LHCb note 2001-112, TRAC

Track following for LHCb

Rutger Hierck
NIKHEF

October 23, 2001

Abstract

A first implementation of a track following algorithm in LHCb is discussed. A reconstruction efficiency of 94.4% is obtained for physics tracks going through the whole spectrometer. The associated ghost rate is 4.6%. Different detector setups are explored by changing the beampipe design, inner tracker size, material thickness of tracking stations, stereo angle of the measurement planes and the hit resolution. All changes have an impact on the reconstruction performance, either improveming or degradating the efficiency. However, the algoritm is able to deal with all of the changes in a flexible and robust way.

Contents

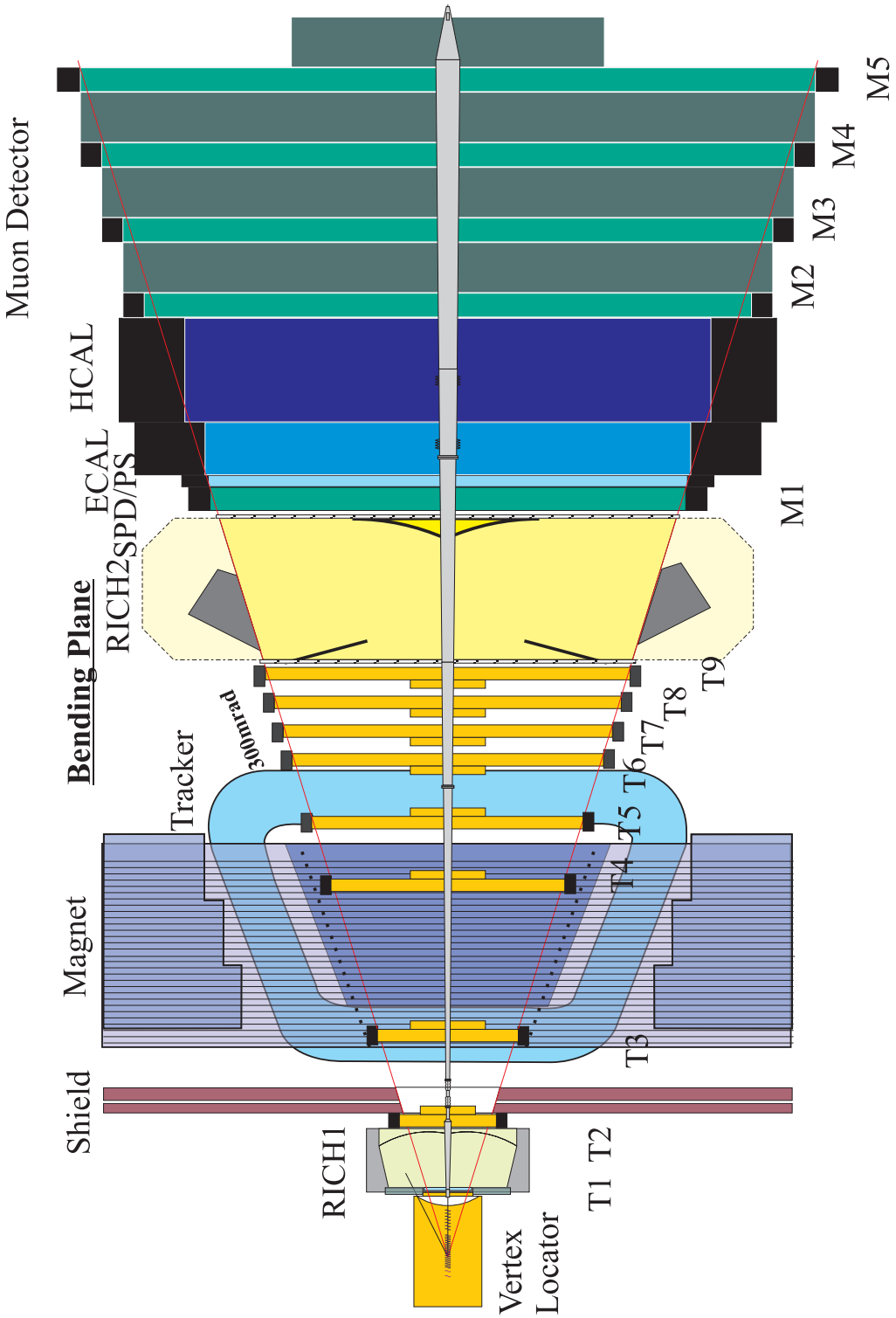

Figure 1: *Layout of the LHCb detector*

1 Introduction

In this note an implementation of a track following algorithm for the LHCb spectrometer is described. A track following algorithm assumes that the track coordinates at a given starting point of the track have been determined in a separate way: this procedure is referred to as track seeding [1]. In LHCb the seeding algorithm provides an initial track state vector $(x, y, t_x, t_y, Q/p)$ (positions, slopes, and curvature) and corresponding covariance matrix at the most downstream tracking station (see fig 1). The track following algorithm follows the track in the upstream direction, assigning the detector hits to the track in a progressive way.

In an environment of high track density, as is the case for LHCb, limiting the amount of combinatorics (i.e. CPU consumption) is crucial to the success of the algorithm. Although in principle each hit combination could be tried, the amount of combinations would render such an algorithm useless.

Track searching algorithms are often classified as being either of "global" or "local" type. Global algorithms use the fact that an overall parametrisation of a trajectory is available and detector hits can be mapped onto coordinates in a track parameter space. The mapped hits originating from a track cluster at a point; the coordinates of which represent the parameters of the track.

In local algorithms the track parameters require an initial ansatz (the *track-seed*). The parameters evolve as the track is followed through the detector. A track model is only required locally in order to assign a hit to the track. The track parameters are updated each time a hit is assigned to the track. The HERA-B track following method [2] is a good example of a local algorithm.

The algorithm described here tries to combine the merits of both local and global methods. It assumes that hits *within* a given station can be mapped to a trajectory with fixed track-parameters, while hits of *different* stations cannot due to the effects of an inhomogeneous B-field, multiple scattering and energy loss. More specific: the hits in a station (*intra-station* pattern recognition) are assigned to the track using a global cluster finding algorithm, while station-to-station track following (*inter-station* pattern recognition) is based on the HERA-B method of concurrent track evolution [2].

Effort is spent to design the method in a flexible way such that it can be applied for tracking station set-ups with a varying number of stations as well as varying numbers of measurement planes per station. Combining inner tracker and outer tracker technologies and the presence of overlap area's require no special treatment in the algorithm. Also, the algorithm can be used for both upstream, as well as downstream track following, by offering data

input in reverse order. However, the upstream method has the advantage that the seeding algorithm already supplies a full track-state including the track momentum, which is not the case for the downstream tracking.

Once the hits corresponding to a track are identified they are added one-by-one on to the track in a standard Kalman filter method, taking into account trajectory "kinks" due to multiple scattering. The Kalman filter implementation in LHCb track reconstruction is described in [3].

In this note first the track following algorithm is described to some detail; discussing both the intra-station algorithm and the inter-station algorithm. The second part addresses the results obtained with the default LHCb detector setup as described in [4] and finally results are given for alternative geometries and detector parameters.

2 Overview of the algorithm

In this section a general overview of the track following algorithm is given. All steps will be discussed in more detail in the following sections.

Figure 2 gives a schematic view of how the algorithm proceeds.

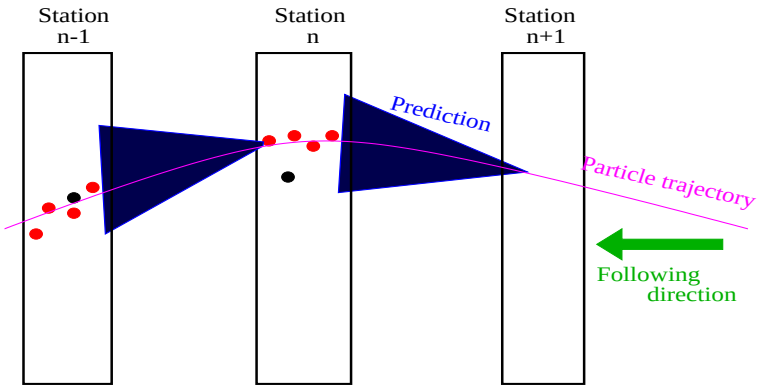

Figure 2: *Schematic picture of the track following algorithm*

Prior to the track following algorithm a seeding algorithm [1] provides a seed to the track state vector at the most downstream tracking station. The seeding is done by a track segment search in the low field region upstream of the RICH 2 detector. The seed includes a momentum estimate as obtained in the p_T-*kick* method [5], which assumes that the track originates from the nominal interaction point.

As the tracks are built in the upstream direction starting from the seeds it should be kept in mind that the algorithm proceeds from station-to-station, building all tracks simultaneously. Compared to an individual track-by-track approach this has the advantage that conflicting situations (e.g. multiple hit assignment) can be dealt with more easily.

The track following procedure can be broken down in two steps:

- intra-station pattern recognition: a cluster search for possible track continuations inside a station.
- inter-station pattern recognition: evolution of track candidates based on the HERA-B concurrent track evolution algorithm.

Several steps in the track following algorithm are schematically depicted in figure 2.

As an overview the track following procedure uses the following algorithms.

