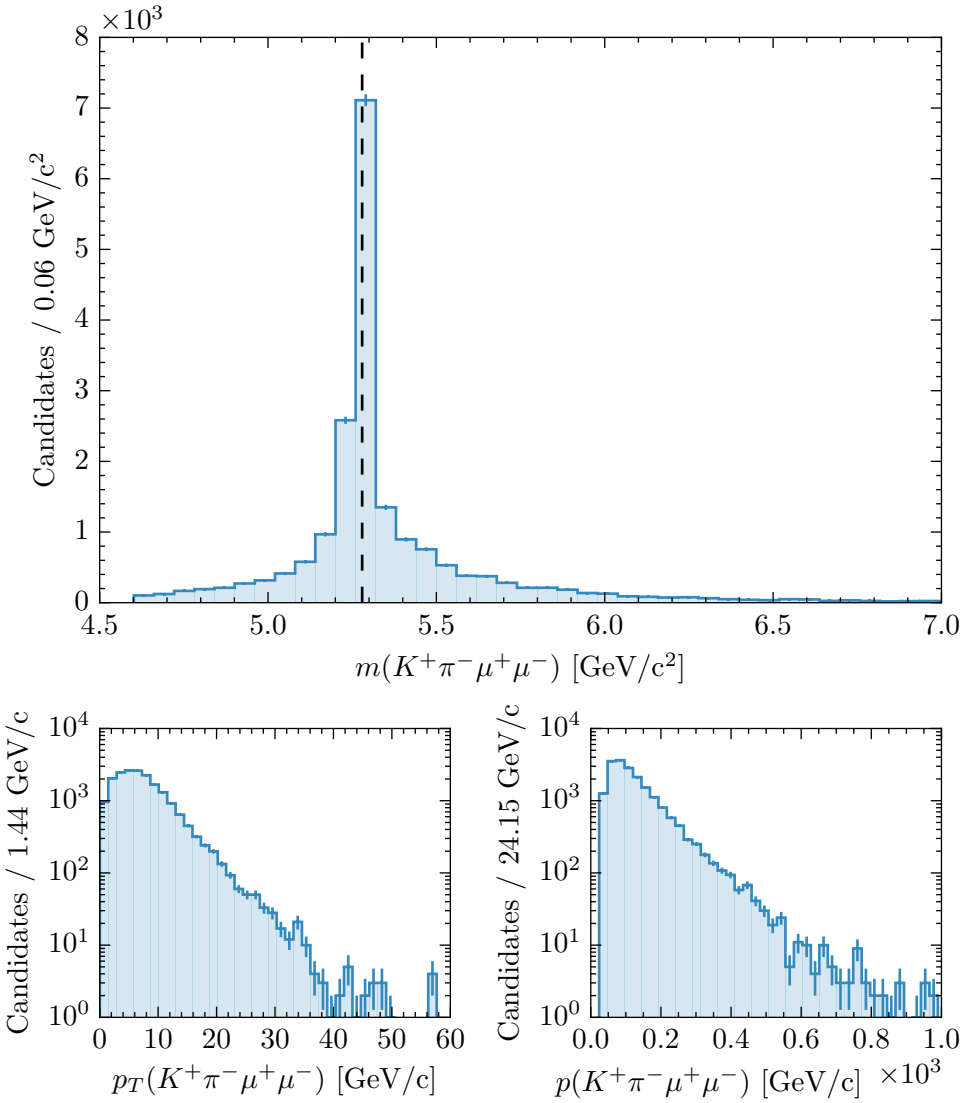

Figure 8: Mass (top), transverse momentum (bottom left), and momentum (bottom right) of the reconstructed $B^0 \to K^{*0}\mu^+\mu^-$ candidates in the simulated sample used to evaluate the efficiency of `Hlt1TrackMuon` in Run 1. The nominal B^0 mass from Ref. [15] is marked by the dashed line at $(5279.58 \pm 0.17)\,\mathrm{MeV/c^2}$. The tails in the mass distribution are a result of the loose selection and the K-π misidentification. Since simulated signal candidates are used, this does not influence the validity of the study.

The efficiency is calculated for several scenarios. Successively, more and more parts of the `Hlt1TrackMuon` line are left out so that the individual contributions to the total inefficiency of the trigger line can be computed. The results of this study can be found

