DISSERTATION

submitted to the
Combined Faculties for the Natural Sciences and Mathematics
of the Ruperto-Carola University of Heidelberg, Germany
for the degree of

doctor rerum naturalium

presented by

Dipl.-Phys. Manuel Tobias Schiller

born in Heidelberg, Germany

Oral examination: July 12th, 2011

Track reconstruction and prompt K^0_S production at the LHCb experiment

Referees: Prof. Dr. Stephanie Hansmann-Menzemer

Prof. Dr. Norbert Herrmann

Abstract

This work presents the implementation and performance of a track reconstruction algorithm for the main tracking system of LHCb. This algorithm has a high reconstruction efficiency and a low fraction of wrongly reconstructed tracks. It is fast enough to be used in the first stage of the LHCb software trigger to quickly confirm or reject the decision made by the hardware trigger. Moreover, it is the only algorithm in LHCb to reconstruct cosmic muons in the main tracking system, an important track sample for alignment and calibration purposes. Due to its good performance and speed, the algorithm reconstructs tracks in the standalone main tracking system at LHCb both in the software trigger and for offline reconstruction.

Furthermore, the first measurement of the cross section for prompt K_S^0 production in pp collisions at $\sqrt{s} = 900\,\text{GeV}$ with the LHCb detector is presented. The K_S^0 is reconstructed via its decay to two charged hadrons. The measurement thus relies heavily on the understanding of the track reconstruction. Covering a range in rapidity of $2.5 \leq y < 4$ and $p_T < 1.6\,\text{GeV}$ in p_T (excluding $p_T < 200\,\text{MeV}$ for $2.5 \leq y < 3$), $\sigma(pp \longrightarrow K_S^0 X) = (6410.3 \pm 169.6 \pm 463.1)\,\mu\text{b}$ has been measured. The measurement can serve as input to validate and optimise Monte Carlo event generators. The correct description of fragmentation and the structure of the underlying event at hadron colliders by the Monte Carlo generators is crucial for many future LHCb measurements. Being the first measurement performed by LHCb, it has served as an important test for the analysis procedures and the understanding of the detector, before more complicated analyses have been performed.

Kurzfassung

In dieser Arbeit wird die Implementierung und Leistungsfähigkeit eines Spurrekonstruktionsalgorithmus für das Hauptspurkammersystem des LHCb-Detektors vorgestellt. Der Algorithmus hat eine hohe Spurrekonstruktionseffizienz, der Anteil falsch rekonstruierter Spuren ist klein. Er ist schnell genug, um in der ersten Stufe des LHCb-Softwaretriggers die Entscheidung der Triggerhardware zu verwefen oder zu bestätigen. Desweiteren ist es der einzige Algorithmus, der kosmische Myonen im Hauptspursystem rekonstruieren kann, die eine wichtige Spurkategorie für die Ausrichtung und Kalibration des Detektors darstellen. Wegen seiner Leistungsfähigkeit und Geschwindigkeit wird der Algorithmus zur Rekonstruktion der Hauptspurkammern bei LHCb verwendet, sowohl im Softwaretrigger als auch für die abschließende Ereignisrekonstruktion.

Desweiteren wird die erste Messung des Wirkungsquerschnitts direkter K_S^0-Produktion in pp-Kollisionen bei $\sqrt{s} = 900\,\text{GeV}$ mit dem LHCb-Experiment vorgestellt. Das K_S^0 wird über den Zerfall in zwei geladene Hadronen rekonstruiert, weswegen die Messung besonders auf ein gutes Verständnis der Spurrekonstruktion angewiesen ist. Im Bereich $2.5 \leq y < 4$ und $p_T < 1.6\,\text{GeV}$ ($p_T < 200\,\text{MeV}$ mit $2.5 \leq y < 3$ ausgenommen) wurde $\sigma(pp \longrightarrow K_S^0 X) = (6410.3 \pm 169.6 \pm 463.1)\,\mu\text{b}$ gemessen. Die Messung kann der Validierung und Optimierung der korrekten Beschreibung der Fragmentation und des zugrundeliegenden Ereignisses in Hadronbeschleunigern in Monte-Carlo-Generatoren dienen. Eine korrekte Beschreibung durch Monte-Carlo-Ereignisgeneratoren ist entscheidend für viele zukünftige Messungen von LHCb. Da dies die erste Messung von LHCb darstellt, war sie wichtiger Prüfstein der Analyseprozeduren und für das Verständnis des Detektors, bevor komplexere Analysen durchgeführt werden konnten.

Contents

1. Introduction

With the Standard Model of particle physics developed during the sixties of the last century, particle physics has an extremely predictive and successful model at its disposal. The Standard Model has been the theoretical work horse for the past forty years and can describe practically all the measurements done at collider experiments so far with astonishing accuracy. However, there are some issues which remain unexplained in the Standard Model:

- The universe today consists of matter and practically no antimatter. The amount of CP violation present in the Standard Model is too small to explain the matter-antimatter asymmetry found in the universe today by several orders of magnitude.
- Arguments from cosmology suggest that only about 5% of the energy density of the universe is contained in matter as we know it. About 22% appear to be made of a form of matter which does not interact with its surrounding and can therefore not be seen — so-called dark matter. The Standard Model of particle physics does not provide any hint as to what dark matter actually is. The remainder of the energy density of the universe seems to be even more exotic and has been dubbed dark energy.
- To give masses to particles, the Standard Model invokes the Higgs mechanism ([1]). This mechanism predicts a scalar boson which should be directly observable by collider experiments. Although a lot of effort has been put into direct searches (e.g. at LEP, Tevatron and, more recently, also at the LHC), the Higgs particle has not been seen yet (April 2011). Measurements exploiting radiative corrections however favour a Higgs which should be light with $m_H < 158\,\text{GeV}$ at 95% confidence level ([2]).

The Large Hadron Collider (LHC) built at CERN[1] in Geneva has been operated since the end of 2009, and has been delivering proton-proton collisions at a centre-of-mass energy of 3.5 TeV since 2010. Four large experiments are currently operated at the LHC: ATLAS, CMS, LHCb and ALICE. Two of them, ALTAS and CMS are general purpose detectors designed for direct searches for physics beyond the Standard Model (New Physics). The LHCb experiment specialises in precision measurements of loop suppressed b hadron decays to indirectly discover New Physics. The ALICE experiment studies heavy ion collisions in order to learn more about the properties of hadronic matter in the very early universe.

LHCb[2] is an experiment which aims to test the Standard Model by precisely measuring decays of so-called b hadrons, i.e. hadrons containing a b quark. The two main fields

[1] Organisation Européenne de la recherche nucléaire
[2] LHC-beauty

in which LHCb is expected to contribute are precision measurements of CP[3] violating decays and with the measurement of rare decays. For both sets of measurements, the Standard model makes firm predictions for loop suppressed decays. Contributions to the radiative corrections from new particles beyond the description of the Standard Model can alter the observed decay rates with respect to the precise Standard Model predictions. Measurements of CP violating decays of b hadrons and rare decays therefore provide a sensitive laboratory to search for New Physics.

Since the quantum loop corrections mentioned above only involve virtual particles, the testable mass range for a potential indirect discovery of a new particle is much higher than for direct searches. In the past, many "direct" discoveries have first been observed in indirect searches. Examples are the existence of the charm quark or the existence of a third quark family. Precision tests of the Standard Model predictions can thus indirectly probe for new effects, nicely complementing the direct searches at the LHC.

Most of the envisaged measurements at LHCb require the reconstruction of charged tracks. The first part of this thesis describes a pattern recognition algorithm for the main tracking system of LHCb which was implemented as part of this thesis. The algorithm reconstructs tracks with high efficiency and executes fast enough to be used in the time-critical first stage of the LHCb software trigger.

Before looking for new effects, the LHCb detector hardware must be thoroughly tested on known phenomena to ensure that the detector and analysis procedures are well understood.

This thesis presents the measurement of the prompt K^0_S production cross section in pp collisions at 900 GeV centre-of-mass energy using data taken during the two month pilot run in 2009. The K^0_S mesons are reconstructed via their decay into two charged pions. A good understanding of the track reconstruction is therefore required to perform the analysis.

The measurement demonstrates that even with this very early data the understanding of the detector is well advanced. The studies of this thesis contributed significantly to the first publication by the LHCb collaboration ([3]).

Besides showing the readiness of the detector and the analysis procedures, this measurement provides interesting input for the tuning of Monte Carlo event generators to better describe the properties of hadronic collisions in the forward region as there have been no previous measurements by other experiments in that region at similar centre-of-mass energies.

The thesis is structured as follows: In the first two chapters, the theoretical foundations of the Standard Model are presented and the LHCb detector is described in the detail necessary for the understanding of the remainder of the thesis. The following chapters form two parts: In the first part (Chapters 4–7), the basics of track reconstruction are explained and the pattern recognition algorithm for the LHCb main tracking system implemented for this thesis is described and its performance is evaluated.

In the second part, the measurement of the prompt K^0_S production cross section is described. Chapter 8 provides an introduction and specifies the analysis strategy. The signal selection and the extraction of K^0_S yields is discussed in Chapter 9. In Chapter 10,

[3] C is a symmetry which stands for charge-conjugation, i.e. exchange of all particles with their anti-particles, P is the parity symmetry which mirrors all spatial coordinates at the origin, thus transforming a vector

the degree of agreement between data and simulated events is quantified, and a method to correct for the observed discrepancy is devised. The combination of selection and reconstruction efficiency is determined in Chapter 11, the trigger efficiency for selected events is studied in Chapter 12. Chapter 13 studies and summarises the different sources of systematic uncertainties, and the final results are presented in Chapter 14.

$\vec{x}$ to $-\vec{x}$.

2. Theoretical background

2.1. The Standard Model of particle physics

The Standard Model of particle physics describes the fundamental constituents of matter and their interactions with astonishing precision and predictive power (see e.g. [4], [5] or [6]). The fundamental building blocks of matter are fermions with spin $\frac{1}{2}$ and bosons with spin 1 which mediate the interaction between particles. Given the matter content and the gauge structure of the Standard Model ($U(1) \times SU(2) \times SU(3)$), the possible interactions are fixed, only the strengths of the gauge couplings, the CKM matrix elements (see below) and the masses are needed as input from experiment to enable predictions.

There are two types of fermions, quarks are the building block of hadrons, and leptons, with the most prominent representative being the electron. The interactions described by the Standard Model are the strong interaction which is responsible for the binding of hadrons, the electromagnetic interaction, and the weak interaction (responsible for nuclear β decay). Gravitation is not included in the Standard Model. Table 2.1 gives an overview over the interactions, the observed masses of the mediating gauge bosons and the relative strengths of the couplings. Table 2.2 gives a corresponding overview over the matter sector described by the Standard Model.

In the fermion sector, both quarks and leptons come in three generations with increasing masses. Each generation contains two types of fermions, up- and down-type quarks or charged leptons and neutrinos, respectively: For the quarks, this results in six flavours called *up* (u), *down* (d), *charm* (c), *strange* (s), *top* (t) and bottom (b, sometimes also called beauty). In the lepton sector, the picture is similar, each generation contains a charged lepton (e, μ or τ) and the corresponding neutrino. For each of these particles, there is a corresponding antiparticle which has the same mass and lifetime) because of CPT invariance, a fundamental property of Lorentz invariant quantum field theories. All these particles have been observed.

The strong interaction, also called Quantum Chromodynamics (QCD), describes the interaction of quarks and gluons among each other. The charge associated with the $SU(3)$ gauge group is called “colour” (charge), so each of the quarks listed in Table 2.2 comes

interaction	mediating gauge boson(s)	mass [GeV]	coupling constant
strong	eight gluons (g)	0	$\alpha_s \sim \mathcal{O}(1)$
electromagnetic	photon (γ)	0	$\alpha_{e.m.} \sim \mathcal{O}(10^{-2})$
weak	$W^{\pm}$	80	$G_F \sim \mathcal{O}(10^{-5})$
	Z^0	91	

Table 2.1.: *Interactions in the Standard Model, approximate masses of mediating gauge bosons, relative coupling strengths at low momentum transfer.*

	type	1st generation	2nd generation	3rd generation
leptons	neutrino	ν_e (< 2 eV)	ν_μ (< 2 eV)	ν_τ (< 2 eV)
	charged lepton	e (511 keV)	μ (106 MeV)	τ (1.78 GeV)
quarks	up	u (2 MeV)	c (1.27 GeV)	t (171 GeV)
	down	d (5 MeV)	s (104 MeV)	b (4.2 GeV)

Table 2.2.: *Matter content of the Standard Model. Approximate particle masses were taken from [7]; neutrino mass limits come from tritium decay measurements, light quark masses (u, d, s) are given as current-quark masses in the* $\overline{\text{MS}}$*-scheme at a scale of* $\mu \approx 2\,\text{GeV}$*, c and b quark masses are running quark masses in the* $\overline{\text{MS}}$*-scheme, and the t mass is obtained from direct observation of top events.*

in one of the three colours charges denoted by red, green or blue or the corresponding "anticolour". Gluons are "bi-coloured", i.e. they carry a colour and an anticolour. Stable particles in nature must be colour-neutral, so quarks are confined to bound states: Mesons contain a quark and an antiquark of opposite colour (e.g. red and anti-red), baryons are built from three quarks (or antiquarks) containing all three colours (anticolours), e.g. red, green and blue. The fact that gluons carry colour gives rise to gluon self-interaction. This gluon self-interaction is responsible for the large value of the strong coupling constant α_s at low energies, and ultimately for the confinement of quarks. Emission or absorption of gluons never changes the quark flavour. Since the main production process for b quarks at the LHC is gluon fusion, the b and anti-b ($\bar{b}$) quarks which form the b hadrons in which LHCb is interested are mostly produced in pairs.

The electromagnetic interaction is mediated by photon exchange and acts on all (electrically) charged particles. All fermions are subject to the weak interaction mediated through the $W^\pm$ and Z^0 bosons. The $W^\pm$ bosons carry one unit of electric charge and change quark flavour from up-type to down-type or vice-versa. Therefore, $W^\pm$ exchange is often referred to as charged current process. The Z^0 which mediates the neutral current process does not change flavour. The $W^\pm$ couples only to the left-handed component of the fermion fields (handedness refers to chirality here). In principle, flavour changing neutral currents could also occur through charged currents at loop-level, however, these contributions are suppressed through the Glashow-Iliopoulos-Maniani (GIM) mechanism (see [8]). Table 2.3 summarises the quantum numbers of quarks and leptons governing electroweak interactions.

Gauge invariance forbids explicit mass terms for fermions and gauge bosons. To solve this issue, the Higgs mechanism has been proposed in [1]. Through the mechanism of spontaneous symmetry breaking, the Higgs acquires a non-zero vacuum expectation value. The coupling of the non-vanishing boson field to the bosons and fermions then generates their masses. Excitations of the Higgs field should be observable as a particle, the Higgs boson. Direct searches at LEP and Tevatron have only been able to produce bounds; if the Higgs mechanism is correct, a Higgs-like particle is expected to be seen at LHC.

generation			T	T_3	Y	Q
$\begin{pmatrix} \nu_{e,L} \\ e_L \end{pmatrix}$	$\begin{pmatrix} \nu_{\mu,L} \\ \mu_L \end{pmatrix}$	$\begin{pmatrix} \nu_{\tau,L} \\ \tau_L \end{pmatrix}$	1/2	$+1/2$ $-1/2$	$-1/2$ $-1/2$	0 -1
e_R	μ_R	τ_R	0	0	-1	-1
$\begin{pmatrix} u_L \\ d'_L \end{pmatrix}$	$\begin{pmatrix} c_L \\ s'_L \end{pmatrix}$	$\begin{pmatrix} t_L \\ b'_L \end{pmatrix}$	1/2	$+1/2$ $-1/2$	$+1/6$ $+1/6$	2/3 $-1/3$
u_R	c_R	t_R	0	0	+2/3	+2/3
d_R	s_R	b_R	0	0	-1/3	-1/3

Table 2.3.: *Quantum numbers of quarks and leptons governing electroweak interactions. Doublets of the weak isospin T are given in brackets, the weak eigenstates d', s' and b' are connected to the mass eigenstates d, s, b via the CKM matrix (equation 2.1). T_3 is the third component of the weak isospin, Q the electric charge in units of the elementary charge e and $Y = Q - T_3$ the hypercharge. The indices L and R indicate left- and right-handed components of the fields.*

2.1.1. Quark mixing

The couplings of the Higgs field to the fermions are also called Yukawa couplings. In the quark sector they give rise to quark mixing because the eigenstates of the weak interaction and the mass eigenstates are not the same. The change from the mass base to the base of weak eigenstates is accomplished by the unitary Cabibbo-Kobayashi-Maskawa matrix (CKM matrix) which rotates the mass-eigenstates of the down-type quarks $(d,s,b)^T$ into the weak eigenstates $(d',s',b')^T$:

$$\begin{pmatrix} d' \\ s' \\ b' \end{pmatrix} = V_{CKM} \cdot \begin{pmatrix} d \\ s \\ b \end{pmatrix} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{td} \end{pmatrix} \cdot \begin{pmatrix} d \\ s \\ b \end{pmatrix} \tag{2.1}$$

Since the CKM matrix is unitary, the 18 free parameters are constrained by 9 unitarity relations ($\delta_{ij} = 1$ for $i = j$, otherwise $\delta_{ij} = 0$):

$$\sum_{k=1}^{3} V_{ki}^* V_{kj} = \delta_{ij}$$

In case of b hadron decay for which more than one amplitude contributes, the interference term is sensitive to the relative phase between the amplitudes. If the phase differences change under CP transformations, this gives rise to CP violation. Given the fact that the amount of CP violation in the Standard Model is small, and that New Physics contributions usually manifest themselves in additional contributions to these amplitudes, precise measurements of CP violating decays are a sensitive probe for physics beyond the Standard Model.

2.2. The LHCb physics programme

LHCb aims to challenge the Standard Model with high precision measurements of rare b decays. There are two main approaches:

- precision measurements of CP violation in B decays
- study of very rare processes (e.g. $B_s^0 \longrightarrow \mu\mu$)

The idea behind both approaches is essentially the same: One performs a measurement of an observable in a loop suppressed B decay for which the Standard Model prediction is well known. Any significant deviation from the prediction can then be interpreted as a sign for New Physics which can enter as additional contribution to radiative corrections.

Since possible new particles only enter as virtual correction, such an indirect search for New Physics can probe the particle spectrum to much higher masses than direct searches at the same centre-of-mass energy.

As the subject of this thesis is not the LHCb core physics programme, the interested reader is referred to other literature for details, e.g. [9].

2.3. Relevance of event structure

Most LHCb analyses use signal and background parameters and/or distributions which are obtained from Monte Carlo simulated events. The quality of the simulation results depends crucially on the general event properties as e.g. multiplicity, angular distributions and momentum spectra.

The total track multiplicity has an influence on practically all efficiencies which are measured in the event: A more occupied detector is more difficult to reconstruct, and the fraction of fake tracks produced by the track reconstruction is also increased. But the effect is not necessarily limited to the track reconstruction efficiency: Having more tracks in the event (or having a slightly different momentum distribution in the underlying event) can also affect trigger efficiencies or the efficiency and purity of the particle identification. Even if such effects amount to only e.g 1.5% of difference in efficiency per track between data and simulation, this can add up to significant differences when the whole decay is concerned; for a decay with four charged tracks, one would thus expect a 6% difference in efficiency between data and simulation — this is clearly significant if precision measurements are to be performed.

Another potential source of uncertainty is the modelling of the fragmentation in the simulation, i.e. the way in which the colour flux of QCD breaks up to form hadronic objects which can be observed in the detector. This modelling can influence a variety of observables; the focus for this discussion will be put on the flavour composition of the generated hadrons, especially the amount of strangeness produced.

One place where this might influence an analysis is again particle identification. The analysis might require a very pure sample of pions or muons. If the simulation predicts a different amount of kaons than observed in data, the pions from K_S^0 decays might influence the selection purity because the signatures in Cherenkov detectors look very similar.

The more pressing problem arises in the determination of the flavour of a neutral B meson at production time, i.e. answering the question if (as an example) a B_S^0 was produced as B_S^0 or as $\overline{B_S^0}$. This is a crucial piece of information for many of the measurements in the LHCb core physics programme (e.g. mixing measurements). At hadron colliders, the distinction of B_S^0 from $\overline{B_S^0}$ at production time can only be done on a statistical basis.

It is crucial to know how effective these methods to determine the flavour are, i.e. how often they can be applied (tagging efficiency) and how often they give the wrong answer (mistag probability).

One method (same side kaon tagging) exploits the fact to create a B_S^0 meson in the fragmentation, an s quark has to be taken from the vacuum. This leaves an anti-s quark which in general forms a kaon in the direct neighbourhood. Observing a K^+ indicates a B_S^0 and not a $\overline{B_S^0}$. Having a poor description of fragmentation in the simulation means that tagging efficiencies and mistag probabilities cannot be evaluated on Monte Carlo, and measuring them on data is difficult.

These examples demonstrate that event shape variables like track multiplicities, momentum distributions or strangeness production cross sections are an important input to verify the agreement of event generators and data and to tune the parameters of the models used to describe the non-perturbative region of QCD in these generators. Without good agreement of event variables, precision measurements such as envisioned by LHCb cannot be performed. The measurement of the prompt K_S^0 production cross section described in the second part is a first step of the validation of these quantities.

3. The LHCb experiment

The LHCb experiment is a precision experiment dedicated to study b hadrons decays at CERN's LHC collider in Geneva (see [10]). These b hadrons are produced in pp collisions of the LHC. The design goal for the LHC was a centre-of-mass energy $\sqrt{s}$ of 14 TeV. Since the end of 2009, the LHC has been delivering collisions for physics at various values of $\sqrt{s}$ between 900 GeV and 7 TeV.

With a nominal luminosity of $2 \cdot 10^{32}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ and an inelastic cross section σ_{inel} is of about 70 mb at 7 TeV, one expects the pp collision rate visible in the detectors to be about 14MHz (cf. Figure 3.1).

The total $b\bar{b}$ production cross section $\sigma_{b\bar{b}}$ at $\sqrt{s} = 7$ TeV which is of interest for LHCb is about 300 μb ([12]); the dominant production mechanism at LHC energies is a fusion process of gluons and partons as sketched in Figure 3.2. The ratio $\sigma_{b\bar{b}}/\sigma_{inel}$ essentially gives the signal to background ratio for the trigger: There is about 1 $b\bar{b}$ pair produced for every 230 pp interactions.

The angles with respect to the beam axis θ_b and $\theta_{\bar{b}}$ under which the b and the $\bar{b}$ are produced are strongly correlated. Due to the dominant production mechanism (quark-gluon fusion) and the low mass of the b quark when compared to the centre-of-mass energy, the $b\bar{b}$ pair is also strongly boosted with respect to the lab frame. Therefore, the $b\bar{b}$ quark pair will tend to fly in a cone around either beam (see Figure 3.3).

This fact dictates the basic design of the LHCb detector as described below: With a single-arm forward spectrometer, one can cover a large fraction of the $b\bar{b}$ pair production. Moreover, the strong boost of the $b\bar{b}$ system with respect to the lab frame ensures that primary vertex and B decay vertex are clearly separated (B hadrons fly typically about 1 cm before they decay), so that decay time-dependent measurements become possible.

The LHCb detector covers the pseudorapidity range $2 \leq \eta \leq 5$, defined as $\eta = -\log\left(\tan\frac{\theta}{2}\right)$ with θ being the polar angle with respect to the beam axis. Roughly, a quarter of all $b\bar{b}$ pairs produced have their decay products inside the LHCb acceptance.

In a nominal year (10^7 s at a luminosity of $2 \cdot 10^{32}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$), the number of $b\bar{b}$ pairs which decays inside the detector is expected to be 10^{11}. Moreover, the special detector geometry complements the kinematic range covered by the other LHC detectors in the forward region. Applications range from studying the properties of pp collisions in that regime (as will be demonstrated in the analysis part of this thesis) to B and charm physics.