- **Intra Station:**
  - The input to the pattern recognition algorithms are the detector digitizations of the inner tracker [6] and outer tracker [7].
  - The fitted track state at a given z position (station $n+1$) is used to predict the trajectory in each of the detection layers of a subsequent station (station n).
  - In each measurement plane of station n a search window (the Region of Interest (RoI)) is opened around the predicted track position. Only hits inside the search window will be considered as candidates for a track continuation.
  - For each hit in the RoI the residual is calculated representing the distance between that hit and the predicted track position. In case the predicted trajectory closely resembles the true trajectory, hits belonging to the track are expected to scatter around the prediction. In case the predicted trajectory slightly deviates from the trajectory the hits will not be centered around the prediction, but in a given projection they will cluster at a similar distance from the track.
  - Grouping all the residual distances of the hits in the station with a clustering algorithm provides possible track continuations in the station.
  - The hits in the cluster are added to the track using the Kalman filter. Outliers are removed.

- **Inter Station:**
  - A maximum number of 10 track branches can be followed per station.
  - Each track branch is subjected to a quality criterion defining it to be "alive", "dead" or "zombie". Track candidates are *alive* until they miss a continuation in a subsequent station: then the label *zombie* is assigned. In case no continuation is found in a second station, they are declared *dead*.
  - Arriving at the last station tracks are subjected to a quality criteria. Track candidates that fail this criterion are killed. In addition, track candidates which share more then a certain fraction of their hits and with track coordinates that closely resemble each other are merged.
  - The final number of reconstructed tracks are compared to the true tracks in order to calculate the efficiency and ghost rate of the method.

2.1 Track seeding

The track following procedure is requires to be initiated with track seeds obtained in the separate track seeding algorithm. To not let the seeding performance interfere with the track following performance, perfect track seeding efficiency has been assumed here. Perfect seeding means seeds have been provided for all tracks of interest.

To mimic the finite resolution of the detector and the track seeding algorithm, the initial parameters (x, y, t_x , t_y and Q/p) are taken from the Monte Carlo truth, but are "smeared" to give initial conservative resolutions as compared to a realistic track seeding algorithm [1]. The Gaussian resolution coefficients are:

- $\sigma_x = 0.5$ mm,
- $\sigma_y = 1.0$ mm,
- $\sigma_{t_x} = 3.0$ mrad,
- $\sigma_{t_y} = 3.0$ mrad,
- $\sigma_{Q/p} = 5.0\%$

2.2 Stations vs. logical layer partitions

A key ingredient of the current track following algorithm is the application of a global pattern recognition algorithm method (clustering) to assign hits inside a station to a track. A station can consists out of OT and IT layers and measurement-layers can be oriented under several measurement angles. Instead of stations, the track following algorithm uses *logical layer partitions*, groups of measurement planes which are treated as identically in the clustering method.

The grouping of these layers has to be chosen with some care. The measurements in the different layers must be correlated, as this is used in the clustering algorithm (e.g. orthogonal measurement planes can not be combined in one logical layer partition). In addition, layers preferentially must be close in z. The inner and outer tracker layers of a given station are close enough in z to be combined

Since the track following algorithm has been used on various detector geometries in LHCb (some of which contain y-coordinate measurement planes) the layer partitioning was different for different geometries. However, in the geometry shown in fig 1, the layer partitions simply correspond to the stations.

3 Intra-station pattern recognition

3.1 Region of Interest

Track following starts with a prediction of the track coordinates of each of the layers in a station. Around the prediction a Region of Interest(RoI) is defined. Only the hit detector cells inside the RoI are used as input for the clustering algorithm, reducing the combinatorics.

For each hit in the search window the distance of the hit to the predicted track (the *residual*) is calculated. Figure 3 shows the momentum dependence of these residuals for the hits in station 4. In figure 4, the *calculated error* in x as function of momentum is shown and next to it the *pull* distribution for the hits is shown.

Table 1 gives the average of the predicted error in x in all the stations, separately for tracks with hits in the inner or outer tracker in that particular station. As expected, the error for tracks with hits in the outer tracker is larger than for the inner tracker. This is attributed to the higher measurement precision of the inner tracker as well as to the fact that tracks in the

inner tracker have on average higher momentum.

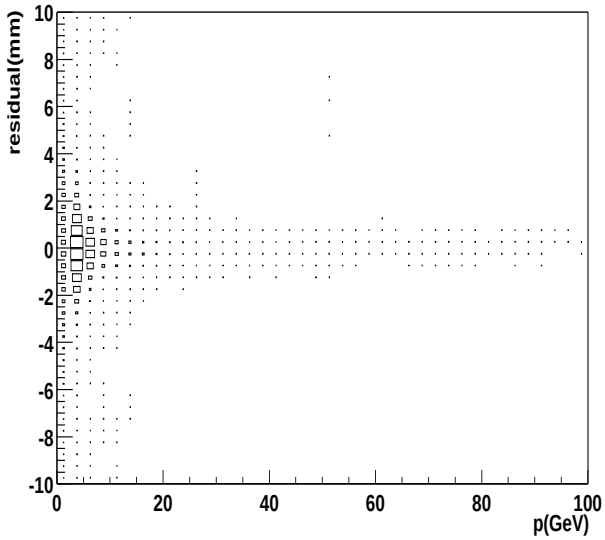

Figure 3: *The residual for all hits in outer tracker in station 4(in mm) vs. momentum.*

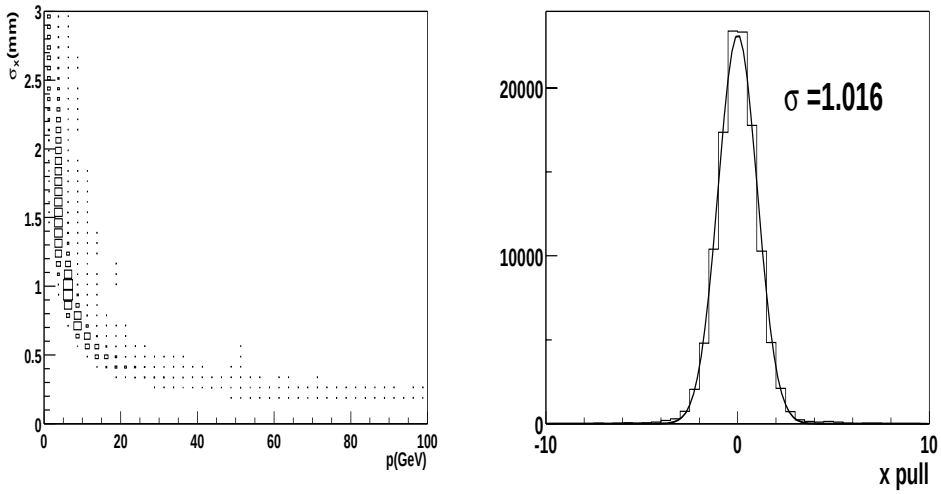

Figure 4: *The left plot shows σ_x vs momentum in GeV for all hits in station 4. The right plot shows the pull distribution for these hits.*

The error in x and y are used to define the size of the RoI. When the errors behave exactly Gaussian, only 5.7×10^{-7} of the hits are expected to have a deviation from the predicted position of more than 5σ.

This, however, is not the case. Table 1 shows the fraction of hits, which have a distance between the predicted state and the closest outer edge of the detection cell, which is larger than $5\sigma_x$. The fact that the outer tracker cells are much bigger than the inner tracker cells explains why the fraction of hits outside $5\sigma_x$ is much larger for the inner tracker hits than for the outer tracker hits.

The numbers in table 1 are obviously worse than expected from Gaussian behaviour. To study this effect a dedicated GEANT run has been done, turning of the non-Gaussian effects in multiple scattering. The fraction of hits outside the 5σ window in that run is halved. The other half of the effect is attributed to wrong estimate of the trajectory due to the finite granularity of the magnetic field map. This is particularly important for low momentum tracks and in regions where the field has a large gradient.

	OT		IT	
Station	$< \sigma_x > (mm)$	$> 5\sigma_x$ x-layer	$< \sigma_x > (mm)$	$> 5\sigma_x$ x-layer
9	0.50±0.00	0.0%	0.50±0.00	0.0%
8	0.81±0.13	0.1%	0.58±0.13	0.9%
7	0.39±0.23	0.2%	0.19±0.14	1.4%
6	0.41±0.27	0.3%	0.18±0.15	0.7%
5	1.20±0.66	0.5%	0.52±0.42	1.5%
4	1.41±0.72	0.7%	0.76±0.57	1.4%
3	4.05±1.42.	0.8%	2.10±0.70	1.0%
2	1.30±0.70	0.8%	0.84±0.57	1.1%
1			1.08±0.70	1.1%

Table 1: *Average resolution for tracks with inner and outer tracker hits. Also shown is the fraction of tracks which are outside a search window 5σ in x. The numbers for station 9 are biased by the cheated track seeding.*

Figure 5 shows $\Delta x - 5\sigma_x$ (Δx is the residual) as function of momentum. The fraction of hits outside the 5σ search window (i.e. they have positive entry in figure 5) is given in table 1.