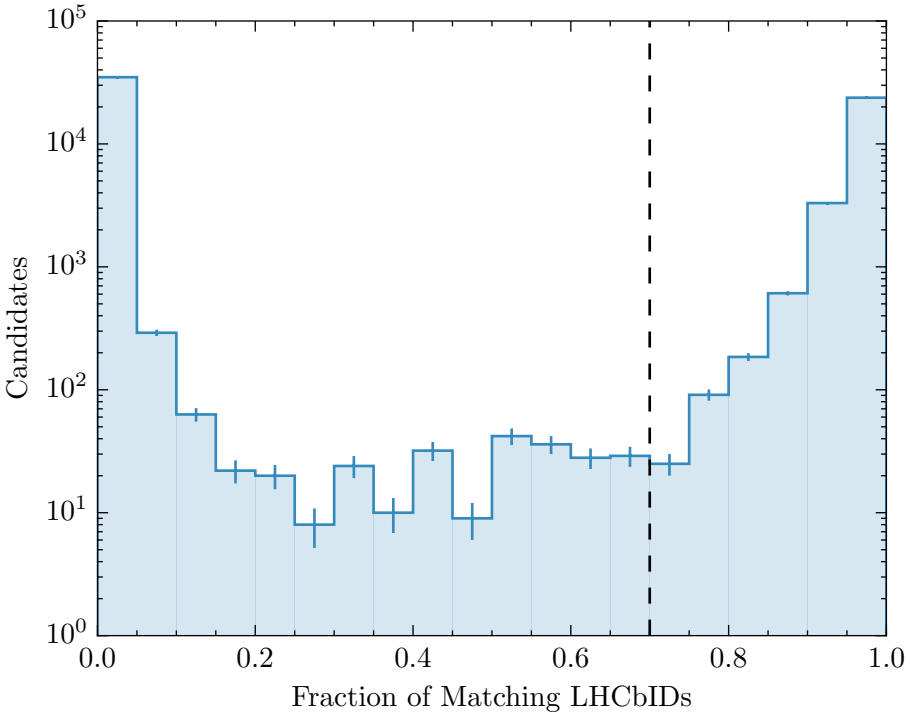

Figure 9: Percentage of matching track hits between trigger tracks and signal muons. On the left are tracks where no matching has been found while on the right a matching becomes more and more apparent. The threshold of 70 % is marked by the dashed line. Please note the logarithmic scale. Effectively, only the edges of the shown range contain a significant amount of events. The drop from the first to the second bin is much stronger than between the last bins.

in Table 2 and Fig. 10. Figure 11 contains a visual representation of the individual contributions to the total inefficiency. It can be seen that cuts are responsible for a large portion of the inefficiency. This is a deliberate choice motivated by timing constraints that can easily be adjusted. More interestingly, (9.4 ± 0.4) % of the inefficiency originates from the simplified reconstruction software. The observed gain in efficiency motivates the adjustment of the trigger strategy for Run 2 as described in subsection 4.2. Checks on the total time consumption of the HLT are routinely performed and the increases found are considered small compared to the gains in signal efficiency.

Table 2: Efficiencies for (parts of) `Hlt1TrackMuon` on single muons from the simulated sample. Successively, parts of the trigger line are removed in order to be able to calculate individual contributions to the inefficiency. The total inefficiency is $(26.33 \pm 0.27)\,\%$, of which $(9.4 \pm 0.4)\,\%$ originate from the simplified reconstruction. The individual contributions are obtained as the pairwise differences $\Delta\varepsilon$. The last two points were added in order to understand the remaining inefficiency after `IsMuon` was removed. The negative contribution from using the offline cuts in the track fit is within the statistical uncertainty. The remaining $(0.79 \pm 0.07)\,\%$ are unexplained.

Scenario (shorthand)	ε(TOS) [%]	$\Delta\varepsilon$ [%]
Original 2012 configuration. (2012)	73.67 ± 0.27	
Use both iterations in `FastVelo`. (FullVelo)	76.09 ± 0.27	2.4 ± 0.4
Remove all cuts in the streamer. (- Cuts)	93.03 ± 0.16	16.94 ± 0.31
Exclude the `MatchVeloMuon` algorithm. (- MatchVeloMuon)	96.85 ± 0.11	3.82 ± 0.20
Swap `IsMuon` and the track fit. (Swapped)	97.47 ± 0.10	0.63 ± 0.15
Exclude the `IsMuon` algorithm. (- IsMuon)	97.92 ± 0.09	0.45 ± 0.14
Use the same cuts as offline in the forward tracking. (+ Offline Cuts in Fwd)	97.87 ± 0.09	-0.06 ± 0.13
Require tracks reconstructed by forward tracking. (+ Fwd tracks only)	99.21 ± 0.07	1.34 ± 0.20

