PatLongLivedTracking: a tracking algorithm for the reconstruction of the daughters of long-lived particles in LHCb

A. Davis\textsuperscript{1}, M. De Cian\textsuperscript{2}, A. Dendek\textsuperscript{3}, T. Szumlak\textsuperscript{3}

\textsuperscript{1}University of Manchester
\textsuperscript{2}École Polytechnique Fédérale Lausanne
\textsuperscript{3}AGH University of Science and Technology

«Live long and prosper» (Spock, 2230 - )

Abstract

The PatLongLivedTracking algorithm is used to reconstruct tracks of daughters of long-lived particles such as $K^0_S$ and $\Lambda$ in Run II of the LHCb data taking period. It uses as input tracks from the standalone tracking in the T stations and propagates them back through the magnetic field, where information from the TT detector is added. This note presents a detailed description of the algorithm and its performance.

© 2018 CERN for the benefit of the LHCb collaboration.
1 Introduction

The LHCb detector [1] consists of subsystems designed to perform track reconstruction with > 95% efficiency [2] and provide momentum resolution of 0.5% for particle momenta \( p < 20 \text{GeV}/c \) [3]. The LHCb tracking detectors are the VertexLocator (VELO) around the interaction point and the T stations, located after the LHCb magnet, consisting of an Inner Tracker (IT) and Outer Tracker (OT). Additionally, a tracking detector exists upstream of the magnet, the Tracker Turicensis (TT). Most of the analyses in LHCb use so-called long tracks, which are particle trajectories traversing the full LHCb tracking volume, and leave clusters in the VELO, the T stations and potentially in TT. Long tracks are e.g. coming from decay products of relatively short-lived particles like \( B \) or \( D \) mesons, and are reconstructed using two dedicated algorithms [4–6].

Decay products of long-living particles, such as \( K_0^0 \) or \( \Lambda \), which decay outside the VELO, only leave charged clusters in TT and the T stations. The tracks formed by these particles are called downstream tracks. Since the start of the 25ns data taking period of Run II of the LHC in 2015, these tracks are reconstructed in LHCb using an algorithm called PatLongLivedTracking. This algorithm superseded an earlier implementation called PatDownstream [7,8]. This document describes the goals and implementation of the PatLongLivedTracking algorithm, as well as its performance.

1.1 Track types

The tracks reconstructed in the LHCb detector are divided into types depending on the subdetectors in which they are reconstructed, as shown in Fig. 1. VELO tracks are defined as those which have measurements only in the VELO subdetector. Upstream tracks are defined as those which have measurements only in the VELO and TT subdetectors. Upstream tracks are also referred to as VeloTT tracks. T tracks are defined as those which have measurements solely in the T stations. Downstream tracks have measurements in the TT subdetector and T stations. Long tracks have measurements in the VELO subdetector and T stations and may also have measurements in the TT subdetector. These tracks provide the best momentum resolution for particles which traverse the full tracking detector and are used in the majority of LHCb analyses.

1.1.1 Track states

The coordinate system of LHCb forms a right-handed system with the \( z \) axis pointing from the collision point along the beampipe through the detector, the \( x-z \) plane being the bending plane (with the \( x \) axis pointing sideways) and the \( y-z \) plane being the non-bending plane (with the \( y \) axis pointing upwards).

In LHCb, a track is modelled as a series of straight line segments called track states. A track state is defined by a state vector of the form:

\[
\begin{pmatrix}
  x \\
  y \\
  t_x \\
  t_y \\
  q/p
\end{pmatrix}
\]

with \( t_x = \frac{\partial x}{\partial z} \) and \( t_y = \frac{\partial y}{\partial z} \) (1)
and a corresponding $5 \times 5$ state covariance matrix at a given position in $z$. Here, $q$ and $p$ are the charge and momentum of the track respectively.

### 1.2 PatLongLivedTracking strategy

Downstream tracks at LHCb are reconstructed in the following way. First a standalone track reconstruction in the T stations with an algorithm called PatSeeding is performed [9], creating T tracks. After filtering with a multivariate classifier to discard bad candidates, i.e. tracks that don’t represent the trajectory of a real particle, these tracks are propagated back through the magnet to the TT station, assuming they are coming from the point of origin of the coordinate system. The TT consists of four layers of silicon microstrip sensors: the first and last layer have strips running vertically (so-called $x1$ and $x2$ layers) and probe the $x$ coordinate. The layers in between, called stereo layers, have the strips tilted by $5^\circ$ ($u$ layer) and $-5^\circ$ ($v$ layer), respectively. This allows for the determination of the $y$ coordinate. The TT is separated into two sections: TT$a$ comprises the $x1$ and $u$ layer, TT$b$ comprises the $v$ and $x2$ layers. One cluster is searched for in any of the two $x$ layers of the TT. This already allows to constrain the flight path of the particle with a good precision. Then a cluster in the other $x$ layer is searched for. Lastly, clusters are searched in the $u$ and $v$ layers. The clusters are then fitted with a $\chi^2$ fit using a parabola as a model. As with a bunch-crossing frequency of 40 MHz spillover can occur, a flag is set for each cluster if it is likely to have been created in a different collision than the one in question (so-called «high-threshold bit»); tracks with a large number of high-threshold clusters are rejected. Finally, the best track is chosen according to the output of another multivariate classifier.

### 1.3 Figures of merit

In the following subsections, the relevant figures of merit for the performance measurement of the PatLongLivedTracking algorithm will be introduced, namely the track reconstruction
efficiency and the rate of wrongly reconstructed tracks.

1.3.1 Reconstruction efficiency

The reconstruction efficiency is measured using simulated data samples by comparing the number of correctly reconstructed tracks, i.e. tracks that can be matched to a simulated particle, with the number of particles that with a perfect algorithm could be reconstructed (called reconstructible). This is made possible using truth information from the simulation of the detector in Geant4. Within the LHCb framework the following definitions are used:

- A particle is reconstructible as a downstream track if it is reconstructible as a T track and has at least one hit\(^1\) in both TTa and TTb.
- A particle is reconstructible as a T track if it has at least one \(x\) and one stereo hit associated to it in each of the three T stations.
- A particle is considered reconstructed as a downstream track if at least 70% of the T-station hits on a track are associated to it and the track has no more than 1 wrongly associated TT hit.

When calculating the efficiency of the downstream track reconstruction algorithm, particles are required to be reconstructible as downstream tracks. Two types of efficiency exist: the first is the overall efficiency which includes losses of efficiency from PatSeeding and PatLongLivedTracking. This efficiency is most useful for physics studies where the most important number is the total reconstruction efficiency. It is given in Eq. 2. The second is the efficiency of the PatLongLivedTracking algorithm itself, asking that a downstream reconstructible T track has already been reconstructed. This efficiency is therefore contingent on the reconstruction of the T seed. It is given in Eq. 3

\[
\varepsilon_{\text{rec,1}} = \frac{N_{\text{downstream reconstructed}}}{N_{\text{downstream reconstructible}}}, \tag{2}
\]

\[
\varepsilon_{\text{rec,2}} = \frac{N_{\text{downstream reconstructed and T reconstructed}}}{N_{\text{downstream reconstructible and T reconstructed}}}. \tag{3}
\]

For all efficiency calculations, electrons are excluded. Electrons undergo much more multiple scattering and energy loss than all other particles whose tracks are reconstructed in LHCb. Tuning for electrons specifically therefore would worsen the performance of a large part of the physics programme of LHCb\(^2\).

1.3.2 Ghost fraction and clone tracks

A ghost track is a track that has no matching simulated particle. The ghost fraction is defined as:

\[
\text{Ghost fraction} = \frac{N_{\text{ghost tracks}}}{N_{\text{tracks}}}. \tag{4}
\]

\(^1\)A hit is a charged cluster with a defined position i.e. the crossing point of a particle with the detector.

\(^2\)Note that this applies here to electrons from long-living particles, which normally are not used in a physics analysis.
A track matched to a simulated particle which has at least one other associated track is said to be a clone.