Since B mesons fly for about a centimetre from their point of production before they decay, a vertex detector with good resolution is of importance to separate the decay vertex from the primary vertex, allowing the signal to be separated from a large amount of background events. In order to facilitate precision measurements, the number of simultaneous inelastic pp collisions in the same bunch crossing (pile-up events) should not become too high because vertex resolution and tracking performance start to degrade with more and more pile-up. Figure 3.4 shows the probability for 0, 1, 2, ... simultaneous

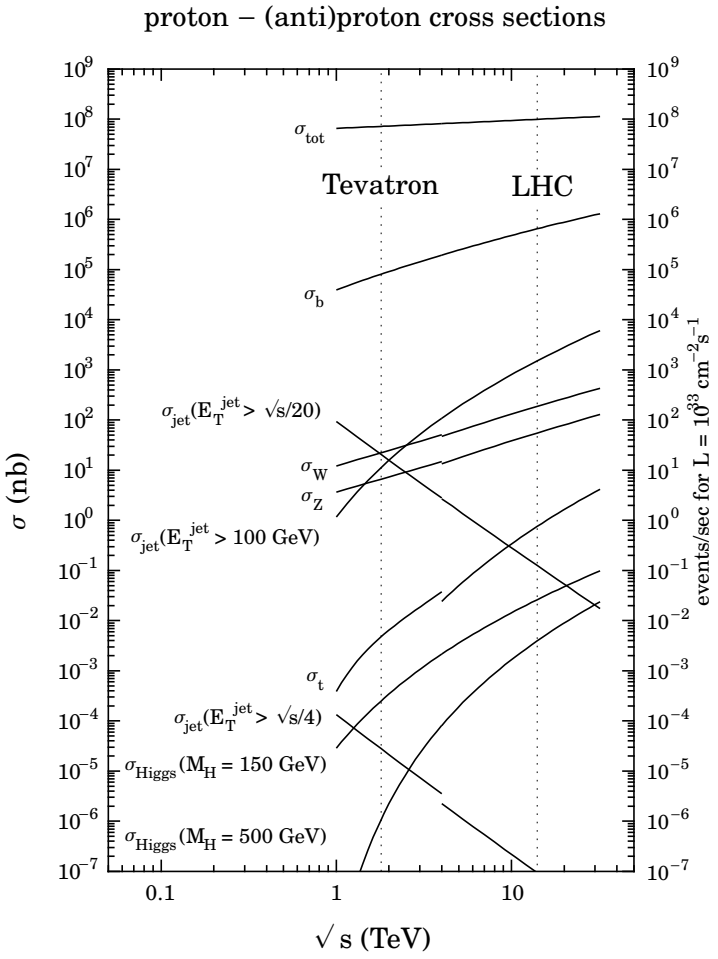


Figure 3.1.: *Cross sections of various hard scattering processes as function of $\sqrt{s}$. The dashed lines indicate the various cross sections at Tevatron energies ($\sqrt{s} = 1.96\,\text{TeV}$) in $p\bar{p}$ collisions and the (final) LHC energy ($\sqrt{s} = 14\,\text{TeV}$) in pp collisions. At the current LHC energy of $\sqrt{s} = 7\,\text{TeV}$, the total pp cross section σ_{tot} is about 95 mb, the $b\bar{b}$ production cross section σ_b about 300 μb. Figure from [11].*

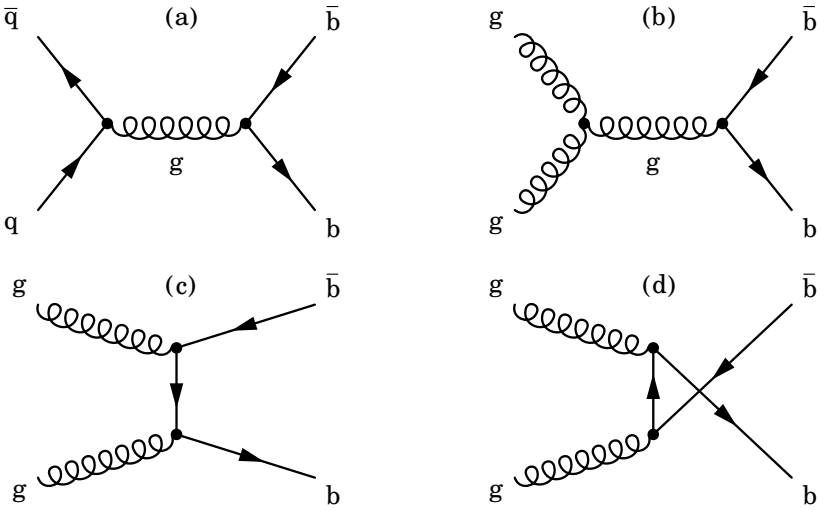


Figure 3.2.: *Leading order Feynman diagrams for $b\bar{b}$ production. (a) shows $b\bar{b}$ pair-production through quark-antiquark annihilation, (b-d) show $b\bar{b}$ production through gluon fusion. Figure from [13].*

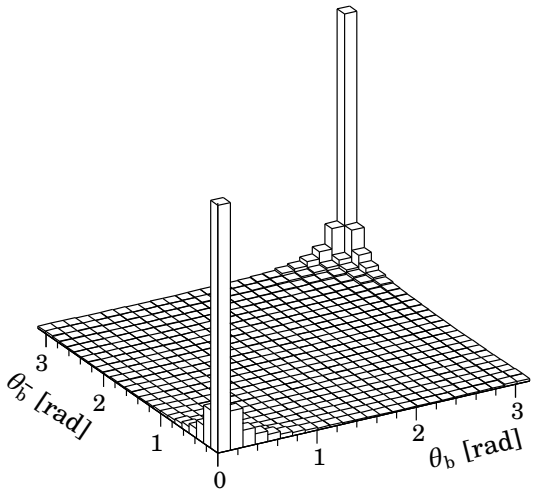


Figure 3.3: *Polar angle correlation of hadrons containing a b and a $\bar{b}$ quark when the b and the $\bar{b}$ quark are produced in pairs. Reproduced from [14].*

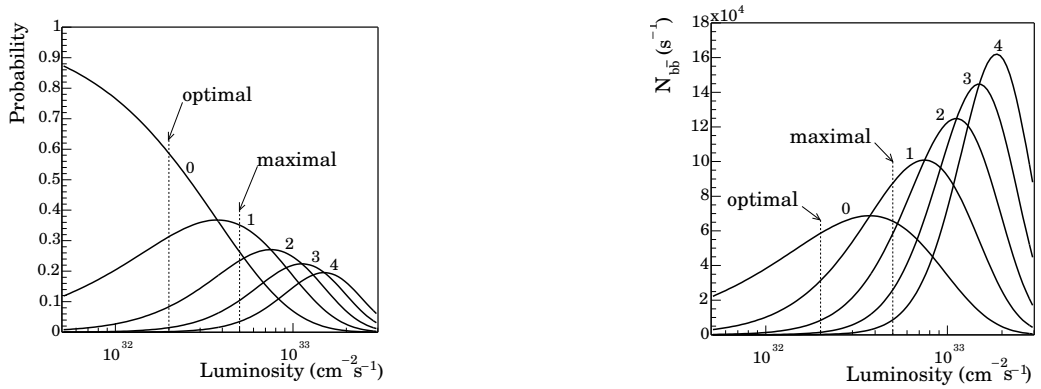


Figure 3.4.: *Probability for 0, 1, 2, ... simultaneous inelastic pp collisions per bunch crossing, number of $b\bar{b}$ pairs produced per second as a function of the luminosity at $\sqrt{s} = 14\,\text{TeV}$. The cross sections for inelastic pp collisions and $b\bar{b}$ production are somewhat smaller at $\sqrt{s} = 7\,\text{TeV}$, but the general trend is similar. The lines labelled "optimal" and "maximal" represent the optimal and maximal luminosities envisaged during the design phase of LHCb. Figure from [15].*

collisions per bunch crossing and the number of $b\bar{b}$ pairs produced per second as a function of the luminosity at the design value of $\sqrt{s} = 14\,\text{TeV}$. A compromise must be chosen to maximise the number of recorded events which are clean enough to be useful in physics analyses.

Detector systems have been designed to operate up to $5 \cdot 10^{32}\,\text{cm}^{-2}\text{s}^{-1}$ at $\sqrt{s} = 14\,\text{TeV}$. For the design scenario, the optimal luminosity was chosen to be at $2 \cdot 10^{32}\,\text{cm}^{-2}\text{s}^{-1}$; this corresponds to 45% of bunch crossings with an inelastic pp interaction or a 14 MHz rate of events with five or more charged tracks in the acceptance, leading to an integrated luminosity of $\mathscr{L} = 2\,\text{fb}^{-1}$ recorded per nominal year (10^7 seconds).

However, the LHC running scenario is not yet the one envisaged in its design, especially the number of bunches filled with protons is much lower. To maximise the recorded luminosity, LHCb was therefore operated with as many as 2.5 inelastic pp collisions per bunch crossing, an increase of a factor of five with respect to the design value. Clearly, this has put the trigger and reconstruction software to the test, and the fact that LHCb is taking data with over 90% efficiency is a testimony of their robustness and flexibility.

As has been said before, LHCb is a single-arm forward spectrometer, its layout is shown in Figure 3.5. It has an acceptance of 10–300 mrad in the bending plane of the magnet (x direction) and 10-250 mrad in the non-bending plane (y direction). It consists of the following components which will be described in more detail in the remainder of this chapter:

- The **tracking system** consists of a vertex detector (Velo – Vertex Locator) and the Trigger Tracker (TT) in front of the dipole magnet, and the main tracking system behind the magnet which in turn consists of Inner Tracker (IT) and Outer Tracker

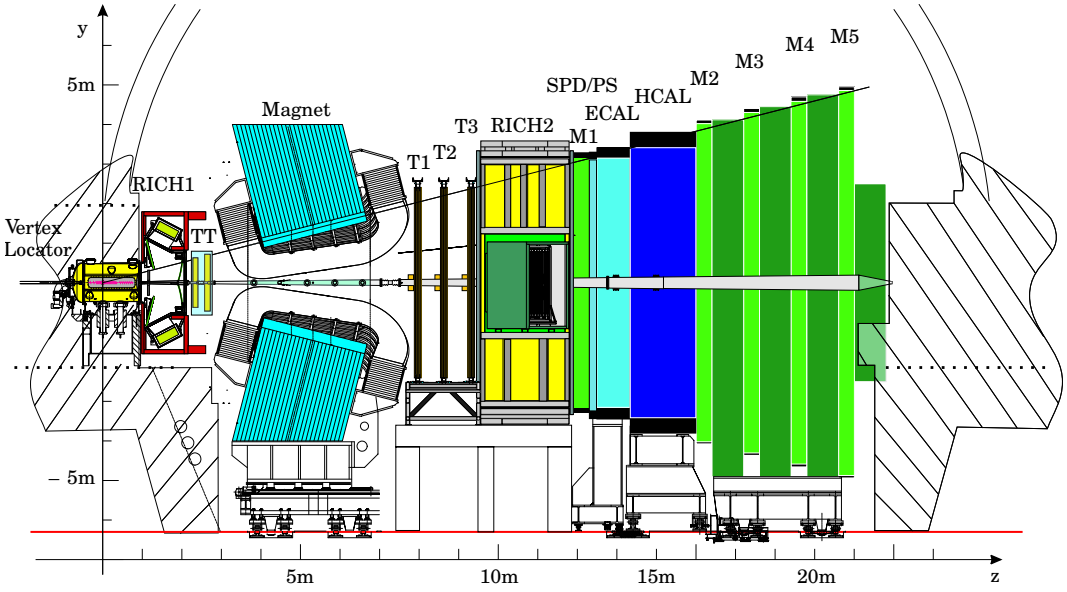


Figure 3.5.: *The LHCb detector layout. The vertex detector (Velo – Vertex Locator) surrounds the interaction point and forms the tracking system of LHCb together with the dipole magnet and the tracking stations (TT and T1, T2, T2; T1-T3 are also known as T stations). Particle identification is provided by two RICH detectors. The calorimeter consists of a Scintillating Pad Detector (SPD), a Preshower (PS) detector, an electromagnetic and a hadronic calorimeter (ECAL and HCAL). Five muon stations (M1-M5) identify muon tracks. Figure from [10].*

(OT).

- **Particle identification** is provided by two Cherenkov detectors (RICH 1 and RICH 2) which mainly serve to provide $\pi^{\pm}/K^{\pm}$ separation over a wide momentum range.

- The **calorimeter system** is formed by a Scintillating Pad Detector (SPD), a Preshower detector (PS) and the electromagnetic and hadronic calorimeters (ECAL and HCAL).

- **Muon identification** is achieved with the muon system, the outermost component of LHCb, which consists of five muon stations; the first station (M1) is situated in front of the calorimeter system, the four remaining stations (M2-M5) behind it.

As can be seen in Figure 3.5, the LHCb coordinate system has its origin in the interaction point and its z axis pointing in beam direction, and the x axis is horizontal; the $x-z$ plane also forms the main deflection plane of the LHCb dipole magnet. The coordinate system is completed by the y axis which is orthogonal to the other two axes, so that a right-handed coordinate system is formed. Since the plane in which the LHC ring

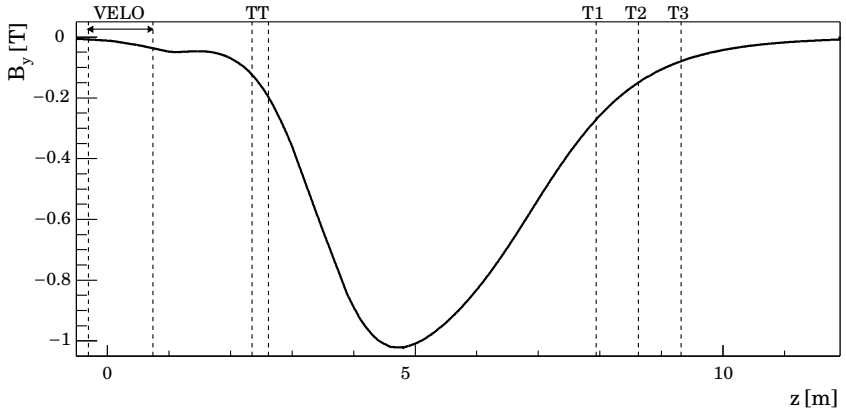


Figure 3.6.: *Main component (in y direction) of the magnetic field as function of the z coordinate; particles are deflected along the x direction. Figure from [10].*

was built is not perpendicular to gravity due to geological constraints, the negative y axis does not point in the direction of gravity. This point is an issue for tracking, since many subdetector systems are mounted hanging from support structures (see the tracking part of this thesis for a more detailed explanation).

3.1. Tracking system

The tracking system consists of the vertex detector (Velo) and the Trigger Tracker (TT) in front of the magnet, the LHCb dipole magnet and the main tracking system behind the magnet which consists of Inner and Outer Tracker (IT and OT). The magnet (see [10], [16]) provides an integrated field of $\int \vec{B} \cdot d\vec{l} = 4.2\text{Tm}$; measuring the bending of charged tracks in the magnetic field allows a relative momentum resolution of $dp/p = 0.6\%$. Figure 3.6 shows the largest component of the magnetic field, B_y, as function of the z coordinate — for particles with their main momentum component in z direction, this causes a deflection in the $x-z$ plane; the deflection caused in the $y-z$ plane is negligible for most purposes.

3.1.1. Vertex detector (Velo)

The vertex detector, also called Velo (from Vertex Locator), is designed to provide good vertex resolution to successfully resolve primary and secondary vertices in the event, and, as a consequence, a good lifetime resolution for B hadron lifetime measurements. To this end, pairs of half-disc shaped sensors mounted back-to-back are positioned left and right of the beam surrounding the luminous region in 21 stations (see Figure 3.7), and two additional stations forming the pile-up system with a single sensor each. The pile-up system is used in the hardware trigger to veto events with multiple interactions, and it can help improve the spatial resolution of reconstructed vertices by providing tracks in backward direction.

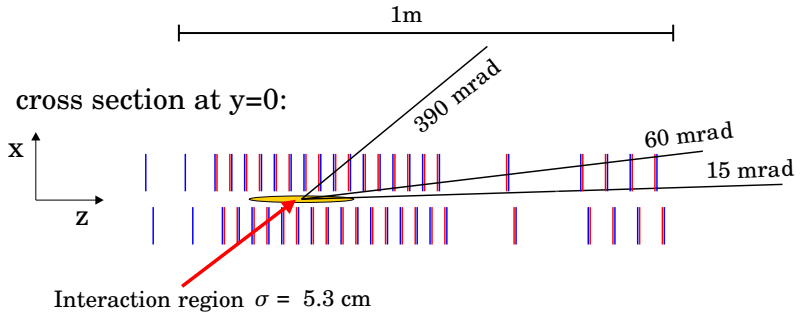


Figure 3.7.: *Top view of the vertex detector. Figure from [10].*

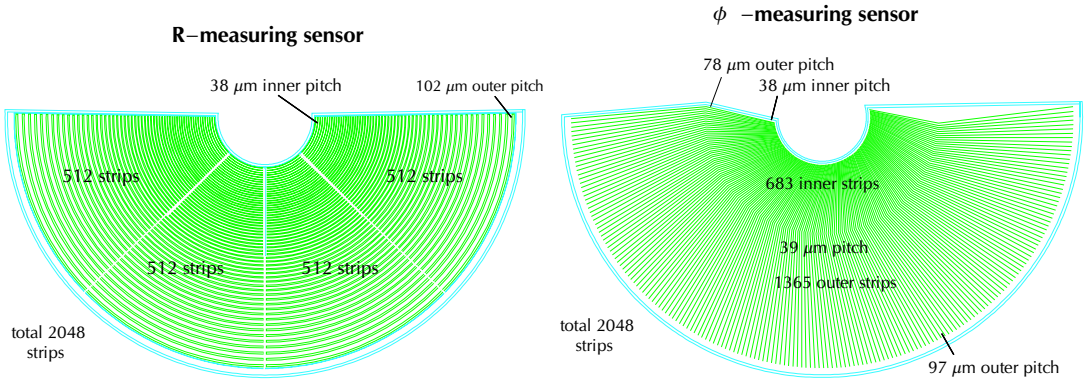
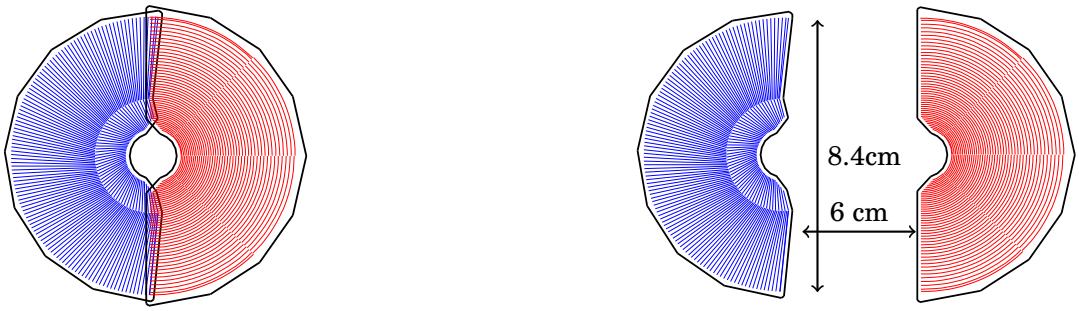


Figure 3.8.: *Sensor geometry of vertex detector sensors for r and ϕ type sensors. Figure based on [17] and [10].*

These sensors are silicon strip sensors of 300 μm thickness which come in two types: First, there are r type sensors which measure the radial coordinate with respect to the beam axis of particles with strips being circle segment shaped, the other type are ϕ type sensors with strips going roughly radially outward. The angle ϕ is zero when the strip direction coincides with the x axis, and increasing ϕ means a counterclockwise rotation when looking in direction of the calorimeters. Figure 3.8 shows the two types of sensors. To keep the occupancy in sensors more or less at uniform levels (well below 1%), r type sensors are divided into four sectors covering about 45° each; the strip pitch increases from about 38 micron at the inner edge to about 102 micron at the outer edge. The figures for the ϕ type sensors are similar.

During detector operation, the silicon sensors are very close to the beam, there are about 8 mm from the beam to the edge of the sensitive area of the silicon (see Figure 3.9). The beam vacuum is only separated from the sensors by a thin aluminium foil which shields the senors from RF pickup from the beam (the geometry of the RF foil accounts for the “staggering” of sensors on left and right side of the beam as visible in Figure 3.7). To avoid beam induced damage in the sensors during beam setup or beam dump, the sensors can be retracted away from the beam to safe positions, as is also shown in Figure



(a) *closed* **(b)** *open*

Figure 3.9.: *Vertex detector in open/closed state. Figure from [10].*

3.9.

3.1.2. Trigger Tracker (TT)

The Trigger Tracker (also called TT) is a tracking detector located in the fringe field in front of the magnet. It consists of two stations separated by about 27 cm along z direction with two layers of silicon strip sensors each. Each sensor covers an area of 9.44 cm × 9.64 cm with 0.5 mm thickness and 512 channels each; the total area covered by TT is about $8.4\,\mathrm{m}^2$. The four layers are arranged in an $xuvx$ configuration, which means that the first and the last layer have their measurement direction along the x axis (i.e. the strips are parallel to the y axis), while the u (v) layer is rotated by a stereo angle of $-5°$ ($+5°$) around the z axis. This allows coordinates to be determined in three dimensions. The strip pitch is about $183\,\mu$m, allowing good resolution in the bending plane of the magnet. Figure 3.10 shows the layout of the first two TT layers. An increasing number of silicon sensors is connected in series for simultaneous readout when going away from the high occupancy regime near the beam pipe towards the outer region, keeping the total number of channels down to 143,360. The spatial resolution in measurement direction (x, u or v depending on the layer in question) is about $50\,\mu$m. More details can be found in [10], [18] and [19].

3.1.3. Inner Tracker

The Inner Tracker is one of the two tracking detectors situated behind the LHCb dipole magnet and is described in [10] and [19]. It consists of silicon strip sensors very similar to those used in the Trigger Tracker. It is arranged in a cross-shaped manner around the beam pipe to cover the area with the highest track density, with three stations of four layers. The layers are in an x, u, v, x configuration with the same stereo angles as used for the Trigger Tracker. Figure 3.11 shows the layout of an x and a stereo layer in the IT. In each station, one distinguishes top, bottom, left, and right boxes depending on which side of the beam pipe the box is located, with each box containing four layers of silicon sensors. The area of a sensor with 384 strips each is 7.6 cm × 11.0 cm with a thickness of 0.32 mm for layers with only one row of sensors or a thickness 0.41 mm for

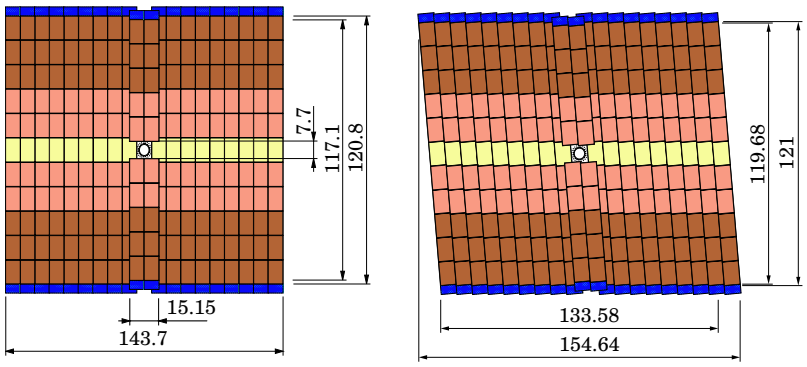


Figure 3.10.: *Layout of the first two Trigger Tracker layers. The colour coding illustrates for which sensors corresponding strips are connected in series to the same readout frontend chip. Figure from [18].*

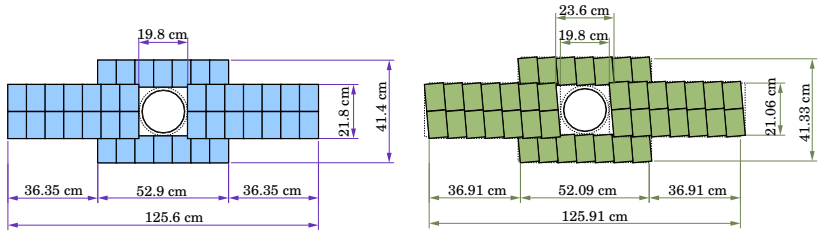


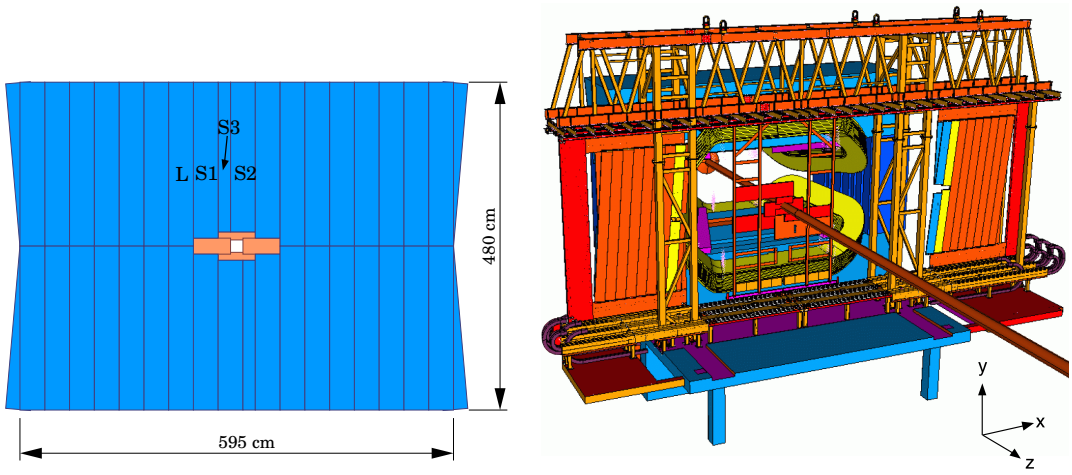
Figure 3.11.: *Geometry of an Inner Tracker layer. Figure from [19].*

layers in which two sensors have corresponding strips connected in series. In total, the Inner Tracker has 129,024 channels and an active area of about $4\,\mathrm{m}^2$. The strip pitch is $198\,\mu\mathrm{m}$, leading to a resolution very similar to the one found in the Trigger Tracker. The occupancy in the IT is below 2%.

3.1.4. Outer Tracker (OT)

The Outer Tracker (OT) is the second of the two tracking subdetectors behind the LHCb dipole magnet and covers the part of the LHCb acceptance that is not covered by the Inner Tracker. Figure 3.12 shows the situation. Like the Inner Tracker, the OT consists of three stations with four layers each; the stereo angle configuration follows the same pattern as the Inner Tracker. Each individual layer is composed of a left and a right half, each containing nine modules. Modules of two successive half-layers are mounted in a so-called C-Frame which can be retracted from its position next to the beam pipe to do maintenance work.

The Outer Tracker is a gas detector operating with a mixture of $\mathrm{Ar}/\mathrm{CO}_2/\mathrm{O}_2$ (70% : 28.5% : 1.5% per volume). Outer Tracker modules contain small drift tubes called *straws* of 5.0 mm outer diameter (4.9 mm inner diameter) which are mounted with a pitch of 5.25 mm. In the middle of each straw, an anode wire is put on a high voltage potential of



(a) *Outer Tracker layer layout* **(b)** *Isometric view of an Outer tracker station*

Figure 3.12.: *Outer Tracker geometry. (a) shows how an Outer Tracker layer is made up of modules of different sizes (the area covered by the Inner Tracker around the beam pipe is marked with a different colour). (b) shows an isometric view of the first Outer Tracker station; the modules for two layers are hanging in so-called C-Frames which form the left and right halves the subdetector and can be retracted (as shown) for maintenance work. Around the beam pipe, an Inner Tracker station with its cable ducts can be seen, the dipole magnet is visible in the background. Figures from [19] and [10], respectively.*

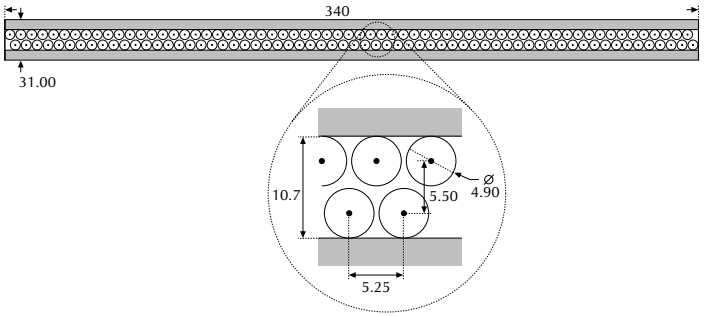
about 1550 V. The walls of the straws are made from conductive material.

Modules are typically 5 m long and are electrically subdivided in the middle to permit separate readout of upper and lower half of the module (modules in the middle of the detectors are shorter to leave a hole for the beam pipe). Straws are arranged in two monolayers per module as shown in Figure 3.13a, with each monolayer having 64 straws (again, in the middle where left and right detector halves touch, there are also narrower modules with fewer straws). The two monolayers are staggered by half the strip pitch to avoid the dead area in between two successive straws in the same monolayer.

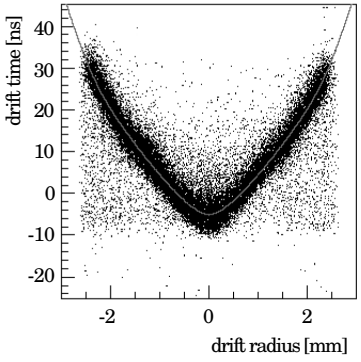
The arrival time of ionisation clusters at the wire with respect to the bunch clock is measured with a TDC; via a relation between drift time and the drift radius like the one in Figure 3.13b, the track position can be determined to about 200 microns. The mean occupancy of the detector can be up to 10% to 15% of the 53,760 channels. More details can be found in [10] and [20].

3.2. Particle Identification system

The particle identification system used by LHCb consists of three different types of detectors: Ring Imaging Cherenkov counters (RICH 1 and RICH 2), the calorimeter system and the muon system. Since these detectors do not play a crucial role for the



(a) *Cross section of an Outer Tracker module. Dimensions given in mm. Figure from [10].*



(b) *The relation between measured drift time and drift radius $r(t)$ is shown (as obtained during the 2005 testbeam campaign). Figure from [10].*

Figure 3.13.: *(a) gives a schematic view of an Outer Tracker module, in (b), the $r(t)$ relation for Outer Tracker is shown.*

work done in this thesis, their description will be brief.