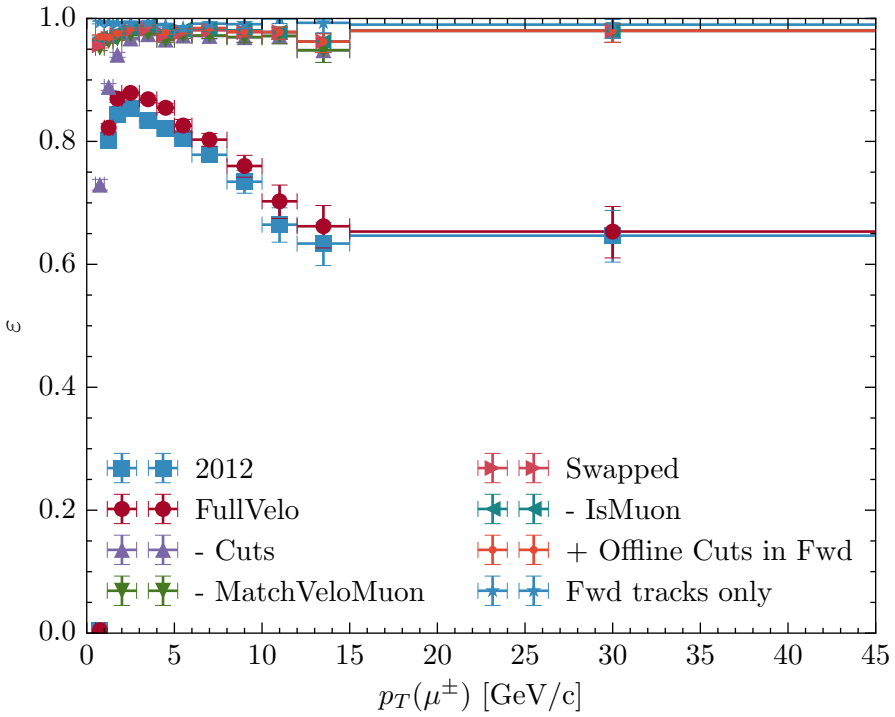

Figure 10: Efficiencies in bins of transverse momentum for `Hlt1TrackMuon` in the several stages of the study. It can be seen that both the cuts and the `MatchVeloMuon` algorithm introduce a p_T dependence.

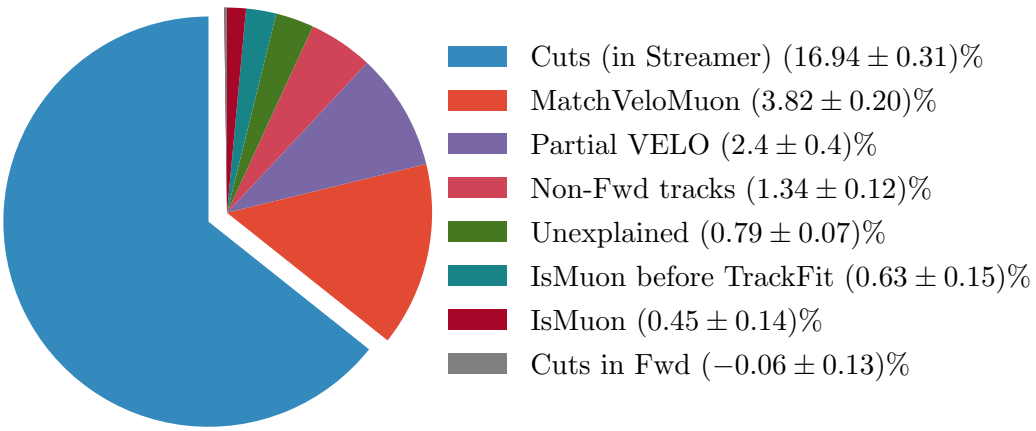

Figure 11: Individual contributions to the total inefficiency. It can be seen that about 17 % of inefficiency are due to the cuts in the streamer. The remaining 9 % stem from timing-driven choices in the software.

5.3 Comparative study for Run 1 and Run 2

In this scenario, a sample of simulated $B^+ \to J/\psi K^+$ with $J/\psi \to \mu^+\mu^-$ signal events is used. Offline selected signal candidates were used [16]. In addition, a cut based selection was added to additionally clean up the sample. Figure 12 shows the distributions of mass, momentum, and transverse momentum of the sample. Three different versions of the trigger sequence are used to understand the changes in efficiency between Run 1 and Run 2: the trigger software and cut values from Run 1 in 2012 (called 2012), the trigger software and cut values for Run 2 in 2015 (called 2015), and the trigger software for Run 2 with cut values as in 2012 (called 2015'). Therefore the resulting differences in efficiencies between 2012 and 2015' stem only from changes in the software, not in the cuts. All events are required to have passed `L0Muon` or `L0DiMuon` with Run 1 2012 threshold values. The efficiencies are calculated in bins of transverse momentum of the single muons for `Hlt1TrackMuon` and `Hlt1SingleMuonHighPT` and in bins of p_T of the reconstructed J/ψ for the `Hlt1DiMuonLowMass` and `Hlt1DiMuonHighMass` lines. Table 3 contains an overview of the efficiencies obtained for the HLT1 muon trigger lines, three regarding single muons and two regarding muon pairs. They are visualised in Fig. 13 in bins of transverse momentum. For single muon tracks, the increase in efficiency due to changes in the reconstruction software is found to be $(8.43 \pm 0.30)\,\%$. This means that $(89.7 \pm 0.5)\,\%$ of the previously mentioned reconstruction-based inefficiencies have been removed. The efficiency for muon pairs is increased by about 15 %. This means a large improvement for analyses on decay channels like $B^0 \to K^{*0}\mu^+\mu^-$ or $B_s^0 \to \mu^+\mu^-$. This effect is slightly mitigated by the cut values used in 2015 to reduce the CPU time consumption. The effects are of sub-percent order except for `Hlt1DiMuonHighMass`.

Table 3: Efficiencies of HLT1 muon lines for 2012 code, 2015 code, and 2015 code with 2012 cuts (labelled 2015'). The important difference which amounts only to changes in the software is $\Delta\varepsilon = \varepsilon(2015') - \varepsilon(2012)$. The `Hlt1SingleMuonNoIP` line is prescaled with a factor of 0.01 for 2012 and 0.1 for 2015. For easier comparison, the numbers presented in this table have been divided by those factors.