2 The track model

2.1 Propagation through the magnetic field

Neglecting multiple scattering, particles propagating through the tracking stations, which all lie outside the magnetic field of LHCb, will follow a straight line path, and the bending of the trajectory inside the field can be approximated by a sharp kink in the flight path at a given $z$ position of a magnet point, called $z_{\text{mag}}$ (see Fig. 2 for an illustration). The $z$ position of this point depends on the parameters of the track itself, and is also affected by inhomogenities in the field. The idea is to parametrize this point as precisely as possible from the properties of the T track. The following empirical formula is used:

$$z_{\text{mag}} = \alpha_0 + \alpha_1 \cdot t_y^2 + \alpha_2 \cdot t_x^2 + \alpha_3 \cdot 1/p + \alpha_4 \cdot |x_T| + \alpha_5 \cdot |y_T| + \alpha_6 \cdot |t_y|. \quad (5)$$

Here, $t_x$ and $t_y$ are the slopes of the last track state in the T stations. The absolute momentum, $p$, is estimated from the T track, using a parametrisation that assumes a kink of the trajectory at the center of the magnet and the track to come from the point of origin. The observables $x_T$ and $y_T$ are the $x$ and $y$ positions of the last state in the T stations, respectively. The variable $z_{\text{mag}}$ depends most strongly on the values in the first line of Eq. 5 while the dependence on the ones from the second row is smaller.

Figure 2: Sketch of the LHCb detector with the tracking system in the $x$-$z$ plane. A downstream track (blue line) and its approximation outside the magnetic field (red dotted line) is shown. In the approximation the track undergoes a sharp kink at the magnet point (big red dot).

The $\alpha$ parameters are determined in simulation, by fitting a straight line to the true position of the hits in TT, and a third order polynomial to the true position of the hits in the T stations. The crossing point of both curves in the $x$-$z$ projection determines the «true» value of $z_{\text{mag}}$. An illustration of $z_{\text{mag}}$ determined in this way, and
the difference between the values obtained with Eq. 5 when using measured instead of simulated quantities are shown in Fig. 3. The values of the $\alpha$ parameters are given in Appendix D.

![Image](image.png)

Figure 3: Left: «True» $z$ position of the magnet point. Right: Difference between true and estimated $z$ position of the magnet point. For the true position, simulated observables, for the estimated position measured observables were used.

The $x$ position of the magnet point is then determined by extrapolating the state in the T stations linearly to $z_{\text{mag}}$.

$$x_{\text{mag}} = x_T + t_x \cdot (z_{\text{mag}} - z_T)$$  \hspace{1cm} (6)

For the $y$ position, an additional correction is applied:

$$y_{\text{mag}} = y_T + t_y \cdot (z_{\text{mag}} - z_T) - \beta_1 \cdot t_y \cdot \Delta^2_{\text{slope}}$$  \hspace{1cm} (7)

with $\Delta_{\text{slope}}$ the difference of the slopes in $x$ before and after the magnet, determined as

$$\Delta_{\text{slope}} = |x_{\text{mag}}/z_{\text{mag}} - t_x|,$$  \hspace{1cm} (8)

and $\beta_1$ an empirical parameter, determined on simulation, to correct for the effect of the magnetic field on the $y$ position. The predicted slope in $y$ in the TT is calculated as:

$$t_{y,TT} = t_y \cdot (1 + \beta_0 |t_y| \Delta^2_{\text{slope}})$$  \hspace{1cm} (9)

with $\beta_0$ an empirical parameter, determined on simulation, using regression with true observables, to correct for the effect of the bending in the magnetic field on the slope in $y$.

A special treatment is needed for tracks with hits in the OT only: As the $y$ resolution and the resolution of the slope in $y$ is rather poor, the track is constrained to come from $y(z = 0) = 0$. The following parametrisation is used:

$$y_{TT} = y_T + t_y \cdot (z_{\text{mag}} - z_T) + t_{y,TT} \cdot (z_{TT} - z_{\text{mag}}) - \beta_1 \cdot t_y \cdot \Delta^2_{\text{slope}};$$  \hspace{1cm} (10)

with $\beta_1$ again determined on simulation using regression with true observables. The default values of $\beta_0$ and $\beta_1$ are given in Appendix D. Assuming that no further correction
is needed between the point of origin of the track and TT, the slope in the T stations can be calculated as:

\[ t_y,\text{constr} = y_T/(z_T + (\beta_0|t_y|z_{\text{mag}} + \beta_1)\Delta_{\text{slope}}^2) \]  

(11)

For tracks with hits in the OT only, \( t_y,\text{constr} \) is used in Eq. 7 instead of the measured value \( t_y \).

As the magnet point will be used as an additional constraint in the \( \chi^2 \) fit (see Sect. 3.7), an uncertainty has to be assigned to it. This is calculated by determining the difference between the values of \( x_{\text{mag}} \) and \( y_{\text{mag}} \) obtained using simulated quantities and reconstructed quantities for the extrapolation. The uncertainty depends on \( \Delta_{\text{slope}} \) and whether the track mainly has IT or OT hits. The following parametrisations of the uncertainty were used:

\[
\begin{align*}
\sigma_{x,\text{OT}} &= (2.0 + 18.0 \cdot \Delta_{\text{slope}}) \text{ mm} \\
\sigma_{y,\text{OT}} &= (5.0 + 20.0 \cdot \Delta_{\text{slope}} + 50 \cdot \Delta_{\text{slope}}^2) \text{ mm} \\
\sigma_{x,\text{IT}} &= (1.0 + 16.0 \cdot \Delta_{\text{slope}}) \text{ mm} \\
\sigma_{y,\text{IT}} &= (2.0 + 15 \cdot \Delta_{\text{slope}}) \text{ mm}
\end{align*}
\]

These parameters were obtained by fitting residual distributions, see Appendix C.

2.2 Determination of the momentum

The momentum of a downstream track mainly depends on the kink it undergoes in the magnetic field, but also shows a dependence on the slopes in \( x \) and \( y \). The following parameterisation results in a momentum resolution of about 2% averaged over the momentum spectrum.

\[ p = \gamma_0 + \gamma_1 \cdot t_x^2 + \gamma_2 \cdot t_y^2 / \Delta_{\text{slope}}, \]  

(12)

where again the parameters \( \gamma_i \) were determined using the true position of the hits in simulation. The default values of the \( \gamma \) parameters are given in Appendix D. This resolution contains both the effect from the detector resolution and from dependencies which are not accounted for in the parametrisation. The resolution is illustrated in Fig. 4.

2.3 Track model in the TT

Despite the fact that the strength of the magnetic field is greatest in the region between TT and the T stations, there is a strong enough residual field in and before the TT such that low-momentum particles undergo deflection from a straight line. A downstream track therefore is represented by a parabola in the TT. However, as a parabola has three degrees of freedom, but one has to be able to fit a track with three clusters in the TT only, the curve-parameter is fixed from the kink of the track through the magnetic field in order for the \( \chi^2/\text{ndf} \) value from the fit to provide a meaningful goodness-of-fit statement. The track model therefore is:

\[ x(z) = x_0 + t_x \cdot (z - z_{\text{mag}}) + c \cdot (z - z_{\text{TT}})^2, \]  

(13)
with \( z_{TT} \) the \( z \) position in the middle of the TT, \( z_{mag} \) the \( z \) position of the magnet point, 
\( t_x \) the slope in \( x \) in TT, and \( c = 1.48 \cdot 10^{-5} \cdot \Delta_{\text{slope}} \). The initial value for \( x_0 \) is \( x_{mag} \). The parameter \( c \) is determined on simulation by fitting the true position of the hits in TT with a parabola and determining it as a function of \( \Delta_{\text{slope}} \). The deflection from a straight line is very small: for a track with a momentum of 3 GeV/c it is about 200 \( \mu m \), which is about the strip pitch of a TT sensor.

