Assessments

Renato Quagliani

June 10, 2015

1 OVERVIEW OF WORK DONE

A description of the work done up to now will be described. It consists in two major blocks. The first one is related to the analysis work and it has been done between June 2014 and October 2014 , period in which i was finishing the master degree in Ferrara Unviersity and i get the NPAC master 2 at Paris Sud University. The second one is related to the work done in the context of the preparation for the upgrade of the LHCb detector consisting in the improvements of the *Seeding* tracking algorithm which actually becomes the development of new algorithm , the *Hybrid Seeding*. Before discussing the work done in this domain a small introduction about the tracking and the upgrade of the *LHCb* detector is mandatory.

1.1 INDTRODUCTION TO LHCB AND TO THE UPGRADE SCINTILLATING FIBRE TRACKER (SCIFI).

The *LHCb* detector [1] aims at search indirect signatures of New Physics through quantum loop induced processes through the measurements of strongly suppressed Standard Model processes. In this context precision plays a fundamental role. Up to now, in *LHCb* and other *LHC* experiments, no strong deviations from the Standard Model (SM) have been observed and even if *LHCb* has already provided the world's best measurements in some channels, it is still limited from improving them by statistics rather than systematics. Due to this, the upgrade of the *LHCb* forward spectrometer is mandatory in order to collect *O(100)* times more data and reduce the statistics uncertainty of a factor 10 in order to be comparable with the theoretical one [2] [3].

The *LHCb* detector is designed to be a single-arm forward spectrometer aiming to detect particles and their decay products, and as the *b* in the name of the detector suggests, it is designed to study particles containing *b* and *c* quarks which are produced strongly boosted in the forward and backward (lost) directions for symmetric energies of colliding protons.[1] The detailed description of the *LHCb* detector can be found in [1]. The data taking at *LHCb* during 2011 and 2012 at LHCb are mainly determined by few steps:

- Interesting events are selected by the *L0 Trigger* which is implemented at the hardware level aiming to reduce the 40 *MHz* bunch crossing rate to 1*MHz* making use of estimations and measurements of the signature of particles having high E_T, p_T through the muon stations and the calorimeters. The main reason why this is done is because the read-out system of some of the detector for Run-I and Run-II can afford an incoming rate of 40*MHz*. In the upgrade infact, all the read-out will be substituted and the *L0* trigger will be replaced by a software one.

- *High Level Trigger*: It consists in an *Online* software trigger where the full reconstruction of tracks is performed. It's at this level that the tracking algorithms are run. After this step , data are stored and an *Offline* reconstruction is also performed before providing usable object for data analysis. [2]

- *LHCb* luminosity is kept constant and it's reduced wrt other *LHC* experiments of 2 orders of magnitudes. This reduction in luminosity is achieved thanks to the *Luminosity Levelling* mechanism which avoid head-on collisions separating beams perpendicularly to the collision plane. This reduction of luminosity is mandatory since the main studies on *b* and *c* hadrons require an extraordinary precise reconstruction of the production vertex of the $b\bar{b}$ pairs, so, the VErtex LOcator (*VELO*) can be placed at very small distance from the interaction point limiting problems coming from radiation damages. [3]

Regarding the upgrade, a brief description on what it consists is mandatory. A small recap is given in table 1.1 where *v* stands for the average number of visible interaction per bunch crossing.

---

[1]The relevant process infact is the quark-gluon fusion which can be obtained in proton proton collisions only with strongly asymmetric PDFs.

[2]During the Run-I, the seeding algorithm (called *PatSeeding*) in the *HLT* was run making use of the left-over hits coming from the *Forward* algorithm. During Run-I the *Seeding* was used in the online reconstruction in tandem with the *Forward* as described before while in the offline reconstruction it was run as a *Standalone* algorithm. For the Run-II online and offline reconstruction gives more or less the same results, but, anyway, the seeding algorithm remains one of the most important one.

[3]The decrease from the maximal designed luminosity of *LHC* of 10^{34}cm$^{-2}s^{-1}$ to the *LHCb* one 10^{32}cm$^{-2}s^{-1}$ permit to reduce the average number of inelastic collisions from 27 to 0.53 and it allows to reconstruct with extraordinary precision the primary vertices.