Trigger line	ε(TOS) [%]			$\Delta\varepsilon$ [%]
	2012	2015	2015'	
DiMuonLowMass	69.86 ± 0.35	85.75 ± 0.26	85.88 ± 0.26	16.0 ± 0.4
DiMuonHighMass	70.68 ± 0.34	81.28 ± 0.29	85.31 ± 0.27	14.6 ± 0.4
TrackMuon	75.25 ± 0.23	84.96 ± 0.19	83.68 ± 0.20	8.43 ± 0.30
SingleMuonHighPT	86.82 ± 0.51	95.07 ± 0.33	95.15 ± 0.33	8.3 ± 0.6
SingleMuonNoIP	74 ± 15	94.3 ± 4.4	94.4 ± 4.4	20.4 ± 15.6

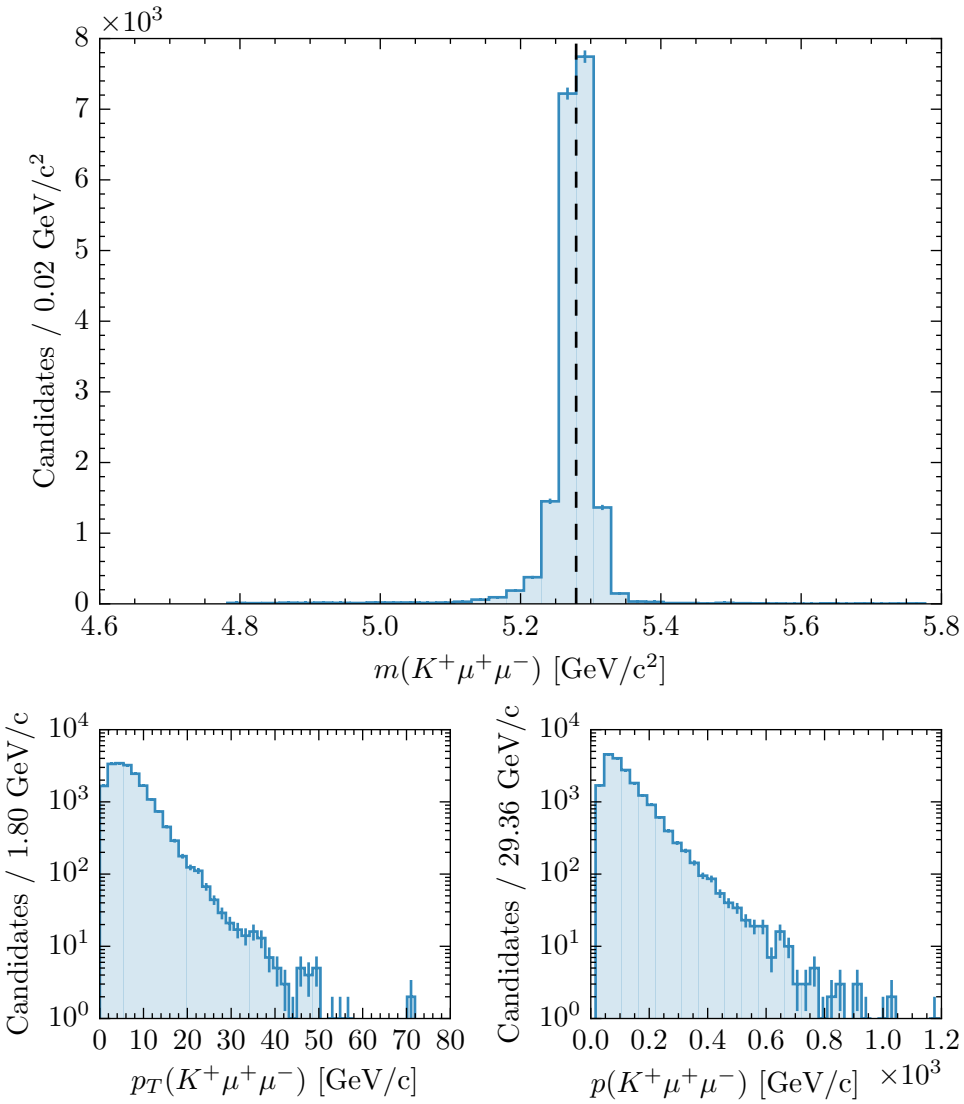

Figure 12: Mass (top), transverse momentum (bottom left), and momentum (bottom right) of the reconstructed $B^+ \to J/\psi K^+$ candidates in the simulated sample used to compare the efficiency of the HLT1 muon identification in Run 1 and Run 2.The nominal B^+ mass from reference [15] is marked by the dashed line at $(5279.26 \pm 0.17)\,\mathrm{MeV/c^2}$.