The initial slope in \( x \) of the track in TT is given by:

\[
\frac{t_{x,TT}}{z_{mag}} = \frac{x_{mag}}{z_{mag}}. \tag{14}
\]

As explained in Sect. 2.1 the slope in \( y \) is given by:

\[
t_{y,TT} = t_y \cdot (1 + \beta_0 |t_y| \Delta_{\text{slope}}^2) \tag{15}
\]

with \( t_y \) being replaced by \( t_{y,\text{constr}} \) in the case of a track with OT hits only. Note that \( t_y \) is not updated in the fit (see Sect. 3.7). The function for the \( y \) position therefore is:

\[
y(z) = y_0 + t_{y,TT} \cdot (z - z_{mag}), \tag{16}
\]

where the initial value for \( y_0 \) is \( y_{mag} \).

\section{Pattern recognition}

Figure 5 gives an overview of the stages of the pattern recognition in PatLongLivedTracking. The arrows depict iterations over compatible tracklets\(^3\) in the individual layers.

\subsection{Selection of T tracks}

As discussed in Sect. 1.2 T tracks are created by the algorithm PatSeeding. All T tracks are reconstructed under the assumption that they originate from the luminous region. T tracks constitute, together with TT hits, the main input of the PatLongLivedTracking algorithm and their quality has a direct impact on the algorithm performance.

\(^3\)A tracklet is an incomplete track object, where not all information has been added yet.
At the beginning of the PatLongLivedTracking algorithm, a filtering procedure (see Fig. 5) is introduced. Its purpose is increasing the purity of the T tracks by rejecting as many ghost tracks as possible. A previous analysis showed that T-track ghost rejection based on a simple linear selection using track quality and kinematical variables is not

\[4\] There is also an execution time gain since the combinatoric is greatly reduced.

Figure 5: Different stages of pattern recognition in PatLongLivedTracking. The arrows depict iterations over individual seed tracks and iterations over compatible tracklets in the individual layers.
Initially, a Fisher discriminant was used to increase the purity of T tracks coming from long-lived particle decays. Further improvement, described in this note, can be achieved by using more sophisticated machine learning algorithms.

The search for the best classifier was performed using a simulated data sample containing signal \( B \rightarrow J/\psi K_s^0 \) events. Any reconstructed track segment that could be associated with a valid Monte Carlo particle not being an electron was considered as a true seed. In addition, a true track candidate had to have associated hits in the TT stations but none in the VELO detector (see also the discussion in Sect. 1.3.1). In turn, a sample containing the fake tracks was prepared using tracks that were not successfully associated with a Monte Carlo particle. A critical step in the pre-processing of input data is balancing the composition of the training sample, that should have an approximately equal number of true and fake T tracks. In our case both training and verification samples contained about 40% of true tracks and were prepared using random dropping. It was decided to use the XGBoost library (Extreme Gradient Boosting [10]), containing the latest developments regarding machine learning techniques. The following input variables were used for the training (see Fig. 6):

- \( \chi^2/\text{ndf} \): T track \( \chi^2/\text{ndf} \) as determined by the PatSeeding algorithm\(^5\)
- \( p \): The absolute momentum of a T track
- \( p_T \): The transverse momentum
- \( N_{\text{hits}} \): The number of hits contributing to a given T track
- \( |x_{\text{T track}}| \): The absolute \( x \) position of the T track’s first state
- \( |y_{\text{T track}}| \): The absolute \( y \) position of the T track’s first state
- \( |t_x| \): The absolute slope of the track in the \( x-z \) plane
- \( |t_y| \): The absolute slope of the track in the \( y-z \) plane
- \( r_{\text{T track}} \): The distance from the \( z \)-axis (beam line) calculated for the first state
- \( \eta \): Pseudorapidity

The final input variables were chosen using their importance metrics, calculated by tools available in the XGBoost library API, and having the compromise between the area under the ROC curve and model complexity in mind. At the end the listed ones were used to train the final model and to optimize its architecture (hyperparameters).

In order to assess the performance of the selected algorithms two figures of merit were chosen: the classification accuracy, \( A_{\text{Classifier}}(th_0) \), and the area under the ROC (Receiver Operator Curve) curve, \( A_{\text{ROC}} \). The former is defined as:

\[
A_{\text{Classifier}}(th_0) = \frac{N_{\text{TP}}^{\text{Classifier}}(th_0) + N_{\text{TN}}^{\text{Classifier}}(th_0)}{N_{\text{Total}}^{\text{Classifier}}} \tag{17}
\]

where \( N_{\text{TP}}^{\text{Classifier}} \) is the number of true tracks classified as true ones (true positives); \( N_{\text{TN}}^{\text{Classifier}} \) is the number of fake tracks classified as fake ones (true negative); \( N_{\text{Total}}^{\text{Classifier}} \) is the total

\(^5\)ndf stands for «number of degrees of freedom».
Figure 6: Input variables for the binned BDT in PatLongLivedTracking extracted using a sample of $B \to J/\psi K^0_s$ decays. The green solid histogram is the signal distribution, while the light blue distribution is the background.
number of tracks used in classification and \( th_0 \) is the classifier discrimination threshold. Two of the tested models that showed the best results were: Boosted Decision Tree (BDT) and a neural network. Having in mind the necessity of porting the algorithm into the global LHCb computing framework the BDT has been selected. The final studies on the selected BDT algorithm focused mainly on the hyperparameter optimization. The most important ones that have been used in the performance analysis were: learning rate \( r \), number of decision trees \( n_{\text{trees}} \) and their maximal depth \( t_{\text{depth}}^{(\text{max})} \). Hyperparameters tuning allowed to find the best working point of the BDT algorithm that offered the optimal performance without overfitting. The XGBoost library offers a convenient facility that helps to detect both under and overfitting of the model and to choose the optimal threshold value. During the training one needs to specify the splitting of the data set into the training and testing samples. Next, the model is preliminarily trained and used against the testing sample and an internal quality metric is estimated (based, among others, on the classification accuracy and the area under the ROC curve). This procedure is repeated until the most optimal working point is found. The final values, estimated by the training procedure, of selected figures of merit obtained for the XGBoost BDT are: \( A^{(\text{BDT})} = 0.78 \) and \( A_{\text{ROC}}^{(\text{BDT})} = 0.94 \), with \( r = 0.1, n_{\text{trees}} = 400 \) and \( t_{\text{depth}}^{(\text{max})} = 6 \). In addition, based on the confusion matrix (see Fig. [7]) determined for the optimal threshold value, we can compute the trained model purity defined as follow:

\[
P_{\text{BDT}}(th_0) = \frac{1}{3} (\text{TPR}_{TP}^{(\text{BDT})}(th_0) + \text{TNR}_{TP}^{(\text{BDT})}(th_0) + \text{ACC}_{TP}^{(\text{BDT})}(th_0))
\]

where: TPR and TNR stands for true positive and true negative event rates. In our case the additional purity metric yielded: \( P_{\text{BDT}}(th_0) = 0.91 \).

![Normalized confusion matrix](image)

Figure 7: Normalised confusion matrix determined for the trained BDT model.

The final step of extending the PatLongLivedTracking algorithm with a BDT based
T-track filter was to port the trained classifier into the global LHCb computing framework. Since the tracking algorithm is a part of the real-time LHCb High Level Trigger system, both the execution time and memory footprint are important, so using the full continuous classifier is not an option. Instead, a binned BDT (called bonsai BDT or bBDT [11]) classifier that meets the speed and memory criteria of the HLT is used. The appropriate C++ code for fetching the corresponding response for a given set of input parameters using the binned classifier was generated with the hep_ml.speedup library [12]. The response of the binned BDT and its ROC curve are presented in Figs. 8 and 9 respectively. A slight drop in the performance of the binned classifier with respect to the full one has been noted. The figures of merit measured for the bBDT algorithm amounted to: $A^{(bBDT)} = 0.73$ and $A^{(bBDT)}_{ROC} = 0.87$.