In order to also include a large part of these hits in the tails, an additional offset on top of a $5\sigma_{x(y)}$ search window is added. The figure shows a momentum dependence, which can be used by making a parametrised function, given in equation 1, which will determine the additional offset:

$$\text{Offset} = \mathrm{C}_1 \frac{1}{(\mathrm{p} + \mathrm{C}_2)^2} \tag{1}$$

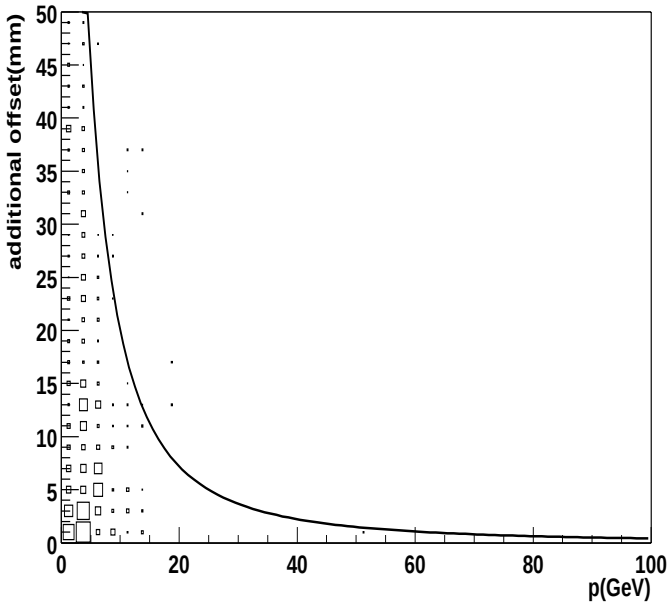

Figure 5: *Additional offset for the OT hits in station 4*

Table 2 summarises the chosen parameters for C_1 and C_2 in the different stations. The last column gives the maximum value of the additional offset, this is done to limit the size, to keep combinatorics reasonable.

Station	C_1	C_2	maximum(mm)
9	5000	12	20
8	5000	12	20
7	3000	2	20
6	4500	5	20
5	4500	5	20
4	4500	5	20
3	5000	6	20
2	3000	2	20
1	1000	05.	20

Table 2: *Additional offset of the region of interest*

3.2 Clustering algorithm

The clustering algorithm uses all hits in the RoI's of each of the layers in a station. The output of the algorithm is a list of possible track continuations.

In a similar notation as in [2], the closest distance between the predicted state and the hit wire (strip) of the OT (IT) is called h. The residual r is calculated according to:

$$r = h - m \tag{2}$$

where m is the actual measurement of the wire (strip). In the case of the outer tracker the left/right ambiguity has not yet been solved and there are two residuals calculated (assuming the measurements $+m$ and $-m$). The residual r differs from 0 due to imperfect measurements or due to a finite resolution of the track position.

For each track prediction the residual values of the correct hits in the search window are expected to group into clusters. Residuals of hits from other tracks will form a background for the cluster search. The clustering is based on two criteria:

- Two hits belong in the same cluster if there difference in residual is smaller than a certain value d_{max}.
- Groups of hits must have a minimum number of hits, N_{min} in order to be called a cluster.

The following geometrical aspects are taken into account:

- For an outer tracker hit it is not possible that both left/right ambiguities are present in the same cluster
- Two inner tracker clusters in the same z plane is impossible.
- At a horizontal module split at y=0, either the upper or the lower hit candidate can be in the cluster, not both.

3.2.1 Determining $\mathbf{N_{min}}$ and $\mathbf{d_{max}}$

The parameters N_{min} and d_{max} are determined as follows:

$\mathbf{N_{min}}$

A track which perpendicularly traverses a 8 layer outer tracker station will on average give rise to

$$\frac{\mathrm{r_{cell}}}{\mathrm{pitch}} * N_{layers} * \epsilon_{cell} \quad (= \frac{5}{5.25} * 8 * 0.965) \quad = \quad 7.3 \text{ hits}$$

For tracks under an angle this number can be higher since two adjacent cells in a layer can be hit.

A track traversing a similar 4 layer inner tracker station will on average generate

$$N_{layers} * \epsilon_{strip} \quad (= 4 * 0.985) \quad = \quad 3.9 \text{ hits}$$

independent of the angle of the track.

Conservatively, for clusters of only inner tracker hits measurements $N_{min} = 2$ is taken, for clusters containing outer tracker hits (or a mix IT + OT), $N_{min} = 3$ is required.

$\mathbf{d_{max}}$

Figure 6 shows the difference in residual between two neighbouring hits in station 5, for hits originating from the same track. Choosing d_{max} too small will give clustering inefficiencies. On the other hand a too large value will increase the combinatorics due to the fact that background hits will be included more easily.

For all stations the value $d_{max} = 700\mu$m is used.

3.3 Cluster quality and selection

The clustering algorithm provides a list of possible track continuations. Prior to fitting these sets of hits to the tracks, each cluster is assigned a *quality factor*. This factor is used to order the cluster candidates and to reject bad ones.

A good cluster will be close to the track prediction and the hits will be lined up in the direction of the track. These properties are used in the calculation of the quality factor.

The average distance is calculated from the hit residuals (see eq 3) in the

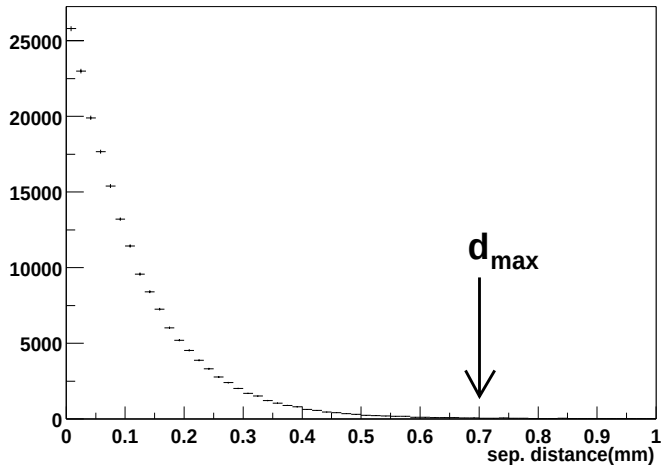

Figure 6: *Residual distance between 2 correct neighbouring hits in station 5*

cluster:

$$R = \frac{1}{N_{hits}} \sum_{i=1}^{N_{hits}} r_i \tag{3}$$

where r_i is the residual value of hit i(see eq 2) in the cluster and N_{hits} is the number of hits in the cluster.

Figure 7 shows the average distance for the clusters which are built from only the correct hits. The distribution is parametrised with the function $f(x; \sigma_1, \sigma_2)$; a double Gaussian centered around zero and normalised to 1. This distribution is used to translate the observed R parameter into a probability that a cluster at distance R belongs to the track:

$$\mathrm{P_1(R)} = 1 - \mathrm{erf}(\frac{|\mathrm{R}|}{\sqrt{2}\vec{\sigma}}) = 1 - \int_{-|\mathrm{R}|}^{|\mathrm{R}|} \mathrm{f(x; \sigma_1, \sigma_2)}\ dx \tag{4}$$

In addition to the average offset the distribution of the hits in the cluster is used in the quality factor. Figure 8 gives a graphical representation of how the hits in a cluster can be distributed around the average distance R, assuming that a track prediction is slightly biased, such that average cluster position is not at 0.

A total χ^2 can be calculated, representing how the hits are distributed in the cluster:

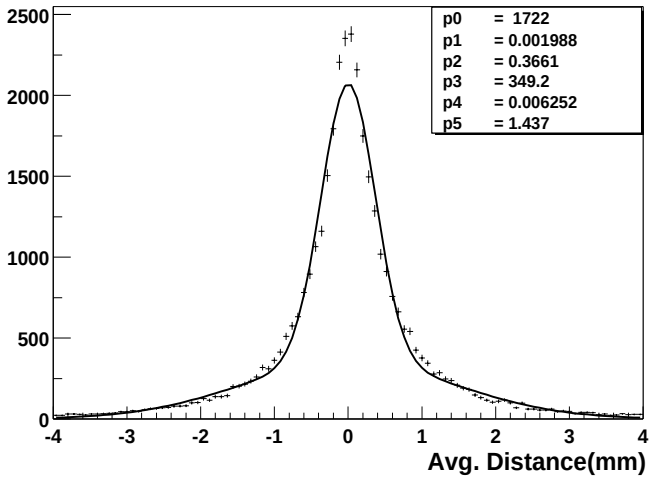

Figure 7: *Average distance of clusters with the correct hits in station 4*

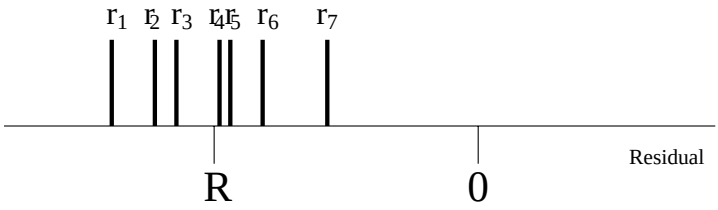

Figure 8: *Graphical representation of a cluster*

$$\chi^2 = \sum_{i=1}^{N} \frac{(r_i - R)^2}{\sigma_{r_i}^2} \tag{5}$$

where r_i is again the residual of hit i, R the average distance of the cluster, and $\sigma_{r_i}^2$ is the measurement error assigned to hit i. A good cluster is expected to have a low χ^2.

The χ^2 is again translated into a probability, with $P_2(\chi^2)$=Prob(χ^2,$N_{hits}-1$) To obtain a flat probability distribution (see figure 9) the measurement errors are multiplied by a fudge factor equal to 1.2. Except for an overall flat behaviour fig 9 contains a narrow peak at 0 probability.

The last variable that is used in the cluster quality factor is the number of hits (N_{hits}), in the cluster. The more hits in the cluster, the more likely it is that this is a correct cluster and not a random combination of background hits.