3.2.1. Cherenkov detectors

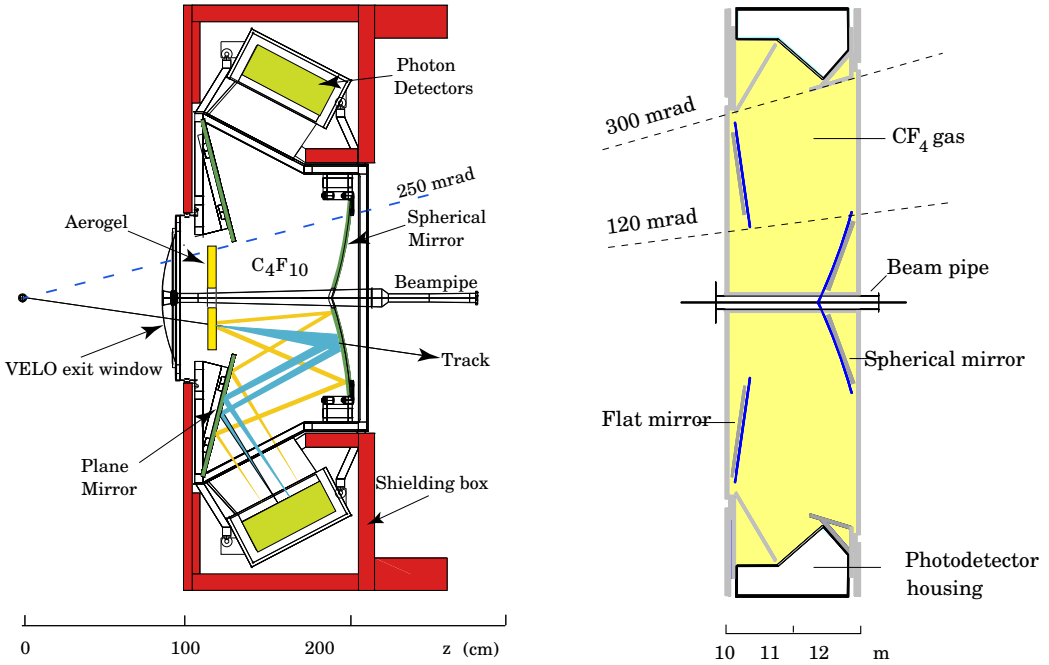
Cherenkov light is light emitted in a cone with opening angle θ_C around the trajectory of a charged particle travelling in a medium at a velocity larger than the velocity of light in that medium. Denoting the refractive index of the material (also called radiator) n and the velocity of the particle $\beta = v/c$, the opening angle θ_C is given by

$$\cos(\theta_C) = \frac{1}{n\beta} \tag{3.1}$$

In LHCb, there are two so-called Ring Imaging Cherenkov detectors which can identify particles of different masses if their momenta are known. The main application is to distinguish charged pions from kaons; often, such particle identification based information is needed to tell different B decay modes apart.

RICH 1 is situated between the vertex detector and the Trigger Tracker; it consists of two radiator materials, 5 cm of silica aerogel ($n = 1.03$) serve as first radiator providing π/K separation in the range of particle momenta from about 1 GeV to about 60 GeV, the second radiator material consists of 85 cm of C_4F_{10} ($n = 1.0014$) which covers the momentum range from 10 GeV to 60 GeV (see Figure 3.14). Emitted photons are detected with Hybrid Photon Detectors (HPDs) which record an image of the resulting rings of Cherenkov light formed by the arrangement of mirrors with a 2.5 mm × 2.5 mm granularity. From the radius of the rings, it is possible to infer θ_C, and, if the momentum of the particle is known from the tracking, its mass.

RICH 2 is situated between the main tracking system and the calorimeter system and is designed for higher momentum particles (which make it past the magnet). It uses CF_4



(a) *RICH 1* **(b)** *RICH 2*

Figure 3.14.: *RICH detectors; side view of RICH 1 (a), top view of RICH 2 (b). Figures from [10].*

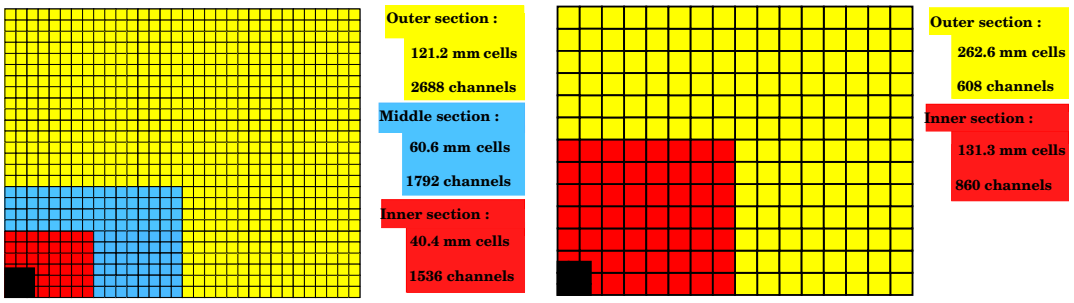
gas with a refractive index of $n = 1.0005$, the usable momentum range starts around 15 GeV and extends well above 100 GeV. Figure 3.14 has a top view.

The acceptance of RICH1 (RICH 2) is from 25 mrad to 300 mrad in x direction and from 25 mrad to 250 mrad in y direction (25 mrad to 120 mrad in x direction and up to 100 mrad in y direction). More details can be found in [10] and [21].

3.2.2. Calorimeters

The calorimeter system is designed to identify photons, electrons and hadrons, measuring their energies and positions. Information from the calorimeter is used in the hardware trigger, so some care must be taken to obtain measurements quickly enough and also with sufficient granularity to be useful in the trigger.

Particles interact with the calorimeter material to produce a cascade of secondary particles. When these secondaries are finally absorbed, the material is ionised. The calorimeters consist of alternating layers of absorber material in which a lot of secondaries are produced and scintillator material in which the ionisation produces light. This light is guided away from the scintillators with wavelength shifting fibres (WLS) and detected with photomultipliers. The number of photons detected is roughly proportional to the amount of energy deposited in the material. The calorimeter system has an acceptance of $\pm$ 300 mrad in x direction and $\pm$ 250 mrad in y direction.



(a) *ECAL/PS/SPD* **(b)** *HCAL*

Figure 3.15.: *Calorimeter segmentation; only one quarter is shown. Colour coding signifies regions of different segmentation (the uninstrumented area left for the beam pipe is black). Lengths of cell sides are given in the Figure. Figure from [10]*

	depth along z [mm]	interaction length X_0/λ_I	segmentation [mm^2] inner part	middle part	outer part
SPD	180	2.0/0.1	40.4 × 40.4	60.6 × 60.6	121.2 × 121.2
PS	180	2.0/0.1	40.4 × 40.4	60.6 × 60.6	121.2 × 121.2
ECAL	835	25/1.1	40.4 × 40.4	60.6 × 60.6	121.2 × 121.2
HCAL	1650	—/5.6	131.3 × 131.3	—	262.6 × 262.6

Table 3.1.: *Depth along z and corresponding electromagnetic and hadronic interaction length (X_0/λ_I) for different parts of the calorimeter system. The segmentation of the different systems is listed as well.*

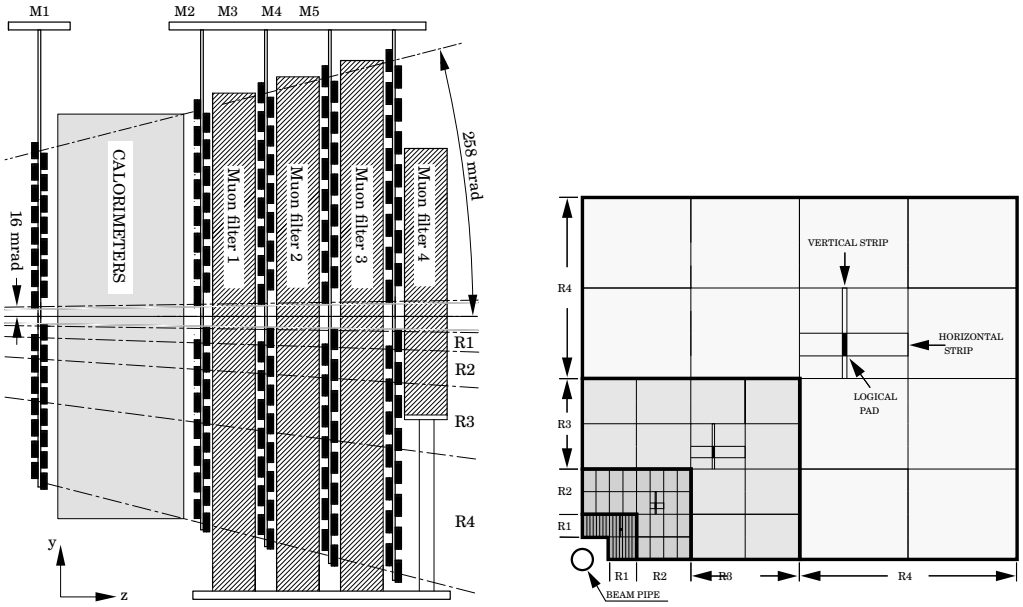
First, particles traverse the Scintillating Pad Detector (SPD); charged particles will produce scintillation light whereas neutral particles will not. After a 12 mm lead wall, the Preshower detector (PS) will cause photons and electrons to produce a shower, allowing hadrons and particles interacting electromagnetically to be distinguished. A shashlik type electromagnetic calorimeter (ECAL) follows, consisting of 66 alternating layers of 2mm of lead and 4mm thick scintillator tiles. This corresponds to 25 electromagnetic interaction lengths (X_0) and 1.1 hadronic interaction lengths (λ_I). Following this, the hadronic calorimeter (HCAL) is designed to absorb the remaining energy of hadrons. It is a sampling device made from layers of 1 cm of iron and scintillating tiles. In total, this sums to 5.6 hadronic interaction lengths (λ_I).

Figure 3.15 shows the segmentation of calorimeter tiles in SPD/PS/ECAL and HCAL depending on the region, with fine segmentation near the beam pipe where the particle density is high and lower granularity in the outward region. Table 3.1 lists the some characteristics of the different subdetectors in the calorimeter system.

The achievable energy resolution is

$$\frac{\sigma(E)}{E} = \frac{10\%}{\sqrt{E/GeV}} \oplus 1\% \quad \text{(ECAL)} \qquad \frac{\sigma(E)}{E} = \frac{80\%}{\sqrt{E/GeV}} \oplus 10\% \quad \text{(HCAL)} \tag{3.2}$$

More details can be found in [10] and [22].



(a) *Muon station layout* **(b)** *Muon tile geometry*

Figure 3.16.: *Muon stations; muon station layout (a), muon tile geometry (b).*

3.2.3. Muon system

The muon system consists of five stations, one in front of and four behind the calorimeter system which are used in the hardware trigger. It covers an acceptance of 20–306 mrad in the bending plane of the magnet and 16–258 mrad in the non-bending plane; this covers about 20% of the muons from inclusive semileptonic B decays. Figure 3.16 shows the layout of the muon stations; absorbers between M2–M5 are 80 cm thick and made from iron. The figure also shows the segmentation of the muon stations, there are four regions with different granularities (see Table 3.2 for details). Muon detection is done with multiwire proportional chambers (MWPCs), only the innermost region uses triple-GEM detectors to cope with the higher particle flux. The different granularities are achieved by combining anode wired and cathode pads into suitable groups called logical pads. In all regions, the spatial resolution in x direction is better than in y direction to help with momentum measurements, and M1 in front of the calorimeters helps to reduce the momentum uncertainty due to scattering in the absorber material of the calorimeters. A muon needs about 6 GeV momentum to successfully travel from M1 to M5.

Since the muon system is also used in the hardware trigger, a fast counting gas is required ($Ar/CO_2/CF_4$ in 40:55:5 parts per volume), and the detector technology used is also suitable for fast readout. More details on the detector and the readout system can be found in [10] and [23].

3.3. Trigger

LHCb employs a three stage trigger. The first stage is the hardware trigger, also called L0 trigger, which gets its input from the calorimeter system, the muon system, and the

region	M1 [mm^2]	M2 [mm^2]	M3 [mm^2]	M4 [mm^2]	M5 [mm^2]
R1	10 × 25	6.3 × 31	6.7 × 34	29 × 36	31 × 39
R2	20 × 50	12.5 × 63	13.5 × 68	58 × 73	62 × 77
R3	40 × 100	25 × 125	27 × 135	116 × 145	124 × 155
R4	80 × 200	50 × 250	54 × 270	231 × 290	248 × 309

Table 3.2.: *Sizes of logical muon pads per station and region.*

pile-up system located upstream of the Velo. The aim of the hardware trigger is to reduce the bunch crossing rate of 40 MHz down to 1.1 MHz, the maximum rate at which the detector can be read out. The hardware trigger uses the pile-up system to veto busy events. With the calorimeter system, the deposit with the highest transverse energy E_T is selected, and a particle hypothesis (electron/photon/hadron) is assigned according to pattern left in SPD and PS. Only the highest candidate of each of the three categories is kept; if it is above threshold, it causes the event to trigger. The muon trigger selects muons with high transverse momentum p_T; p_T is reconstructed by forming a track from muon hits in all five stations and estimating p_T under the assumption that the track came from the interaction point (with about 20%–30% relative resolution). Again, if the reconstructed p_T is over a configurable threshold, the event is triggered.

The next two stages of the trigger, HLT 1 and HLT 2, run on a computing farm (HLT stands for High Level Trigger). The task of HLT 1 is to quickly confirm or reject the L0 trigger decision by partial event reconstruction in a region around the object that caused L0 to trigger, reducing the rate by roughly a factor 30. Section 7.1 discusses how the track reconstruction algorithm implemented for this thesis accomplishes part of the necessary reduction by taking muon triggers as an example. For surviving events, HLT 2 has sufficient time to perform a (fast) full event reconstruction, selecting either specific decay modes (triggering exclusively on certain decays) or triggering inclusively by selecting special event topologies (e.g. heavy displaced two-, three- or four-prong vertices). The rate must be reduced down to about 2 kHz which can be written to tape.

More details on the trigger can be found in [10] and [24], and the L0 confirmation process is also explained in some detail in Section 7.1 in this thesis.

3.4. Software framework

LHCb software runs within the GAUDI framework (see [25]) and consists of several applications:

- GAUSS: GAUSS is the generation and simulation package used by LHCb (see [26]). Internally, it uses PYTHIA ([27]) to generate events, B decays are generated using EVTGEN ([28]). The propagation of the resulting particles through the material of the detector and the magnetic field is done with GEANT4 ([29]).

- BOOLE: BOOLE (see [30]) takes the output of GAUSS and simulates the detector response in the sensitive area of the individual subdetectors. This process is also known as digitisation. Wherever possible, this process relies on results from test

beam data or calibration using data taken more recently with the full detector. The resulting output data has the same format as the data gathered by the real detector, and the aim is of course that simulated events should be as close as possible in its properties to the data taken with the LHCb detector.

- MOORE: MOORE is an application to run the software part of the trigger (see [31]). It is used both in the computing farm which processes data from the detector and for trigger emulation on simulated events.

- BRUNEL: This is the reconstruction package ([32]). Based on hit level information (either from the detector or from simulated events), tracks are reconstructed and fitted. Then, particle identification algorithms are run on the resulting tracks, and the result is written out.

- DAVINCI: The final step is the physics analysis software ([33]) which combines different stable particles to form the decay chain one is interested in. Different selection cuts can be applied, and the results can be written either to summary data files which can be re-read using any of the LHCb software applications, to tuples, or plots can be output to ROOT files.

- PANORAMIX: This is the visualisation software used by LHCb ([34]); it can display, for example, detector geometry, individual hits in subdetectors, tracks and vertices. It can decode and reconstruct data by itself, so it can be run independently of MOORE, BRUNEL and DAVINCI; this makes it a valuable tool to understand the detector and the data.

Most of the software is written in C++, although there are also bits written in other languages like Python.

Part I.

Tracking

4. Tracking in the LHCb experiment

LHCb is an experiment which aims at studying decays of hadrons containing b quarks. This is reflected in the entire detector design which favours very good vertexing and momentum resolution in the forward region in which a large fraction of all produced $b\bar{b}$ pairs travel. It is also reflected in the tracking strategy employed by LHCb which is characterised by complementarity in pattern recognition algorithms and an emphasis on keeping the reconstruction flexible enough to run fast versions of many offline reconstruction algorithms in the software trigger. These traits are due to several characteristics of decays containing b quarks:

First, many interesting physics channels (e.g. $B_s \longrightarrow J/\psi(\mu\mu)\phi(KK)$) decay into final states with four or more charged particles. This means that for an average tracking efficiency ε and n charged tracks in the final state, the efficiency to reconstruct the whole decay is ε^n, so a high track reconstruction efficiency is required.

The second point has to do with the fact that due to the relatively low mass of the b quark (compared to the mass scales in direct Higgs or Beyond the Standard Model physics searches which form a focus of activities at the general purpose detectors at LHC), B decay products look very similar to the overwhelming soft QCD background that is present in the detector.

Apart from the displaced vertex due to the long B lifetime, the main distinguishing factor between B daughter tracks and the QCD background is a slightly harder p_T spectrum. Figure 4.1 shows the p and p_T spectra of charged b daughter particles in the LHCb acceptance region[1] in 50,000 simulated inclusive $b\bar{b}$ events together with the spectra of charged particles in the LHCb acceptance from 50,000 simulated minimum bias events.

Both spectra in Figure 4.1 are made with the same number of events, however, the total inelastic pp cross section at $\sqrt{s} = 7$ TeV is on the order of 60 mb while the $b\bar{b}$ production cross section is only on the order of 300 μb ([12]). This pushes down the spectra for B daughters by a factor of about 200 with respect to what is shown in the plot.

This translates into the need to have highly selective triggers which exploit displaced vertices and/or partially reconstruct the event to look for interesting B decays. The hardware trigger cannot read out the entire detector at the rate at which pp collisions occur; the data transfer rates are not manageable. Instead, a few subdetector systems (muon system, calorimeters and pile up veto) are read out at the full speed while a certain backlog of events is kept in the readout frontend chips. The hardware trigger can reduce the rate sufficiently based on partial information. In case of a positive decision by the hardware trigger, a signal is sent to the readout front end chips which will in turn send their part of the detector data to one of the nodes of the software trigger farm of about

[1] In fact, we require the tracks to be "long track reconstructible" (see 4.3.2), a slightly stronger requirement; for the discussion here, the difference between being in the acceptance and being long track reconstructible is negligible

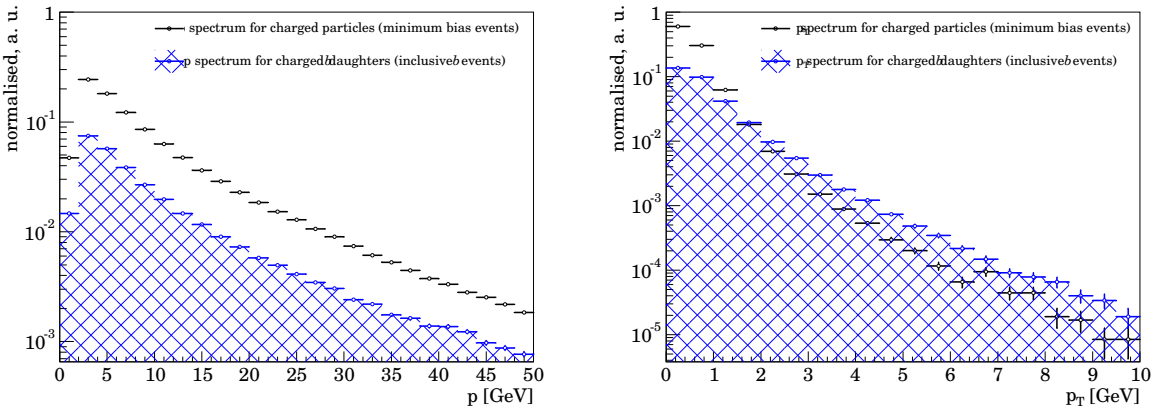


Figure 4.1.: *p and p_T spectra for charged particles in a minimum bias sample (black curve) and for charged b daughter tracks from an inclusive $b\bar{b}$ sample (blue curve, area below is hatched). Both samples contain 50,000 events. The minimum bias spectra have been scaled to have unit area. The same scaling factor was applied to the inclusive $b\bar{b}$ spectra to illustrate the amount of QCD background with which B daughters have to compete which is interesting for trigger purposes. The reason why the spectrum for B daughter tracks is typically below the one for minimum bias is that the B daughter tracks are only fraction of all the tracks in events which do contain a B. The shape of the spectra is rather similar, although the p_T spectrum of the B daughters is slightly harder than the one for minimum bias.*

2,000 cores. There, the 1.1 MHz output rate of the hardware trigger must be reduced to about 2 kHz which can be written to tape. This has to happen in less than 2 ms per event (on average).

Reconstruction software must therefore not only accommodate the need to provide efficient offline event reconstruction, it must also be flexible enough to provide shortcuts which meet the timing requirements in the software trigger and can be activated at the "flip of a switch". The aim of this chapter is to give an overview over and to introduce the ideas behind the algorithms in the LHCb track reconstruction. It also serves to introduce metrics to judge the performance of the pattern recognition algorithm introduced in chapter 5.

4.1. General overview

The track reconstruction as implemented in LHCb software consists conceptually of three distinct stages:

- The *pattern recognition* recognises the patterns of detector signals (e.g. silicon strips or Outer Tracker straws with charge deposits over the respective readout thresholds — these will be called "*hits*") typically produced by charged particles on their way through the detector and forms tracks from these hits.

- The *track fit* uses a Kalman filter ([35], [36], [37], [38]) to obtain the best possible estimate of the true trajectory of the corresponding particle, incorporating corrections due to energy loss and multiple scattering. Typically, the CPU time spent in fitting tracks is about four times of what is needed for pattern recognition.
- An additional stage follows which removes tracks with fit failures and duplicate tracks (which exist either as short version of a longer track or because there is a certain level of redundancy in the reconstruction, i.e. a track can be found by more than one pattern recognition algorithm).

4.2. Track types and tracking strategies

The aim of this section is to introduce different types of tracks reconstructed by the LHCb experiment and pattern recognition strategies, and to familiarise the reader with their basic properties.

4.2.1. Track types and representation

It is useful to have names for tracks which satisfy some common criteria, so a few track types are introduced. The names follow the mode of speech used by the collaboration. Figure 4.2 gives a graphical impression of the track types.

- Velo tracks: tracks which use measurements in the vertex detector (Velo) only (in both r and ϕ sensors); these tracks serve as starting point to form longer tracks and for vertexing
- T tracks: tracks with measurements in the main tracker ("T stations") which are sometimes also called Seed tracks because these tracks can be used to "seed" the long track reconstruction
- Long tracks: tracks which go through the whole detector, from Velo to T stations (this type of track is most useful for physics analyses since it traverses the entire magnetic field and therefore has the most accurate momentum information)
- Upstream tracks: tracks with hits in the Velo and the TT stations only (tracks of this type are often low momentum tracks which are bent out of the acceptance by the magnet)
- Downstream tracks: tracks with hits in the TT and T stations only (downstream tracks often arise from the charged decay products of long-lived resonances such as the K_S^0)

Each track is represented by a set of *state vectors* $\vec{s}_i = (x, y, tx, ty, q/p)^T$ consisting of x and y coordinates, slopes in $x-z$ and $y-z$ projections and the inverse track momentum times its charge q at certain z positions z_i along the track. Together with the associated covariance matrices, this representation captures the knowledge about both the track trajectory and its uncertainty.

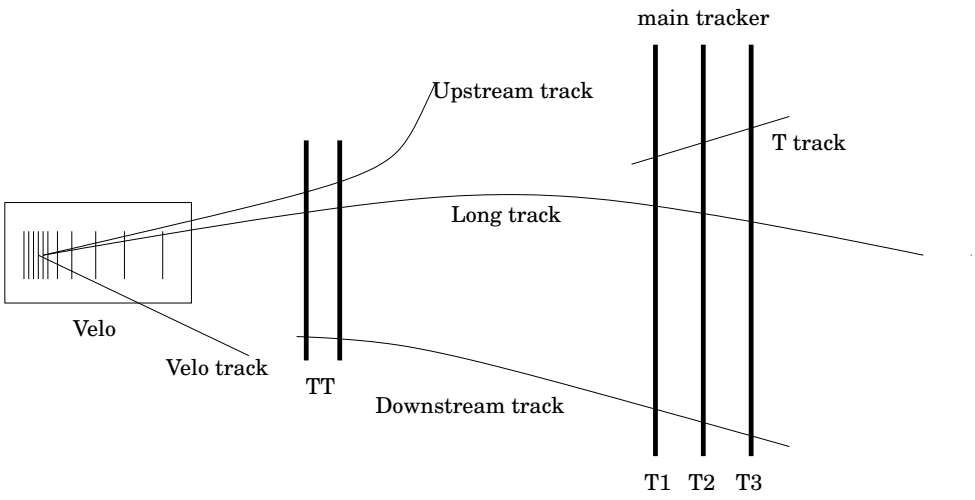


Figure 4.2.: *Schematic view of the track types in the LHCb tracking system.*

4.2.2. Tracking in the vertex detector

Pattern recognition in the Velo is relatively straightforward in the sense that the magnetic field in the Velo is negligible, so it is sufficient to look for straight line tracks. The task of Velo pattern recognition algorithms is complicated by the r-ϕ geometry of the Velo sensors and the fact that the algorithms have to find tracks when the two halves of the Velo are open.

Information on the Velo reconstruction algorithms used until the end of 2010 can be found in [39], [40] and [41], from 2011 on, a faster and more flexible algorithm will be used ([42]).

4.2.3. Standalone track reconstruction in the main tracking system

The main tracker is inside the fringe field of the LHCb dipole magnet, so the pattern recognition has to account for track curvature. There are two competing pattern recognition algorithms (both of which are described in Chapter 5 in more detail) which are very similar in their approach. TsaSeeding has been around for a long time and was the default in the reconstruction for reconstruction of data and simulation until the end of 2010. Its sibling, PatSeeding, has been developed as part of the work for this thesis from a prototype by O. Callot into a mature, efficient and very fast algorithm to accomplish the same task. Due to its speed, PatSeeding has been the default algorithm in the software trigger from the start of data taking by LHCb, and it will become the default pattern recognition algorithm for T station standalone reconstruction also for the offline reconstruction from 2011 on.