All T tracks with a bBDT value larger than SeedCut are considered.\[6\]

![Figure 8: Binned BDT classifier output distribution, blue is background and orange is signal.](image)

### 3.2 T tracks as input

The pattern recognition starts by considering all T tracks that were filtered with the bBDT as potential parts of downstream tracks.\[7\]

In a first step, the particle is assumed to have come from the origin of the coordinate system. The momentum is then determined by extrapolating the T track to the center of the magnet (see Sect. 2.1 for details) and then taking the difference in the slope of the track when constraining to the origin. Track candidates are rejected if their estimated momentum is below 1.4 GeV/c (MinMomentum) and their estimated $p_T$ is below 40 MeV/c (MinPt), as the vast majority of real downstream reconstructible particles have values above this minimum, as illustrated in Fig. 10. The loss in real tracks due to these cuts is

---

\[6\] Quantities than can be set via options in the algorithm are denoted in TrueType. Their default values is given in Appendix D.

\[7\] A filtering option exists which discards T tracks that are already used as part of long tracks.
Figure 9: Comparison of ROC curves for selecting true T tracks using a simulated $B \rightarrow J/\psi K_s^0$ sample before and after binning.

about 0.15%. Furthermore, track candidates are rejected if they are likely to originate from inside the beam-pipe, i.e. their $x$ or $y$ coordinates at TTa are compatible with being in the beam-pipe. They are further rejected if their estimated momentum, evaluated under the assumption of being a downstream track with Eq. (12) is incompatible with the original momentum estimate from the T track, which is optimized for long tracks. These criteria are illustrated in Fig. 11. The left hand figure shows $x$ and $y$ position of truth-matched tracks at $z=2350\text{mm}$, which is the middle position of TTa. The right hand figure shows the difference in estimated momentum as a function of momentum. The cut on the difference between the original momentum estimate from the T track and the momentum estimate in PatLongLivedTracking is given by:

$$\Delta p = \text{MaxDeltaPConst} \cdot p + \text{MaxDeltaPOffset},$$

where $p$ is the momentum estimate in PatLongLivedTracking. $\text{MaxDeltaPConst}$ and $\text{MaxDeltaPOffset}$, as well as all following parameters, were optimized on simulated $B^0 \rightarrow J/\psi K_s^0$ decays.

3.3 First search for compatible hits

Track candidates at this stage have the magnet point (see Sect. 2.1) as the only constrain, and the track model at this point is a straight line from $(0,0,0)$ to the magnet point. To correct for the fact that most particles of interest do not originate from the global LHCb
origin (0,0,0), a correction to the $x$ position of the search window is applied, which is also derived from simulation. The value is given by:

$$
\delta x = \text{sign}(p \cdot \text{magPol}) \cdot (\text{XCorrectionConst}/p + \text{XCorrectionOffset}),
$$

(20)

with magPol the magnet polarity. An illustration can be seen in Fig. 12 left.

To search for hits compatible with this first track estimate, momentum-dependent search windows in the TT are opened. An illustration can be seen in Fig. 12. The dependence of the window size on the momentum is:

$$
\Delta x = \text{XPredTolConst}/p + \text{XPredTolOffset},
$$

(21)

where $p$ is the absolute value of the momentum. The parameters determining the size of the search window, $\text{XPredTolConst}$ and $\text{XPredTolOffset}$, were derived on simulation, illustrated on the right of Fig. 12.
Furthermore, it is checked if the hit is compatible with the expected $y$ position in the TT with a tolerance $Y_{\text{Tol}}$. As the position of a hit in the TT can only be determined up to the length of a sensor module, this provides only a weak constraint. The $x$ positions of the hits in the stereo layers are then updated, assuming the $y$ position from the T track extrapolation.

All hits in all four layers which lie inside these tolerances are then stored in a container, and sorted according to the projection distance, i.e. the absolute distance of the hit to the predicted position.

The T track is not considered if less than 3 hits in the TT are found in total, or the $x$ or the stereo layers do not contain any hit.

### 3.4 Search for hits in $x$ layers

In this step, the algorithm iterates over all hits in the $x$ layers. The first hit is chosen in one of the two $x$ layers, if a hit is present. This allows for a more precise determination of the slope of the track and the curvature, and therefore also of the momentum of the track – these quantities of the track candidate are therefore updated. Next, corresponding hits in the other $x$ layer are searched for. This essentially means that, if several hits are present in both $x$ layers, all possible combinations between the hits are iterated over. This leads to the duplication of hit combinations, which however given the good timing performance of the algorithm (see Sect. 4.3) is negligible.

The search window in the other $x$ layer is defined as follows:

$$\Delta x = \frac{\text{TolMatchConst}}{p} + \text{TolMatchOffset},$$

if $\Delta x$ is smaller than a given value $\text{MaxWindowSize}$, otherwise $\text{MaxWindowSize}$ is taken as the tolerance. This serves as a sanity check to exclude unphysically large values of the window size for low momentum particles. All hits within this window are then considered.
for further processing. As illustrated in the left hand plot of Fig. 13 essentially all hits from true particles in simulation are enclosed within this region.

Figure 13: Left: Distance between extrapolated track with one x hit and (truth-matched) hit in the other x layer. The red line represents the cut on the distance. Right: $\chi^2$ values for a fit to the x hits (truth-matched), with the cut represented as a red line.

A $\chi^2$/ndf fit to the x coordinate is then performed for each possible candidate, consisting of the first hit in the x layer, and one of the matching hits in the other x layer. The magnet point is used as a further point to add enough degrees of freedom for the fit. All track candidates are then sorted according to their $\chi^2$/ndf value, and track candidates are discarded if the $\chi^2$/ndf value is above:

$$\chi^2/\text{ndf}_{\text{max}, x \text{ hits}} = \text{FitXProjChi2Const}/p + \text{FitXProjChi2Offset}.$$  \hfill (23)

If no hit could be found in the other x layer, the track candidate is not rejected. It is kept without fit to search for hits in the u layer. This allows for hit inefficiencies in the TT.

Due to the large combinatorial background, there can be many ghosts in this selection. Tracklets with only two x hits are prone to be ghost tracks, and it is possible that the true combination of hits does not have the smallest $\chi^2$/ndf. Therefore, in the next steps, the first MaxXTracks tracklets are considered whose $\chi^2$/ndf value is within the range ($\text{MaxChi2DistXTracks}$) to the lowest $\chi^2$/ndf value.

3.5 Search for hits in the u layer

As the parameters of a track with (mostly) two hits are reasonably well constrained, the search window in the u layer can be smaller than those in the x layers. Hits are searched around the track extrapolated to the u layer, where the x position of the hit is updated by using the extrapolated y position. All hits within a search window are considered, where the window size is defined as

$$\Delta x = \text{TolUConst}/p + \text{TolUOffset}.$$  \hfill (24)

The parameters TolUConst and TolUOffset are determined from simulation, illustrated in the left hand plot of Fig. 14. The hits inside the window are then sorted according to their distance between the extrapolation of the track and their actual position. For each
hit which passes this tolerance, a new tracklet is formed, until the maximum number of xu tracks (MaxXUTracks) is reached. Each of these tracklets is then fitted with a \( \chi^2 \) fit to obtain the best parameters for the following search in the \( v \) layer.

### 3.6 Search for hits in the \( v \) layer

As a last step in the hit search, hits in the \( v \) layer are searched for. If the track candidate already has 3 hits, a momentum dependent window is created according to

\[
\Delta x = \text{TolVConst}/p + \text{TolVOffset}.
\]  

See Fig. 14 (right) for an illustration. In case the track only contains 2 hits, Eq. 24 is used to determine the size of the search window. The hit that is closest to the extrapolation is added.