	Current *LHCb*	Upgrade *LHCb*
Trigger	Hardware(*L0*) + Software(*HLT*)	Only Software
Read-Out rate	*L0*: 40 *MHz* to 1*MHz* for readout	40 *MHz* Full software trigger for every 25 ns bunch crossing
Luminosity	$4 \times 10^{32} \mathrm{cm}^{-2} s^{-1}$	$2 \times 10^{33} \mathrm{cm}^{-2} s^{-1}$
ν	2	7.6

Table 1.1: Main differences between actual LHCb and the Upgrade

The reason why the *L0* trigger will be removed in the upgrade is mainly because lot of analysis looking for deviations from Standard Model at *LHCb* are limited by statistics and a fixed 1 *MHz* readout at the upgrade running condition will be too limiting. In order to reach the physics goals the *LHCb* detector will be upgraded and the installation of the new detectors and the new read-out is expected to happen during the long shutdown 2 in 2018/2019. Each subdetectors will be replaced as shown in Fig. 1.1.
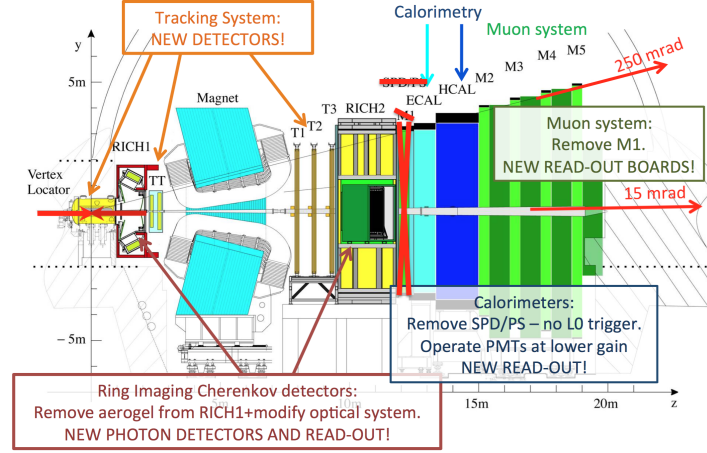

Figure 1.1: LHCb detector modifications for the upgrade. The Vertex Locator (*VELO*), the Tracker Turicensis(*TT*) and the 3 T-Stations are the main resposible of the tracking at LHCb. The Particle Identification is ensured by the two ring cherenkove detectors (*RICH1* and *RICH2*). The particles energy measurement is provided by the hadronic and the electromagnetic calorimeters (HCAL and ECAL respectively). The properties of the muons are then mainly determined by the Muon System.

The actual *VELO* will be replaced by a lightweight hybrid pixel detector capable of 40 *MHz* readout at the upgrade luminosity which is 5 times greater than the actual one. The trackers downstream and upstream the magnet will be also completely replaced. In particular, the tracker downstream the magnet which is now made by the so-called Inner tracker (*IT*) and the Outer Tracker (*OT*) will be replaced by the scintillating fibre tracker (*SciFi*) detector to cope for the higher luminosity, higher occupancy and the 40 *MHz* read-out.

The current tracker system downstream the magnet implement two different technologies: the Outer Tracker system is based on gaseous straw tube detector for a global resolution on the bending plane (*x-z*) of around 200 μm while the Inner Tracker is made of Silicon microstrip detectors (also employed for the *Upstream Tracker* upgrade). The main reason why it's necessary to replace the *IT* and *OT* is related to the occupancy being too high in upgrade conditions and the fact that the electronics for them was designed for a 1MHz read-out rate. On top of that, the upgrade phase of LHCb is designed to collect an integrated luminosity greater than 50 fb^{-1} and the detector itself is required to be resistent to the corresponding radiation damages.