4.2.4. Reconstruction of tracks traversing the entire spectrometer

Hough transform based approach (“Forward tracking”)

The Forward tracking is one of two methods to reconstruct long tracks. Its name stems from the fact that in the process of pattern recognition the algorithm moves “forward”

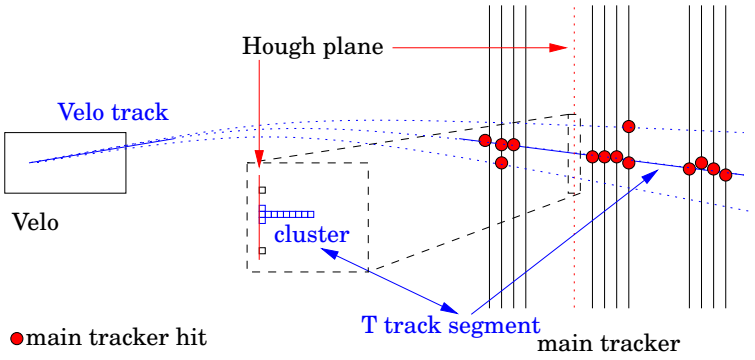


Figure 4.3.: *Sketch illustrating Forward tracking: Based on a Velo track and a main tracker hit, one can calculate the position of a (hypothetical) particle in a plane inside the main tracker. Hits from true tracks produce a cluster of very close positions, while the position of other hits in that plane (e.g. noise hits) is more isolated. The Hough space for Forward tracking is the position in this Hough plane.*

along a track from the end of the Velo to the end of the main tracker.

The Forward tracking algorithm employs a *Hough transform* to find long tracks: Basically, one seeks a method to transform the space of observables (i.e. measurements or hits) to a more abstract space, the *Hough space*. The transformation is picked such that the mutual distances of measurements from the same particle are small in the transformed space. The concept will be illustrated using the Forward tracking as example.

The idea behind the Forward tracking is that in the absence of scattering and energy loss, the trajectory of a particle in a magnetic field is completely determined by the knowledge of the field and the equations of motion of a charged particle in it. Only the direction of the particle at one position along its trajectory and its momentum need to be known, or, alternatively, the direction at a position in front of the magnet and a second position behind the magnet. Once these quantities are known, the position and heading of a particle can be calculated at any point along its trajectory.

The algorithm exploits this fact by calculating the x position of the trajectory defined by a Velo track and a T station measurement in a plane at $z = 8520\,\text{mm}$ which will also be called the Hough plane. If the Velo track and a T measurement were produced by the same particle, the x position calculated will be close to the true x position of that particle in the Hough plane, if not, the result of the calculation is more random, see Figure 4.3. Thus, the method will form a cluster in the calculated Hough plane x positions from T measurements belonging to the same particle, provided that the correct Velo track was used for the calculation.

It is sufficient to focus on the x component of the trajectory because the main part of the deflection is in that direction. The field acting on the y component is negligible for this discussion.

To reduce algorithm execution time, an effective parametrisation is derived which takes track parameters of a Velo seed and the x and z positions of a measurement in one of the T stations, giving the x position in the Hough plane of the magnet. Details on how such an effective parametrisation is obtained and how the algorithm proceeds in detail

can be found in [43] or [44], for example.

Track matching

Track matching is the second algorithm to reconstruct long tracks. In contrast to the forward tracking above, it relies on T tracks which have been produced by TsaSeeding or PatSeeding. Since these tracks have a momentum estimate, they can be propagated through the magnet using numerical integration of the equations of motion. They are matched to Velo tracks in a matching plane at the exit window of the Velo (around $z \approx 87$ cm).

This matching is done using a quantity called χ^2_{match}:

$$\chi^2_{match} = (x_{Velo} - x_T)^T (C_{Velo} + C_T)^{-1} (x_{Velo} - x_T)$$

Here, x_{Velo} and x_T are the track parameters of the Velo track and the extrapolation of the T track into the matching plane, C_{Velo} and C_T are the corresponding covariance matrices. The T station part of the track is fitted using the Kalman Fitter before the Matching algorithm runs, so x_T and C_T are available. For x_{Velo} and C_{Velo}, one has to rely on estimates provided by the Velo pattern recognition. So in principle, χ^2_{match} just measures how well the two sets of track parameters agree in the matching plane. If the pair passes selection cuts, a single Long track is formed from the two tracks. A search for TT hits which are compatible with the track found follows. Details can be found in [45] and [46].

There is also a faster version of the algorithm, called PatMatching (the slower variant is called TrackMatching), which uses a suitably weighted sum of the squared differences in positions and slopes of both extrapolations in the matching plane. The advantage here is that the tracks need not be fitted since the covariance matrices are not needed, offering potential time savings both offline and in the software trigger.

4.2.5. Downstream tracking

Downstream tracks are reconstructed by using a T track to define a region in which potential TT hits belonging to that track might exist ("region of interest") and using a Hough transform on any contained hits to form a track. The idea is in principle the same as for the Forward tracking, the main difference is that the algorithm starts with a track on the other side of the magnet. Downstream tracks are important to reconstruct tracks of long-lived particles like K^0_S and Λ which often do not decay before they leave the vertex detector.

It should be noted that the Downstream tracking has a more difficult task to accomplish than the Forward tracking because the TT consists of only four layers of silicon, so there is not much redundancy in TT to give confidence in a collection of hits (the two x layers are needed to define the track slope and intercept in the $x-z$ plane, and one of the two stereo layers is needed to obtain the y position of the track). Moreover, the track density is higher than in the main tracker because the magnet acts as a kind of "momentum filter" which prevents very low momentum tracks from reaching the main tracker by either bending them out of the acceptance or causing them to curl inside the magnet.

More details on the Downstream tracking can be found in [47] and [48].

4.2.6. Upstream tracking

Upstream tracks are reconstructed from Velo tracks and TT hits. In a manner very similar to the Downstream tracking, each Velo track defines a region of interest in TT; hits contained in that region are again subjected to a Hough transform. The difference with respect to Forward and Downstream tracking is that one has to deal with an "effective magnet" which consists of the stray magnetic field between the Velo and the end of TT. The algorithm is described in detail in [49], with tunings for application in the software trigger and offline reconstruction. In the former case, the added momentum information can used to trigger on displaced high p_T tracks; however, in the present incarnation of LHCb software, the p_T information in the trigger typically is extracted using long tracks. Within the framework of offline reconstruction, the focus is on recovering low momentum tracks which did not make it through the magnet. Just as for the downstream tracking, the high track density and the lack of redundancy in TT is the main challenge.

4.2.7. Momentum measurement

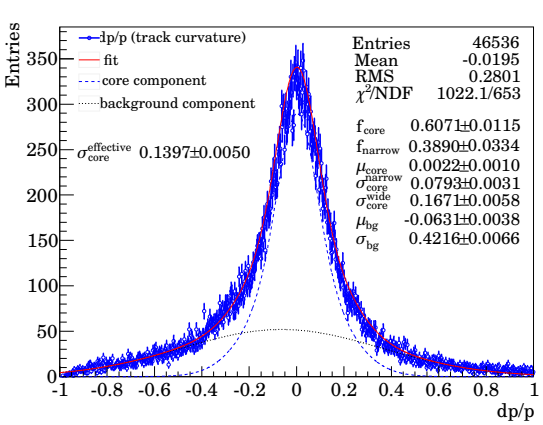
In the LHCb detector, track momentum is measured with different methods depending on the application context (trigger/offline reconstruction). This subsection serves to introduce these methods and give a feeling for their respective performance. Specifically, the following methods are investigated:

- Momentum from track curvature: In the main tracking system, track curvature is inversely proportional to track momentum, so the calculation is extremely fast (a single division). This method does not take into account local field variations, energy loss or multiple scattering, so the relative momentum resolution $dp/p = p_{\text{reconstructed}}/p_{\text{true}} - 1$ is relatively poor. For T tracks, $dp/p \approx 14\%$ (cf. Figure 4.4a), and there is a large tail which is mostly due to local field variations being ignored.

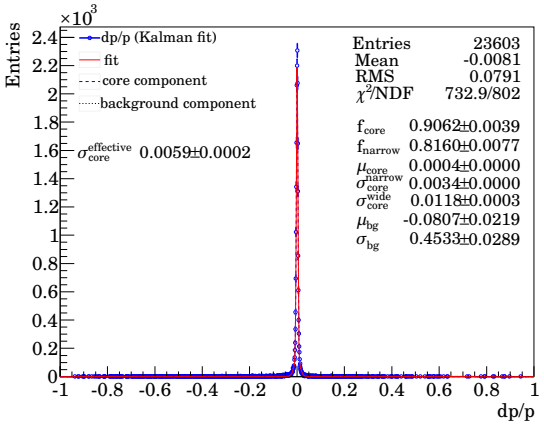
- Kalman filter: The full Kalman filter fitted tracks have a momentum estimate which includes the full information about the magnetic field and is corrected for energy loss and multiple scattering; For long tracks, $dp/p \approx 5.9‰$ (cf. Figure 4.4b); T tracks have a $dp/p \approx 8.1\%$ which is better than the estimate obtained from T track curvature above because local field variations are taken into account, but a lot worse than dp/p for long tracks because T tracks do not have any information about the track position in front of the magnet.

- Momentum from Forward tracking parametrisation: The parametrisation used by the Forward tracking can be used to predict track momentum from x and y positions and slopes of a track at a given z. T tracks are assumed to come from the origin which results in a $dp/p \approx 2.6\%$ (Figure 4.4c). For T tracks, the parametrisation together with the assumption of the track coming from the origin is very good compared to the two other methods available, and the calculation is still very fast because it only consists in the evaluation of a few polynomials.

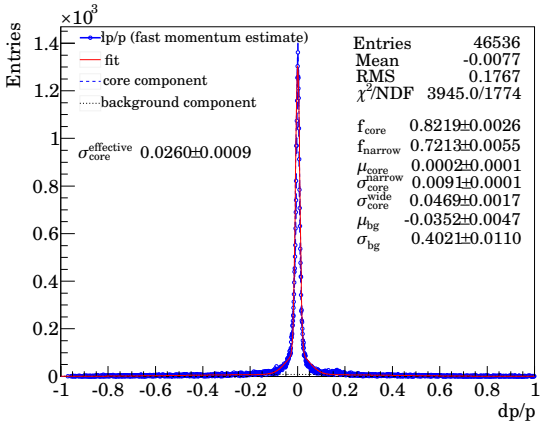
The distributions of relative momentum resolution $dp/p = p_{\text{reconstructed}}/p_{\text{true}} - 1$ shown in Figure 4.4 are characterised by a peak, which I will call the *core* of the distribution, and some background which is sitting below the core and has a much broader distribution.



(a) dp/p *from T track curvature*



(b) dp/p *from Kalman fit*



(c) dp/p *from parametrisation*

Figure 4.4.: *Relative momentum resolution dp/p for the measurement of track momentum determined from (a) track curvature of a T track, (b) using the Forward tracking parametrisation on T tracks, (c) result of the full Kalman filter fitted track for long tracks. See Appendix A for the precise meaning of the parameters.*

In order to give some quantitative results on the narrower peaking component, a fit is performed to disentangle core and background. The plots contain the full set of fit parameters on the right; the detailed meaning of the parameters and the fit method are described in Appendix A. Usually, the aim of the fit is to extract an effective width of the core, $\sigma_{\text{core}}^{\text{effective}}$, and sometimes also the fraction of entries in the core, f_{core}, so a detailed understanding of the fit model is not required to understand the point of the plot. It is worth noting that the fit need not be perfect as long as it roughly describes the shape of the distribution, since we are only interested in obtaining an estimate of the core width and the corresponding fraction of entries.

If the time taken by the Kalman fitter can not be tolerated or the high precision is not needed, the faster methods mentioned above can be used. Inside the pattern recognition code, the inverse proportionality between track curvature and momentum can be used to quickly adjust the region searched for hits depending on estimated track momentum, while the parametrisation based on the Forward tracking is used extensively in the software trigger and to obtain initial starting parameters for the Kalman filter fit.

4.2.8. Final track output

Since there are two algorithms which reconstruct long tracks and several ways to obtain shortened versions of long tracks (e.g. for each reconstructed long track with hits in TT, it should be possible to find a corresponding downstream track), there must be a way to chose the longest (most hits) possible alternative of each track. In LHCb, it is the task of the *Clone Killer* to check the output of all reconstruction algorithms for such clone track and compile a combined track sample as final output of the track reconstruction. To check for clone tracks, the Clone Killer checks for common hits, if two tracks share more than a certain threshold value (see 4.3.4 for details on the precise definition), the shorter of both tracks is discarded. More details on the Clone Killer can also be found in [50] or [51].

For completeness, it seems prudent to name the order in which the different stages of the pattern recognition are run during the normal offline reconstruction: The process starts with the Velo reconstruction which is followed by the Forward tracking to produce long tracks. One of the standalone main tracker reconstruction packages (either TsaSeeding or PatSeeding) is run next, to be followed by Track Matching which also produces long tracks. Then, Downstream and Upstream tracking are run in that order. Finally, the Clone Killer is used to produce the final output as described above.

In older versions of LHCb software, each pattern recognition algorithm was immediately followed by an instance of the Kalman filter producing fitted tracks for subsequent use. Since the fitting takes the major share of the total time spent in track reconstruction (typically more than five times the time taken by the pattern recognition), an effort was made to work with unfitted tracks during track reconstruction, postponing the fitting procedure until after the Clone Killer has produced its final output. This has helped reduce the time spent in track fitting significantly, since fitting time is no longer wasted on clones.

4.3. Performance indicators

4.3.1. Hit purity, hit efficiency, association to Monte Carlo truth

The hit purity of a track is defined as the fraction of hits on the track which are produced by the charged particle which gives rise to the track:

$$\text{hit purity} = \frac{\text{\# of hits produced by corresponding particle}}{\text{total \# of hits on track}}$$

Likewise, the hit efficiency of a pattern recognition algorithm is the efficiency with which it manages to pick up the hits which belong to a track that was found:

$$\text{hit efficiency} = \frac{\text{\# hits on track produced by corresponding particle}}{\text{all hits produced by corresponding particle}}$$

While these definitions seem to be intuitively clear and straightforward, the notion of a "corresponding particle" has to be clarified before the definitions make sense.

On real data, it is quite difficult to tell with any certainty if an individual hit which happens to be on a track is actually caused by the "corresponding particle" (which may not even exist if the pattern recognition accidentally forms a track from unrelated hits). While the notion of hit efficiency and purity is clearly an important and convenient concept when thinking about tracks in data, no part of this thesis actually needs these quantities for data, so hit purity and hit efficiency will be defined formally in terms of Monte Carlo truth only.

To arrive at such a definition, it is convenient to define the fraction of hits a particle p contributes to all hits of a track t as the weight $w(t,p)$ with which that particle contributes to the track. As an example, Figure 4.5 shows the distribution of the maximum weight (i.e. $\max_p w(t,p)$) for Seed tracks[2].

By convention, a Monte Carlo particle p is said to be *associated* to a track t if its weight $w(t,p)$ is at least 0.7. While the cutoff value of 0.7 in the definition above may seem somewhat arbitrary, it should be noted that the distribution of maximum particle weights is concentrated in the peak at weight 1, so the definition above will not mean something radically different if the cutoff is shifted a bit in either direction.

With this definition, it is possible to write down well defined expressions for hit purity and hit efficiency[3]:

$$\text{hit purity of track t} = \max\left(0, \max_{\substack{\text{particles } p \\ \text{contributing to } t}} w(t,p)\right)$$

For a track t that is associated to a particle p, one can define the hit efficiency as

$$\text{hit efficiency} = \frac{\text{\# of hits from } p \text{ on } t}{\text{total \# of hits from } p}$$

[2] A maximum weight of 0 is assigned if no hit on the track was caused by a particle (e.g. for a track formed from noise hits alone).

[3] In the formula above, the value of 0 is used when there are no particles contributing to the track.

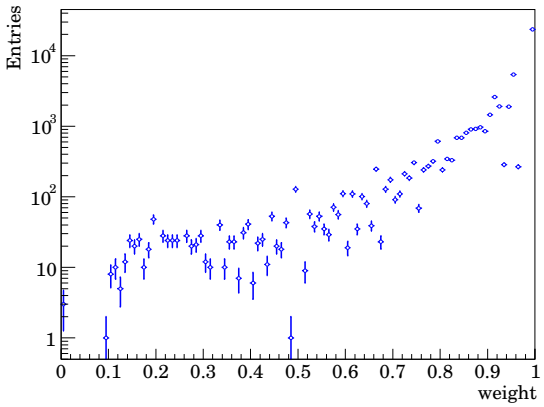


Figure 4.5: *Distribution of maximum (particle) weight distribution for Seed tracks. There is a large peak at weight 1 for well reconstructed particles which dies out quickly when moving away from 1. Note the logarithmic scale.*

4.3.2. Track reconstruction efficiency

The track reconstruction efficiency is the probability to reconstruct a charged particle travelling through the detector, and can be determined by forming the ratio of the number of *reconstructed* charged particles $N_{\text{reconstructed\&reconstructible}}$ among the the number of *reconstructible* charged particles $N_{\text{reconstructible}}$ (those which can reasonably be expected to be reconstructed):

$$\varepsilon_{tr} = \frac{N_{\text{reconstructed\&reconstructible}}}{N_{\text{reconstructible}}}$$

There are several possible definitions in use which differ in what goes into the numerator and denominator of the fraction above, for example one can look at tracking efficiencies in different subdetectors, often as function of quantities like the momentum of the charged particle or the total multiplicity in the event.

Broadly speaking, such definitions fall into two classes:

On the one hand, there are definitions which use Monte Carlo truth and thus work only on simulated events. This kind of definition is mostly used to develop and validate pattern recognition algorithms and is explained in more detail below.

On the other hand, there are data-driven methods to measure efficiency. They typically use some form of the “tag-and-probe” approach. The idea is to “tag” a track which should be present in one subdetector with information from a subdetector which is not under investigation and then “probe” if the track is actually found in the subdetector of interest. The advantage here is that the same method can be applied unmodified to both real and simulated events, thus enabling data/Monte Carlo agreement to be checked with the same method. Several examples of this approach can be found the the analysis part of this thesis.

When developing pattern recognition algorithms, the class of Monte Carlo based definitions is most frequently used because the truth is known and can be used to help to isolate problems.

To better understand which Monte Carlo particles produce tracks which can reasonably be expected to be found by a pattern recognition algorithm, it is useful to define “reconstructible” particles:

- a particle is reconstructible in the Velo if it has at least three r and three ϕ hits

- a particle is reconstructible in the main tracker ("T reconstructible") if it has at least one x and one stereo hit in each station
- a particle is reconstructible as long track if it is reconstructible in both the Velo and the main tracker
- a particle is reconstructible as downstream track if it is reconstructible in the main tracker and has hits in three of the four layers of TT
- a particle is reconstructible as upstream track if it is reconstructible in the Velo and has hits in three of the four layers of TT

It is important to realise that a particle can be in the acceptance of a subdetector and even produce a track there that is reconstructed while not being reconstructible in that subdetector. This usually happens for tracks which have too few hits in a subdetector to be found reliably because they enter or leave the acceptance of that subdetector prematurely. By excluding these particles, the efficiency denominator is cleaned up somewhat, and it is possible to focus on tracks which have to be found during development of pattern recognition algorithms.

For the very same reason, one also excludes particles which interact hadronically before they reach the calorimeter. Electrons are also excluded because they tend to emit Bremsstrahlung which makes things more difficult for the pattern recognition algorithms. Unless explicitly noted otherwise, any tracking efficiency quoted for simulated events in this thesis has these restrictions applied.

There is one additional complication: Rare events with very high particle multiplicities tend to be more difficult to reconstruct. Since these events have more tracks than the average event, the *track-averaged* reconstruction efficiency calculated by counting reconstructed and reconstructible particles and dividing is biased towards lower efficiencies, i.e. the track-averaged reconstruction efficiency tends to be lower than the reconstruction efficiency in the average event.

To work around the feature, one can define the *event-averaged* track reconstruction efficiency which is calculated by averaging the per-event track reconstruction efficiencies over all events.

4.3.3. Fake track fraction ("ghost rate")

A certain fraction of reconstructed tracks is not associated to a Monte Carlo particle (these tracks arise from combinations of unrelated hits which sometimes pass the quality cuts in pattern recognition algorithms). The *fake track fraction* (colloquially also referred to as "ghost rate") is the fraction of these tracks in the entire output of a pattern recognition algorithm.

These fake track fractions are typically on the level of a one to four percent when reconstructing the Velo or the main tracker standalone. Since there are no tracking stations inside the magnet, the matching of track segments in front of the magnet (which have poor momentum information) and the ones behind the magnet introduces an additional uncertainty which raises the fake track fraction to about 15 percent for long tracks.

High multiplicity events tend to produce more fake tracks than low multiplicity events. For this reason, it makes sense to distinguish track-averaged and event-averaged fake track fractions, defined in very much the same way as above for track reconstruction efficiencies.

4.3.4. Clones

Clones arise when there are several tracks which are associated to the same particle. Typically, the longest of these tracks (i.e. the one with most hits) is used for subsequent analysis because it contains more information than the other tracks, the others are called clones. Naturally, the fraction of clone tracks in the output of a pattern recognition algorithms should be low to avoid unnecessary work in later stages of the reconstruction and double-counting later in the analysis stage. Clone fractions are below the percent level in the output of pattern recognition algorithms.

The clone definition above can be extended to work without Monte Carlo truth information. There are two ways to do this, and both ways have their applications in LHCb software:

- One can define two tracks to be clones if they share a certain fraction of hits. The convention used by the Clone Killer mentioned above is that two tracks are clones if they share 70 % of the hits on the shorter track. (For tracks spanning several subdetectors, the criterion is applied to each subdetector separately, and the two tracks are called clones if they are clones in each subdetector.) This definition catches clones which share hits and is used to discard short versions of tracks (e.g. a Velo-only track when a long track with the same Velo hit content is available) and tracks which have been found by both of the long track reconstruction algorithms.

- One can also define two tracks as clones if their track parameters are compatible within their uncertainties. This method also catches tracks which do not share hits and is interesting in physics analyses which cannot tolerate a small clone track contamination (e.g. searches).

4.3.5. Hit residuals

To check how well the track fits work which are done inside the pattern recognition, one can look at the (signed) distance $r = x_{\text{hit}} - x_{\text{track}}$ between the reconstructed track and a hit which is also called *residual*. Such a distribution is shown for Outer Tracker hits in Figure 4.6 for simulated events. If the hit for which the residual is calculated is included in the fit, the residual is biased towards smaller values because the hit pulls the track towards itself. If the hit for which the residual is calculated is excluded from the fit, the residual is called unbiased. The residuals in Figure 4.6 were calculated from the fit inside the pattern recognition.

Thus, the residuals give an impression how close the hits are to the track. Assuming the track fit works, it is possible to learn something about the resolution of the detector in question by observing the width of the residual distribution.

To judge if the fit works properly, one can also look at the *pull* r/σ where σ is the hit resolution. If the fit works properly (and the detector resolution is well described), the

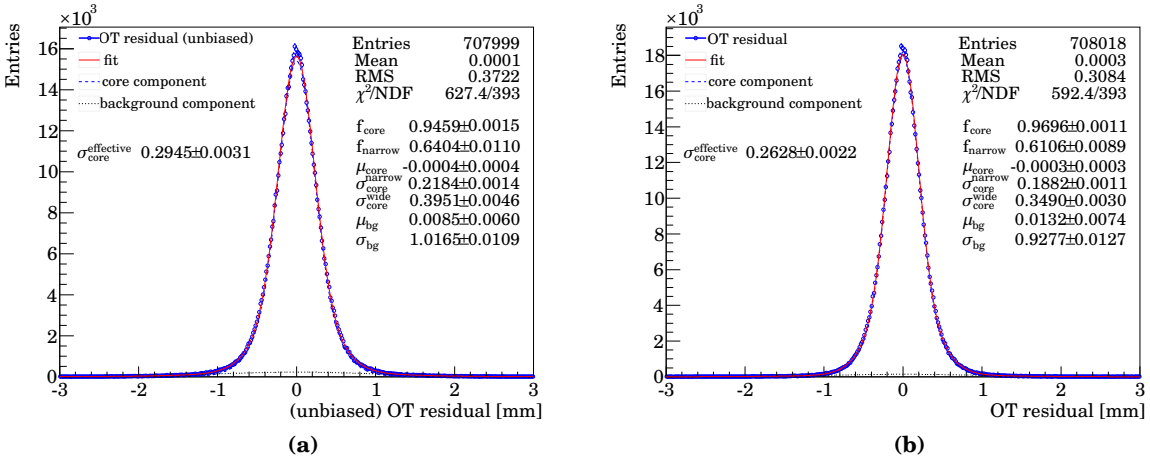


Figure 4.6.: *Unbiased (a) and biased (b) Outer tracker hit residuals in simulated events. The fits and parameters are explained in Appendix A. The width of the unbiased residual distribution is slightly wider, as expected.*

resulting distribution should roughly be a Gaussian centred at zero with unit width. Figure 4.7 shows biased and unbiased pull distribution for the Outer Tracker. The widths are close to unity, so the detector resolution appears to be well described in simulation events; this is expected and reflects the fact that the reconstruction uses the same resolution model parameters which the detector simulation uses.

4.3.6. Practical application

To fill the preceding definitions with life, it is probably best to show some tables and figures to serve as an example. I chose the final output of the track reconstruction step because it serves as starting point for any physics analysis. Specifically, long reconstructible tracks will be investigated. For performance evaluations based on simulated events, the aim is usually to better understand the behaviour of the pattern recognition algorithms. For this reason, the definitions used for tracking efficiency, ghost and clone rates are those based on Monte Carlo truth given in the previous subsections. Table 4.1 shows the performance obtained with the default settings for 2010 Monte Carlo; the sample[4] used contains 50,000 simulated inclusive $b\bar{b}$ events. Track reconstruction efficiencies are given as track-averaged quantities in these kind of tables. One is especially interested in the following subcategories of tracks:

- High momentum tracks ($p > 5$ GeV): High momentum tracks are interesting because they are less affected by multiple scattering and thus give an impression of the performance under optimal circumstances.
- B daughters: Since LHCb is a dedicated b physics experiment, one is naturally interested in the B daughter efficiency (for daughters which are inside the LHCb acceptance).

[4] `2010-Beam3500GeV-VeloClosed-MagDown-Nu1-Sim03Reco03-withTruth-10000000-incl_b`

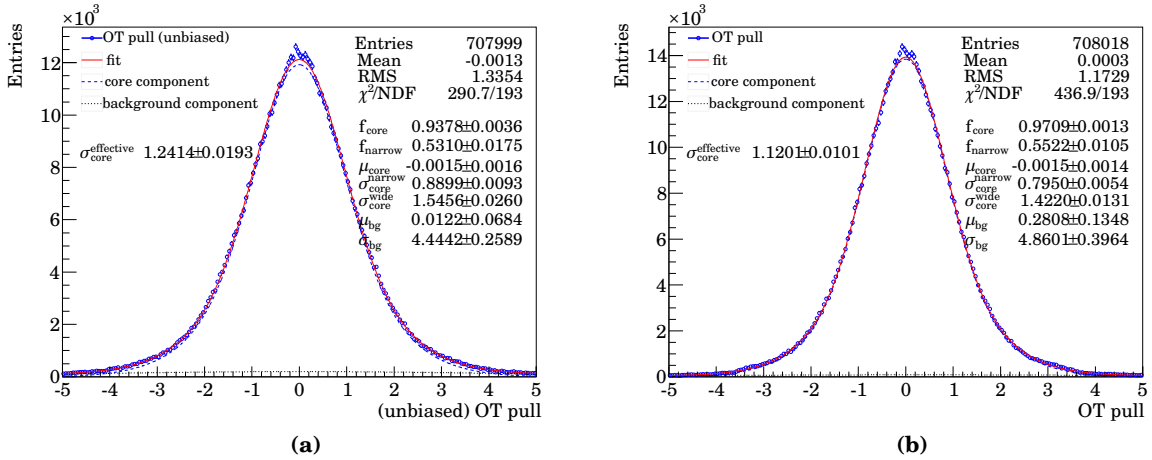


Figure 4.7.: *Unbiased (a) and biased (b) Outer tracker hit pulls in simulated events. The fits and parameters are explained in Appendix A. The width of the unbiased pull distribution is slightly wider, as expected. In both cases, the widths are close to unity, because detector simulation and track reconstruction use the same resolution model parameters.*

- Good B daughters: This is a subgroup of the former item which is characterised by the fact that all daughters of the B mesons are inside the acceptance of the detector, i.e. the efficiency denominator is restricted to daughters from B which one can hope to fully reconstruct (these are obviously the most important B mesons for precision measurements).