![Figure 14: Left: Distance between extrapolated \( x \)-hits track and (truth-matched) hit in \( u \) layer. The red line represents the cut on the distance. Right: Distance between extrapolated \( xu \)-hits track and (truth-matched) hit in \( v \) layer. The red line represents the cut on the distance.](image)

### 3.7 Calculation of \( \chi^2 \) and outlier removal

At this stage, all possible hits in TT were added to the track candidates (see Sect. 3.9 for adding hits in the overlap region of TT), and a \( \chi^2 \) fit is performed. If the fit \( \chi^2/\text{ndf} \) is smaller than a given value (MaxChi2 and MaxChi2ThreeHits for candidates with 4 and 3 hits, respectively), the candidate is accepted and passed on to the next stage, otherwise the hit that contributes most to the \( \chi^2 \) is removed and the fit is repeated. The procedure is repeated until either the \( \chi^2/\text{ndf} \) is small enough, or there are hits in less than 3 planes left. Furthermore, for each iteration of the fit, an explicit check is made that each hit is still compatible with the estimated \( y \) position of the track.

The \( \chi^2 \) fit only fits in 1 dimension by converting the displacement in \( y \) in a displacement in \( x \). The \( \chi^2 \) is therefore defined as:

\[
\chi^2 = \sum_{i \in \text{hits}} \left( \frac{x_i - (x_0 + t_x \cdot (z - z_{mag}) + c \cdot (z - z_{TT})(z - z_{TT}) + \left( \frac{dx}{dz} \right)_0 y_0)}{\sigma_i} \right)^2
\]  

(26)
with \( \left( \frac{dx}{dy} \right)_i \) the slope of the stereo plane and \( y_0 \) the displacement in \( y \). Note that \( c \) is fixed, see Sect. 2.3. Forming partial derivatives with respect to \( x_0, t_x \) and \( y_0 \), a set of linear equations can be solved to obtain the three free parameters. For details of this calculation, see for example Ref. [8].

![Figure 15: Left: \( \chi^2 \) distribution for truth-matched tracks with 4 or more hits. The maximum allowed value is 30. Right: \( \chi^2 \) distribution for truth-matched tracks with 3 hits. The maximum allowed value is 50.](image)

### 3.8 Accepting the candidates

The track candidate is accepted in the final selection of track candidates, if it fulfills the following criteria. All cut values were trained on simulated data, with the goal to keep a balance between signal retention and background rejection.

- It has at least three hits in at least three different layers of the TT.
- The \( \chi^2/\text{ndf} \) is below a given threshold. This value is different for three-hit tracks (\( \text{MaxChi2ThreeHits} \)) and tracks with four or more hits (\( \text{MaxChi2} \)), see Fig. 15.
- It contains at least as many hits as any other candidate for a given T track.
- The track has a pseudorapidity larger than 1.8 and smaller than 5.2.
- At least \( N_{\text{MinHighThresHits}} \) of the hits have the high-threshold bit set.

The following checks are repeated, this time using the fit result instead of the initial estimates.

- The momentum estimate is compatible with the momentum estimate from the T track.
- The track has a minimum momentum \( \text{MinMomentum} \) and a minimum \( p_T \) \( \text{MinPt} \).
- The track does not point into the beampipe.
3.9 Addition of overlap hits

For all track candidates which are retained at this stage, hits in the overlap regions of the TT detector are searched for. TT modules are staggered in the $z$ direction with an overlap, in order to cover the insensitive region of the modules and to not lose efficiency. A hit qualifies as an overlap hit if it is within a certain distance to the extrapolated track ($OverlapTol$), in the same layer as an already existing hit, but at a different $z$ position.

As adding an additional hit will change the properties of the track, the track is again fitted with the potential removal of outliers. It is then again checked that it fulfills the criteria in Sect. 3.8.

3.10 MVA for the final selection

In the final stage, a neural network is employed to reject ghost tracks and find the best downstream track candidate for a given input T track. The network used is a multilayer perceptron (MLP) with one hidden layer and 15 nodes in the hidden layer, using the Broyden-Fletcher-Goldfarb-Shannon training (see e.g. Ref. [13]) and a Bayesian regulator from the TMVA package [14]. A rectified linear unit was used as an activation function for the hidden layer as it showed good timing and discrimination performance. For the output layer a sigmoid function with a fast calculation of the exponential was implemented. The following input variables were used:

- $\log(\chi^2/ndf)$: Natural logarithm of track $\chi^2/ndf$ given by the fit
- $\log(p)$: Natural logarithm of the total momentum
- $\log(p_T)$: Natural logarithm of the transverse momentum
- $\Delta p$: Difference between momentum estimate from the T track and final momentum estimate of downstream track candidate.
- $\log(|\Delta x \text{ magnet}|)$: Natural logarithm of $x$ displacement with respect to the point in the magnet after the fit
- $\log(|\text{dist}_{\text{initial}}|)$: Natural logarithm of the distance of the hits with respect to the initial track estimate
- $\log(|y_{\text{track}}(0)|)$: Natural logarithm of the absolute $y$ position at $z = 0$
- $\log(|x_{\text{track}}(0)|)$: Natural logarithm of the absolute $x$ position at $z = 0$
- $\# \text{ fired Layers}$: Number of layers in the TT with hits on this track

The variables were chosen by their importance as reported by TMVA after the training. Variables that only showed little discrimination power, or that are known to be poorly modeled on simulation (e.g. occupancy dependent variables) were not included in the set.

The network was trained on simulated $D^*+ \rightarrow D^0 \pi^+$ decays, with $D^0 \rightarrow K^0_s K^+ K^-$. The sample was chosen such that also tracks from $D$ decays are considered, which tend to have a softer momentum spectrum than tracks from $B$ decays and are more similar to low-momentum background tracks. Only downstream reconstructible tracks coming from
a strange particle decay which are not electrons were used as signal events. As background sample, tracks that could not be matched to a single simulated particle were used. The distributions of input variables is shown in Fig. 16. The classifier output distribution and the ROC curve is shown in Fig. 17.

Figure 16: Input variables for the MVA in PatLongLivedTracking on a sample of $D^{*+} \rightarrow D^0 \pi^+$ decays. The blue solid histogram is the signal distribution, while the hashed red distribution is the background.

A candidate is written out if its MVA value is larger than FinalMvaCut. It is possible to write out a maximum of MaxOutputCands, where the the lowest MVA value has to be closer to FinalMvaCut than FinalMvaSpread.
Figure 17: Left: Classifier output distribution, blue is signal, red is background. The default cut is at 0.1. Right: ROC curve for the MVA in PatLongLivedTracking. Both are obtained on a sample of $D^{**} \rightarrow D^0\pi^+$ decays.
4 Performance

As explained in Sect.[1.3.1] two benchmarks are quoted: a global efficiency and an algorithm performance. The global efficiency is the efficiency of reconstructing a charged particle in the acceptance as a downstream track. This comprises the efficiency of PatSeeding and of PatLongLivedTracking. The algorithm efficiency is the efficiency that is attributed to PatLongLivedTracking, i.e. the downstream track reconstruction efficiency for particles that are reconstructed as T tracks. The first number is of interest for physics analyses where one is interested in the efficiency to reconstruct a downstream track as such, while the second mainly for a verification of the algorithmic performance of PatLongLivedTracking. Note though that due to differences between simulation and collision data, these numbers are not necessarily a very precise representation of the efficiency of PatLongLivedTracking in collision data.

The efficiency numbers were obtained from simulation, using 50'000 events of both magnet polarities. The difference in the numbers between the polarities is 0.1 – 0.4%, depending on the type, and well compatible with statistical uncertainties. Furthermore, it was checked that inverting the magnetic field does not change the performance. The numbers for both magnet polarities were therefore averaged. Two types of decay were simulated: \(B^0 \rightarrow J/\psi K^0_s\) as a representative of a \(B\) decay, and \(D^{*+} \rightarrow D^0 \pi^+\), with \(D^0 \rightarrow K^0_s K^+ K^-\) for charm. The efficiency is only given for simulated particles that do not leave enough information in the VELO to be reconstructed as long tracks (i.e. they are not long reconstructible). There are four categories, all of them only contain tracks from daughters of particles with a non-zero strangeness:

- \(\varepsilon_{TT+T}\): Efficiency for all downstream reconstructible particles
- \(\varepsilon_{TT+T, p>5\text{ GeV}/c}\): Efficiency for all downstream reconstructible particles with \(p > 5\text{ GeV}/c\)
- \(\varepsilon_{TT+T, \text{from}B/D}\): Efficiency for all downstream reconstructible particles from a decay of a \(B\) or \(D\) meson
- \(\varepsilon_{TT+T, \text{from}B/D, p>5\text{ GeV}/c}\): Efficiency for all downstream reconstructible particles with \(p > 5\text{ GeV}/c\) that are from a decay of a \(B\) or \(D\) meson

No uncertainties are given, as the numbers are not a precise measure of the efficiency on collision data. The statistical uncertainty is at the permille level.