The adopted solution for the upgrade is the *Scintillating Fibre Tracker*. The active and light-transport (also wavelenght shifter) material for the detector are the fibres themselves and the read-out is provided by arrays of silicon photomultipliers. The pitch of a single channel is designed to be equal to 250 μm and each module of the detector consist of 5 (6 in central region) closed packed fibres ($\Phi \sim 250 \mu m$) layer.

The *SciFi* is placed in the downstream region with respect to the magnet (*B* field lines direction are along the *y* direction) consists of 3 stations. Each of them is composed by four layers of scintillating fibres detectors oriented in the so-called stereo *x-u-v-x* configuration.[4]. The stereo configuration allows the reconstruction of tracks using their projection in the bending *x-z* plane and the evaluation of the *y* information is obtained through the combined measurement of the *u-v* layers.

Each layer is divided in two halves (roughly *y>0* and *y<0*) equipped at *y=0* by a mirror to improve the light yield in the higher occupancy area (the region close to the beam pipe).

A skematic view of how a single station look like is given in 1.2 and it will consists of around 10.000 *Km* of fibres and 560.000 read-out channels distributed over the 12 layers (6 *x*, 3*u*, 3*v*).

The requirements for the *SciFi* to satisfy the physics program of *LHCb* can be listed in few concise points:

- The hit efficiency has to be ~ 99%

- The noise cluster rate should be less than 10%, reason why the *SiPM* read-out has to be cooled

---

[4]The x - layer contains fibres running perpendicularly to the x-z plane while the u(v) layers are exactly the same as the x-layers but rotated by a "stereo" angle of +5(-5) degree in order to reconstruct also the y-information of the tracks.
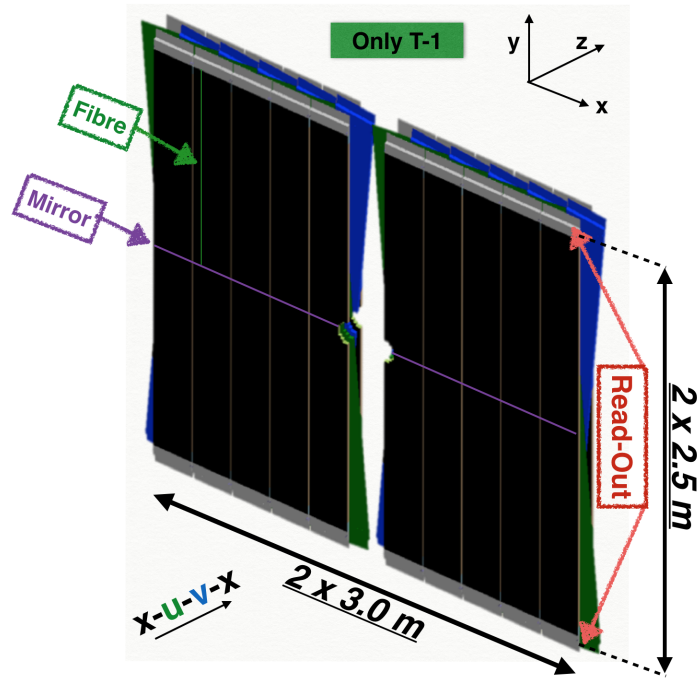
Figure 1.2: Sketch of a single station of the SciFi tracker. The Station is made of 4 different layers in x-u-v-x configuration, the central region is mirrored and the read-out is done at the edges of the layers far from the beampipe.

- The position resolution in the bending plane has to be ~ $100\mu m$ and the material budged has to be as reduced as possible $\frac{X}{X_0} < 1\%$ per detector layer

- The readout has to be performed at 40 MHz

- The tracker has to be efficient for at least an integrated luminosity of 50 fb^{-1}

- The *SciFi* tracker has to replace the actual one, so additional goemetrical constraints must be satisfied.

1.2 PATTERN RECOGNITION AT LHCB

The track reconstruction at *LHCb* is performed in different steps. The idea is to firstly reconstruct tracks collecting list of hits from sub-detectors following some given criteria (track model in a subdetector, expected behavior of tracks in between one detector and another). Each of the found track is then stored in a temporary container and a global fit and finer treatment is performed afterwards (*Kalman Filter* and *Ghost and Clone killer*). We will then refer to the first step as *pattern recognition* and the second one as the *final fit*.