5.4 `MatchVeloTTMuon`

For the calculation of the `MatchVeloMuon` efficiency a sample of fully simulated $\Sigma \to p\mu^+\mu^-$ events accepted by the hardware trigger lines, is used. After reconstructing the `VeloTTCandidates` signal muons are passed to `MatchVeloMuon` and `MatchVeloTTMuon`. The ratio of the accepted over signal muons gives the efficiency of those two tools. Signal muons are defined as all the VELO segments that match the Monte Carlo true, and overlap more than 70% with the VELO segments reconstructed offline, which is quantified by the

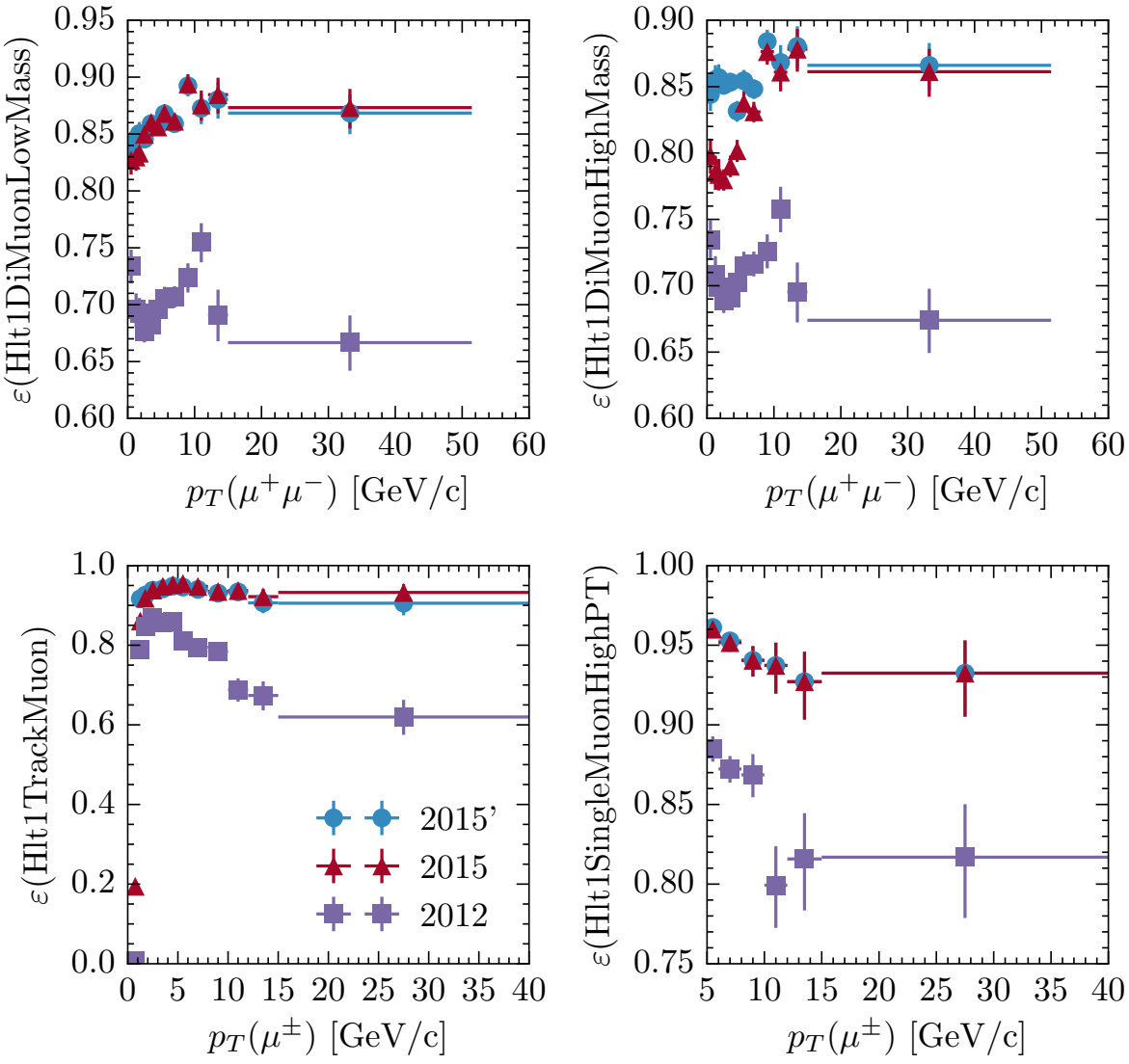

Figure 13: HLT1 TOS efficiencies evaluated on simulated $B^+ \to J/\psi K^+$ events. The efficiencies for 2012 are shown as purple squares, the ones for 2015 as red triangles, and those for the 2015 trigger with 2012 cuts (labelled as 2015') are shown as blue circles. It can be seen that the changes from 2012 to 2015 lead to significant gains in all discussed trigger lines. In addition, the p_T dependence improves for 2015. For the dimuon lines the curves are smoother and don't feature a sharp drop after 10 GeV/c anymore. The efficiency of the single track muon line is now almost constant for $p_T > 1$ GeV/c.

number of overlapping track hits. The efficiency curves for muons from the $B^0 \to K^{*0}\mu^+\mu^-$ and $\Sigma \to p\mu^+\mu^-$ decays are shown in Figs. 14 and 15, respectively. Top row shows the `MatchVeloMuon` efficiency as a function of p and p_{T}, whereas the bottom that of the `MatchVeloTTMuon` algorithm is displayed. Both algorithms get the same input tracks,

namely `VeloTTCandidates`. For low momenta the improvement is clear, whereas for high momenta there is no efficiency loss within the statistical uncertainties. However any potential loss after 500 MeV/c in p_{T} is not relevant since, as stated in Sect. **??**, `MatchVeloTTMuon` runs only on tracks with low momenta. In addition, the χ^2 distributions for the old and updated matching tool are shown in Fig. 16, where it can be seen the improvement of the discriminating power of the χ^2. A ROC curve of the updated algorithm is shown in Fig. 17, based on which the χ^2 cut-off value for the `MatchVeloTTMuon` algorithm is 2.