4.1 Global efficiency numbers

The global efficiency numbers to reconstruct a downstream track are given in Table[1] upper half, and the ghost fraction is given in Table[2] upper half. Plots of the efficiency and the ghost fraction as a function of momentum, transverse momentum and pseudorapidity are shown in Figs. [18] and [19].

The downstream tracks are fitted with a Kalman Filter, where a cut on the \(\chi^2/\text{ndf}\) is applied, and a clone killing algorithm is run that removes duplicates, mostly between reconstructed long and downstream tracks, where the tracks with more subdetectors involved is kept. Clones with the same number of involved subdetectors are identified by checking their quality and number of shared hits. In addition a weak cut on the so-called
ghost probability, a neural network using quantities from the Kalman Filter to reject ghost tracks [15], is employed. Plots of the efficiency and the ghost fraction as a function of momentum, transverse momentum and pseudorapidity after the Kalman Filter, the clone killer and the ghost probability are shown in Figs. 20 and 21. Plots evaluated on the \( D^{*+} \rightarrow D^0 \pi^+ \) sample are given in Appendix A.

The performance after Kalman Filter, clone killer and ghost probability cut is given in Table 1, lower half, and Table 2, lower half. The benchmark after these three steps seems worse than before, as the \( \chi^2/ndf \) cut in the Kalman Filter and the ghost probability also reject correct tracks, and the clone killer is inefficient, i.e. it sometimes falsely classifies downstream tracks as long tracks (for example when the VELO part is a ghost), which then get removed from the downstream numbers. Furthermore, clone killing prefers long tracks over downstream tracks: of all downstream tracks, the non-ghost ones get clone-killed and one is left with a larger proportion of ghosts than before.

The largest drop in reconstruction efficiency comes from the fact that a wrong combination of hits is taken as the final track. Running the algorithm in simulation, only considering hits which actually belong to the simulated particle one wants to reconstruct, the efficiency is higher by 8% – 11% for daughters of strange particles and by 6% – 7% for daughters of strange particles from \( B \) or \( D \) decays.

Table 1: Reconstruction efficiency of downstream tracks on simulated samples of \( B^0 \rightarrow J/\psi K_S^0 \) and \( D^{*+} \rightarrow D^0 \pi^+ \) decays. The upper half are non-filtered tracks, the lower half is after the clone killer, Kalman Filter and ghost probability. This efficiency includes the efficiency of PatSeeding and PatLongLivedTracking. Due to a softer momentum spectrum, the efficiency is lower in the \( D^{*+} \rightarrow D^0 \pi^+ \) sample.

<table>
<thead>
<tr>
<th>filter</th>
<th>decay type</th>
<th>( \varepsilon_{TT+T} )</th>
<th>( \varepsilon_{TT+T, \ p&gt;5 \text{ GeV}} )</th>
<th>( \varepsilon_{TT+T, \ \text{from} B/D} )</th>
<th>( \varepsilon_{TT+T, \ \text{from} B/D, \ p&gt;5 \text{ GeV}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>no ( B^0 \rightarrow J/\psi K_S^0 )</td>
<td>73.3%</td>
<td>80.1%</td>
<td>81.4%</td>
<td>85.4%</td>
<td></td>
</tr>
<tr>
<td>( D^{*+} \rightarrow D^0 \pi^+ )</td>
<td>71.3%</td>
<td>78.0%</td>
<td>76.8%</td>
<td>81.4%</td>
<td></td>
</tr>
<tr>
<td>yes ( B^0 \rightarrow J/\psi K_S^0 )</td>
<td>70.0%</td>
<td>76.7%</td>
<td>79.0%</td>
<td>83.2%</td>
<td></td>
</tr>
<tr>
<td>( D^{*+} \rightarrow D^0 \pi^+ )</td>
<td>67.3%</td>
<td>73.7%</td>
<td>73.1%</td>
<td>77.3%</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Ghost fraction of downstream tracks on simulated samples of \( B^0 \rightarrow J/\psi K_S^0 \) and \( D^{*0} \rightarrow D^- \pi^+ \) decays. The upper half are non-filtered tracks, the lower half is after the clone killer, Kalman Filter and ghost probability. This ghost fraction includes the ghosts produced in PatSeeding and PatLongLivedTracking. Due to a softer momentum spectrum, the ghost fraction is higher in the \( D^{*+} \rightarrow D^0 \pi^+ \) sample.

<table>
<thead>
<tr>
<th>filter</th>
<th>decay type</th>
<th>fraction of ghosts</th>
</tr>
</thead>
<tbody>
<tr>
<td>no ( B^0 \rightarrow J/\psi K_S^0 )</td>
<td>29.5%</td>
<td></td>
</tr>
<tr>
<td>( D^{*+} \rightarrow D^0 \pi^+ )</td>
<td>30.3%</td>
<td></td>
</tr>
<tr>
<td>yes ( B^0 \rightarrow J/\psi K_S^0 )</td>
<td>39.2%</td>
<td></td>
</tr>
<tr>
<td>( D^{*+} \rightarrow D^0 \pi^+ )</td>
<td>40.2%</td>
<td></td>
</tr>
</tbody>
</table>
Figure 18: Efficiencies to reconstruct downstream tracks as a function of momentum (top row), transverse momentum (middle row) and pseudorapidity (bottom row). The left column is for all downstream reconstructible tracks, the right column for all downstream tracks from a decay chain of a $B$ or $D$ meson. The efficiencies are obtained on a simulated sample of $B^0 \rightarrow J/\psi \ K_S^0$ decays.
Figure 19: Ghost fraction in downstream tracks as a function of momentum (top left), transverse momentum (top right) and pseudorapidity (bottom). The ghost fractions are obtained on a simulated sample of $B^0 \rightarrow J/\psi K^0_S$ decays.
Figure 20: Efficiencies to reconstruct downstream tracks, after clone killing, Kalman filtering and the cut on the ghost probability, as a function of momentum (top row), transverse momentum (middle row) and pseudorapidity (bottom row). The left column is for all downstream reconstructible tracks, the right column for all downstream tracks from a decay chain of a $B$ or $D$ meson. The efficiencies are obtained on a simulated sample of $B^0 \rightarrow J/\psi K_S^0$ decays.
Figure 21: Ghost fraction in downstream tracks, after clone killing, Kalman filtering and the cut on the ghost probability, as a function of momentum (top left), transverse momentum (top right) and pseudorapidity (bottom). The ghost fractions are obtained on a simulated sample of $B^0 \to J/\psi K^0_S$ decays.
4.2 PatLongLivedTracking efficiency numbers

The efficiency numbers of the PatLongLivedTracking algorithm are given in Table 3. Plots of the efficiency as a function of momentum, transverse momentum and pseudorapidity are shown in Fig. 22. Given the very high efficiency of the PatSeeding algorithm, these numbers largely correspond to the ones reported in Sect. 4.1. No Kalman Filter, clone killing or ghost probability is applied here.

Table 3: Reconstruction efficiency of the PatLongLivedTracking algorithm on simulated samples of $B^0 \to J/\psi K_s^0$ and $D^{*+} \to D^0 \pi^+$ decays. Due to a softer momentum spectrum, the efficiency is lower in the $D^{*+} \to D^0 \pi^+$ sample.