The pattern recognition at *LHCb* is done depending on the path the track goes through, so it's based on the datector's hit content as shown in Fig. 1.3.
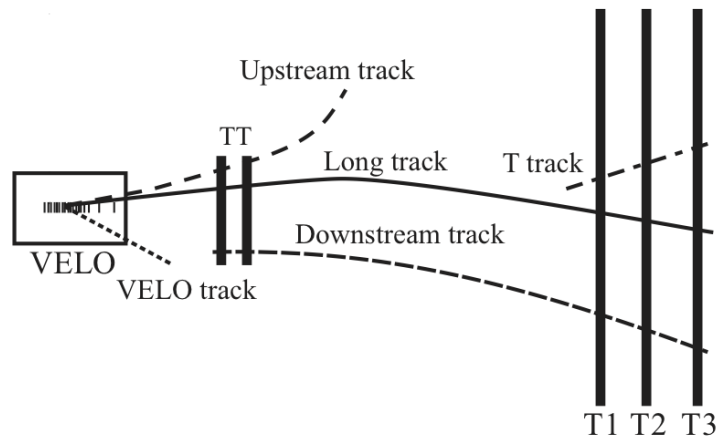

Figure 1.3: Track type at *LHCb*. Velo tracks are basically straight lines since the magnetic field is almost 0 in that region. Tracks are mainly bended in the x-z [5] plane between the Tracker Turicensis (Upstream Tracker for the upgrade) and the T-Stations which is composed by the Inner Tracker(IT) and the Outer Tracker(OT) (Run-I and Run-II), while for the upgrade the stations will be replaced by the Scintillating Fibre tracker (SCIFI).

---

[5] z- is the beam axis direction and the y axis is the B field line direction.

In the tracking system of *LHCb* each track type is reconstructed by a dedicated algorithm and a schematic layout of how it is done is given in table 1.2.

Track type	Used detector	Algorithm(s)	Input tracks	Output tracks
Velo Tracks or Velo-Segment	Velo	Velo algorithm	/	Velo
Seed Tracks or T-Tracks	T-Stations (SciFi in upgrade)	Seeding algorithm	Allow the possible usage of the leftover hits of forward. If Not: Standalone Algo	Seed
Long Tracks (1)	Velo + TT + T-Stations (TT → UT T-Stations → SciFi in upgrade)	1) Forward tracking: Search in T-Stations knowing Velo-Segment (adding also TT) 2)Matching algorithm: Merge T-Tracks with Velo-Segment 3)BestSelector= Forward+Matching	Velo Velo and Seed	Long
Downstream Tracks	T-Stations and TT (SciFi and UT)	Downstream algorithm: Use T-Tracks and add TT (UT upgrade) hits	Seed	Downstream
Upstream Track	Velo and TT	Upstream algorithm: Use Velo segment and add TT(UT upgrade) hits	Velo Container	Upstream

Table 1.2: Tracking algorithm at LHCb

All the tracks produced by the algorithms provided in Table 1.2 are processed afterwards by the *Kalman Filter* which fit the tracks assigning a sort of *univoque* χ^2 to the track (each algorithm infact has an internal track model parametrisation resultin in different χ^2 computation) taking into account the *B(x,y,z)* field map and the material budget computing for instance corrections due to multiscattering.

From a more technical point of view, in the *Kalman Filter* and in the *LHCb* framework each track is defined by a vector of track state $\left(x, y, t_x, t_y, \frac{q}{p}\right)^T_z$ ($t_x = \frac{dx}{dz}, t_y = \frac{dy}{dz}$) which is then propagated by a 5x5 matrix through the detector considering the interactions and the B-Field map [6]. So, the main goal of the listed algorithm in Table 1.2 is to provide a preliminary set of tracks composed by compatible hits in a given sub-detector and only at the end of the algotithms the found tracks are converted into a vector of track states, which can be handled by the *Kalman Filter*.