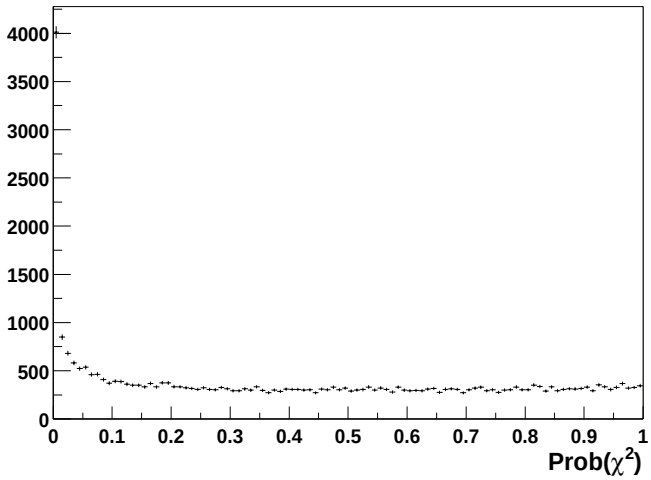

Figure 9: *Probability distribution for the χ^2 of the correct clusters*

These three variables are combined into one quality factor as in equation 6, where both probabilities have equal weight.

$$Q_{cluster} = N_{hits} * (\mathrm{P}_1(\mathrm{R}) + \mathrm{P}_2(\chi^2)) \tag{6}$$

It should be noted that in principle instead of the average, the product of the two probabilities should be taken. However, in that case, the peak at 0 probability in $P_2(\chi^2)$ gives rise to inefficiencies. The procedure of averaging the probabilities gives a better performance and is adopted for now.

All clusters found for a track prediction are ordered on quality. All cluster candidates with a quality less then 70% of the best cluster are rejected. Clusters with higher quality are kept and form track continuation candidates.

3.4 Hit rejection and track continuation quality

Once clusters have been selected based on cluster quality criteria the hits are added to the tracks and new quality criteria, based on the full tracks are defined. At this stage the terminology changes: once added to the track the clusters are now referred to as track continuation candidates.

Individual hits in the clusters are added to the track using the Kalman filter technique. The hits now have a χ^2 assigned with respect to the full track.

(Previously the χ^2 was defined only relative to the cluster). Based on the χ^2 individual hits can be rejected:

- Reject all the hits which have a χ^2 >20
- Refit the track with the remaining hits.
- Reject all the hits which have a χ^2 >16

This procedure rejects many of the background hits.

After hits have been assigned to the track a new quality factor $Q_{candidate}$ is defined for each track continuation:

$$Q_{candidate} = \left[N_{Outer} - w_1 * \sum_{i=0}^{N_{Outer}} \chi_i^2 \right] + 2. \left[N_{Inner} - w_1 * \sum_{i=0}^{N_{Inner}} \chi_i^2 \right] \quad (7)$$

The quality contains a separate sum over the outer and inner tracker hits, where the inner tracker hits have double weight compared to outer tracker hits (related to the average number of expected hits per station, see section 3.2.1). The weight factor w_1 is set to 0.1, implying that a hit with a χ^2 of 10 will have a contribution of zero to the quality.

The track continuation with the highest quality is referred to as Q_{best}. All clusters satisfying $Q/Q_{best} > 0.6$ are accepted as a track continuation. A maximum of 10 continuations per station are allowed. In the rare case that there are more then 10 continuations, only the 10 best ($Q_{candidate}$) are kept.

4 Inter-station pattern recognition

The station-to-station pattern recognition algorithm uses ideas of the concurrent track evolution method developed by HERA-B.

As before the quality of each track candidate is continuously monitored with a quality factor as in eq 7, but now summed over all hits on the complete track. In order to reduce the amount of track branches at each station is kept limited using the following criteria:

- The track parameters of all the candidates must be different. Tracks are declared identical if *all* track parameters are the same within 10σ. This roughly means that they share the same detector elements and

their slope differs by less then a mrad. If two candidates are the same, the one with highest quality is kept, the other candidate is deleted.

- A maximum of 10 best candidates for each *track-seed* can be followed.

In the process described up to now, a track candidate must always find at least one continuation in the next station. There are several reasons why tracks can fail to have continuations in one or more stations:

- The particle can go out of the acceptance (for example passing through the beam pipe)
- The sensitive part of the detector was inefficient for a track, either due to a detector problem, or due to screening of hits by other tracks.
- The clustering algorithm can fail to find a cluster satisfying the selection criteria

The inter-station algorithm is designed such, that it can cope with these losses. In the algorithm a track candidate is declared to be either: "Alive", "Zombie" or "Dead":

- A track is "Alive" when there are hits added in the previous station
- A track is "Zombie" when there are no hits added in the previous station but there are added hits in the station prior to that.
- A track is "Dead" when there are no hits added in the two previous stations.

It should be remarked that in case no "alive" candidates exist for a given track seed, the "zombie" candidates are tried to be revived. Once declared dead, however, a track candidate cannot be revived: the particular branch in the pattern recognition ends at this point.

4.1 Final selection of tracks

When the whole detector is traversed each initial seed will have several possible track candidates, either alive, zombie or dead. The best candidate from the tree will be selected, which is the track with the highest quality as specified in equation 7. All the hits from the outer and inner tracker on the track are considered.

It is possible that there is more than one "acceptable" track is found for a single seed. Besides the best candidate, also additional candidates are selected, if they satisfy two criteria:

- The quality of any acceptable candidate has to be within 95% of the best candidate: i.e. $Q/Q_{best} > 0.95$, and
- For two tracks to be both selected they must have at least have 15% different hits. In practice, the following formula is used:

$$\frac{N_1 \cup N_2}{min(N_1, N_2)} < 0.85 \tag{8}$$

  Where N_1 and N_2 are the number of hits on the 2 tracks and $N_1 \cup N_2$ is the number of hits they share.

5 Performance Indicators

For a track to be efficiently found, a large fraction of the hits must be found, very few background hits should be on the track and the track parameters must be correct.

To test the performance, two sets of tracks will be reconstructed each event. For one set ideal pattern recognition is assumed, which simply implies that all hits are correctly assigned to tracks. The second set of tracks is built using the actual real track following algorithm. In both cases the track states are calculated up to station T1 using the same Kalman filter method.

A track found with the pattern recognition is attributed a hit efficiency and a hit purity in comparison to the cheated tracks.

- The hit *efficiency* is defined as the fraction of correct hits on the pattern recognition track with respect to the number of hits on the cheated track:

$$\epsilon_{eff} = \frac{N_{pr} \cup N_{ch}}{N_{ch}} \tag{9}$$

- The hit *purity* is the fraction of correct hits on the pattern recognition track with respect the total number of hits on this PR track:

$$\epsilon_{pur} = \frac{N_{pr} \cup N_{ch}}{N_{pr}} \tag{10}$$

The trackfinding effiency and ghost rate is calculated by "matching" pairs of reconstructed and cheated tracks. For each cheated track, the pattern track with the highest purity is declared to be the "match".

A cheated track is considered as efficiently found in the pattern recognition if:

- both the hit efficiency and hit purity of this best match are higher then 70%. This also taking into account the left/right ambiguities in the outer tracker.
- each of the reconstructed track parameters (x, y, t_x and t_y) at the upstream surface of station 2($z = 2111$mm) are the same within 10σ of the true track parameters; i.e.:

$$\frac{\Delta i}{\sigma_i} < 10 \quad \text{with } i = \ x,\ y,\ t_x \ and\ t_y \tag{11}$$

5.1 Ghosts and clones

Ghosts are reconstructed tracks which cannot be assigned to any true track. In practice, in this note the ghost rate is defined as:

$$\epsilon_{gh} = \frac{N_{pr.\ not\ matched}}{N_{pr}} \tag{12}$$

Where $N_{pr.\ not\ matched}$ is the number of not matched tracks, and N_{pr} is the total number of reconstructed tracks in the track following.

Ghosts can occur both in the seeding algorithm and in the following algorithm. A ghost seed results in an initial track state with for which no corresponding MC true state exists. The ghost rate in the seeding algorithm is about 10% [1], however it remains to be studied which fraction of these ghosts are killed in the track following algorithm.

In the trackfollowing algorithm ghosts can occur due to the fact that a reconstructed track is inefficient, when one of the matching criteria is not satisfied (see prev section), or when more then one track is reconstructed from a single seed.

Clones are multiple reconstructed tracks which match to the same MC true track. As an example, consider the (extreme) case of two reconstructed tracks which have all hits but one in common. These two tracks could be clones. In this study the clone rate is suppressed by criterium in equation 8, which in practice reduces the clone rate to zero. In practice the requirement means

that for two track clones the best quality track is kept, the other track is rejected.

6 Performance

The performance of the track following algorithm is presented for the baseline LHCb detector layout of 9 tracking stations, see fig 1. The simulated beam pipe corresponds to the aluminium-beryllium alloy pipe design.

In the subsequent section of this note the performance is reported for varying detector parameters; beam pipe,luminosity, station thickness, resolution, size, orientation, etc. For each study a sample of 1000 inclusive B events is used, piled up with minimum bias events corresponding to a luminosity of $5{\cdot}10^{32}\mathrm{cm}^2\mathrm{s}^{-1}$).

The track following algorithm is optimised for efficient reconstruction of what we call *physics tracks*. Physics tracks are expected to represent typical tracks originating from the primary *pp* collision.