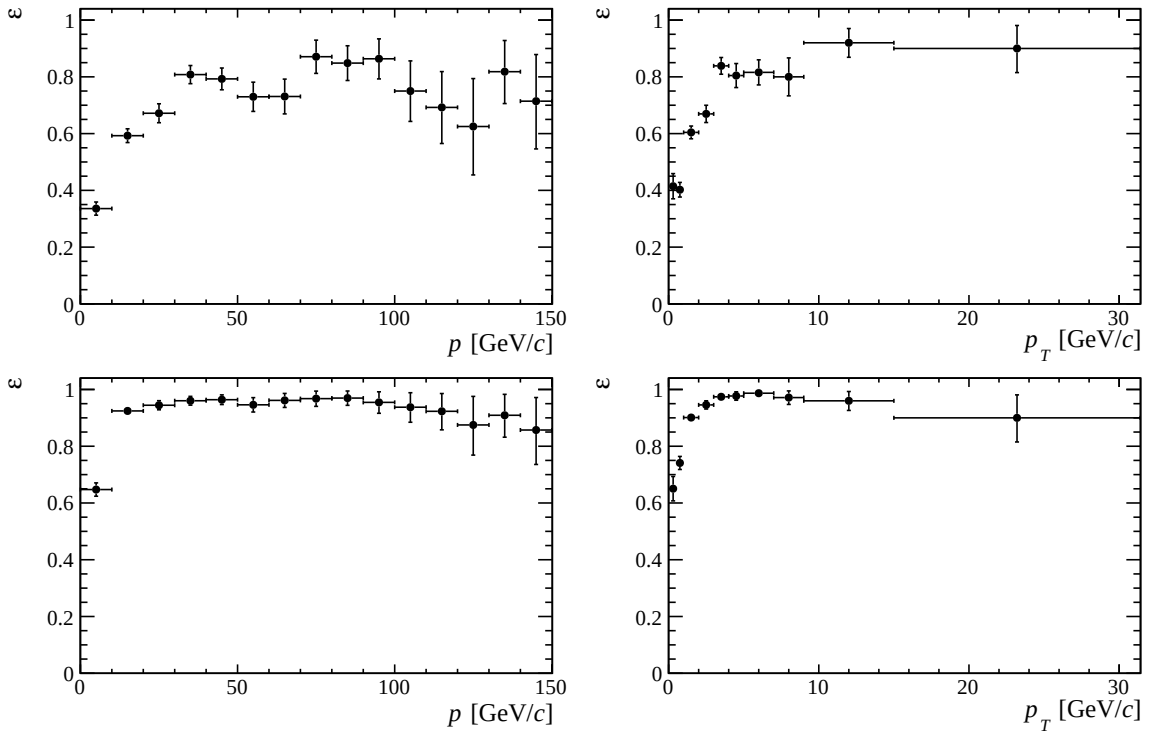

Figure 14: Efficiency comparison between the old `MatchVeloMuon` (top) and the new `MatchVeloTTMuon` (bottom) algorithms. The efficiency is projected versus total (left) and transverse (right) momentum. Muons from the $B^0 \to K^{*0}\mu^+\mu^-$ decay are used as input.

5.5 Timing

The timing spent by the new and old algorithms in the HLT sequence has been checked using 4000 minimum bias events collected in 2016. All tests have been performed on a dedicated HLT performance testing node, `hltperf-asus-amd6272`, and the results are summarized in Table 4.

As it possible to see, the replacement of the `MatchVeloMuon` with the `MatchVeloTTMuon` algorithm, speeds the reconstruction of low p_{T} muons by a factor ~ 1.6. This timing reduction has mainly to be attributed to the most efficient muon identification performed