<table>
<thead>
<tr>
<th>decay type</th>
<th>$\epsilon_{TT+T}$</th>
<th>$\epsilon_{TT+T, p&gt;5\text{ GeV}}$</th>
<th>$\epsilon_{TT+T, \text{ from }B/D}$</th>
<th>$\epsilon_{TT+T, \text{ from }B/D, p&gt;5\text{ GeV}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B^0 \to J/\psi K_s^0$</td>
<td>77.6%</td>
<td>85.1%</td>
<td>86.3%</td>
<td>90.4%</td>
</tr>
<tr>
<td>$D^{*+} \to D^0 \pi^+$</td>
<td>75.7%</td>
<td>83.1%</td>
<td>81.6%</td>
<td>86.5%</td>
</tr>
</tbody>
</table>

4.3 Processing time

As the processing time depends a lot on the type of machine and the event multiplicity it is hard to give an absolute number. On a simulated signal sample for Run II on a worker node at CERN, PatLongLivedTracking takes $O(5\text{ ms})$ to process an event. This time is not significant for the overall timing budget of HLT 2.

Comparisons using callgrind (see e.g. Ref. [16]) show a decrease in processing time compared to PatDownstream of about 45%.
Figure 22: Efficiencies of the PatLongLivedTracking algorithm, as a function of momentum (top row), transverse momentum (middle row) and pseudorapidity (bottom row). The left column is for all downstream reconstructible tracks, the right column for all downstream tracks from a decay chain of a $B$ or $D$ meson. The efficiencies are obtained on a simulated sample of $B^0 \to J/\psi K^0_S$ decays.
4.4 Notes on performance

As can be seen from Figs. 20 and 22, the efficiency strongly depends on the momentum and transverse momentum of the tracks. This has two main reasons: first, the search windows do not cover the full region necessary to find all tracks (see e.g. Fig. 12, right), and second, for low momentum tracks, search windows are generally larger than for high momentum tracks, which increases the number of hits in the search window, and increases the chances that the MVA identifies the wrong hits.

In addition, the downstream tracks that are also reconstructed as long tracks are less prone to be ghosts, which is why the ghost fraction actually increases after the Kalman Filter, the clone killer and the ghost probability cut. However, it should be noted that in a physics analysis of a decay channel, a strict cut is normally placed on the ghost probability of the downstream tracks, which reduces the ghost fraction significantly.

Furthmore, in Appendix B, a comparison to PatDownstream is made, which is the predecessor of this algorithm. The comparison shows that PatLongLivedTracking is superior in all relevant aspects: track reconstruction efficiency, ghost fraction and timing. It has to be noted however that PatDownstream was never optimized for Run II data taking.

5 Ideas for further improvements

PatLongLivedTracking was optimized for data taking in Run II. However, for running in LHCb upgrade conditions (with a factor five increase in instantaneous luminosity), several improvements can be foreseen:

- A lot of CPU time is spent in calculating the $\chi^2$ fits. This is no problem for Run II, but it starts to be relevant for Run III. A possible approach would be to use a simple Kalman filter, which would propagate from layer to layer, instead of fitting the full track for each layer with a $\chi^2$ fit. Possibly also a more correct $\chi^2$/ndof distribution could be obtained.

- A multivariate classifier could be employed while making the tracks, and not only at the beginning and the end. This could help getting rid of bad tracks at an early stage.

- So far, the uncertainties on the extrapolations are parameterized. A better method would be to use information from the fit in the T stations to have a track-by-track uncertainty, allowing for better tuned search windows.

- The algorithm could be staged: in a first stage, the high-momentum tracks, which are less likely to be ghost tracks, could be found and their hits flagged as being used, while in the second stage, the lower momentum tracks could be found, with a lower probability for using wrong hit combinations due to the removal of hits used in the first stage. This would rely on a very clean selection of tracks (i.e. having only a small ghost fraction in the first stage) in order not to compromise the efficiency.
6 Conclusion

The algorithm PatLongLivedTracking, which is the default algorithm to reconstruct tracks of daughters of long-lived particles in LHCb, was presented. It considerably improves on the performance in Run II data taking conditions over the previous algorithm in terms of track reconstruction efficiency, ghost track rate, and CPU time consumption.
A Efficiency and ghost fraction plots for $D^{*0} \to D^- \pi^+$ sample

Figure 23: Efficiencies to reconstruct downstream tracks as a function of momentum (top row), transverse momentum (middle row) and pseudorapidity (bottom row). The left column is for all downstream reconstructible tracks, the right column for all downstream tracks from a decay chain of a $B$ or $D$ meson. The efficiencies are obtained on a simulated sample of $D^{*0} \to D^- \pi^+$ decays.
Figure 24: Efficiencies to reconstruct downstream tracks, after clone killing, Kalman filtering and the cut on the ghost probability, as a function of momentum (top row), transverse momentum (middle row) and pseudorapidity (bottom row). The left column is for all downstream reconstructible tracks, the right column for all downstream tracks from a decay chain of a $B$ or $D$ meson. The efficiencies are obtained on a simulated sample of $D^{*0} \rightarrow D^- \pi^+$ decays.
Figure 25: Ghost fraction in downstream tracks as a function of momentum (top left), transverse momentum (top right) and pseudorapidity (bottom). The ghost fractions are obtained on a simulated sample of $D^{*0} \rightarrow D^- \pi^+$ decays.
Figure 26: Ghost fraction in downstream tracks, after clone killing, Kalman filtering and the cut on the ghost probability, as a function of momentum (top left), transverse momentum (top right) and pseudorapidity (bottom). The ghost fractions are obtained on a simulated sample of $D^{*0} \rightarrow D^- \pi^+$ decays.
Figure 27: Efficiencies of the PatLongLivedTracking algorithm, as a function of momentum (top row), transverse momentum (middle row) and pseudorapidity (bottom row). The left column is for all downstream reconstructible tracks, the right column for all downstream tracks from a decay chain of a $B$ or $D$ meson. The efficiencies are obtained on a simulated sample of $D^{*0} \rightarrow D^- \pi^+$ decays.
B Comparison between PatLongLivedTracking and PatDownstream

Figures 28 and 29 show the efficiency and ghost fraction of downstream tracks reconstructed with PatLongLivedTracking or PatDownstream in a sample of simulated $B^0 \rightarrow J/\psi K_S^0$ decays as a function of different quantities, before Kalman filtering and clone killing. Figures 30 and 31 show the same quantities after the Kalman filter and clone killer. In both cases a gain of efficiency and a strong reduction in the ghost fraction is visible for the new algorithm. The overall numbers are given in Tables 4, 5 and 6. Note that the comparison is somewhat unfair, as PatDownstream was never optimized to run on Run II data. However, it still serves as an estimate of the improvements made.

Table 4: Comparisons of reconstruction efficiency of downstream tracks made with PatLongLivedTracking or PatDownstream, on simulated samples of $B^0 \rightarrow J/\psi K_S^0$. This efficiency includes the efficiency of PatSeeding and PatLongLivedTracking.

<table>
<thead>
<tr>
<th>algorithm</th>
<th>$\varepsilon_{TT+T}$</th>
<th>$\varepsilon_{TT+T, p&gt;5\text{ GeV}}$</th>
<th>$\varepsilon_{TT+T, \text{ from } B/D}$</th>
<th>$\varepsilon_{TT+T, \text{ from } B/D, p&gt;5\text{ GeV}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PatLongLivedTracking</td>
<td>73.3%</td>
<td>80.1%</td>
<td>81.4%</td>
<td>85.4%</td>
</tr>
<tr>
<td>PatDownstream</td>
<td>68.3%</td>
<td>74.4%</td>
<td>77.1%</td>
<td>81.6%</td>
</tr>
</tbody>
</table>

Table 5: Comparisons of reconstruction efficiency of downstream tracks made with PatLongLivedTracking or PatDownstream, on simulated samples of $B^0 \rightarrow J/\psi K_S^0$, after the Kalman Filter and clone killer. This efficiency includes the efficiency of PatSeeding and PatLongLivedTracking. The efficiency decreases compared to Table 4 due to inefficiency of the clone killer.