In this note the efficiencies are calculated for two classes of tracks. The track from both classes must have at least one vertex detector hit, and additionally:

- Type 1 : Tracks which have main tracker hits before z_{first} <1200mm (in T1) and with hits after z_{last} >9000mm(in T9). These are considered the "cleanest" physics tracks.
- Type 2 : Tracks with main tracker hits before $z_{first} < 3600$ mm (in or before T3) and with hits in the seeding region: z_{last} >7500mm (in or after T6).

Table 3 shows the observed average number of tracks per event for each of these types and their average momentum. Electrons are shown separately.

Type	$< \frac{nr.tracks}{event} >$	$< p >$(GeV)
1	22.1	9.37
2	36.6	12.36
electrons		
1	0.48	7.58
2	0.97	8.49

Table 3: *Properties of track types samples*

It should be kept in mind that the inner angular acceptance boundary of station T1 is 25 mrad. This implies that the requirement of having a hit in T1 selects tracks of relatively large angle. This in turn means that the average momentum for this sample of tracks is relatively low, as can be seen in the table.

6.1 Hit efficiency and parameter resolutions

For these two types of tracks the track following algorithm is applied. Figure 10 shows for all matched tracks the hit efficiency, hit purity and the absolute value of the four individual pull contributions. It can be seen that the matching criteria defined in section 5 are chosen conservatively.

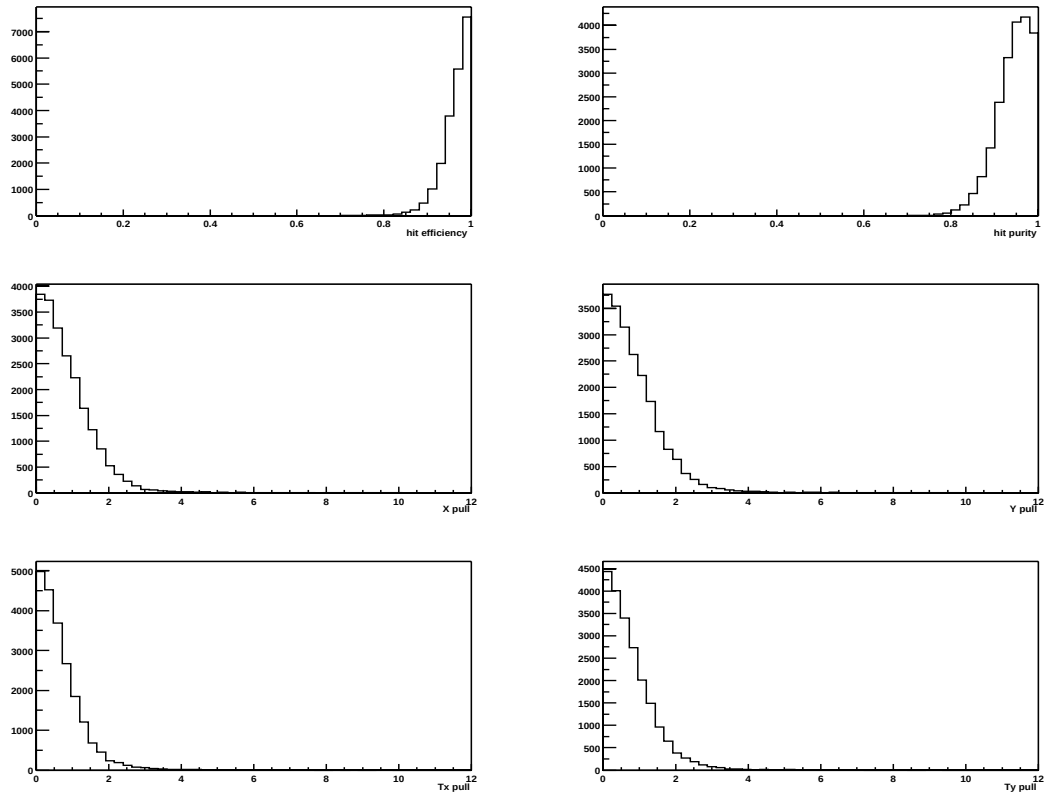

Figure 10: *Matching criteria. Top left: the hit efficiency. Top right: the hit purity. The remaining four plots are the individual pull distributions for x, y, t_x and t_y*

Figure 11 shows that the parameter resolutions of the pattern recognition tracks are similar to those of the cheated tracks.

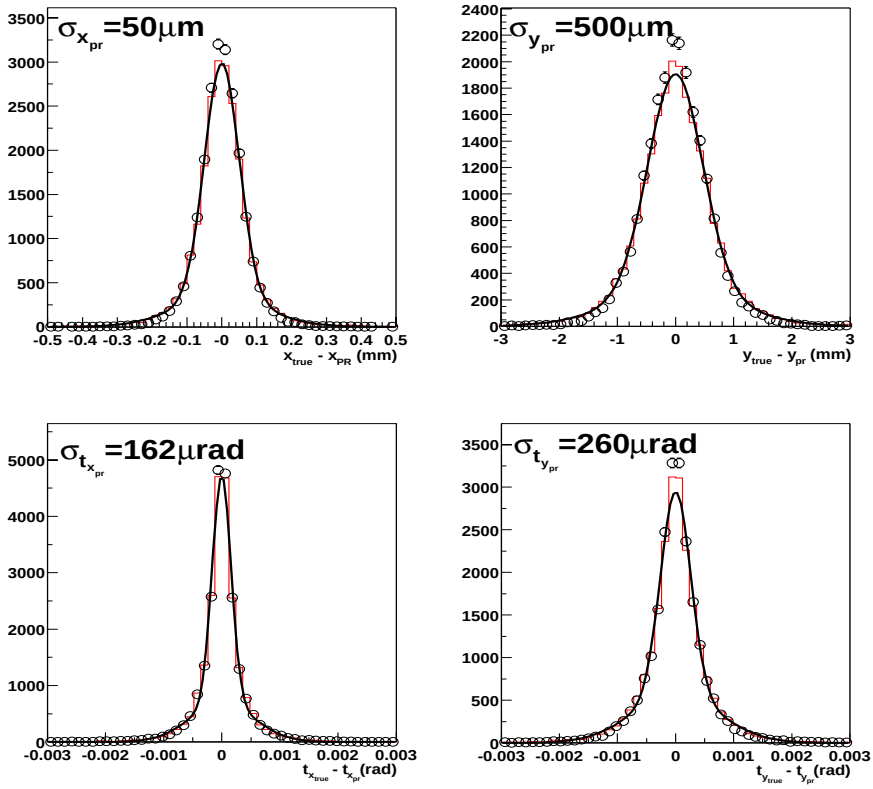

Figure 11: *Track parameter resolutions for the found tracks(solid line). The ∘ show the resolution for these tracks when the track following is cheated*

6.2 Track efficiency

Figure 12 shows the efficiency as function of momentum for tracks of type 1. As expected, the reconstruction efficiency for low momentum tracks (<10GeV) is somewhat lower than for high momentum tracks. The integrated reconstruction efficiency for all these tracks is 94.5% while for track with momentum larger than 10GeV, the efficiency is 97.1%. Table 4 shows the reconstruction efficiency for the different type of tracks.

Type	Efficiency(%)	ghost rate(%)
1	$(94.5 \pm 0.15)\%$	$(7.5 \pm 0.13)\%$
2	$(92.1 \pm 0.13)\%$	

Table 4: *Efficiencies for the different track types*

The track following efficiency not only depends on the momentum of the

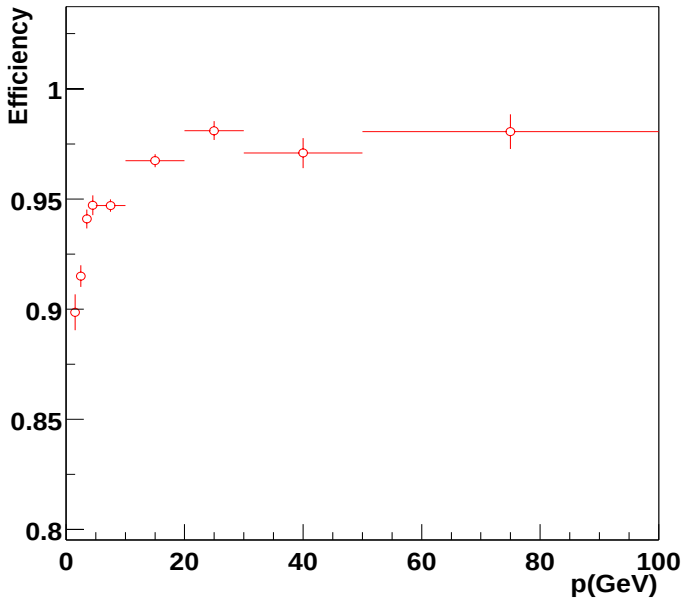

Figure 12: *Efficiency as function of momentum.*

tracks, but obviously also on the total number of hits present in the detector, since more hits increase the number of combinatorics and the probability to assign wrong hits. Figure 13 shows the dependence of the efficiency versus the total number of outer and inner tracker hits in the detector. Fitting a linear dependency to the data gives the following relations between the number of hits and the efficiencies:

$$\begin{aligned} \text{OT}: \quad \epsilon &= 97.1\% - \frac{4.0\%}{10.000\text{hits}} \\ \text{IT}: \quad \epsilon &= 97.0\% - \frac{10.2\%}{10.000\text{hits}} \end{aligned} \tag{13}$$

The efficiency for events with very few hits is close to 100% (∼97%). Only some very low momentum tracks fail to be found even in low hit densities.

6.3 Ghost rate

The definition of ghost rate is given in equation 12. It corresponds to the number of tracks which are not assigned to a true track, using the matching criteria from section 5.