- K^0_S/Λ daughters: This group leads to heavily displaced tracks due to the long lifetimes of both K^0_S and Λ which are more difficult to reconstruct for some algorithms.

The table also contains fake track fraction (i.e. the fraction of fake tracks among all tracks, irrespective of track type), both event and track averaged. As explained above, the track averaged fake track fraction is higher than the event-averaged one because busy events (which have a higher fake track fraction) contribute more than less busy events. While the numbers in the table sound alarming, these fakes are mostly at low apparent momentum. Clone fractions (again track averaged) are well below 2% in most track categories, which means that the Clone Killer is doing its job. Hit efficiencies and purities are again track averaged quantities and are given (mostly) for the sake of completeness. The (statistical) uncertainties on the numbers in this kind of table are up to 0.2% in the worst case (usually this worst case is realised for good B daughters with $p > 5$ GeV because that is the smallest track sample), but are usually much better, i.e. 0.1% or below. To prevent information overflow in these tables, the statistical uncertainties will be suppressed and the considerations above apply unless specifically noted otherwise.

It is also possible to plot track reconstruction efficiency as function of p, p_T and η as is shown is Figure 4.8 for good B daughters. The figure also contains a plot of the (event-averaged) track reconstruction efficiency and fake track fraction as function of the number of visible interactions, which is essentially a measure of how many primary

Table 4.1: *Track reconstruction efficiencies, fake track fractions ("ghost rate") and hit efficiencies and purities for long-reconstructible tracks at the final stage of the track reconstruction phase.*

	eff. (clone) [%]	hit pur./eff. [%]
all tracks	89.4 (1.3)	98.5 / 95.0
— $p > 5$GeV	93.9 (0.9)	98.7 / 96.7
B daughters	89.6 (1.2)	98.7 / 94.9
— $p > 5$ GeV	94.1 (1.0)	98.7 / 96.6
good B daughters	93.1 (1.3)	98.7 / 95.9
— $p > 5$ GeV	95.0 (0.9)	98.7 / 96.8
K^0_S/Λ daughters	80.9 (2.4)	98.2 / 92.6
— $p > 5$GeV	88.0 (1.4)	98.3 / 95.6
	tr. avg.	ev. avg.
ghost fraction	15.9	12.3

vertices can be reconstructed in the luminous region. It can be seen that events with many visible interactions are more difficult to reconstruct than those with few, and the fake track fraction also rises accordingly. The error bars in the plots which show reconstruction efficiency as function of p, p_T and η reflect the statistical uncertainties of the calculated efficiencies in each bin.

For plots which show reconstruction efficiency and fake track fraction as function of the number of visible interactions, things are a little less straightforward because event-averaged quantities are shown. In principle, it is possible to calculate uncertainties for track reconstruction efficiency and fake track fraction for each event, however the question on how to properly average these per-event readings in a manner that works well even at low statistics is not trivial. Moreover, a weighted mean efficiency or fake track fraction does not include the event-to-event fluctuations in these quantities. I therefore opted to show mean values and RMS of all per-event readings in a single bin which is conceptually easy to understand and also easy to implement (technically, this is realised as a profile histogram). However, it has the drawback that the error bars essentially become more and more meaningless if there are very few events in a bin. In our example plot, this happens for seven and more visible interactions in the event. Since it is fairly obvious what is going on, I chose to live with this limitation of the plotting technique.

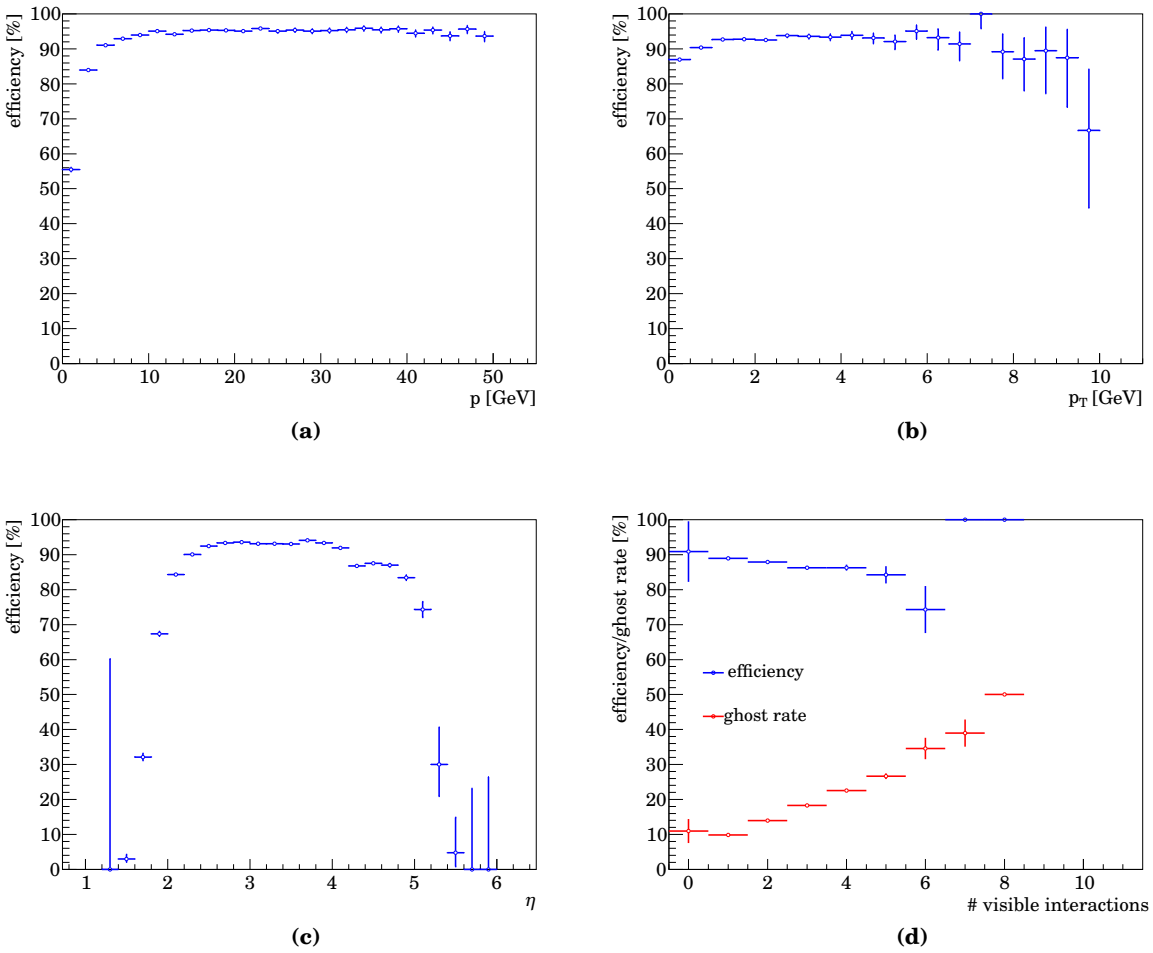


Figure 4.8.: *Reconstruction efficiency after the final stage of track reconstruction as function of (a) p, (b) p_T, (c) η and (d) the number of visible interactions for good B daughters. (a), (b), (c) contain track-averaged quantities, while (d) is plotted event-averaged. (d) also contains the fake track fraction (or ghost rate) as function of the number of visible interactions.*

5. Overview over standalone track reconstruction in the main tracking system

There are the two algorithms, TsaSeeding and PatSeeding, which compete in the field of standalone T station track reconstruction. The aim of this chapter is to give a brief overview over similarities and differences of these two algorithms without losing oversight in the details of implementation. The chapter ends with some performance figures to show the effect of the differences between the algorithms.

More detailed information on TsaSeeding can be found in [52], [53] and [54].

Most of PatSeeding has been developed as part of this thesis, applying many of the ideas that led to TsaSeeding's success, while putting the focus on speed. Because of its good performance, it was chosen as default standalone main tracker reconstruction algorithm in 2011. The implementation of PatSeeding is described in detail in the next chapter, and there is a multitude of tunings and special applications which are introduced in chapter 7.

5.1. Pattern recognition in the main tracker

The main tracker of the LHCb experiment is situated in the fringe field of the dipole magnet, and the induced track curvature cannot be neglected by any pattern recognition algorithm. Since the main component of the magnetic field is parallel to the y direction of the coordinate frame, tracks are bent in the $x-z$ plane, while a straight line is an excellent approximation in the $y-z$ projection.

On the hardware side, it is important to recall that both Inner and Outer Tracker consist of three stations of four layers each. For the Inner Tracker, each layer consists of a single layer of silicon, so a track can have up to 12 hits in the Inner Tracker. An Outer Tracker layer consists of modules with two so-called *monolayers* of straw tubes in each module, leading to a maximum of 24 hits. Tracks may sometimes leave more than one hit per layer, for example if a track has large slopes so two or more adjacent silicon strips or Outer Tracker straws in the same (mono-)layer are hit. The distribution of the number of hits per track is shown in Figure 5.1.

The individual layers in a station are arranged in an x, u, v, x configuration (cf. Figure 5.2), with stereo angles of 0°, −5°, +5°, 0° respectively. If there is a hit in a layer at the point (x, y) in the LHCb frame, it will be measured in the "layer coordinate", i.e. along x,

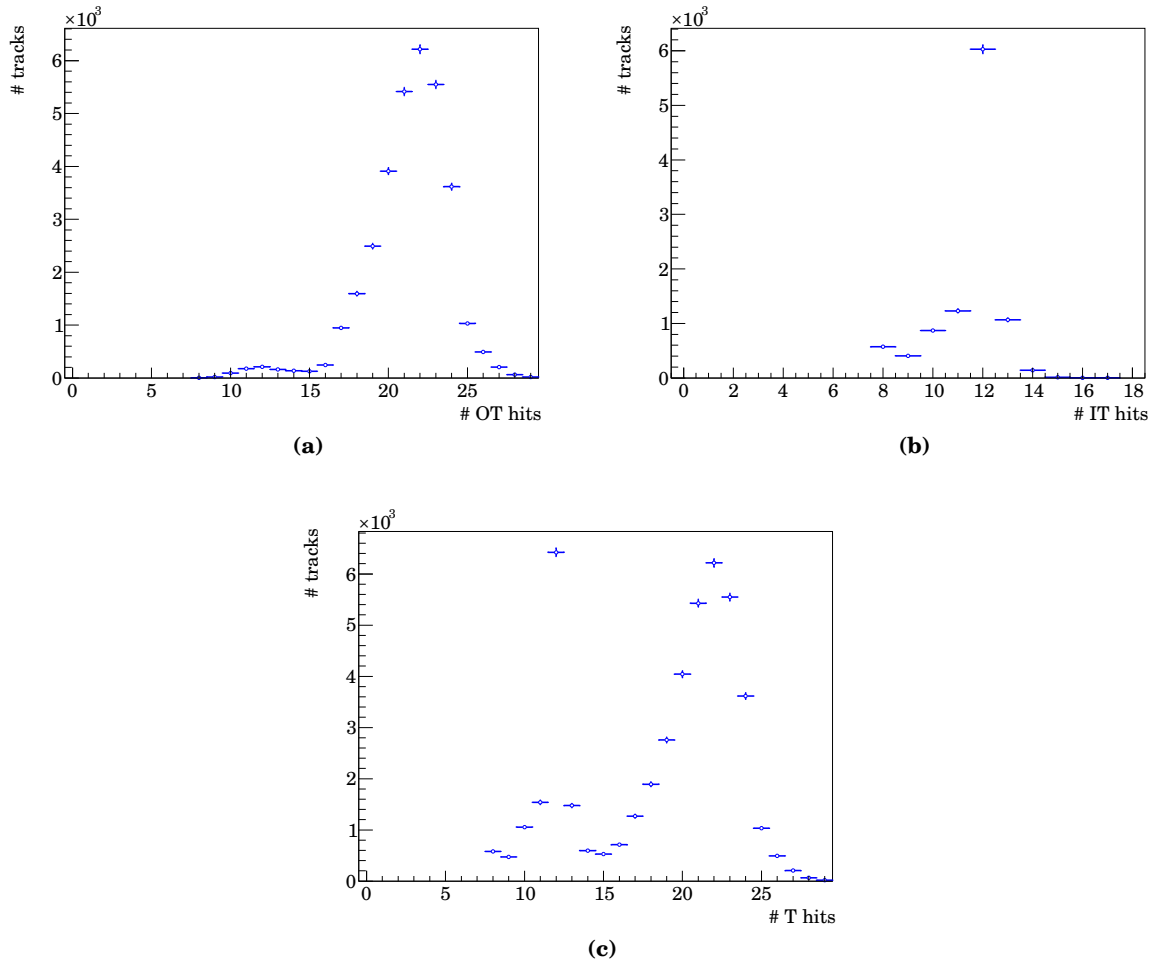


Figure 5.1.: *Number of main tracker hits for (a) tracks entirely in the Outer Tracker, (b) tracks entirely in the Inner Tracker, (c) all T tracks including tracks migrating from IT to OT. Tracks were reconstructed with PatSeeding, only truth-matched tracks are shown.*

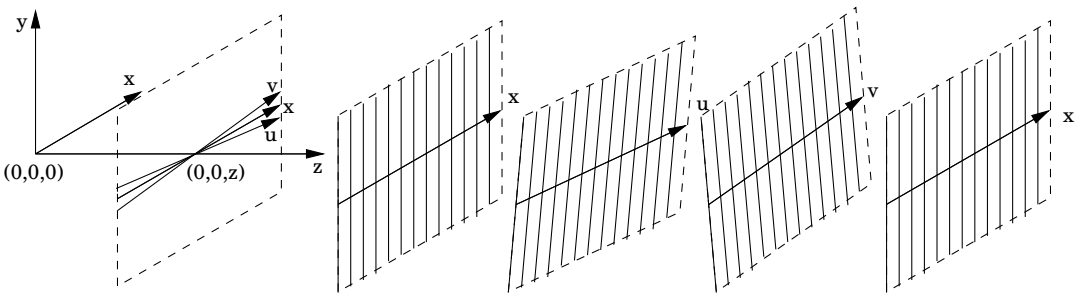


Figure 5.2.: *Sketch illustrating x, u and v directions (left, together with coordinate axes), along with a sketch of the silicon strip/Outer Tracker straw orientation of layers in an x, u, v, x configuration (right).*

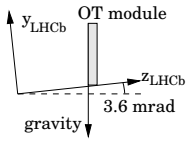
u or v, respectively.[1] The transformation equations read:

$$x_{layer} = x$$

$$u_{layer} = x\cos(5^\circ) - y\sin(5^\circ)$$

$$v_{layer} = x\cos(5^\circ) + y\sin(5^\circ)$$

There is another caveat to watch out for: Because of geological constraints, the LEP tunnel which houses the LHC is in a plane whose normal vector is not parallel to gravity. The LHCb coordinate frame has its z axis pointing along the tangent to the ring, and the x axis is horizontal. That leads to a 3.6 mrad tilt of the y axis with respect to gravity. With both Inner and Outer Tracking being relatively heavy, the decision was made to have them hanging from support structures which align with gravity to minimise material stress. Unfortunately, this means that a layer is no longer at constant z in the LHCb frame, so the y coordinate of the track must be known in order to deduce the correct z coordinate of the hit.



tilt of LHCb frame (sketch)

These constraints dictate much of the pattern recognition approach used by both TsaSeeding and PatSeeding:

- First, the algorithms prepare their hits, e.g. by discarding physically impossible drift radii in the Outer Tracker; very busy events (more than 10,000 hits in the Outer tracker or more than 3,000 hits in the Inner Tracker) can be discarded to prevent both online and offline reconstruction from virtually getting stuck because of a very busy (and rare) event. These cuts on the total hit multiplicity are also known as Global Event Cuts (GEC) and are typically enforced by the trigger before the pattern recognition is run. Since most of the studies shown in this and the next two chapters will be done on simulated events without running trigger emulation code to enforce the GEC, they are applied (and mentioned) explicitly. Similar cuts are also applied for other subdetectors.

[1] This is only true for the "nominal" LHCb detector, the real detector might have tiny rotations around the z axis even for x layers due to the finite precision available when constructing a piece of hardware. Obviously, the pattern recognition must be smart enough to tolerate this.

- Both algorithms perform a so-called "hit cleaning" step which gets rid of very hot regions in the detector (see Section 6.2.1) which are very unlikely to be reconstructed correctly but generate a huge combinatorial load.[2]

- Pattern recognition starts by reconstructing tracks in the $x-z$ projection within a single subdetector. To reduce combinatorics, the main tracking system is divided into regions (upper/lower half of the Outer Tracker and the top, bottom, left and right boxes of the Inner Tracker) which are treated separately, since most tracks stay within their region. Tracks migrating between regions will be reconstructed in a separate step (see below). Tracks are constructed from a hit in an x layer in each station by trying all reasonable combinations of three hits, giving rise to a parabola (see Figure 5.3). Other hits in the vicinity of this parabolic track candidate are collected, and if the resulting candidate passes certain quality cuts, it is passed on to the next stage. Since both algorithms potentially produce clone tracks in this stage, they keep only tracks which do not share a large fraction of hits with each other. Usually, the track with more hits is kept (for tracks with the same number of hits, the track with lower χ^2 is kept).

- Once a set of track candidates in $x-z$ projection is available, the algorithm proceeds with adding the $y-z$ projection to the track. The stereo angle of the u and v layers is $\pm 5^\circ$, so an estimate of the track trajectory in $x-z$ projection is needed to obtain an estimate of the y position of a track. Both algorithms differ in the way the track candidate in $y-z$ projection is extracted from stereo hits, see below for details. Once the $y-z$ projection of the track is known, tracks are refitted, taking into account the 3.6 mrad tilt of the detector layers with respect to the y and z axes. Tracks passing certain quality cuts are saved for a final track selection.

- Tracks which migrate between different Inner Tracker boxes or from Inner to Outer Tracker are found by building "space-points with direction" from four Inner Tracker hits in different layers of the same station (i.e. a point (x,y,z) in the middle of that station, along with a measurement of its slope in $x-z$ projection and the assumption that the track came from the origin to define the slope in $y-z$ projection). These "space-points with direction" are then extrapolated into adjacent stations, and more hits in the vicinity are searched for. If the resulting candidate passes certain quality cuts, tracks are saved for a final selection.

- The final track selection is done by sorting the tracks according to a quality metric. Tracks are then selected and written to the output in order of decreasing quality, discarding tracks which share too many hits with other candidates. Tracks which are written to the output have their hits tagged as used.

Having laid out the similarities between PatSeeding and TsaSeeding in the approach used to find tracks, the differences are summarised below:

- TsaSeeding kills track candidates in busy regions of the detector by discarding

[2] In the reconstruction software (BRUNEL v37r8) used for these studies, TsaSeeding uses hit cleaning by default, while PatSeeding does not. To make a fair comparison, hit cleaning was also activated for PatSeeding, a change that is likely to become the default in the future. The effect of enabling or disabling

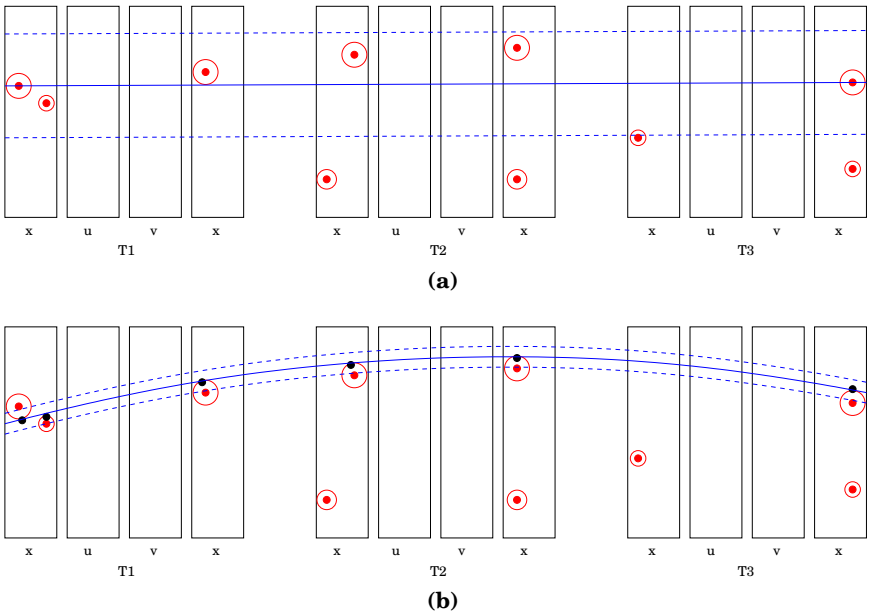


Figure 5.3.: *Sketch illustrating $x-z$ projection track search: In (a), two x hits (red dots, in this example, the red circles symbolise the drift circles of Outer Tracker hits) in the outermost T stations are used to define a line; a window around that line wide enough to enclose curved tracks in the main tracker is used as a region of interest. Using a hit in T2 in that region, it becomes possible to define a parabola and collect hits in its vicinity with much tighter tolerances (b).*

those which have more than a certain number of hits in the search window around the candidate when collecting hits in the vicinity of that candidate. Note that this is not the same as the hit cleaning mentioned above (which looks for combinatoric buildups in a single detector layer), while this method considers (in turn) all x layers and all stereo layers in the main tracker.

- PatSeeding uses a slightly modified track model in $x-z$ projection which adds a small cubical admixture proportional to the parabolic term to better describe low momentum tracks (see 6.3).

- PatSeeding takes into account known (small) rotations of detector layers around all three axes when calculating search windows. Such small rotations can arise from the 3.6 mrad beam tilt and the software alignment for real data (software alignment is the process which estimates the true position and orientation of detector layers in the cavern by studying track residuals). TsaSeeding only accounts for the 3.6 mrad tilt and the stereo angle (if applicable) when calculating search windows.

- For the track search in $y-z$ projection, PatSeeding uses a Hough transform, exploiting the fact that tracks in $y-z$ projection are well approximated by straight

hit cleaning is investigated in more detail in Section C.1.

lines coming from the origin. This makes the method take $\mathcal{O}(n\ln(n))$ time in terms of the hit density n: Each hit must only be touched once to calculate its coordinate in Hough space ($\mathcal{O}(n)$), sorting the list of interesting hits takes at worst $\mathcal{O}(n\ln(n))$ time, and looking for a cluster of hits in the sorted list takes again time linear in n. The combinatorial approach used by TsaSeeding takes at least quadratic time (i.e. $\mathcal{O}(n^2)$), because two hits are needed to define a straight line, and compatible hits in the vicinity of the straight line need to be collected for each two-hit combination tried.

- The final χ^2 fit to obtain track parameters differs between TsaSeeding and PatSeeding. The former decouples the $x-z$ and $y-z$ projections, treating correlations solely by iterating the fit until parameters become stable. PatSeeding does a simultaneous fit to $x-z$ and $y-z$ projections to obtain optimal parameter estimates, thus treating potential correlations among track parameters correctly.
- When searching for tracks migrating from Inner to Outer Tracker, TsaSeeding links straight line segments to each other to obtain longer tracks. PatSeeding uses its track model to estimate the expected track curvature from the line segment obtained, and it updates its parameter estimates and search windows accordingly when searching for hits in adjacent stations, permitting tighter search windows and better performance for low momentum tracks.
- PatSeeding has an extra pass over unused hits which recover tracks at high $|y|$ in the Outer Tracker where the magnetic field (and hence track curvature) is essentially zero.
- During track search, PatSeeding saves tracks which do not point back to the origin in $y-z$ projection into an extra container. These tracks are usually discarded because many are fake and only few are real tracks. After all well-pointing tracks have been examined and their hits have been tagged as used, these non-pointing tracks can be rechecked quickly with stringent quality cuts applied, discouraging tracks which share hits with any of the well-pointing tracks. This helps to recover daughter tracks from long lived neutral particles like K_s^0 or Λ which do not originate in the primary vertex.
- During the final phase of track competition, TsaSeeding uses a sophisticated figure of merit for track quality which includes contributions from the χ^2 probabilities of the fits in $x-z$ and $y-z$ projection, information about expected hits (obtained by tracing the track through the detector and checking for intersections with sensitive volumes). In contrast, PatSeeding uses a much simpler weighted combination of the reduced χ^2, i.e. χ^2/NDF, and the number of hits; the advantage here is that the method is very fast because it uses only information which was already calculated by earlier phases of the algorithm.

5.2. Performance comparison

This section shows performance figures for both PatSeeding and TsaSeeding. Both algorithms will be used in comparable scenarios (same samples, comparable reconstruction

setup as far as possible, algorithm execution time is measured on the same computer), and performance is investigated in detail on simulated events.

Quantifying the performance of PatSeeding on real data is more difficult, see Sections 5.2.2 and 5.2.3 for details.

5.2.1. Monte Carlo simulation

Event multiplicity is an important quantity affecting both reconstruction efficiency and timing of pattern recognition algorithms. For simulated events, the number of pp collisions in a crossing of two non-empty bunches scales with the instantaneous luminosity and is characterised by a Poisson distribution with a mean value ν.

We will investigate two samples of 50,000 inclusive $b\bar{b}$ events with values of $\nu = 1$ and $\nu = 3$. The former[3] closely corresponds to the LHCb design value which favours one pp collision per event because clean events are easier to reconstruct, the latter[4] is close to how the detector was actually operated at the end of the 2010 data taking period to obtain a reasonably sized data sample for analysis despite the low number of bunches in the machine. Both samples were re-reconstructed with a recent version of the LHCb reconstruction software[5].

Low multiplicity events ($\nu = 1$)

Tables 5.1 and 5.2 show the tracking efficiency for PatSeeding and TsaSeeding, respectively. We show the tracking efficiency for both long reconstructible tracks and tracks reconstructible in the main tracker. Specifically, the subsamples of B daughters, good B daughters (fully reconstructible B, i.e. those B decays which have all decay products inside the LHCb acceptance) and K^0_S/Λ daughters are investigated in two momentum ranges, once for the entire spectrum, and once for tracks with $p > 5$ GeV. Clone fractions, hit efficiencies and hit purities are quoted for these categories, and the track and event averaged ghost fraction is given for the overall output.