1.3 Hybrid Seeding Algorithm

Since no real data is available at the upgrade condition, a huge effort must be done to reproduce the real data taking conditions and what the real output of the detector would be. This task is obtained through different steps in the simulation. Without going into details we can make a short summary of how the *LHCb* software is working in order to provide, for a given algorithm the final efficiencies. In this document efficiencies will be given for a simulated sample of $B_s \rightarrow \phi\left(K^+K^-\right)\phi\left(K^+K^-\right)$ event.

- The *Gauss* application is responsible of the physics simulation, starting from the $pp \rightarrow b\bar{b}$ production (package doing that is called *Pythia*) given the 25 *ns* bunch crossing, the $\sqrt{s} = 14$ *TeV* centre-of mass energy collision and the ν^7=7.6 and the subsequent $B_s \rightarrow \phi\left(K^+K^-\right)\phi\left(K^+K^-\right)$ (package *EvtGen*) decay chain are provided. Inside *Gauss* also the interaction of the produced final states in the detector is simulated by the *Geant* package. At this level of the simulation the so-called *MCHits* are produced, corresponding to the energy deposition in the active material of the detector. A true particle is then given by a certain set of *MCHits*.

- The read-out, the *SiPm* response , the light attenuation in the fibres and the *Clusterization* together with the front-end boards (data - encoding) is simulated by the *Boole* application. The output of this step are the *Clusters* containing binary information about the corresponding position in the tracker.

- Once the digitization is done, the resulting objects that pattern recognition algorithm can handle are the so called *PrHit* (in the SciFi context) and making use of them one can finally reconstruct tracks. The *PrHit* are obtained converting the binary information of the *Clusters* into a geometrical information usable for pattern recognition. The track reconstruction is done inside the *Brunel* application.

- In each of the previous steps one can store the *Linkers* which allows to figure out the list of *MCHit* associated to a true particle which are finally converted to cluster (so into a *PrHit*). Thanks to them one can define the *reconstructible*[8] tracks and compare them with the output of the pattern recognition algorithm (*reconstructed* tracks) in order to provide tracking efficiencies numbers.

---

[6] The formalism is quite similar to accelerator physics.

[7] Average number of visible interaction per bunch crossing

[8] For the SciFi a reconstructible track is defined if it leaves at least 1 x-layer hit in each of the three stations together with at least one hit in the u or v layer for each of the three stations

The most important figures of merit classifying a pattern recognition algorithm are the following:

- *Reconstruction efficiency* : how many of the tracks we find in the algorithm are actually tracks we expect to reconstruct? For instance,for analysis requiring n final states, the yield will be reduced by a factor ϵ^n.

- *Ghost rate* : how many tracks we reconstruct are associated to pesudo-random combination of hits?. This is important because we don't want to recontruct fake tracks and in addition we don't want to feed the *Kalman Filter* with too many fake tracks for timing constraints.

- *Timing* : how fast we are finding the tracks. This is particualry important since in the upgrade the tracking algorithms has to be run online at an incoming rate of *40 MHz*.

The work done from October 2014 to now is encoded in this framework.

At the beginning i was involved in the debugging of the simulation code, putting my hands in the detector description, the simulation of the clusterisation and the encoding/decoding of the clusters (basically the *Boole* and *Brunel* applications and packages). Some problems and inefficiencies in the software were infact not well understood passing from the default detector description (the one of the technical design report [**?**]) to the new one where bigger gaps between channels were simulated, together with a new shape of detector layers, and a more realistic model of the clusterisation were implemented.

In that context my effort was fundamental (i took part of the *DILBERT* task force for the software fixing and bug correction) to recover part of the loss in efficiency and we finally provide a stable version of the simulation of the detector from which the development of a new pattern recognition algorithm becomes possible.

In the meanwhile i also took part to the *Test Beam* data analysis aiming to measure the radiation lenght of the fibres doing light yield measurements scanning along the active material at different distances from the read-out and also evaluating the dependency of the cluster size against track incident angles. I also analyse some of the data from the test beam to study the cluster size properties as a function of the incident angle of tracks in the detector in order to tune the simulation to be the most reliable possible. [**?**] [**?**]. The results of how the efficiencies for the seeding algorithm change in time will be shown at the end of this section to highlight my contribution to the upgrade *SciFi simulation and reconstruction* working group. I presented regularly my work in the internal *SciFi simulation and reconstruction* working group and i presented once the results of the group in the parallel session of the *LHCb week*.