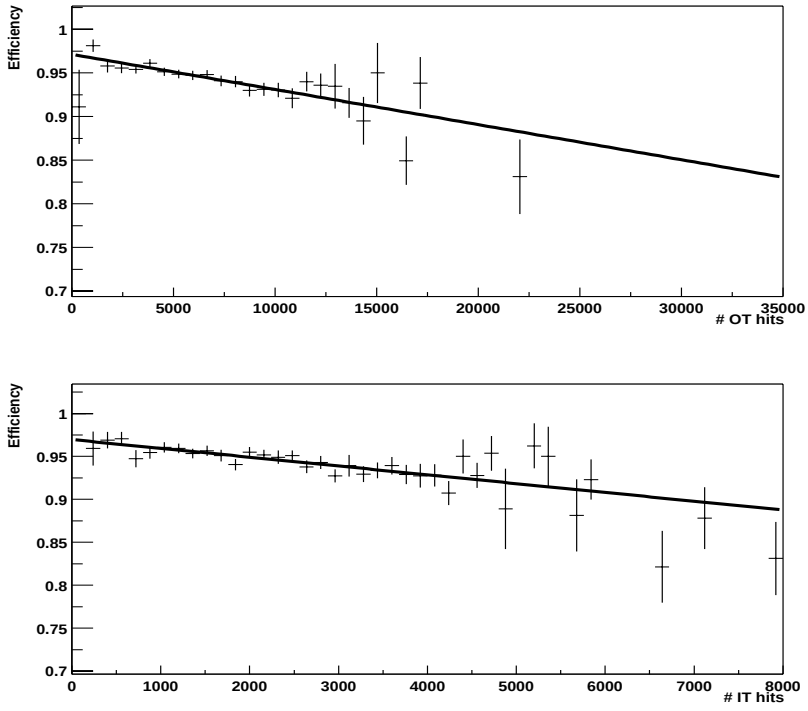

Figure 13: *Efficiency as function of the total number of outer(top) and inner(bottom) tracker hits*

The ghost rate is related to the amount of combinatorics, thus one expects a rising ghost rate versus the number of hits in the detector. Figure 14 shows the ghost rate plotted versus the total number of outer and inner tracker hits in the event. Integrated over the whole spectrum, the ghost rate is 7.5%.

The track following ghost rate can be optimised versus the efficiency by varying the minimum track quality criterion (see equation 7) of the tracks. Asking for a higher quality reduces the ghost rate, but also the track following efficiency.

Figure 15 shows how the ghost rate versus efficiency when this cut is applied. This is done for a Q_{min} of 0., 30., 40., 50., 55. and it is shown for tracks of type 1 and 2.

Setting the cut on the quality to 40 instead of no cut, will about half the ghost rate, going from 7.5% to 4.6%. On the other hand the efficiency for tracks of type 1 will only go from 94.5% to 94.3% and for the tracks of type 2, the efficiency goes from 92.1% to 89.9%.

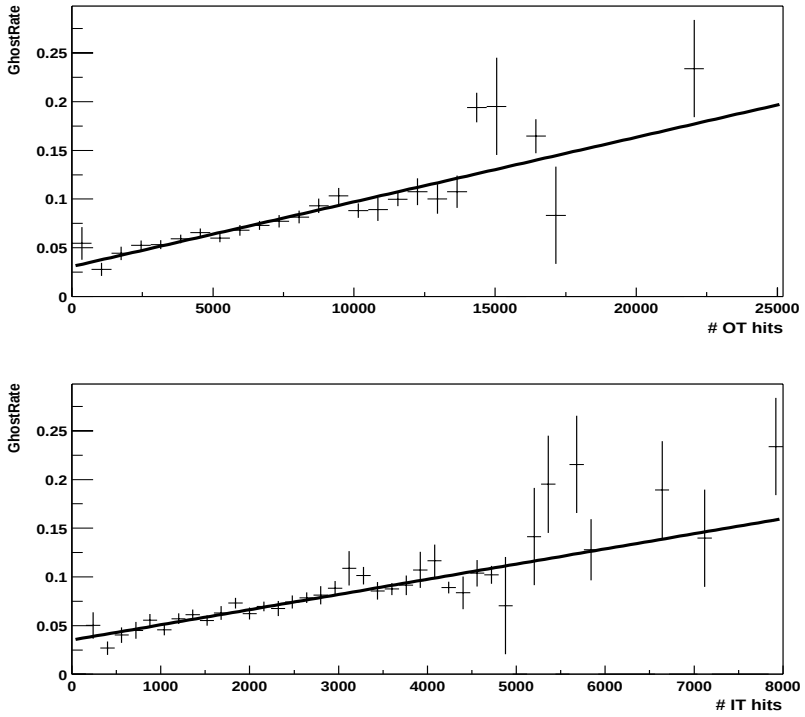

Figure 14: *Ghost rate versus the number of outer and inner tracker hits*

6.4 Electron reconstruction

Electrons are subject to photon bremsstrahlung when they traverse a material of sufficient radiation thickness. If a hard photon is radiated inside the magnetic field region track following might become difficult due to a sudden change in curvature of the track.

Figure 16 shows the momentum dependence of the electron following efficiency for electrons of type 1. The integrated efficiency is 65.2% $\pm$ 2.1%, indeed significantly worse than for non-electron tracks.

7 Optimization studies

The track following algorithm is exercised on various geometry and detector settings. These studies serve partly to understand the sensitivity of the algorithm and also to help in detector optimization.

These studies are done with different samples of events compared to the

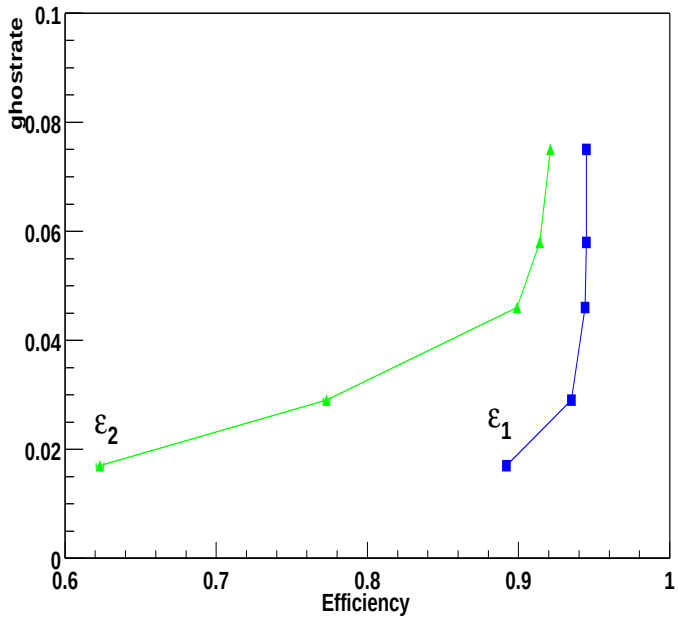

Figure 15: *Efficiency vs. ghost rate for two types of tracks. The ghost rate is reduced by using different cuts on minimum quality, respectively, 0., 30., 40., 50. and 55.*

previous performance studies. One of the main differences is that the default radiation length per station is 2% compared to 3% before. Many other things are changed to optimize the detector and the algorithm.

- Changing the occupancy in different ways:
  - Lowering the luminosity
  - Changing the material of the beampipe
  - Increasing the size of the inner tracker
- Changing the stereo angle.
- Changing the amount of material in the experiment.
- Study the effect of decreased resolution in the inner and outer tracker.

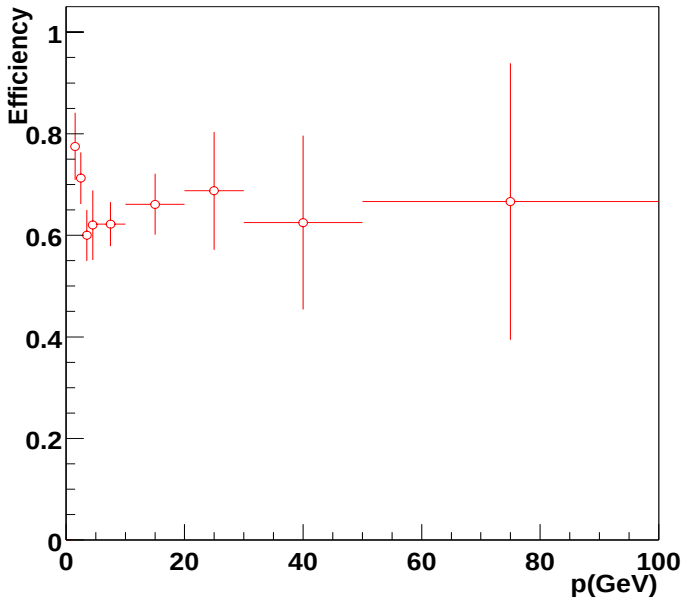

Figure 16: *Efficiency as function of momentum for electrons*

7.1 Luminosity

The number of tracks (and thus also hits) in an event is obviously of crucial importance for pattern recognition performance (see also fig 13 and 14).

As a standard setting all results have been produced assuming high luminosity mode running, i.e. including the pile-up hits from the additional minimum bias interactions in the bunch-bunch interaction. Table 5 compares the performance of high luminosity ($\mathcal{L} = 5 \cdot 10^{32}\text{cm}^2\text{s}^{-1}$) running with that of nominal ($\mathcal{L} = 2 \cdot 10^{32}\text{cm}^2\text{s}^{-1}$) luminosity running. A difference of 1% in efficiency and ghost rate is observed.