The distribution of tracking efficiency as function of p, p_T, η and the number of visible interactions can be found in Figures 5.4 (all long reconstructible tracks), 5.5 (long reconstructible B daughters) and 5.6 (long reconstructible K^0_S/Λ daughters) for both algorithms. Since the number of visible interactions is a quantity of the event, the plots as function of the number of visible interactions are event-averaged, while all other plots and figures give track averaged quantities.

The figures and tables show that PatSeeding is better than TsaSeeding in terms of reconstruction efficiency and ghost fractions in all the categories shown. The hit purity for PatSeeding is typically a little bit lower (about a percent or so) than for TsaSeeding; on the other hand, the hit efficiency is typically about 5% higher than that of TsaSeeding (that is about one hit more on a typical Outer Tracker track with about 20 hits), so the slightly lower hit purity of PatSeeding can be compensated by the outlier removal of the

[3] `2010-Beam3500GeV-VeloClosed-MagDown-Nu1-Sim03Reco03-withTruth-10000000-incl_b`

[4] `2010-Beam3500GeV-VeloClosed-MagDown-Nu3-Sim04Reco04-withTruth-10000000-incl_b`

[5] BRUNEL v37r8; To make a fair comparison between PatSeeding and TsaSeeding, hit cleaning was also activated for both algorithms. For the former, that is likely to become the default in the future, while hit cleaning already is used by default in TsaSeeding. The effect of enabling or disabling hit cleaning is investigated in more detail in Section C.1.

subsequent Kalman filter fit which is stricter than the one in the pattern recognition algorithms without significantly degrading track quality due to too few hits being left.

Especially for low-momentum tracks with $p < 2$ GeV, PatSeeding performs significantly better: In the bin $0\,\text{GeV} < p < 2\,\text{GeV}$, the efficiency rises from below 20% for long-reconstructible tracks found by TsaSeeding to above 80% for PatSeeding. This proves that the improved way of accounting for the stray field in the main tracker with a cubic admixture to the parabolic term in $x-z$ projection does make a huge difference.

Another interesting observation is that for TsaSeeding, the reconstruction efficiency as function of η starts to leave its plateau around $\eta \approx 2.5$ when going from large towards small values of η, whereas for PatSeeding this doesn't happen until $\eta \approx 2$. Since low η means large angles of the track with respect to the beam pipe, these tracks are the tracks that go through the outer regions of the Outer Tracker — these are precisely the tracks at high $|y|$ for which PatSeeding introduces special treatment to account for the lower fringe field in this region.

	long reconstructible		T reconstructible	
	eff. (clone) [%]	hit pur./eff. [%]	eff. (clone) [%]	hit pur./eff. [%]
all tracks	95.8 (0.2)	98.0 / 93.9	81.9 (0.2)	97.8 / 93.1
— $p > 5$GeV	96.8 (0.3)	98.0 / 94.8	96.3 (0.3)	98.1 / 94.4
B daughters	96.1 (0.1)	98.2 / 94.5	86.7 (0.1)	98.0 / 93.9
— $p > 5$ GeV	97.1 (0.2)	98.2 / 95.6	96.7 (0.2)	98.2 / 95.3
good B daughters	96.5 (0.2)	98.1 / 95.1	94.9 (0.2)	98.1 / 94.9
— $p > 5$ GeV	96.8 (0.1)	98.1 / 95.7	96.8 (0.1)	98.1 / 95.5
K^0_S/Λ daughters	95.4 (0.1)	97.9 / 93.2	79.8 (0.1)	97.8 / 92.6
— $p > 5$GeV	96.4 (0.2)	97.9 / 94.4	96.0 (0.3)	98.1 / 94.1
	tr. avg.	ev. avg.	tr. avg.	ev. avg.
ghost fraction	2.9	2.2	2.9	2.2

Table 5.1.: *Tracking efficiency for both long reconstructible tracks and tracks reconstructible in the main tracker only for tracks found by PatSeeding. Specifically, the subsamples of b daughters, good b daughters (fully reconstructible b, i.e. those b decays which have all decay products inside the LHCb acceptance) and K^0_S/Λ daughters are investigated in two momentum ranges, once for the entire spectrum, and once for tracks with $p > 5$ GeV. Clone fractions, hit efficiencies and hit purities are quoted for these categories, and the track and event averaged ghost fraction is given for the overall output.*

	long reconstructible		T reconstructible	
	eff. (clone) [%]	hit pur./eff. [%]	eff. (clone) [%]	hit pur./eff. [%]
all tracks	90.9 (0.2)	98.2 / 88.2	75.7 (0.1)	98.0 / 87.9
— $p > 5$GeV	95.7 (0.3)	98.2 / 90.0	93.2 (0.3)	98.3 / 90.4
B daughters	91.1 (0.1)	98.4 / 88.7	80.7 (0.1)	98.3 / 88.3
— $p > 5$ GeV	95.8 (0.1)	98.4 / 90.6	94.5 (0.2)	98.4 / 90.7
good B daughters	94.1 (0.1)	98.4 / 90.0	92.2 (0.1)	98.4 / 89.9
— $p > 5$ GeV	95.8 (0.1)	98.4 / 91.1	95.7 (0.1)	98.4 / 91.2
K^0_S/Λ daughters	88.7 (0.1)	98.1 / 86.8	72.9 (0.1)	98.0 / 87.2
— $p > 5$GeV	95.0 (0.2)	98.1 / 89.1	92.2 (0.2)	98.2 / 89.9
	tr. avg.	ev. avg.	tr. avg.	ev. avg.
ghost fraction	3.7	2.6	3.7	2.6

Table 5.2.: *Tracking efficiency for both long reconstructible tracks and tracks reconstructible in the main tracker only for tracks found by TsaSeeding. Specifically, the subsamples of b daughters, good b daughters (fully reconstructible b, i.e. those b decays which have all decay products inside the LHCb acceptance) and K^0_S/Λ daughters are investigated in two momentum ranges, once for the entire spectrum, and once for tracks with $p > 5$ GeV. Clone fractions, hit efficiencies and hit purities are quoted for these categories, and the track and event averaged ghost fraction is given for the overall output.*

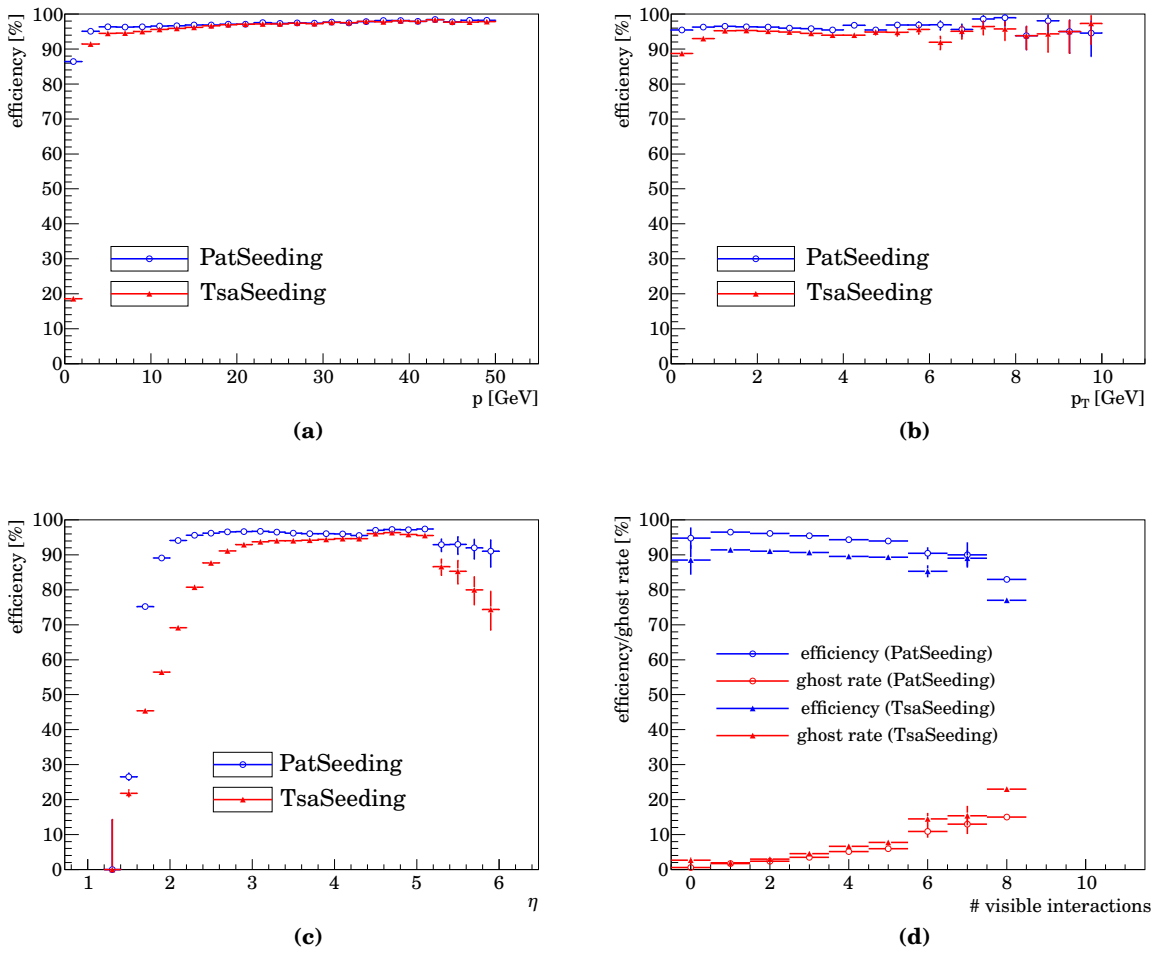


Figure 5.4.: *Efficiency versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for tracks found by PatSeeding (circles) and TsaSeeding (triangles).*

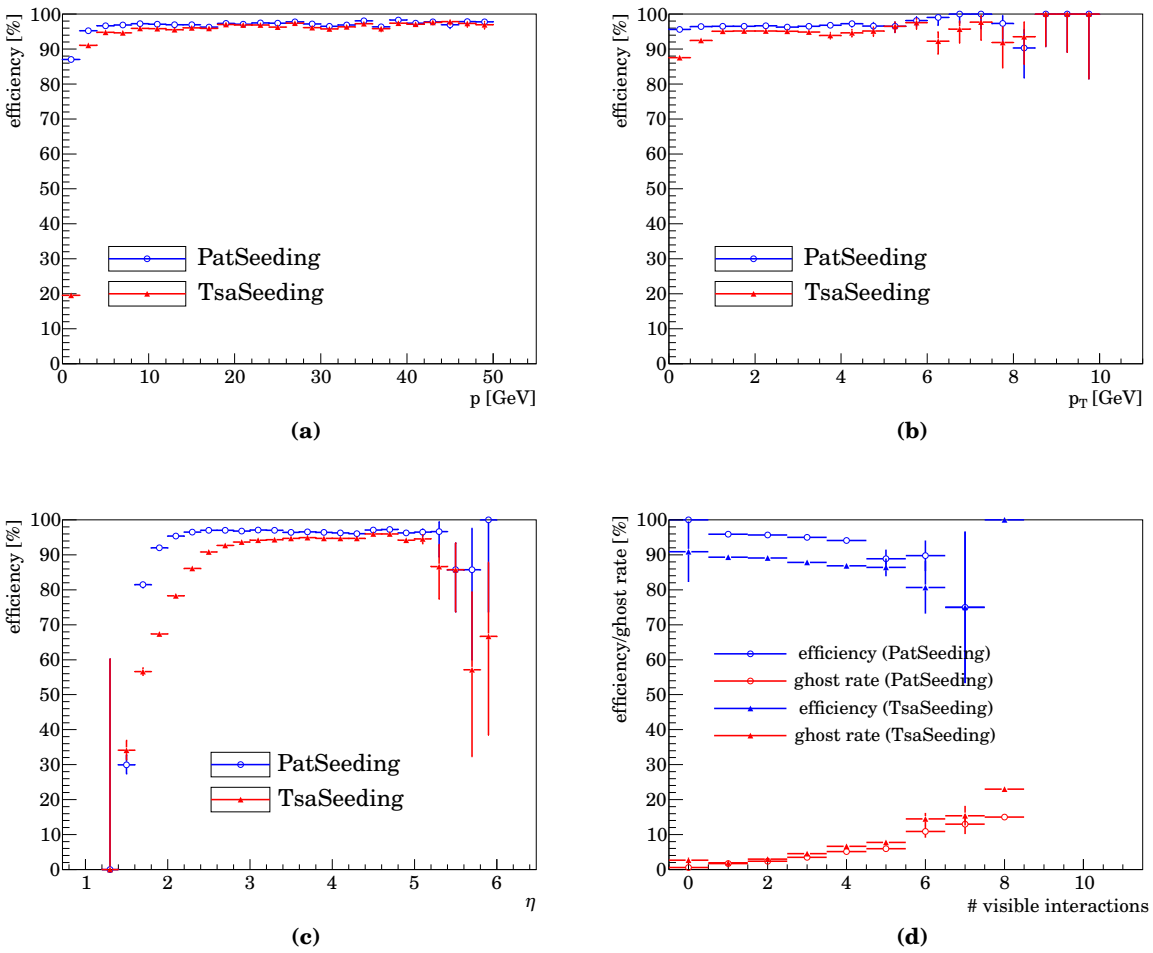


Figure 5.5.: *Efficiency versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for B daughter tracks found by PatSeeding (circles) and TsaSeeding (triangles).*

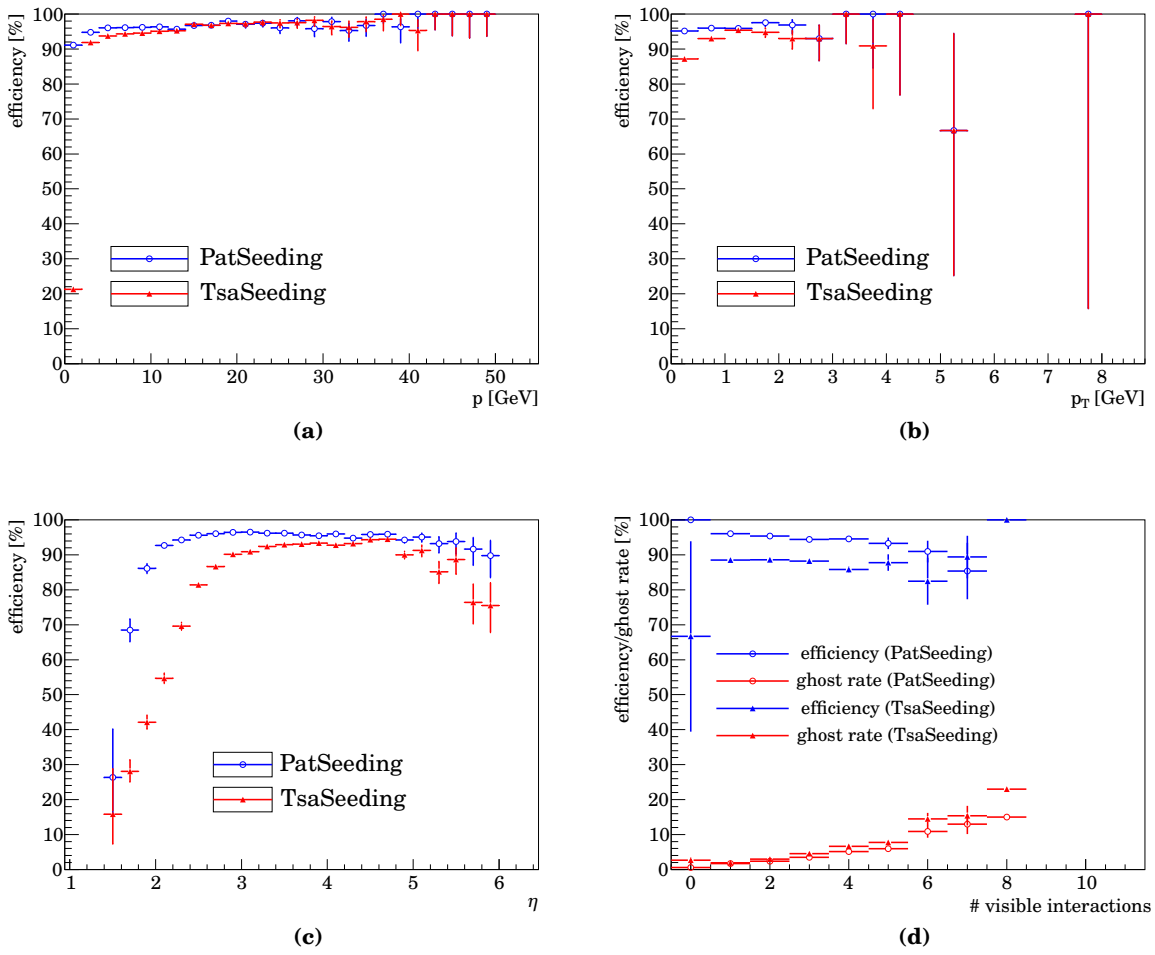


Figure 5.6.: *Efficiency versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for K^0_S/Λ daughter tracks found by PatSeeding (circles) and TsaSeeding (triangles).*

	eff. (clone) [%]	hit pur./eff. [%]
all tracks	90.4 (1.1)	98.6 / 96.1
— $p > 5$GeV	94.0 (0.8)	98.7 / 97.2
B daughters	90.6 (1.1)	98.7 / 96.2
— $p > 5$ GeV	94.3 (0.8)	98.7 / 97.3
good B daughters	93.5 (1.1)	98.7 / 96.7
— $p > 5$ GeV	95.1 (0.8)	98.7 / 97.4
K^0_S/Λ daughters	82.2 (2.0)	98.3 / 94.0
— $p > 5$GeV	88.1 (1.2)	98.3 / 96.1
	tr. avg.	ev. avg.
ghost fraction	15.3	12.0

(a)

	eff. (clone) [%]	hit pur./eff. [%]
all tracks	89.4 (1.3)	98.5 / 95.0
— $p > 5$GeV	93.9 (0.9)	98.7 / 96.7
B daughters	89.6 (1.2)	98.7 / 94.9
— $p > 5$ GeV	94.1 (1.0)	98.7 / 96.6
good B daughters	93.1 (1.3)	98.7 / 95.9
— $p > 5$ GeV	95.0 (0.9)	98.7 / 96.8
K^0_S/Λ daughters	80.9 (2.4)	98.2 / 92.6
— $p > 5$GeV	88.0 (1.4)	98.3 / 95.6
	tr. avg.	ev. avg.
ghost fraction	15.9	12.3

(b)

Table 5.3.: *Efficiency for the final long track sample at $\nu = 1$ when using PatSeeding (a) or TsaSeeding (b)*

Since the T station standalone reconstruction is not the only reconstruction algorithm running in the LHCb reconstruction framework, it makes sense to look at the final sample of long tracks after the Clone Killer has done its work. This final sample consists of the tracks that are typically used for physics analyses. Table 5.3 presents a breakdown for both PatSeeding and TsaSeeding. Since there are two complementary strategies to find long tracks (Forward tracking and standalone main tracker reconstruction followed by track matching), the influence of difference in efficiency between PatSeeding and TsaSeeding is less apparent in the final output of the entire reconstruction chain, but the improvement due to the higher reconstruction efficiency of PatSeeding is still clearly visible.

Another interesting quantity to judge the performance of a pattern recognition algorithm is the time it takes to reconstruct an event. Clearly, the main driver of combinatorics and thus execution time in the event is the number of hits in the main tracker. Therefore, execution time will be measured as function of the number of main tracker hits. It is also interesting to look as another quantity, the number of visible interactions, which is basically the number of primary vertices which can be "seen" by the detector. For illustration, suppose these is a single elastic pp collision in an event, and both protons escape detection through the beam pipe in opposite directions — this would be an example for an event with zero visible interactions (i.e. an example for an interaction which cannot be seen). The distribution of the number of main tracker hits for a given number of visible interactions is quite broad, so there is no strong correlation between the number visible interactions and the number of main tracker hits. The number of visible interactions is nevertheless a useful quantity because it allows to turn predictions for the number of pp collisions in a bunch crossing into estimates for the average execution time needed for the main tracker pattern recognition.

The following timing measurements have all been done on the same computer[6] to ensure comparable results. The results on other machines will differ in the actual

[6]The machine in question contains an Intel Xeon E5440 processor clocked at 2.83 GHz.

algorithm	average time per event [ms]	maximum time per event [ms]
TsaSeeding	19.63	386.5
PatSeeding	9.09	331.0

Table 5.4.: *Average and maximum time needed by PatSeeding and TsaSeeding for the reconstruction of a single event at $\nu = 1$.*

numbers by an overall scaling factor, but the general trend will be the same. Table 5.4 contains the average and maximum time PatSeeding and TsaSeeding spent reconstructing events. On average, PatSeeding is twice as fast as TsaSeeding. In the worst case, i.e. the maximum time spent for a single event, PatSeeding is still faster than TsaSeeding.

Figure 5.7 shows the time needed to reconstruct the main tracker as function of the number of main tracker hits and the number of visible interactions in the event. From these plots, is becomes apparent that PatSeeding is generally faster than TsaSeeding. It can also be seen that two events above about 11,500 hits in the main trigger do not pass the Global Event Cuts (GEC) mentioned earlier (and therefore need practically zero time).

The plots also contain fitted parametrisations (polynomials) of the execution time; events which do not pass the GEC are excluded from the fits. The fitted parametrisations of execution time t given in ms as function of the number of main tracker hits n_{hits} and number of visible interactions $n_{vis.}$ are give below.

For PatSeeding, the following parametrisations were fitted:

$$\begin{aligned} t(n_{\text{hits}}) &= (0.1995 \pm 0.0006) \cdot 10^{-9} \cdot n_{\text{hits}}^3 + (-0.2102 \pm 0.0067) \cdot 10^{-6} \cdot n_{\text{hits}}^2 + \\ &\quad (1.8505 \pm 0.0230) \cdot 10^{-3} \cdot n_{\text{hits}} + (0.2156 \pm 0.0224) \end{aligned} \tag{5.1}$$

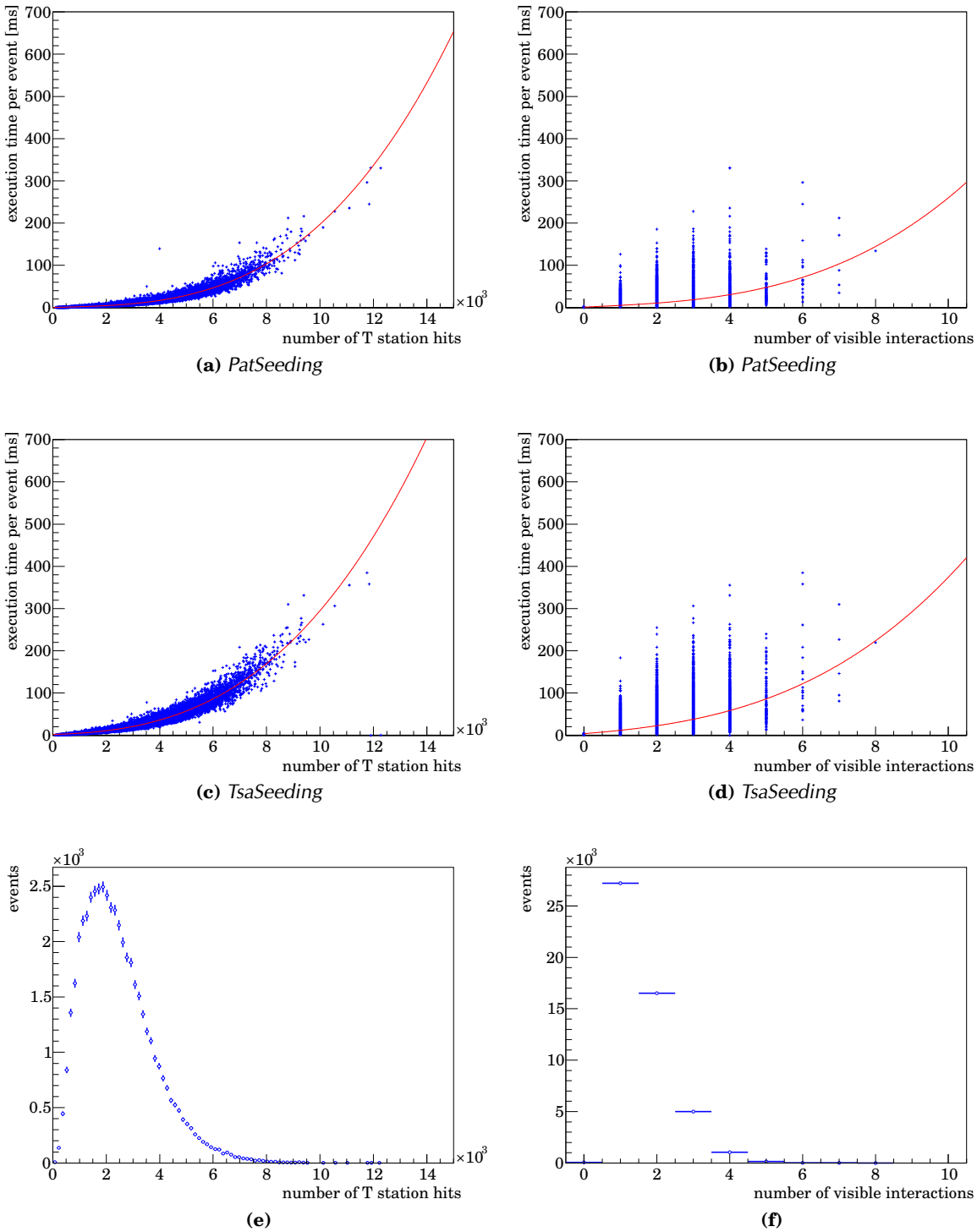
$$\begin{aligned} t(n_{\text{vis.}}) &= (0.2246 \pm 0.0031) \cdot n_{\text{vis.}}^3 + (-0.0593 \pm 0.0251) \cdot n_{\text{vis.}}^2 + \\ &\quad (4.0269 \pm 0.0601) \cdot n_{\text{vis.}} + (1.0955 \pm 0.0407) \end{aligned} \tag{5.2}$$

For TsaSeeding, the fit resulted in the following parametrisations:

$$\begin{aligned} t(n_{\text{hits}}) &= (0.1768 \pm 0.0006) \cdot 10^{-9} \cdot n_{\text{hits}}^3 + (0.9875 \pm 0.0072) \cdot 10^{-6} \cdot n_{\text{hits}}^2 + \\ &\quad (1.7514 \pm 0.0241) \cdot 10^{-3} \cdot n_{\text{hits}} + (2.0235 \pm 0.0230) \end{aligned}$$

$$\begin{aligned} t(n_{\text{vis.}}) &= (0.2210 \pm 0.0031) \cdot n_{\text{vis.}}^3 + (0.8014 \pm 0.0251) \cdot n_{\text{vis.}}^2 + \\ &\quad (6.9330 \pm 0.0600) \cdot n_{\text{vis.}} + (3.8627 \pm 0.0407) \end{aligned}$$

The effect of the hit cleaning step (explained in detail in Section 6.2.1) on tracking efficiency and algorithm execution time is discussed in more detail in Section C.1.