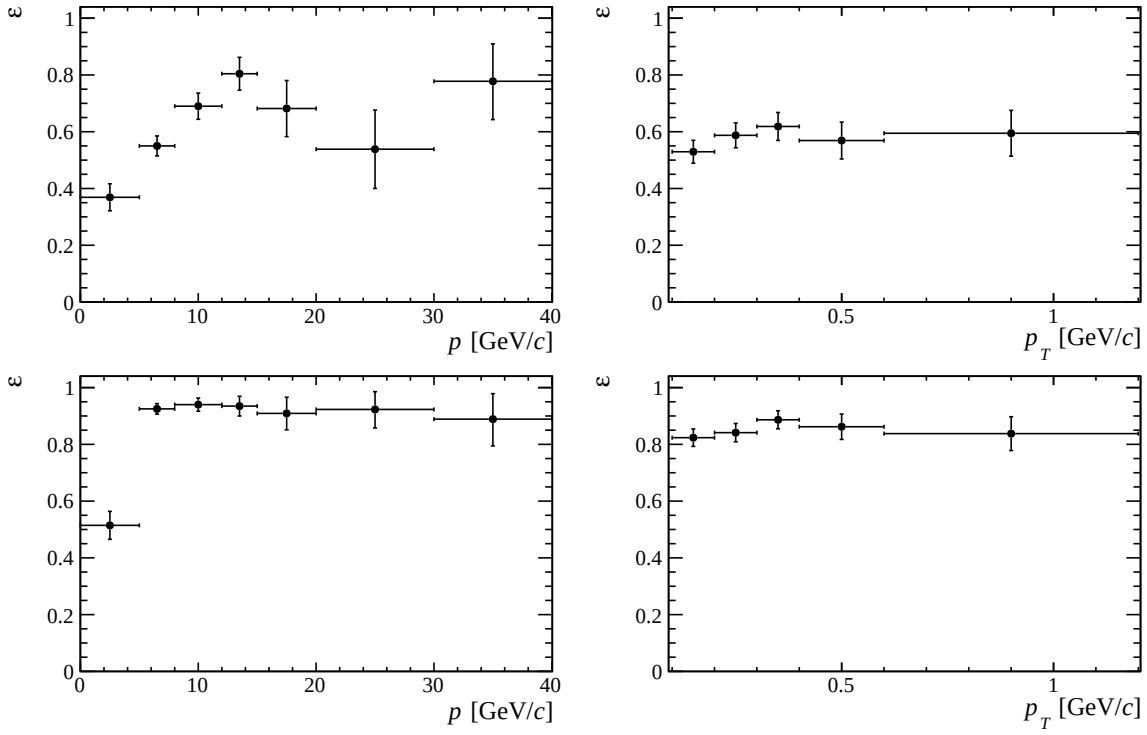

Figure 15: Efficiency comparison between the old `MatchVeloMuon` (top) and the new `MatchVeloTTMuon` (bottom) algorithms. The efficiency is projected versus total (left) and transverse (right) momentum. Muons from the $\Sigma \to p\mu^+\mu^-$ decay are used as input.

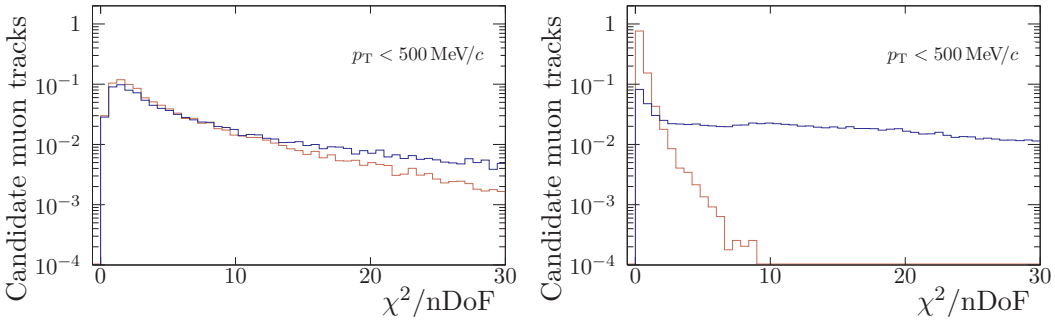

Figure 16: Distribution of the χ^2 between the old (left) and the new (right) matching algorithms. Simulated $\Sigma \to p\mu^+\mu^-$ decays are used. Matched (signal) and Fake (background) muon tracks are shown with orange and blue colours respectively.

by the `MatchVeloTTMuon`, which reduces considerably the amount of fake muon candidates that are passed to the Loose Forward tracking algorithm.

A remarkable improvement is also observed at the HLT2 level, where the new `MuonIDAlgLite` algorithm is measured to be 2.7 faster than its older version, per event. For reasons not discussed here, in Run 2 a larger number of tracks are passed to the muon

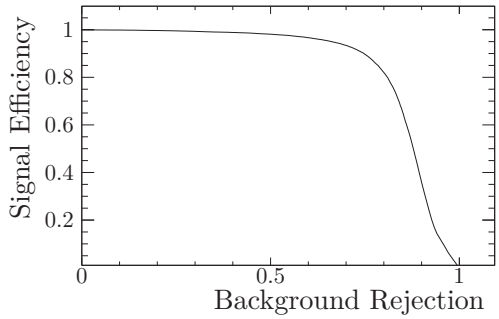

Figure 17: ROC curve for the updated `MatchVeloTTMuon` algorithm based on the χ^2 distributions of Fig. 16.

identification. If normalized to the number of input tracks analysed by the algorithm, an even larger gain in timing, equal to 7.3, is observed.

A similar study has been performed with the offline reconstruction sequence ((Brunel)) on a `lxplus` node over 5000 minimum bias 2012 events. The offline timing went from 2.9 ms/evt to roughly 0.9 ms/evt, which is compatible with what has been measured in the online environment.