<table>
<thead>
<tr>
<th>algorithm</th>
<th>$\varepsilon_{TT+T}$</th>
<th>$\varepsilon_{TT+T, p&gt;5\text{ GeV}}$</th>
<th>$\varepsilon_{TT+T, \text{ from } B/D}$</th>
<th>$\varepsilon_{TT+T, \text{ from } B/D, p&gt;5\text{ GeV}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PatLongLivedTracking</td>
<td>70.0%</td>
<td>76.7%</td>
<td>79.0%</td>
<td>83.2%</td>
</tr>
<tr>
<td>PatDownstream</td>
<td>63.7%</td>
<td>71.3%</td>
<td>74.5%</td>
<td>79.2%</td>
</tr>
</tbody>
</table>

Table 6: Ghost fraction of downstream tracks on simulated samples of $B^0 \rightarrow J/\psi K_S^0$ for PatLongLivedTracking and PatDownstream, once before and after the clone killer, Kalman Filter and ghost probability (CKG) were applied. This ghost fraction includes the ghosts produced in PatSeeding and PatLongLivedTracking (PatDownstream). The ghost fraction is higher after the clone killer as tracks, also reconstructed as long tracks, are not present in that number.

<table>
<thead>
<tr>
<th>algorithm</th>
<th>fraction of ghosts</th>
<th>fraction of ghosts after CKG</th>
</tr>
</thead>
<tbody>
<tr>
<td>PatLongLivedTracking</td>
<td>29.5%</td>
<td>39.2%</td>
</tr>
<tr>
<td>PatDownstream</td>
<td>46.3%</td>
<td>52.1%</td>
</tr>
</tbody>
</table>
Figure 28: Efficiencies to reconstruct downstream tracks as a function of momentum (top row), transverse momentum (middle row) and pseudorapidity (bottom row), in red for PatLongLivedTracking and in black for PatDownstream. The left column is for all downstream reconstructible tracks, the right column for all downstream tracks from a decay chain of a $B$ or $D$ meson. The efficiencies are obtained on a simulated sample of $B^0 \to J/\psi K^0_S$ decays.

Due to several improvements in the algorithmic structure of PatLongLivedTracking compared to PatDownstream, the algorithm is 30 - 40% faster. Its timing is not critical for running in the second stage of the software trigger.
Figure 29: Ghost fraction in downstream tracks as a function of momentum (top left), transverse momentum (top right) and pseudorapidity (bottom), in red for PatLongLivedTracking and in black for PatDownstream. The ghost fractions are obtained on a simulated sample of $B^0 \rightarrow J/\psi K^0_S$ decays.
Figure 30: Efficiencies to reconstruct downstream tracks, after clone killing, Kalman filtering and the cut on the ghost probability, as a function of momentum (top row), transverse momentum (middle row) and pseudorapidity (bottom row). The left column is for all downstream reconstructible tracks, the right column for all downstream tracks from a decay chain of a $B$ or $D$ meson. The efficiencies are obtained on a simulated sample of $B^0 \rightarrow J/\psi K^0_S$ decays.
Figure 31: Ghost fraction in downstream tracks, after clone killing, Kalman filtering and the cut on the ghost probability, as a function of momentum (top left), transverse momentum (top right) and pseudorapidity (bottom). The ghost fractions are obtained on a simulated sample of $B^0 \to J/\psi K^0_S$ decays.
C Residuals of the magnet point

Figure 32 shows the x and y residuals between the true and the reconstructed position of the point in the magnet, both, for tracks originating in the OT and in the IT. The functional shapes are the lowest order polynomials that provide a good fit. The algorithm performance only depends weakly on these parametrisations, hence the given treatment is justified.

Figure 32: Residuals between the true and the reconstructed position of the point in the magnet, for tracks reconstructed in the OT (top row) and the IT (bottom row). The residuals are obtained on a simulated sample of $D^{*0} \rightarrow D^{-}\pi^{+}$ decays.
### D Parameters and their default values

Tables 7 and 8 show the most important parameters in PatLongLivedTracking which can be set, and their corresponding default values. Some more parameters, mostly for more technical functionality like timing measurements, etc. are not listed here.

Table 7: Most important parameters in PatLongLivedTracking and their default values.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPredTo1Const</td>
<td>200'000 mm MeV/c</td>
</tr>
<tr>
<td>XPredTo1Offset</td>
<td>7.0 mm</td>
</tr>
<tr>
<td>TolMatchConst</td>
<td>20'000 mm MeV/c</td>
</tr>
<tr>
<td>TolMatchOffset</td>
<td>1.5 mm</td>
</tr>
<tr>
<td>TolUConst</td>
<td>20'000 mm MeV/c</td>
</tr>
<tr>
<td>TolUOffset</td>
<td>2.5 mm</td>
</tr>
<tr>
<td>TolVConst</td>
<td>2000 mm MeV/c</td>
</tr>
<tr>
<td>TolVOffset</td>
<td>0.5 mm</td>
</tr>
<tr>
<td>MaxWindowSize</td>
<td>10.0 mm</td>
</tr>
<tr>
<td>MaxChi2</td>
<td>30.0</td>
</tr>
<tr>
<td>MaxChi2ThreeHits</td>
<td>50.0</td>
</tr>
<tr>
<td>MinPt</td>
<td>40.0 MeV/c</td>
</tr>
<tr>
<td>MinMomentum</td>
<td>1400 MeV/c</td>
</tr>
<tr>
<td>MaxDeltaPConst</td>
<td>4 \cdot 10^{-6}</td>
</tr>
<tr>
<td>MaxDeltaPOffset</td>
<td>0.075 MeV/c</td>
</tr>
<tr>
<td>XCorrectionConst</td>
<td>23'605 mm MeV/c</td>
</tr>
<tr>
<td>XCorrectionOffset</td>
<td>0.4 mm</td>
</tr>
<tr>
<td>MaxXTracks</td>
<td>2</td>
</tr>
<tr>
<td>MaxChi2DistXTracks</td>
<td>0.2</td>
</tr>
<tr>
<td>MaxXUTracks</td>
<td>2</td>
</tr>
<tr>
<td>FitXProjChi2Offset</td>
<td>7.0</td>
</tr>
<tr>
<td>FitXProjChi2Const</td>
<td>35000.0</td>
</tr>
<tr>
<td>NMintHighThresHits</td>
<td>2</td>
</tr>
<tr>
<td>SeedCut</td>
<td>0.07</td>
</tr>
<tr>
<td>FinalMvaCut</td>
<td>0.1</td>
</tr>
<tr>
<td>FinalMvaSpread</td>
<td>0.1</td>
</tr>
<tr>
<td>MaxOutputCands</td>
<td>1</td>
</tr>
<tr>
<td>OverlapTol</td>
<td>2.0 mm</td>
</tr>
<tr>
<td>YTo1</td>
<td>2.0 mm</td>
</tr>
</tbody>
</table>
Table 8: Track parameters in PatLongLivedTracking and their default values.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_0$</td>
<td>5363.5 mm</td>
</tr>
<tr>
<td>$\alpha_1$</td>
<td>-3615.15 mm</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>399.87 mm</td>
</tr>
<tr>
<td>$\alpha_3$</td>
<td>97205 mm MeV/c</td>
</tr>
<tr>
<td>$\alpha_4$</td>
<td>-0.01081</td>
</tr>
<tr>
<td>$\alpha_5$</td>
<td>-0.15562</td>
</tr>
<tr>
<td>$\alpha_6$</td>
<td>1624.3 mm</td>
</tr>
<tr>
<td>$\beta_0$</td>
<td>5.588</td>
</tr>
<tr>
<td>$\beta_1$</td>
<td>1260.0</td>
</tr>
<tr>
<td>$\gamma_0$</td>
<td>1217.0 MeV/c</td>
</tr>
<tr>
<td>$\gamma_1$</td>
<td>487.5 MeV/c</td>
</tr>
<tr>
<td>$\gamma_2$</td>
<td>2864.7 MeV/c</td>
</tr>
</tbody>
</table>
References