$\mathcal{L}$	<# type 1 tracks>	efficiency	ghost rate
$2 \cdot 10^{32}\text{cm}^2\text{s}^{-1}$	18.4	$(95.5 \pm 0.15)\%$	$(7.1 \pm 0.14)\%$
$5 \cdot 10^{32}\text{cm}^2\text{s}^{-1}$	23.4	$(94.4 \pm 0.15)\%$	$(8.1 \pm 0.15)\%$

Table 5: *Track following performance for nominal luminosity and high luminosity*

7.2 Beam pipe design

In addition to the number of tracks of the primary interaction the event reconstruction efficiency can be endangered if there are many *secondary tracks* present in the event. It was realised (see e.g. [8]) that a potentially dangerous source of secondaries is the beam pipe. The pattern recognition performance has been tested for 6 different beam pipe designs, based on the following materials:

- Beryllium-Aluminium alloy (current default solution) (pipe v.7.1)
- Aluminium (pipe v.4.1)
- Aluminium with stainless steel flanges (pipe v.4.0)
- Beryllium beampipe with Aluminium flanges (pipe v.4.2)
- Full Beryllium (pipe v.5.0)
- "Air" pipe (no pipe)

Note that last option ("air" pipe) is purely added in order to compare the performance with and without the presence of a beampipe in the setup. Also it should be mentioned that the "full" Beryllium option is not based on a realistic technological design.

Table 6 shows the efficiency and ghost rate for tracks of type 1 and 2 with the different kinds of beam pipe.

	Type 1	Type 2	
Beam pipe	Efficiency(%)	Efficiency(%)	ghost rate
Al +SS	$(91.0 \pm 0.21)\%$	$(86.2 \pm 0.20)\%$	$(11.8 \pm 0.20)\%$
All Al	$(92.7 \pm 0.16)\%$	$(90.1 \pm 0.15)\%$	$(9.9 \pm 0.45)\%$
Be + Al	$(94.2 \pm 0.15)\%$	$(91.8 \pm 0.14)\%$	$(8.7 \pm 0.15)\%$
Al-Be	$(94.4 \pm 0.15)\%$	$(92.2 \pm 0.15)\%$	$(8.1 \pm 0.15)\%$
Be	$(94.3 \pm 0.15)\%$	$(92.5 \pm 0.13)\%$	$(7.9 \pm 0.14)\%$
Air	$(95.1 \pm 0.15)\%$	$(93.3 \pm 0.10)\%$	$(7.1 \pm 0.12)\%$

Table 6: *Efficiencies for the different beam pipe designs. They are ordered from worst to best*

As expected, the more "massive" designs give a lower efficiency and a higher ghost rate.

7.3 Radiation thickness

The amount of radiation length present in the LHCb spectrometer is of crucial importance for physics performance. There are two effects which can lead to a reduced overall efficiency for reconstructing B decays:

- tracks might interact with detector material and thus fail to traverse the complete spectrometer (governed by the interaction length λ_I)
- tracks might multiple-scatter in the detector and might be lost in the recognition algorithm (governed by the radiation length X_0)

To study both these effects four separate detector simulations have been done, respectively with tracking detector thicknesses equivalent to: 1% X_0, 2% X_0, 4% X_0 and 8% X_0 per station.

In these studies the simulated beam pipe was of the full Aluminium type, i.e. not with the (now baseline) Al-Be alloy pipe. However, the effect on particle absorption will be independent on the pipe, as are the relative comparisons of track following efficiencies.

7.3.1 Track absorption

In [9] a more detailed description is given of the causes of losing tracks in the detector. Table 7 shows the number of tracks per event and the average energy per track, when the radiation length per station is 1%, 2%, 4% and 8%. The left plot of figure 17 represents these numbers in a graphical way. For presentation purposes the number of electrons per event is multiplied by a factor of 10.

In the figure one sees that the loss of hadrons is much more significant than the loss of electrons. The amount of hadrons lost due to hadronic interactions in the material are proportional to the interaction length(λ_I) traversed. If there is a hadronic interaction, the track is lost. Electrons loose energy in bremsstrahlung or delta ray events. However, the relative amount surviving (with different energy and momentum) is higher. The latter will be strongly felt in pattern recognition.

7.3.2 Recognition efficiency

The resulting track following efficiency, assuming that a tracks has traversed (and survived!) the whole spectrometer (i.e. tracks of type 1), is shown in

Type	X_0(per station)	$< \frac{nr.tracks}{event} >$	$< p >$(GeV)
1	1%	24.0	8.43
	2%	23.3	8.59
	4%	22.1	8.68
	8%	17.6	8.66
electrons			
1	1%	0.63	9.44
	2%	0.62	9.28
	4%	0.57	9.29
	8%	0.49	10.13

Table 7: *Properties of track types samples*

the right side plot of figure 17 for each of the detector radiation thicknesses. One can see that in this case the losses of efficiency for electrons are higher than those of the hadrons.

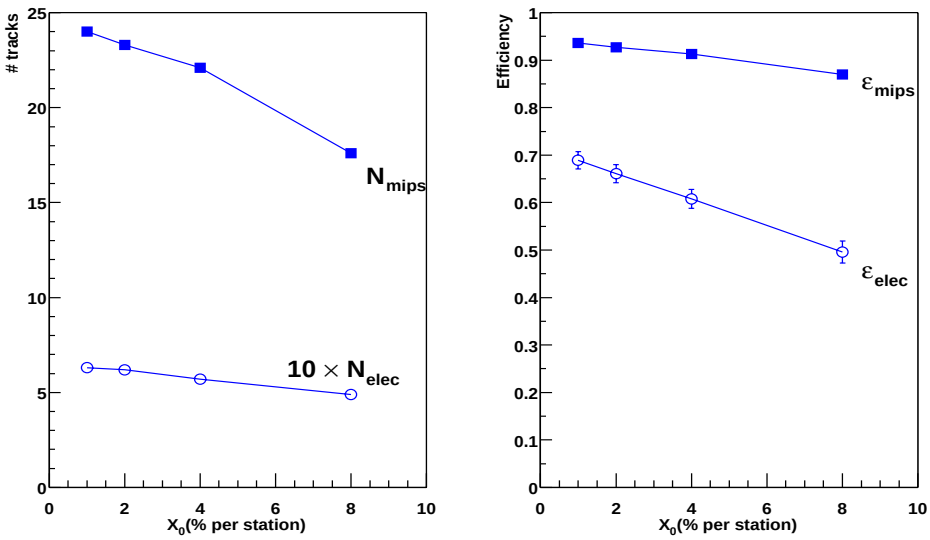

Figure 17: *Number of hits per event and efficiency as function of* X_0

7.3.3 Overall effect

Both the number of tracks as well as the efficiency are affected by the radiation length in the experiment. The effective overall efficiency is equal to

the product of these two effects. In order to illustrate the dependency of the overall efficiency versus the amount of material in the tracking stations let us renormalize the track absorption factor to 0 for an LHCb set-up with 1% X_0 tracking stations. The reduction in number of physics tracks present with other tracking thicknesses are normalised to this 1% case. In other words, the point at 1% X_0 contains only the efficiency due to track following reconstruction. Figure 18 then shows then shows the total efficiency, calculated as the product of the normalised number of physics tracks per event and the reconstruction efficiencies. Again, this is done for both hadrons and electrons. For hadrons the efficiency ranges from 94.4% to 65% and for electrons from 71% to 40% when going from 1% to the 8% of a radiation length. Interestingly, the slope of the hadron and electron curves are similar, however the individual effects are very different.

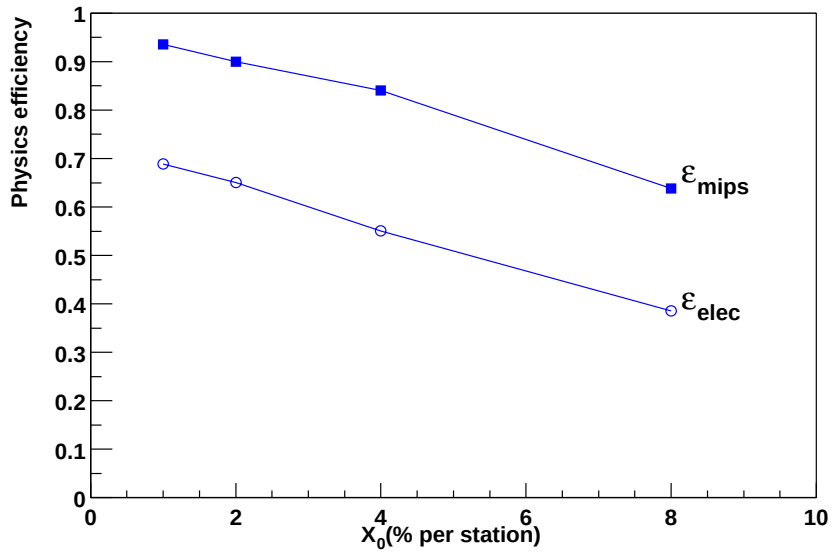

Figure 18: *Physics Efficiency for Mips and electrons for different radiation lengths per station*

7.4 Stereo angle

The measurement planes under a stereo angle will provide a measurement of the y coordinate of the tracks. Larger stereo angles thus will result in better precision. On the other hand, larger angles lead to larger combinatorics and might result into higher ghost rates and lower efficiencies.