Table 4: Per-event execution time per event of the muonID in the HLT1, HLT2 and offline levels. `MuonIDAlg` is compared to `MuonIDAlgLite` and `MatchVeloMuon` is compared to `MatchVeloTTMuon`. The total timing is measured running on 4000 minimum bias events. Absolute number cannot be compared between different levels.

Level	Algorithm	total (ms)
HLT1	`MatchVeloMuon`	0.191
	`MatchVeloTTMuon`	0.121
HLT2	`MuonIDAlg`	2.133
	`MuonIDAlgLite`	0.790
Offline	`MuonIDAlg`	2.966
	`MuonIDAlgLite`	0.919

6 Conclusions

During the first LHC shutdown, there has been a huge activity in the alignment of the software between the trigger and offline reconstruction at LHCb. In this work, a detailed analysis of the muon identification procedure and its inefficiencies in the LHCb experiment was performed. In Run 1 26 % of single muon tracks were not selected in the first stage of the software trigger. Part of this inefficiency (9.4%) was reconstruction-based and not due to selection cuts. The origin of most of those reconstruction-based inefficiencies has

been precisely determined. As a consequence, the majority of the inefficiencies have been removed. This translates into an absolute increase in efficiency of about 15% (only when triggering on muons) for some of the most important signal modes at the LHCb experiment, e.g. $B^0 \to K^{*0}\mu^+\mu^-$ and $B_s^0 \to \mu^+\mu^-$. The code for the muon identification procedure has been revised for Run 2. The software infrastructure has been unified such that now a common tool is used both in the offline reconstruction and in the both parts of the trigger. This improves maintainability and removes possible and unwanted differences between the online and offline muon identification.The timing spent by the muon identification on the online reconstruction has been evaluated on a computing farm node. It has been found that the changes in the algorithms reduced the execution time by more than a factor of three, from approximately 2.5 ms per event to less than 0.8 ms per event. This improvement has allowed to include in the online sequence the evaluation of the muon track χ^2, ensuring the perfect alignment between the online and offline execution sequences.

References

[1] G. Lanfranchi *et al.*, *The Muon Identification Procedure of the LHCb Experiment for the First Data*, Tech. Rep. LHCb-PUB-2009-013. CERN-LHCb-PUB-2009-013, CERN, Geneva, Aug, 2009.

[2] A. Sarti, S. Furcas, G. Lanfranchi, and M. Palutan, *Calibration Strategy and Efficiency Measurement of the Muon Identification Procedure at LHCb*, LHCb-PUB-2010-002.

[3] R. Aaj and J. Albrecht, *Muon triggers in the High Level Trigger of LHCb*, LHCb-PUB-2011-017.

[4] A. A. Alves Jr *et al.*, *Performance of the LHCb muon system*, J. Instrum. **8** (2012) P02022. 29 p.

[5] LHCb, R. Aaij *et al.*, *LHCb Detector Performance*, Int. J. Mod. Phys. **A30** (2015), no. 07 1530022, `arXiv:1412.6352`.

[6] M. Gandelman and E. Polycarpo, *The Performance of the LHCb Muon Identification Procedure*, Tech. Rep. LHCb-2007-145. CERN-LHCb-2007-145, CERN, Geneva, Mar, 2008.

[7] LHCb collaboration, *LHCb VELO (VErtex LOcator): Technical Design Report*, Technical Design Report LHCb, CERN, Geneva, 2001.

[8] LHCb collaboration, *LHCb reoptimized detector design and performance*, Tech. Rep. CERN/LHCC/2003-030, CERN, 2003.

[9] J. Albrecht, V. V. Gligorov, and S. Raven, G.Tolk, *Performance of the LHCb High Level Trigger in 2012*, LHCb-PROC-2014-005.

[10] R. Aaij, *Triggering on CP Violation Real-Time Selection and Reconstruction of $B_s \to J/$ Decays*, PhD thesis, Vrije U., Amsterdam, Jan, 2015.

[11] R. Aaij, *Triggering on CP Violation Real-Time Selection and Reconstruction of $B_s \to J/\psi\varphi$ Decays*, CERN-THESIS-2015-102.

[12] M. J. Morello, S. Neubert, and S. Stracka, *Studies towards an HLT1 tracking sequence for 2015*, LHCb-PUB-2015-005.

[13] A. Puig, *The LHCb trigger in 2011 and 2012*, Tech. Rep. LHCb-PUB-2014-046. CERN-LHCb-PUB-2014-046, CERN, Geneva, Nov, 2014.

[14] LHCb collaboration, R. Aaij *et al.*, *Measurement of form-factor-independent observables in the decay $B^0 \to K^{*0}\mu^+\mu^-$*, Phys. Rev. Lett. **111** (2013) 191801, arXiv:1308.1707.

[15] Particle Data Group, K. A. Olive *et al.*, *Review of Particle Physics*, Chin. Phys. **C38** (2014) 090001.

[16] LHCb, R. Aaij *et al.*, *Measurement of the $B_s^0 \to \mu^+\mu^-$ branching fraction and search for $B^0 \to \mu^+\mu^-$ decays at the LHCb experiment*, Phys. Rev. Lett. **111** (2013) 101805, arXiv:1307.5024.