(a) *PatSeeding* **(b)** *PatSeeding*

(c) *TsaSeeding* **(d)** *TsaSeeding*

(e) **(f)**

Figure 5.7.: *Execution time at $\nu = 1$ of PatSeeding (a, b) and TsaSeeding (c, d) as function of the number of main tracker hits (a, c) and the number of visible interactions (b, d). A third order polynomial has been used to fit the graphs in (a-d); the large variation of execution time in (b, d) for a given number of visible interactions is due to the fact that the number of tracks per pp interaction is itself quite variable. (e) and (f) contain the distribution of the number of main tracker hits and the number of visible interactions in the events which entered the plots.*

High multiplicity events ($\nu = 3$)

Towards the end of the 2010 data taking period, the number of proton filled bunches in a beam of the LHC was a factor of roughly ten below the design value, mostly for safety reasons. In order to allow the experiments to collect a data sample of reasonable size, the per-bunch beam current was increased, and beams were more tightly focused to obtain more pp collisions per bunch crossing. LHCb was designed to work well in clean environments with typically a single pp interaction per bunch crossing to do precision measurements. However, towards the end of the 2010 data taking period, the experiment was running at around $\nu = 3$ simultaneous pp interactions per bunch crossing. It is therefore interesting to do the same study as in the previous subsection, but on a sample with three simulated pp collisions per non-empty bunch crossing instead of one.

Tables 5.5 and 5.6 give a tabular overview of the performance figures for PatSeeding and TsaSeeding. The corresponding plots as function of p, p_T, η and the number of visible interactions are in Figures 5.8 (all long reconstructible tracks), 5.9 (long reconstructible b daughters) and 5.10 (long reconstructible K^0_S/Λ daughters) for both algorithms. Table 5.7 shows a summary of the track reconstruction efficiency for long tracks in the final output of the track reconstruction software.

Generally speaking, one can confirm two things one would expect for samples with increased average multiplicity: The efficiency is slightly lower than in the $\nu = 1$ case above (about 1% for high momentum tracks), while the ghost rate increases, in this case almost by a factor of two. It continues to be true that the reconstruction efficiency of PatSeeding is slightly higher while its ghost rate remains a little lower than that of TsaSeeding.

It also becomes clear (from the plot of efficiency and ghost rate as function of the number of visible interactions) that track reconstruction becomes an extremely challenging task above the regime of six to eight visible interactions, and both algorithms have their difficulties there with reconstruction efficiencies dropping significantly below 90% and ghost fractions above 10%. At $\nu = 3$, only about 3.4% of non-empty bunch-crossings have more than six pp collisions. Due to Global Event Cuts (GEC) being applied at the trigger level, busy events with many pp collisions are discarded, so that the total multiplicity in triggered events stays manageable. In this sense, GEC are a useful tool to maximise the luminosity recordable by LHCb because they offer a possibility to favour the “clean” multi-pp-collisions events without introducing huge amounts of dead time in the trigger due to a few percent of very busy events.

	long reconstructible		T reconstructible	
	eff. (clone) [%]	hit pur./eff. [%]	eff. (clone) [%]	hit pur./eff. [%]
all tracks	94.3 (0.2)	97.2 / 93.1	80.2 (0.2)	97.1 / 92.3
— $p > 5$GeV	95.1 (0.3)	97.3 / 94.0	94.7 (0.4)	97.5 / 93.7
B daughters	95.1 (0.1)	97.5 / 94.0	85.6 (0.1)	97.4 / 93.3
— $p > 5$ GeV	95.9 (0.2)	97.4 / 95.0	95.5 (0.2)	97.5 / 94.7
good B daughters	95.7 (0.1)	97.3 / 94.5	94.4 (0.2)	97.4 / 94.3
— $p > 5$ GeV	95.8 (0.1)	97.2 / 95.0	95.8 (0.1)	97.4 / 94.9
K^0_S/Λ daughters	94.1 (0.1)	97.1 / 92.4	78.3 (0.2)	97.1 / 91.8
— $p > 5$GeV	94.5 (0.2)	97.1 / 93.5	94.5 (0.4)	97.5 / 93.3
	tr. avg.	ev. avg.	tr. avg.	ev. avg.
ghost fraction	5.3	3.7	5.3	3.7

Table 5.5.: *Tracking efficiency for both long reconstructible tracks and tracks reconstructible in the main tracker only for tracks found by PatSeeding. Specifically, the subsamples of b daughters, good b daughters (fully reconstructible b, i.e. those b decays which have all decay products inside the LHCb acceptance) and K^0_S/Λ daughters are investigated in two momentum ranges, once for the entire spectrum, and once for tracks with $p > 5$ GeV. Clone fractions, hit efficiencies and hit purities are quoted for these categories, and the track and event averaged ghost fraction is given for the overall output.*

	long reconstructible		T reconstructible	
	eff. (clone) [%]	hit pur./eff. [%]	eff. (clone) [%]	hit pur./eff. [%]
all tracks	89.6 (0.2)	97.4 / 87.3	74.3 (0.1)	97.3 / 87.1
— $p > 5$GeV	94.2 (0.3)	97.5 / 89.1	91.7 (0.2)	97.6 / 89.6
B daughters	90.1 (0.1)	97.7 / 87.9	79.7 (0.1)	97.6 / 87.6
— $p > 5$ GeV	94.7 (0.2)	97.7 / 89.7	93.4 (0.2)	97.8 / 89.8
good B daughters	93.1 (0.1)	97.6 / 88.9	91.6 (0.1)	97.7 / 88.9
— $p > 5$ GeV	94.7 (0.1)	97.6 / 90.0	94.7 (0.1)	97.7 / 90.1
K^0_S/Λ daughters	87.5 (0.1)	97.4 / 86.0	71.7 (0.1)	97.3 / 86.4
— $p > 5$GeV	93.5 (0.3)	97.3 / 88.1	90.8 (0.3)	97.6 / 89.1
	tr. avg.	ev. avg.	tr. avg.	ev. avg.
ghost fraction	7.0	4.8	7.0	4.8

Table 5.6.: *Tracking efficiency for both long reconstructible tracks and tracks reconstructible in the main tracker only for tracks found by TsaSeeding. Specifically, the subsamples of b daughters, good b daughters (fully reconstructible b, i.e. those b decays which have all decay products inside the LHCb acceptance) and K^0_S/Λ daughters are investigated in two momentum ranges, once for the entire spectrum, and once for tracks with $p > 5$ GeV. Clone fractions, hit efficiencies and hit purities are quoted for these categories, and the track and event averaged ghost fraction is given for the overall output.*

	eff. (clone) [%]	hit pur./eff. [%]
all tracks	88.6 (1.1)	98.0 / 95.6
— $p > 5$GeV	92.4 (0.9)	98.1 / 96.7
B daughters	89.4 (1.2)	98.2 / 95.8
— $p > 5$ GeV	93.3 (1.0)	98.3 / 96.9
good B daughters	92.9 (1.1)	98.2 / 96.3
— $p > 5$ GeV	94.6 (0.9)	98.2 / 97.0
K^0_S/Λ daughters	78.6 (1.6)	97.6 / 93.6
— $p > 5$GeV	85.1 (1.1)	97.7 / 95.6
	tr. avg.	ev. avg.
ghost fraction	22.2	17.2

(a)

	eff. (clone) [%]	hit pur./eff. [%]
all tracks	87.6 (1.3)	98.0 / 94.6
— $p > 5$GeV	92.4 (1.0)	98.1 / 96.3
B daughters	88.4 (1.4)	98.2 / 94.6
— $p > 5$ GeV	93.2 (1.1)	98.3 / 96.2
good B daughters	92.4 (1.4)	98.2 / 95.4
— $p > 5$ GeV	94.5 (1.0)	98.2 / 96.4
K^0_S/Λ daughters	77.3 (2.0)	97.6 / 92.3
— $p > 5$GeV	85.2 (1.3)	97.7 / 95.2
	tr. avg.	ev. avg.
ghost fraction	23.5	17.8

(b)

Table 5.7.: *Efficiency for the final long track sample at $v = 3$ when using PatSeeding (a) or TsaSeeding (b)*

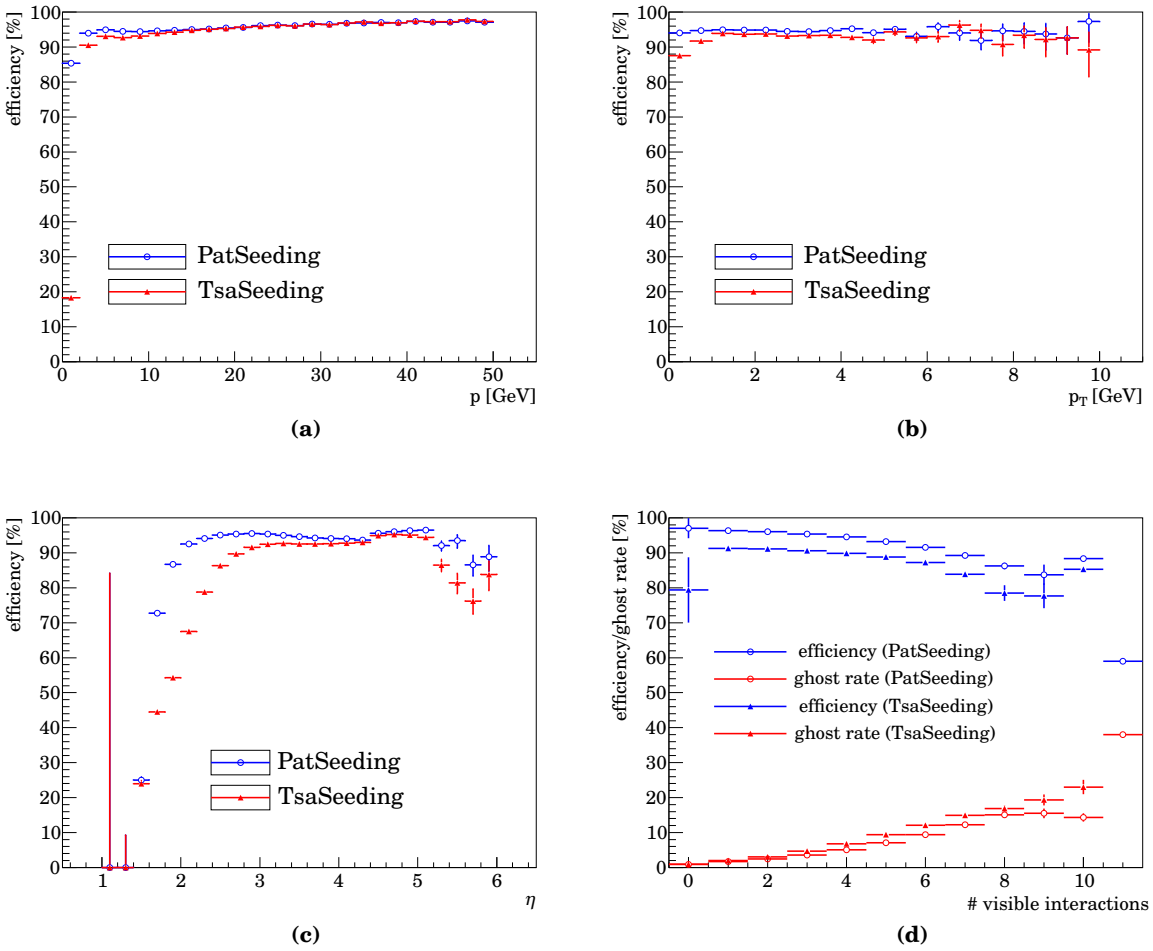


Figure 5.8.: *Efficiency versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for tracks found by PatSeeding (circles) and TsaSeeding (triangles).*

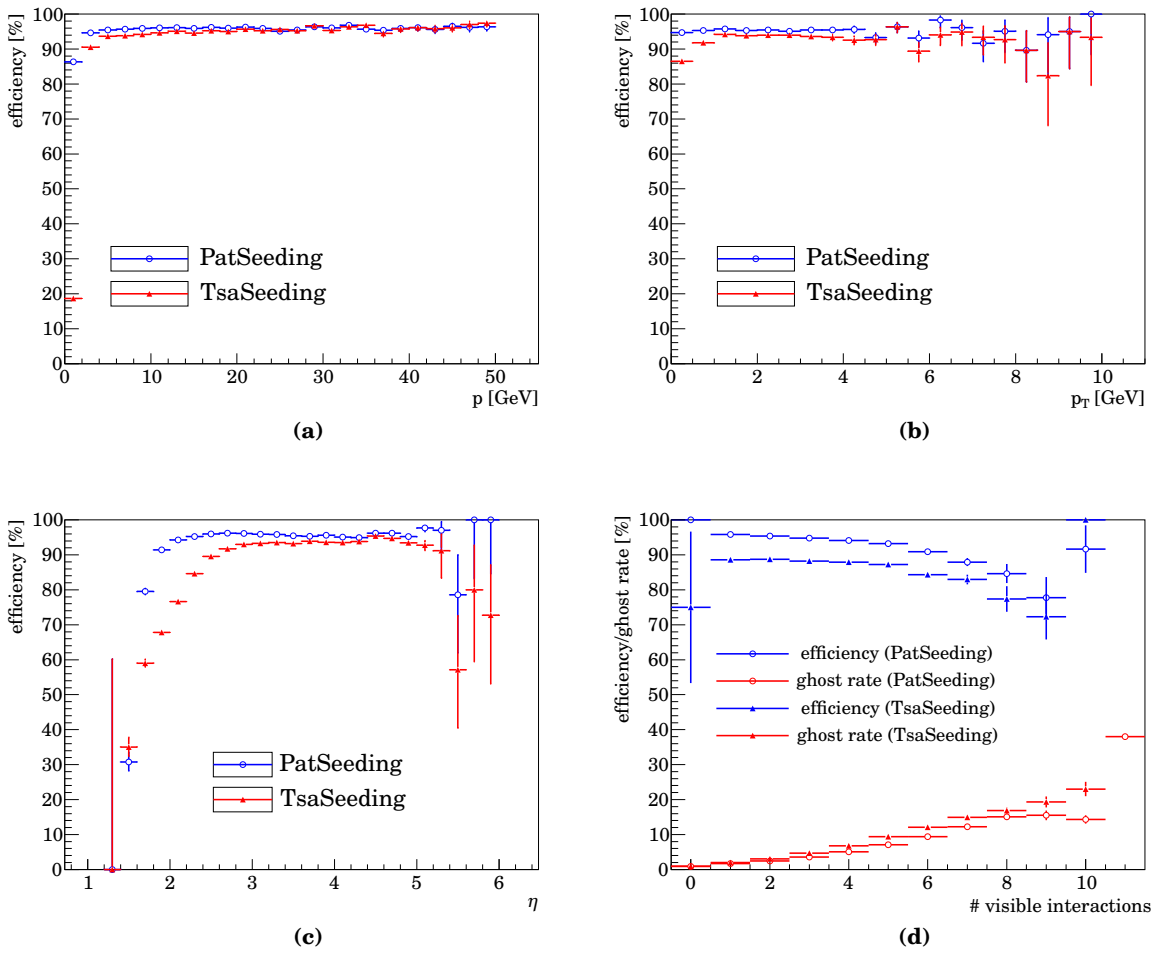


Figure 5.9.: *Efficiency versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for b daughter tracks found by PatSeeding (circles) and TsaSeeding (triangles).*

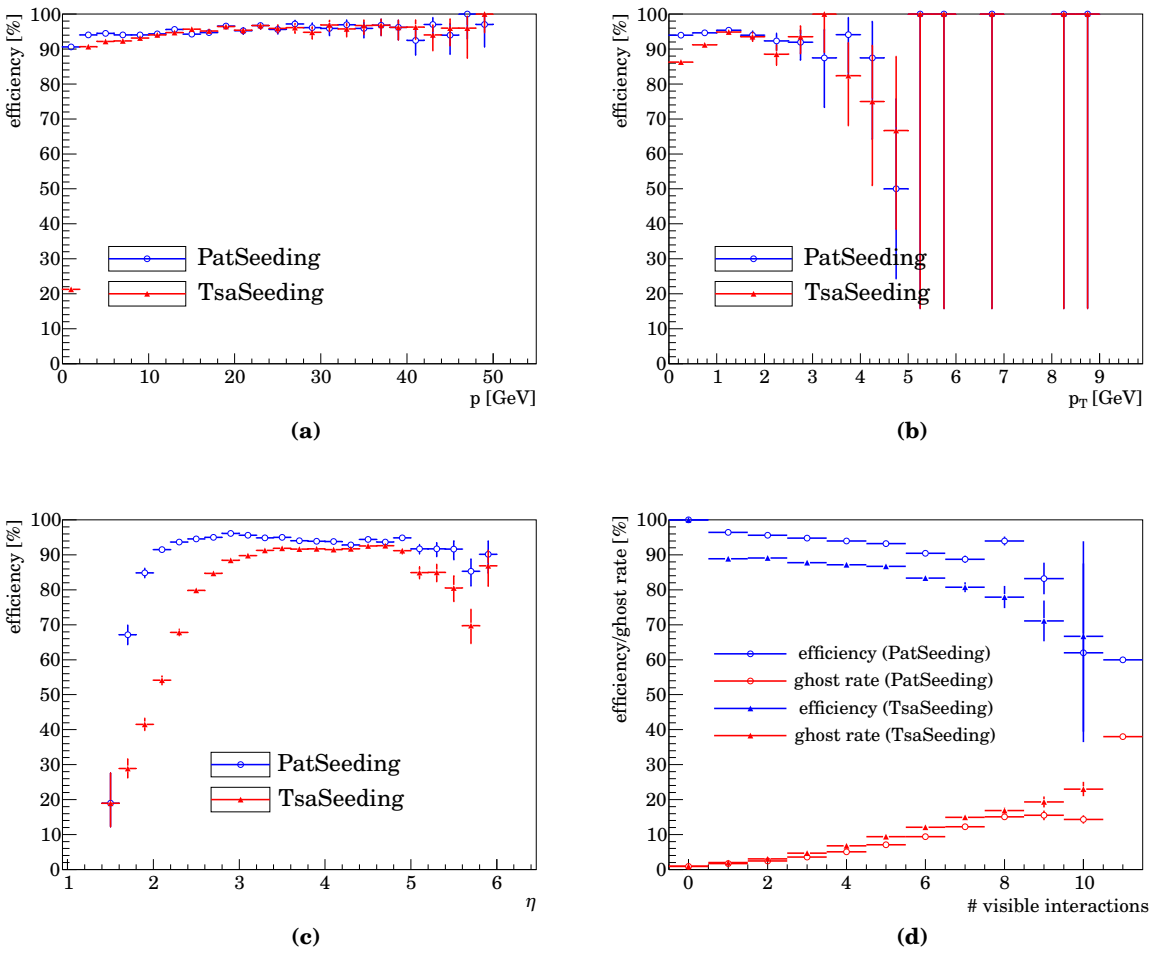


Figure 5.10.: *Efficiency versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for K^0_S/Λ daughter tracks found by PatSeeding (circles) and TsaSeeding (triangles).*

Just as in the $\nu = 1$ case, one can study the timing behaviour of both algorithms, and the same setup (same computer etc.) was used to obtain these results. Table 5.8 shows average and maximum time needed to reconstruct an event. Figure 5.11 shows the execution time t in ms of PatSeeding and TsaSeeding as function of the number of main tracker hits n_{hits} and number of visible interactions n_{vis}.

For PatSeeding, the following parametrisations were obtained:

$$\begin{aligned} t(n_{\text{hits}}) &= (0.1803 \pm 0.0003) \cdot 10^{-9} \cdot n_{\text{hits}}^3 + (0.3310 \pm 0.0043) \cdot 10^{-6} \cdot n_{\text{hits}}^2 + \\ &\quad (-0.8196 \pm 0.0197) \cdot 10^{-3} \cdot n_{\text{hits}} + (3.4762 \pm 0.0264) \end{aligned}$$

$$\begin{aligned} t(n_{\text{vis.}}) &= (0.0586 \pm 0.0007) \cdot n_{\text{vis.}}^3 + (1.6411 \pm 0.0087) \cdot n_{\text{vis.}}^2 + \\ &\quad (-0.0761 \pm 0.0302) \cdot n_{\text{vis.}} + (4.3013 \pm 0.0307) \end{aligned}$$

For TsaSeeding, the fit resulted in these parametrisations:

$$\begin{aligned} t(n_{\text{hits}}) &= (0.1417 \pm 0.0003) \cdot 10^{-9} \cdot n_{\text{hits}}^3 + (1.7176 \pm 0.0048) \cdot 10^{-6} \cdot n_{\text{hits}}^2 + \\ &\quad (-1.4916 \pm 0.0213) \cdot 10^{-3} \cdot n_{\text{hits}} + (5.5363 \pm 0.0274) \end{aligned}$$

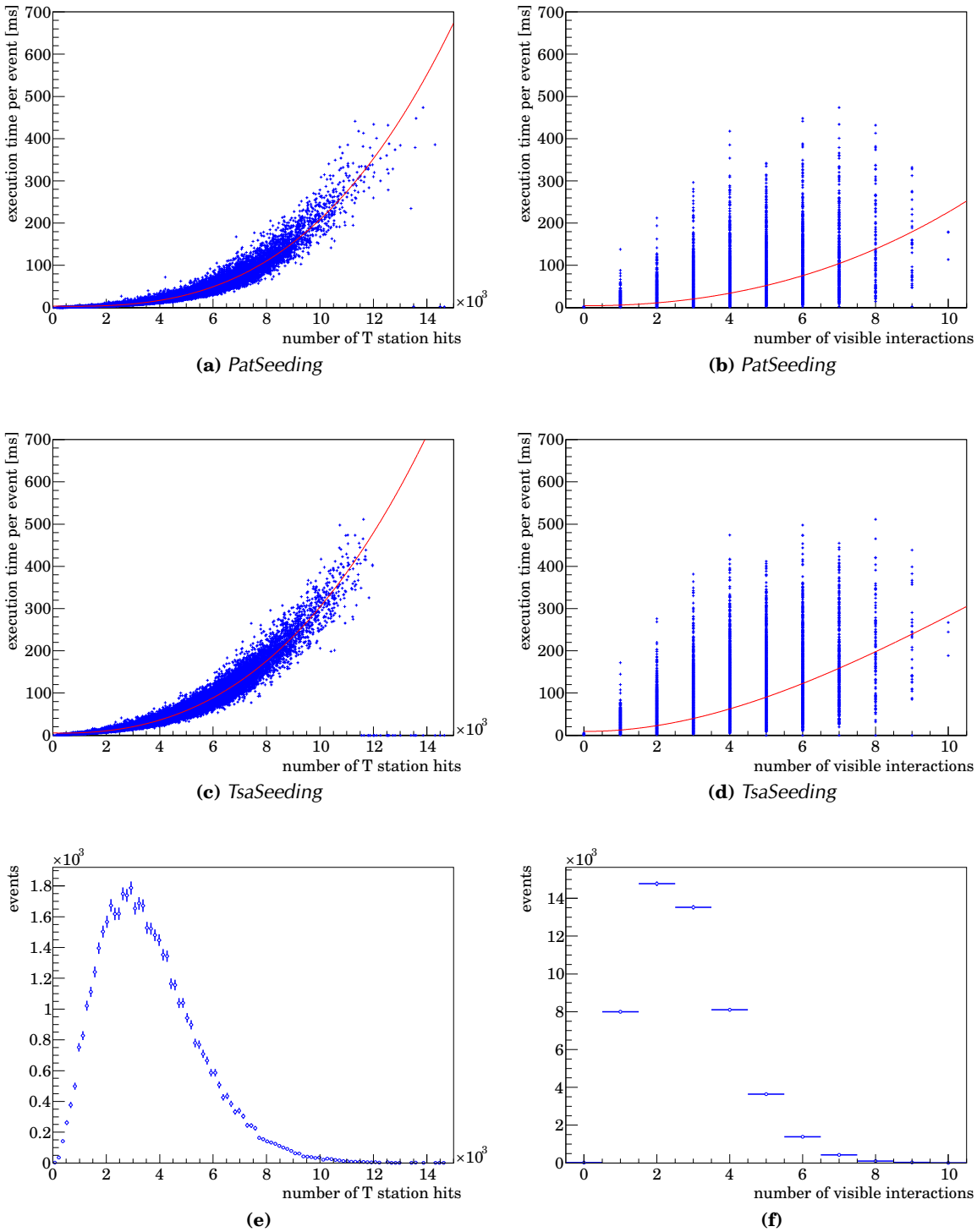
$$\begin{aligned} t(n_{\text{vis.}}) &= (-0.1128 \pm 0.0008) \cdot n_{\text{vis.}}^3 + (3.9216 \pm 0.0089) \cdot n_{\text{vis.}}^2 + \\ &\quad (-0.5745 \pm 0.0304) \cdot n_{\text{vis.}} + (9.2582 \pm 0.0307) \end{aligned}$$

Both the table and the figures do not add anything surprising to the results at $\nu = 1$. PatSeeding continues to be faster on average, while TsaSeeding has the better worst-case timing behaviour because it uses hit cleaning (cf. Section 6.2.1) and it kills track candidates in busy regions of the detector. A detailed study of both algorithms with and without hit cleaning will be done in Section C.1.

However, it is interesting to note that explicit Global Event Cuts (as done in PatSeeding) do have an advantage in terms of predictability, because an event is either accepted or rejected; in the latter case a global flag in the event is set to indicate that occurrence, and practically no time is used. In case of TsaSeeding, there is no indication if track candidates have been discarded because of a busy region in the detector. In fact, such a busy region might even arise due to two particles travelling alongside each other, with one of the resulting track candidates in the search windows around the other; TsaSeeding would not reconstruct either track, nor would there be any indication of what happened. Moreover, these kind of events still take considerable time to process for TsaSeeding, which is another disadvantage.

algorithm	average time per event [ms]	maximum time per event [ms]
TsaSeeding	42.07	513.8
PatSeeding	22.50	473.9

Table 5.8.: *Average and maximum time needed by PatSeeding and TsaSeeding for the reconstruction of a single event at $\nu = 3$.*



(a) *PatSeeding* **(b)** *PatSeeding*

(c) *TsaSeeding* **(d)** *TsaSeeding*

(e) **(f)**

Figure 5.11.: *Execution time at $\nu = 3$ of PatSeeding (a, b) and TsaSeeding (c, d) as function of the number of main tracker hits (a, c) and the number of visible interactions (b, d). Third order polynomials have been used to fit the graphs (a-d). (e) and (f) contain the distribution of the number of main tracker hits and the number of visible interactions in the events which entered the plots.*

5.2.2. Performance in 2009 data (K_S^0 production)

To check the performance of PatSeeding on data, the 2009 data sample was used to extract the K_s yield as explained in the analysis part of this thesis. For the analysis part, the official reprocessing is taken which uses TsaSeeding (in its tuning for early data taking conditions). The entire data sample was reprocessed with PatSeeding (which uses its own tuning for early data taking conditions, see Section 7.2). Apart from the different algorithms for the main tracker standalone reconstruction, both reprocessings are treated exactly alike, and identical selection and yield extraction procedures are used.