In figure 19 the improvement in resolution of the track parameters y and t_y

at the entry of station T2 is shown when increasing the stereo angle.

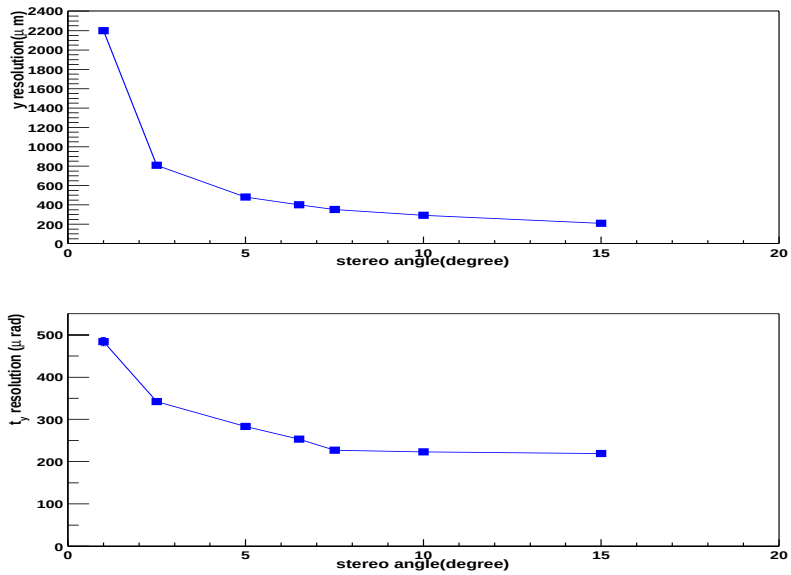

Figure 19: *Resolution in y and t_y function of stereo angle*

Figure 20 shows the dependence of the track following efficiency and ghost rate for stereo angles between 1 and 15 degrees. The plots indeed show a degradation in efficiency and increase in ghost rate for stereo angles larger then 5 degrees. For stereo angles less then 2 degrees the performance also seems to become worse, due to rather bad resolution in the y direction.

7.5 Outer tracker occupancy and inner tracker size

The detector occupancy is defined as the fraction of channels that are hit in an event. The occupancy is proportional to the number of tracks in the event which affects the pattern recognition as is shown earlier.

Another way of reducing occupancy is to increase the detector granularity. The inner tracker has a much smaller channel pitch as compared to the outer tracker, and a much lower occupancy. The overall detector occupancy can be reduced by increasing the size of the inner tracker as a larger inner tracker reduces the occupancy in the outer tracker.

Due to the single hit readout of the outer tracker electronics detector channels a high occupancy leads to detector inefficiency. Once a channel is hit by a track a second hit of a different track cannot be observed. This is referred

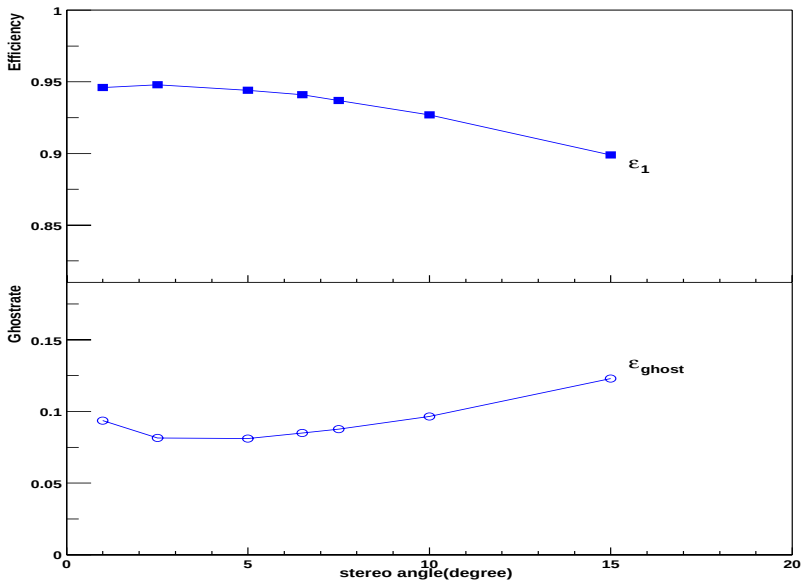

Figure 20: *Efficiency as function of stereo angle*

to as "hit masking". As a consequence of this hit masking, the individual detector channel inefficiency is roughly equal to the occupancy. The track following performance of has been simulated for three different rectangular inner tracker sizes ($x \times y$): $60{\times}40\text{cm}^2$, $80{\times}60\text{cm}^2$ and $120{\times}80\text{cm}^2$. In [8] one can find the effect on Outer Tracker occupancy in each of these cases. The resulting efficiency for the track following procedure given in table 8.

IT Size	efficiency	ghost rate
$60{\times}40\text{cm}^2$	$(94.4 \pm 0.16)\%$	$(8.1 \pm 0.15)\%$
$80{\times}60\text{cm}^2$	$(95.3 \pm 0.14)\%$	$(7.7 \pm 0.13)\%$
$120{\times}80\text{cm}^2$	$(95.6 \pm 0.13)\%$	$(7.2 \pm 0.13)\%$

Table 8: *Track following performance for different sizes of the inner tracker*

7.6 Resolution

Current testbeam results show that the the expected resolution for the outer tracker will be 200μm while for the inner tracker about 80μm is expected.

However, it can happen that in the full-scale experiment, this design resolution is not met, for example in the first running period. Here, the degradation

in track following efficiency is studied for the case that the detector resolution will be twice or even four times worse.

In table 9 the track following efficiencies for these reduced resolutions are summarized. The efficiency for a factor of two worse resolution is still acceptable, while the efficiency for a factor four worse resolution would clearly be unacceptable.

OT/IT resolution	efficiency	ghost rate
200/80μm	$(94.4 \pm 0.16)\%$	$(8.1 \pm 0.15)\%$
400/160μm	$(90.7 \pm 0.19)\%$	$(11.9 \pm 0.16)\%$
800/320μm	$(48.3 \pm 0.30)\%$	$(47.2 \pm 0.25)\%$

Table 9: *Track following performance for different resolutions of inner and outer tracker*

Algorithm Retuning

It should be stated that in this study the parameters of the track following algorithm have not been retuned. Especially the parameter d_{max} (see section 3.2.1) should be retuned in case the resolutions become much worse than anticipated.

Increasing d_{max} from 700 μm to 1 mm results in an increase of efficiency from 48.3% to 66.2%.

An additional modification to the algorithm (in the case of worse detector resolution) could be the way the left/right ambiguity in the outer tracker is treated. In the standard algorithm a wrong left/right hit assignment is counted as a bad hit. Clearly, in case the resolutions are 800 μm, a correct left/right choice is getting very difficult.

In the case of a four time worse resolution, ignoring the left/right ambiguity of straw hits in the efficiency definition, the efficiency goes up from 48.3% to 81.0% and reduces the ghost rate from 47% to 21%.

It is to be expected that a combination of the above mentioned improvements will give even al larger improvement in efficiency.

8 Conclusions

The first implementation of a track following algorithm in LHCb has been described in this note. Also many optimisation studies have been done to test the performance of the algorithm as well as the detector performance.

Assuming ideal seeding efficiency, a track originating from the vertex and going through station T1 and T9 can be reconstructed with an efficiency of 94.4%, of these, tracks with momentum >10GeV are reconstructed with an efficiency of 97%. A 90% efficiency is obtained for all tracks originating from the vertex and reaching the seeding region(T6). The reconstruction results in a ghost rate of 4.6%.

Electron reconstruction in LHCb is difficult due to the presence of material in the spectrometer. The track following efficiency for electrons is 65%. It must be noted that no dedicated study has been done for the electron reconstruction and future studies will improve the performance.

The amount of material seen by tracks is important for the experiment. An increased amount of material in the tracking station will absorb tracks due to λ_I, increase the multiple scattering and the amount of bremstrahlung of electrons is much higher.

The number of tracks/hits in the detector is of importance for the performance of the pattern recognition. An increase of the number of primary tracks in the experiment by increasing the luminosity, or the number of secondary tracks by e.g. a different beampipe, will lower the efficiency and increase the ghost rate. Reducing the occupancy by increasing the inner tracker size can compensate part of these losses.

In the last part the effect of stereo angle and resolution is studied. This results in an optimal stereo angle of 5^0 from the point of view of track following.

A factor of 2 worse resolution will reduce the efficiency, but not kill the track following performance. However, the algorithm requires some retuning to regain part of the losses.

References

[1] R.Forty, "Track seeding", LHCb 2001-109.

[2] R. Mankel, "RANGER- A Pattern Recogntion Algorithm for the HERA-B Main Tracking System", HERA-B note 97-082.

[3] R. van der Eijk *et al*, "Track Reconstruction for LHCb",LHCb 1998-45

[4] LHCb Outer Tracker Technical Design Report, CERN/LHCC 2001-024, 14 September 2001.

[5] R. van der Eijk *et al*, "Performance of the LHCb OO Track Fitting Software",LHCb 2000-86

[6] A. Polouektov *et al*, "First results from LHCb inner tracker performance studies using new digitization software", LHCb 2001-118

[7] M. Merk *et al*, "An improved digitisation procedure for the outer tracker", LHCb 2001-055

[8] R. Hierck, M. Merk and M.Needham, "Outer Tracker occupancies and detector optimization", LHCb 2001-093

[9] R. Hierck, M. Merk and M.Needham, "Particle interactions in the LHCb detector", LHCb 2001-090