1.4 IMPROVEMENTS APPORTED TO THE ALGORITHM

Several trials in changing the previous algorithm (named here as *Old*) has been done and we finally decide to develop a new one from scratch, once we figured out the limitations of the previous one. The developing of the new one started from very basics things related to track properties, detector occupancy and requirements on tracks.. First of all we need to separate different aspects of the algorithm:

- The way we collect the hits to generate tracks.

- The way we fit the tracks.

- The tolerances we allow in both the fitting and the hit search.

- The requirements we put on tracks for their storage and clone/ghost killing.

. In order to do that, several tools have been developed: one of them, for instance, aims to extract the *Monte-Carlo* information of tracks before the clusterisation step in order to evaluate the track properties and their behavior in the *SciFi* stations.[9] The strategy adopted once the track property was extracted was then the following: instead of using a look-up table for the *B* field information one can directly use the *B*-field effect on tracks parametrising their motion by some fixed constants reflecting the magnetic field behavior.[10]

The main limitations, or problems, of the previous algorithm can be summarized in few points

- It was trying to find everything at once, meaning that it was not disentangling the cases where tracks are easy to finde to the cases where the tracks are harder to find. Harder to find stands for find it paying an high prize in terms of ghost rate.

- The tracks behavior in the *SciFi* tracker was roughtly parametrised.

- The different detector area were treated in the same way even if where we expect a lower efficiency we can loosen the cuts for the track selection.

The strategy adopted for the track search is to **split the algorithm in nested sub blocks** following the philosophy of the *projective tracking* and *progressive cleaning* of the tracking environment. Since the detector is divided into two halves (*y>0* and *y<0*) a natural choice is to split the algorithm in a first step where tracks are searched using only the hits in the upper modules of the stations and a second iteration is done using hits from the lower half. From *MC* study we figure out that the amount of tracks going from the upper to the lower half is a negligible fraction of the full set of tracks $\sim O(0.1\%)$.

Inside the upper(or lower) half search we run the second main block of the algorithm 3 times which we will call here *case* since for each of this run different configurations are applied. Each of this *3 cases* is dependent on the previous one since at the end of each of them (only the first and the second) we mark the hits on the found tracks using some specific criteria. The final result is a progressive clean of the tracking environment moving from one case to the next one, since for instance in the second *case* the hits found by the previous *case* will not be available anymore. Each of the case is made by some different step:

---

[9]Thanks to that an unofficial tracking algorithm running over true hit position was developped to figure out which is the best track model to use in order to fit the tracks going through the *SciFi* and also to have an estimation of "how much" one can squeeze some search windows without starting to loose to much in efficiency.

[10]The B field in the Sci-Fi tracker region is a fringe field which is not easly parametrisable, but still, some caracteristic behavior can be extracted.

- **x-z search**: Hits are collected using only the x layers in the 3 stations. The maximal number of hits on track is then 6 and the minimal one can be set to 4 or 5 depending on the case. The lower is the minimal number of hit requirement here the higher is the ghost rate. [11]

  1. 2 Hit combinations are obtained looking to pairs of hit in the x layers in *T1* (1st *SciFi* station downstream the magnet) and *T3*. Given a hit in the first station (*T1*) the infinite momentum prediciton is used to open a search window in *T3* ($x_{T3} \sim \frac{x_{T1}}{z_{T1}} \cdot z_{T3}$ under the $p^{\inf}$ assumption). All the possible 2 hit combination is then collected. Tighter search window is applied for the first *case* and bigger ones are used for the second and third case in order to find lower momentum tracks.

  2. A third hit from *T2* is added to each single pair taking the linear prediction from $x_{T2} = \frac{x_{T3} - x_{T1}}{z_{T3} - z_{T1}} \cdot z_{T2}$. At this step, the small is the momentum of the track we are searching for and the bigger the deviation from the linear predicted position we should look at. This search window is somehow related to the p_x and it's related to the sagitta of the track itself in the bending plane.

  3. Given the 3 hit combination (one per station), the parameters of a parabola can be computed solving the linear system given by $x_i(z_{T-i}) = a_x + b_x \cdot dz_{T-i} + c_x dz_{T-i}^2$) where $dz_{T-i} = z_{T-i} - z_{ref}$ and z_{ref} is picked to be $8530mm$ ($\sim z_{T2}$) for numerical stability of track parameters (a_x, b_x, c_x).[12].

  4. Once we have the a_x, b_x and c_x we can compute the predicted x position at any z and collect the found hits around the expected position inside a given tolerance. At this step the hit at the smallest distance from the prediction is picked and added to the track.

  5. At this stage , a list of hits is collected which should correspond to the x-z projection of the true track in *x-layers*.

  6. A track object is created using the hits found in the *SciFi* and a fit is performed. The fit uses as track model for the x-z projection the following parametrisation $x(z) = a_x + b_x dz + c_x dz^2 (1 + dRatio \cdot dz)$.

     a) The *Backward projection* is computed, being the estimation of the $x(z = 0)$ accounting for the integrated magnetic field from $z = 0$ to $z = z_{ref}$ under the assumption that the particle is generated at $(x = 0, z = 0)$. The formula used is then $X_0 = a_x^{fit} + b_x^{fit} + c_x^{fit} \cdot C^{Const}$ where C^{Const} is a fixed parameter found from *MC* accounting for the kick the tracks receive in the region before the *SciFi*.

     b) Combined cuts are applied in the plane given by the χ^2 versus X_0 to return the fit status. At this step also the occupancy is taken into account. For tracks going far from the beampipe ($|X_{T1}| > 500mm$) looser cuts are applied since the chance to find compatible hits in the **AddStereo** step is much lower than in the higher occupancy detector area ($|X_{T1}| < 500mm$).

     c) If the x-z *projection* track satisfy the requirements they are stored as good tracks, if not, they are *refitted* after the removal of the worst hit (defined as the hit giving the higer contribute to the χ^2). So, for example, if we find a x-z *projection* track with 6 hit and at the first fit the fit is not successful because of the requirements we put on the χ^2 and the X_0, we remove one hit and we refit it again. This process is done recursively as soon as we don't reach the limit given by the minimal number of hits we require. At this step of the fitting and minimal hit requirements, we make some differenciation between the first *case* and the following ones, being that, for the first and the second one we keep only track containing 4 hits on x layers at the second and the third *fit* while for the third *case* we also keep 4 hits on track at the first iteration.

  7. The track are stored (at this stage we can only have tracks containing 4,5 or 6 hits) and a *clone removal step* is performed based on a minimal number of shared hits, which is set to 2 by default. If two tracks shares 2 hits or more, we keep the one with the higher number of hits, if they have the same amount of hits, we keep the one with the better $\frac{\chi^2}{D.O.F}$

- **Add Stereo**

  1. For each of the x-z candidates found in the x-z *search* step, a list of compatible hits in the u-v layers is collected once the x predicted position x_{pred} is computed using the usual track model $x(z) = a_x + b_x \cdot dz + c_z \cdot dz^2 (1 + dRatio \cdot dz)$. Since the $u(v)$ layers have the fibres bended in the x-y plane of $\alpha = +5°$ ($\alpha = -5°$) what can be actually handled in the algorithm is the $\frac{u - x_{pred}}{sin(\alpha)}$ which is related to the y position we allow.

  2. Once the vector of compatible hits in withitn the y tollerance we set is created, the collected hits are sorted by β, where the $\beta_{Hit} = \frac{u - x_{pred}}{sin(\alpha) z_{Hit}} \sim \frac{y}{z}$.

  3. Since the magnetic field is almost zero in the x and z direction the track motion is basically a line in the non-bending plane y-z. So if a group of hits have the same β this is an hint that we are actually finding the good combination of stereo hits to add to our *x-candidate*. Infact what we pass to the fitting method is the track created by the x-z condidate plus all the hits satisfying $\beta_{Hit^i} - \beta_{Hit^j} < Tolerance$. This is the way the *hough transformation* is implemented[13].

  4. For each of the track candidates we generate adding the stereo hits, we make a fit and , as in the x-z *projection* search, a fit is performed, but at this stage the track model is extended also to the y-z plane. The adopted model becomes then

$$\begin{cases} x(z) = a_x + b_x dz + c_x dz^2 \cdot (1 + dRatio \cdot dz) \\ y(z) = a_y + b_y z \end{cases}$$

.

---

[11]Tracks with only 4 x-layer hits are much less constrained by tracks with 6 x-layer hits.

[12]In reality we solve for (a_x, b_x, c_x) using $x(z) = a_x + b_x dz + c_x dz^2 (1 + dRatio \cdot dz)$ where the *dRatio* is fixed by MC studies

[13]The technical implementation is the followign: we pick the first hit in the β sorted array and we go at the sixth element after it in the array. If it satisfy the tolerance condition we extend the picked hits from 6 untill the $\beta_{Hit^i} - \beta_{Hit^j} < Tolerance$ is not satisfied anymore. If the search fails we move in steps of 6 inside the array and we repeat the search

5. In the fit method the status of the fit (failed or successful) is based on different properties: the maximal contribution of the hit to the χ^2, the χ^2 of the track, the number of hits on the track, the value of the backward projection X^0, the value of the y backward projection (i.e. $track.y(z=0.mm)$) and the region of the detector interested ($x(zT1),x(zT3),y(zT1),y(zT3)$). The last bit is taken into account because knowing where the track is going through in the detector we can apply stronger requirements for tracks going through the higher occupancy area.

    a) If the track satisfy the requirements, it is stored in memory as a good candidate, if not, we remove the worst hit and we redo the fit.

6. Since the starting point was a *x-Candidate* and there is the possibility to find more than one stereo segment associated to that, the best one is selected, based on the number of hits on track (favouring the one with more hits) and keeping the one with the best $\frac{\chi^2}{D.O.F}$ if the comparison is done between the same amount of hits on the track.

- Once we have the track candidates produced by the *Case* tracks undergoes a clone removal step which has the same logic as the one in *x-z* search.

- As discussed, the algorithm is sub-divided in 3 cases, and the passage between the first one and the next is done performing a *Flagging* of the hits found on the tracks associated to the case. The flagging step has a crucial role in the of *progressively cleaning* of the tracking environment because the algorithm is re-run without considering the hits found in the previous case.[14]

The idea for the first case is to find tracks with high momentum which are easy to find since the kick they get in the bending plane from the *B-field* both in the *SciFi* tracker region and in the propagation between its origin vertex and the *SciFi* is relatively small ($\propto \frac{qB}{p}$). Hits are collected in a relatively small search windows under the infinite momentum assumption given an hit found in the first layer.

The infinite momentum assumption relies on the fact that very high momentum tracks motion is almost a straight line in the *x-z* bending plane, resulting in a linear pointing to the origin vertex (*(x,z)=(0,0)*).

1.4.1 ANALYSIS WORK : $B^0 \to D^0\overline{D}^0 K^{*0}$ ANALYSIS

2 CHOICE OF THESIS TOPIC

3 TIMETABLE FOR FUTURE

REFERENCES

[1] LHCb Collaboration , *The LHCb Detector at the LHC, JINST 3(2008) S08005*

[2] LHCb collaboration, *Letter of Intent for the LHCb Upgrade*, CERN-LHCC-2011-001, March 2011.

[3] LHCb collaboration, *Framework TDR for the LHCb Upgrade*, CERN-LHCC-2012-007, May 2012.

[4] LHCb collaboration, *LHCb Scintillating Fibre Tracker Technical Design Report*, CERN-LHCC-2014-001; LHCb TDR 15.

---

[14] Each case is carachterized by the selection of the 2-hit combination: where for case 1 we have small search windows since we are trying to reconstruct high momentum tracks which are the easiest one to build.