Figure 5.12 shows the yields extracted from the same data sample when reconstructed once with TsaSeeding and once with PatSeeding. Downstream tracks were used to form the K_S^0 candidates because the Velo was not fully closed during the 2009 data taking; the full reasoning for this decision is explained in the analysis part of this thesis. The fit model is a double Gaussian with a linear contribution for the background; the detailed meaning of the fit parameters is also explained in the analysis part of the thesis (Section 9.3.1). The important quantity for this discussion is the signal yield extracted from the fit, N_{signal}. The uncertainties given are of a statistical nature only. When the 2009 data sample is reconstructed using TsaSeeding. the K_S^0 signal yield is 4801.3 ± 84.3 candidates, when reconstructing with PatSeeding, the yield is slightly higher at 4841.8 ± 86.2 candidates. Since the two yields have been extracted from an identical sample of events, the difference in yields is significant, despite the relatively big statistical uncertainty of the yields.

From the performance figures obtained on simulated events, one might have expected PatSeeding to take a more prominent lead. However, the selection of K_S^0 daughter tracks demands $p > 2$ GeV in order to reduce combinatorial background. For these high-momentum tracks, the performance of PatSeeding and TsaSeeding is actually compatible with the result obtained here.

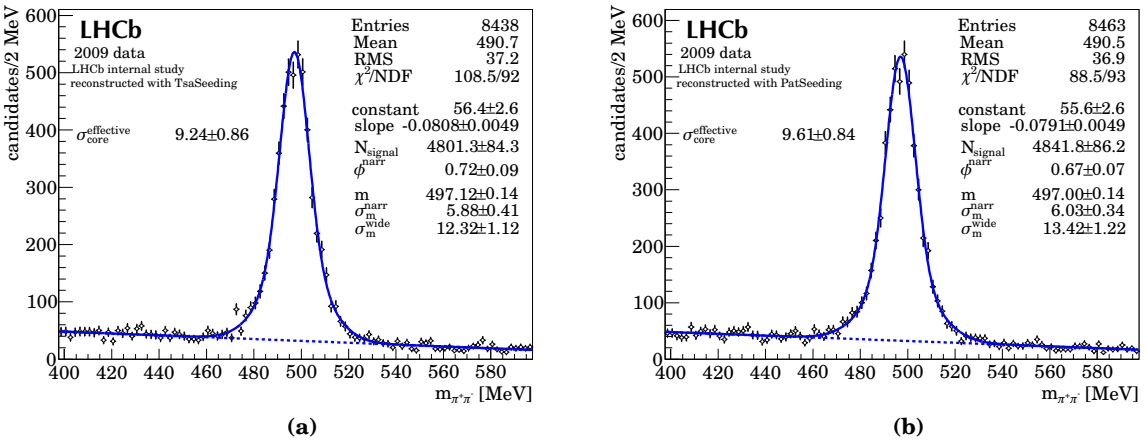


Figure 5.12.: *K_S^0 yield in 2009 data when reconstructed with either TsaSeeding (a) or PatSeeding (b). The signal yield, N_{signal} is slightly higher for PatSeeding.*

5.2.3. Performance in 2010 data ($B_u^+ \longrightarrow J/\psi K^+$)

The check of PatSeeding performance in the last section was done on a rather special track sample: K_S^0 daughter tracks are typically very displaced tracks, and the momentum spectrum is also different from that of tracks from a typical B decay. For this reason, a cross-check with B daughters seems to be appropriate.

For the entire 2010 data sample, TsaSeeding was used in all official reprocessings. The fastest way to obtain figures for the signal yield for both seeding algorithms is to just re-reconstruct a sample of selected signal events with PatSeeding. Unfortunately, this method will give biased results if both algorithms do not reconstruct exactly the same tracks: Obviously, if a daughter track of the selected decay is found by TsaSeeding, but not by PatSeeding, the signal candidate is lost. However, if a daughter track would be found by PatSeeding but not by TsaSeeding, the yield cannot increase because these events do not pass the selection which was done based on tracks found by TsaSeeding. Thus, the method of re-reconstructing a preselected sample using PatSeeding will see any losses from a switch to PatSeeding, but it will not see gains.

Thus, a full reprocessing of the 2010 data sample with PatSeeding would be needed to obtain an unbiased figure for the signal yield. Unfortunately, this is prohibitive in terms of CPU and storage requirements.

Therefore, the bias of the fast method will be predicted using the overlap between tracks found by PatSeeding and TsaSeeding as measured in simulated events. Then, a biased yield estimate is given based on a preselected sample of 2010 data re-reconstructed using PatSeeding.

The signal decay which will be used is $B_u^+ \longrightarrow J/\psi(\mu^+\mu^-)K^+$ which will be reconstructed from long tracks alone. To show the effect of differences between the two seeding algorithms more clearly, the Forward tracking will be disabled in the re-reconstruction, so that long tracks can only be found if one of the two standalone T station reconstruction algorithms finds the track. This means that a re-reprocessing of preselected events is also needed for TsaSeeding to switch off the Forward tracking which is active in the official reconstruction.

There is a reason to favour decays with an intermediate resonance like the J/ψ in $B_u^+ \longrightarrow J/\psi K^+$: Cutting on the mass difference between the the dimuon combination and the nominal J/ψ mass allows to reduce many backgrounds using purely kinematical quantities; in decays without such an intermediate resonance, these backgrounds have to be suppressed using other means, for example using particle identification. Since we are specifically interested in the behaviour of the track reconstruction efficiency, it seems prudent to chose channels whose selection depends as much as possible on quantities which are provided by the track reconstruction itself.

Bias estimate

To estimate the per-track bias, 10,000 simulated $B_u^+ \longrightarrow J/\psi X$ events are used. As explained above, signal yield cannot be gained by tracks which are found by PatSeeding but not by TsaSeeding. The more significant figure is the prediction for the loss which is caused by tracks which are found by TsaSeeding but not by PatSeeding.

Since only long tracks are used to reconstruct the B_u^+ and the Velo segment of the final long track is the same for both re-reconstructions, it is sufficient to study the overlap of

	number of tracks			TsaSeeding		PatSeeding	
category	only Tsa	both	only Pat	only Tsa [%]	both [%]	both [%]	only Pat [%]
long tracks	7928	240047	13275	3.20(0.04)	96.80(0.04)	94.76(0.04)	5.24(0.04)
long $B_u^+ \longrightarrow J/\psi K^+$ daughters	292	11332	300	2.51(0.15)	97.49(0.15)	97.42(0.15)	2.58(0.15)

Table 5.9.: *Overlap in track sample between PatSeeding and TsaSeeding (abbreviated "Pat" and "Tsa"). Columns 2 to 4 show the number of long-reconstructible tracks found only by TsaSeeding, by both and only by PatSeeding. The following two pairs of columns give details on how big a percentage of the output tracks of one algorithm is also reconstructed by the other and how big a fraction is actually unique; statistical uncertainties are given in parentheses.*

long-reconstructible tracks found by PatSeeding and TsaSeeding. The result of such a study can be seen in Table 5.9.

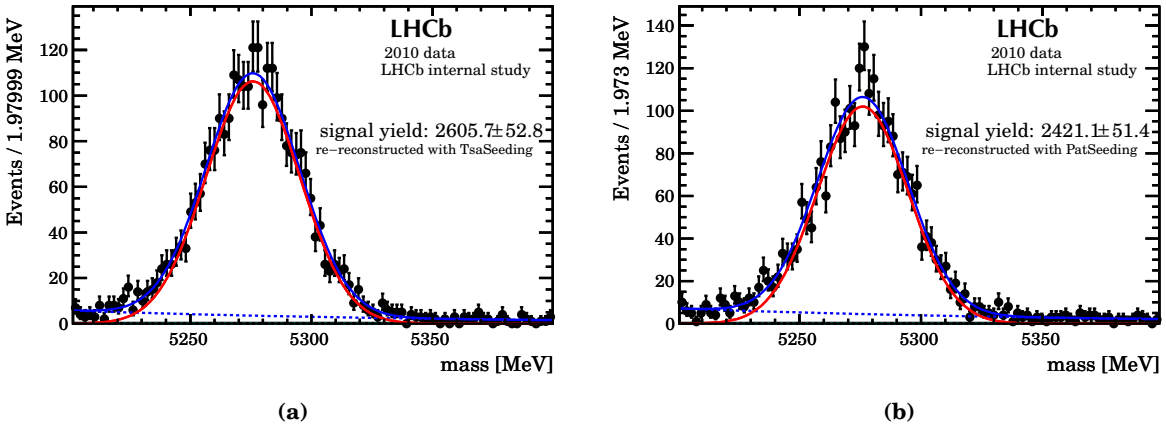
From the fraction of tracks only found by TsaSeeding, one would predict a mean loss of $2.51 \pm 0.15\%$ per track for the $B_u^+ \longrightarrow J/\psi K^+$ decay products. Since there are three tracks in the final state of the $B_u^+ \longrightarrow J/\psi K^+$ decay, the loss in yield of a sample re-reconstructed with PatSeeding over the same sample re-reconstructed with TsaSeeding is estimated to be 92.66%±0.43%.

Yield in $B_u^+ \longrightarrow J/\psi K^+$

A sample of preselected $B_u^+ \longrightarrow J/\psi K^+$ was taken from an ongoing analysis effort within LHCb (see [55] for the final results of that analysis). The event sample used for the yield determination corresponds to roughly $21 \pm 3\ \mathrm{pb}^{-1}$ and was re-reconstructed once with PatSeeding and once with TsaSeeding. To make sure that the long track reconstruction reflects the properties of these two main tracker reconstruction algorithms, the Forward tracking was deactivated because it provides an alternate path to produce long tracks. For the two resulting samples, the B_u^+ selection was run again, and a fit was made to each to extract the B_u^+ yield, see Figure 5.13. Details on the fit model used can also be found in [55] (Appendix D1, equation 101); the selection used is also discussed there (Section 7.1 in [55], but contrary to what is described there, PID cuts were used to select the K^+ used for this study to obtain a cleaner signal). The extracted yields are 2605.7 ± 52.8 for the sample re-reprocessed with TsaSeeding and 2421.1 ± 51.4 for the sample re-reprocessed with PatSeeding; the quoted uncertainties are the statistical uncertainties from the fit.

As expected, the signal yield is higher for TsaSeeding; the loss of PatSeeding with respect to TsaSeeding is $(92.9 \pm 2.7)\%$ which is in excellent agreement with the prediction of $(92.66 \pm 0.43)\%$ from above. The excellent agreement between the predicted and the observed yield loss means that it is possible to predict performance improvements due to the use of PatSeeding with simulated events, so the advantages of PatSeeding over TsaSeeding exist not only for simulated events but also with real data under realistic conditions.

Given the fact that PatSeeding is more efficient than TsaSeeding on simulated events,



(a) **(b)**

Figure 5.13.: *Signal yield in $B_u^+ \longrightarrow J/\psi K^+$ for a preselected sample of 2010 data (preselection based on events reconstructed with TsaSeeding), re-reconstructed without the Forward tracking, once with TsaSeeding, once with PatSeeding. The background contribution is the dashed blue line, the red solid line is the signal part, and the blue solid line is the sum of both. The uncertainties for the yield which are given in the plot are the statistical uncertainties from the fit.*

one expects a slightly improved track reconstruction efficiency when PatSeeding will be used instead of TsaSeeding for the 2011 data. According to Table 5.9, the gain in signal yield in $B_u^+ \longrightarrow J/\psi K^+$ will be tiny, but long track reconstruction in general should profit (the exact amount will of course depend on the analysed decay).

6. Implementation of PatSeeding

The aim of this chapter is to give a detailed view of PatSeeding, most of which was developed during this thesis. This chapter is rather technical in its nature, so that it can serve as reference or manual for PatSeeding, since detailed documentation is crucial for the long-term maintenance of the algorithm.

PatSeeding is steered by several job options, which allows much of its behaviour to be modified on a per-job basis. For clarity, the names of these job options and their use will be mentioned in the text, with the names set in a `typewriter font`. Names which correspond to classes or names of data types have been set using a sans serif font.

In this chapter, a lot of cuts inside PatSeeding are mentioned (about 50), and not all of them will be justified in detail. However, the values chosen are the result of the experience gained during years of study with both simulated events and real data.

6.1. General structure of the algorithm

The *algorithm*[1] PatSeeding performs the pattern recognition using a *tool*[2], PatSeedingTool, which can be reused by other algorithms (e.g. in the trigger). Therefore, this chapter effectively describes PatSeedingTool. A list of options with their default values is given in the Appendix.

The algorithm consists of five distinct steps:

1. hit preparation
2. track search per detector region
3. track search for tracks migrating from IT to OT
4. track search per Outer Tracker region for tracks at large $|y|$
5. validation of low quality tracks left over from the per region search

At the end of steps 2-5, there is a phase in which the resulting tracks have to compete (see Section 6.6); this final competition stage keeps the number of clones produced by the algorithm low and ensures that only the best track candidates are passed on for subsequent processing.

Each layer of the main tracking stations is divided into six different regions (cf. Fig. 6.1). Because there are few tracks migrating between regions, the second step is executed per detector region which reduces the amount of combinatorics which needs to be considered.

[1] An instance of an *algorithm* (when the term is used in the framework of LHCb software) is an instance of a class which runs at most once per event.

[2] The term *tool* is used in LHCb software to denote a reusable piece of software with a well defined interface;

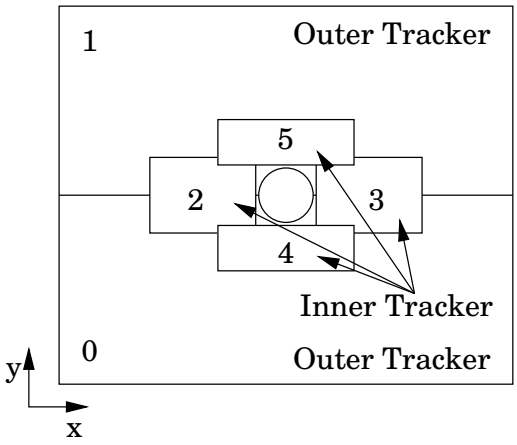


Figure 6.1: *All layers in the main tracking stations are divided into six regions (OT: 0-1, IT: 2-4). Not to scale.*

Once all tracks staying within a single region have been found, tracks are formed from the remaining hits. This step allows for tracks which leave the Inner Tracker and continues to search for hits in the Outer Tracker.

The remaining steps recover tracks that are difficult to find in previous stages.

6.2. Hit preparation

PatSeedingTool obtains a range of hits from the T station hit manager [3], a tool which allows access to the decoded hit data and geometry information for Inner and Outer tracker. Moreover, the hit manager performs a hit cleaning step by default (see Section 6.2.1) and it forms *clusters* of adjacent Outer Tracker straws by marking hits belonging to a cluster with a flag. These clusters of adjacent OT hits are typically small (single hits or two-hit clusters being the most common), see Figure 6.2. For the Inner Tracker, a similar procedure is already applied during the readout process, so IT clusters already appear as "single hits" to the pattern recognition.

The hits returned by the hit manager have also been sorted by increasing x coordinate at $y = 0$. Due to the 3.6 mrad tilt of the LHCb coordinate system with respect to gravity, the x and z coordinates of silicon strips and Outer Tracker straws depend on the y coordinate at which they are given; $y = 0$ is chosen as a reference to start from until the y coordinate of a track is known.

Each hit has a flag indicating if it is used. Per default, this flag is initialised to indicate an unused hit, unless the hit has an unphysical drift time; for OT hits, a cut is placed on the drift radius r obtained from the drift time such that $-0.6\,\text{mm} < r < 2.8\,\text{mm}$ (the corresponding job option is the array `DriftRadiusRange`). If PatSeeding is used under "special circumstances", e.g. in the software trigger (see Section 7.1), it also marks hits as used if

- if `ReusePatSeeding` is false, hits used by a previous run of PatSeedingTool are marked used; this may be useful in the trigger to save some time if PatSeedingTool is invoked more than once per event

an instance of a tool can be shared among algorithms and run several times in the same event.

[3] Tf::TStationHitManager<PatForwardHit>

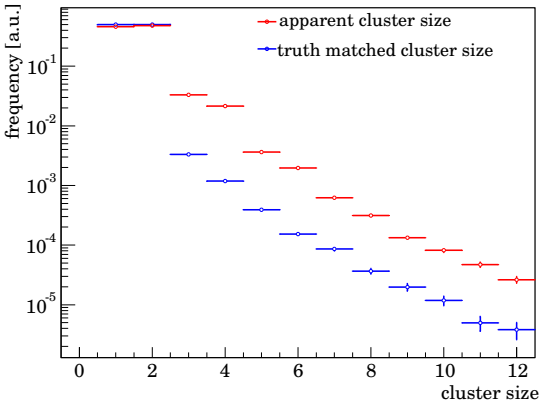


Figure 6.2: *Outer Tracker cluster sizes in simulated events, either found by marking hits in adjacent straws ("apparent") or by marking hits in adjacent straws if produced by the same Monte Carlo particle ("truth matched"). One and two hit clusters are the most common for geometrical reasons, but longer chains of hits do occur for tracks with large slopes, tracks which curl inside an OT module, high local track density, cross-talk or noise.*

- if `UseForward` is true, hits used by a previous run of the PatForward algorithm are marked as used hits, thus preventing tracks already found by PatForward from being found again; this is useful if algorithm execution time is deemed too long, which allows to save time with only a small loss in reconstruction efficiency (see Section 7.3 for details)

6.2.1. Hit cleaning

Hit cleaning removes certain classes of hits before pattern recognition algorithms have a chance to see and work with them. The idea is that there are certain configurations of hits from which the pattern recognition cannot draw sensible conclusions, but spends considerable time trying to do so. By removing these hits beforehand, the efficiency of the pattern recognition remains practically unaffected, while the worst case timing behaviour improves a lot.

If hit cleaning is used, it removes the following types of hit configurations:

- Outer tracker modules with occupancies larger than 40%; since occupancies are typically much lower, one is forced to conclude that either the read-out process was faulty, or there is some form of noise affecting all channels in a read-out front-end. In any case, a pattern recognition algorithm cannot solve such a problem, only make it worse by creating ghost tracks.

- Readout from Beetle chips, the Inner tracker frontend chip, is discarded if the occupancy in that chip is larger than 37.5%; the reasoning behind this decision being the same as for the Outer tracker above.

- Chains of 6 or more neighbouring hits in the Outer Tracker; these chains are rare and unlikely to be produced by a single particle (cf. Figure 6.2), so there is little information in these chains. If a pattern recognition algorithm happens to pick the correct hit(s) from such a chain, then that is because the track trajectory was known well enough from the other hits on that track, so there is little gain in spending time trying to use hits in such a chain.

6.3. Track model and fit method

The track model and the fit are basic building blocks of the pattern recognition strategy. They are explained ahead of the actual pattern recognition algorithm to have them available when needed.

6.3.1. Track model constraints from geometry and field shape

Without magnetic field in the main tracking stations, and ignoring multiple scattering and energy loss, tracks would just be straight lines. The actual field in the main tracker is not negligible and has its largest influence on the trajectory of a particle in $x-z$ projection. The T station geometry and the magnetic field have some properties which influence the shape of the track segments which the pattern recognition attempts to find:

- A single main tracker station spans a range in z direction which, when multiplied by typical track slopes, is much larger than the spatial resolution of the detectors. Therefore, one cannot form "space-points" from hits inside a single station without taking the direction of the particle (i.e. the track slope) into account.
- The field is decreasing with z, which means that the curvature between the first and the second station of the main tracker is higher than the one between the second and the third. The ratio of the integral of the magnetic field in the two gaps between stations (i.e. the field integral in the gap between T1 and T2 divided by the field integral in the gap between T2 and T3) is reasonably constant over the sensitive area of the main tracking system. This means that the „loss" of track curvature towards the end of the main tracker at high z is essentially proportional to the curvature itself.
- The field varies with y, and in the uppermost (and lowermost) part of the T stations, it is essentially zero. Therefore, one expects tracks at high $|y|$ to be almost straight lines.

These properties guide the choice of track model described below.

6.3.2. Track model

The algorithm employs a straight line description of the trajectory of a particle in $y-z$ projection. For $x-z$ projection, a cubic model with three track parameters a, b, c is used:

$$x(z) = c(z - \texttt{zReference})^2(1 + \texttt{dRatio}(z - \texttt{zReference})) + b(z - \texttt{zReference}) + a \quad (6.1)$$

$$y(z) = b_y z + a_y \quad (6.2)$$

`zReference` is in the middle of the main tracker at $z = 8250$ mm; the shift by `zReference` makes the fit more numerically stable because the amount of correlation among the track parameters is minimised.

`dRatio` describes the correlation between the quadratic and cubic terms and is determined from Monte Carlo studies. As described above, it is negative (resulting in a curvature loss with increasing z which is proportional to the curvature itself), and has a

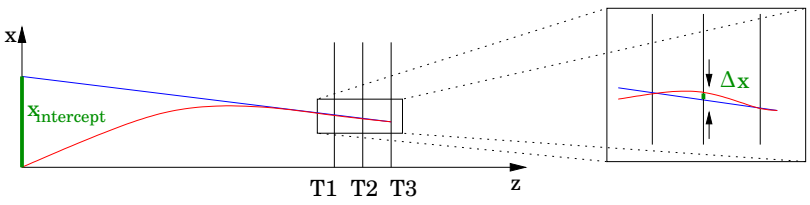


Figure 6.3.: *Assuming a track comes from the origin, one can estimate the position of a hit in T2 relative to a straight line joining hits in T1 and T3. This displacement Δx (best visible in the magnified portion on the right) is proportional to the intercept of the straight line with the x axis, $x_{intercept}$. Not to scale.*

numerical value of $-3.2265 \cdot 10^{-3}$ (in units of mm^{-1}, with lengths measured in mm as is done in the LHCb software stack). The distance dz from `zReference` to either end of the main tracker is of the order of a metre, so the order of magnitude of $\texttt{dRatio} \cdot dz$ is unity, so the curvature loss is a sizable fraction of the total track curvature. For high momentum tracks, this is less important (they have a small curvature anyway), but the reconstruction of low momentum tracks benefits, as has been shown in Section 5.2.1.

For brevity, the cubic track model in $x-z$ projection will also be referred to as "parabola".

So far, the properties influencing the choice of track model have had their origin in the shape of the magnetic field in the main tracking system. There is, however, one other important property of the global shape of the field of the LHCb dipole (and hence the track model chosen) which can be exploited: Under the assumption of a track coming from the primary vertex and due to the magnetic field in the T stations being roughly proportional to the integral of the field along the trajectory of a particle through the magnet, one can derive a linear relation between the x coordinate of a line joining two x hits in stations 0 and 2 at $z = 0$, $x_{intercept}$, and estimate the curvature c:

$$c \approx \texttt{InitialArrow} \cdot x_{intercept} \tag{6.3}$$

`InitialArrow` has a numerical value of $4.25307 \cdot 10^{-9}$ (again, all lengths in equation 6.3 are measured in mm) and is determined from Monte Carlo studies. This is used in several places of the algorithm. The most important application (and probably also the easiest to understand) is to predict the sagitta Δx, i.e. the x deviation from the straight line mentioned above in station 1 (i.e. T2) which is essentially proportional to the track curvature (see sketch in Figure 6.3). Figure 6.4 shows a clear correlation between $x_{intercept}$ and Δx for ideal pattern recognition (i.e. reconstruct truth-matched tracks with 100% hit purity), which nicely illustrates the correlation which leads to equation 6.3.

This relation holds rather well in the centre of the T stations, but not near the top or the bottom where the field inside the T stations is essentially zero. Therefore, the algorithm has an extra pass over the hits with `InitialArrow` put to zero to recover tracks at high $|y|$ (see 6.5.2).

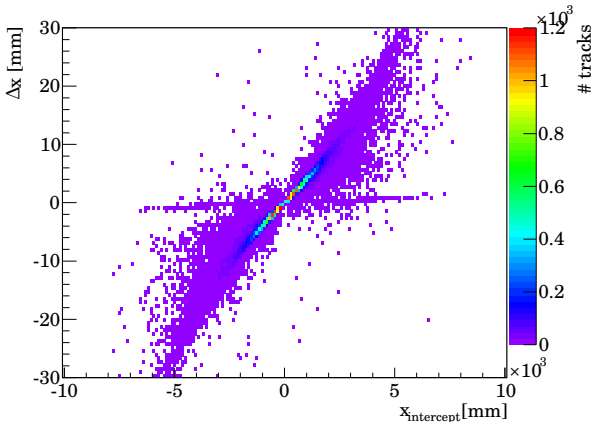


Figure 6.4: *Correlation between $x_{intercept}$ and track sagitta Δx in T2 for ideal pattern recognition (i.e. reconstruct truth-matched tracks with 100% hit purity). A clear correlation between $x_{intercept}$ and Δx exists, with some scatter due to low momentum tracks, and a weak almost horizontal structure caused by the finite spatial resolution in the OT which is important for low curvature tracks.*

6.3.3. Fit method

The fit is a standard weighted least-squares fit; an estimate of the intrinsic spatial resolution of the hits is provided by the hit classes themselves.[4] Throughout the fitting procedure, the coordinates of the hits are updated according to the changes in track parameters to account for shifts in x and z coordinates along the wire or silicon strip due to stereo angles, the 3.6 mrad tilt of the main tracker coordinate frame with respect to the LHCb one and known misalignments.

In a first step, initial parameters are obtained by just fitting first the $x-z$ and then $y-z$ projection separately, using only x hits (i.e. hits from x layers) in the first case and only stereo hits (i.e. hits from u and v layers) in the second.

Then, the fit is iterated up to ten times each to account for correlations and resolve ambiguities in the Outer Tracker (see below for details). Two possibilities for the actual fit are examined:

- to be fast, the fits for the $x-z$ and $y-z$ projections are kept separate and are repeated (up to ten times) in this order until fit parameters become stable
- a simultaneous fit to x and stereo hits is used to obtain the best possible track parameters (slightly slower since the resulting system of equations is larger, but correlations between x and stereo hits are correctly accounted for)

The two possibilities are compared near the end of the section.

The fit algorithm minimises

$$\chi^2 = \sum_{\text{hits}\, i} \left(\frac{x_i - x_{\text{track}}(z_i) - y_{\text{track}}(z_i)(\frac{dx}{dy})_i \pm r_{\text{drift}}/\cos(\alpha)}{\sigma_i/\cos(\alpha)} \right)^2 \tag{6.4}$$

Here, (x_i, z_i) is the x coordinate of the i-th hit at the $y_{\text{track}}(z_i)$ predicted from the track model[5], $(\frac{dx}{dy})_i$ is the change of the x coordinate per change in y due to a potential stereo

[4] The Tf-framework is described in [56].

[5] The x coordinates of hits on a track are shifted by $y \cdot (dx/dy)$ as soon as an estimate of y at the z position

angle of the wire/strip. The factor $\cos(\alpha)$ accounts for the fact that the drift radius r_{drift} (and the position uncertainty) have to be modified due to the slope of a track in $x-z$ projection: In the Outer Tracker, the nominal hit position (x_i, z_i) is the wire position, while the true hit is somewhere on a drift circle around that wire. Dividing the drift distance by $\cos(\alpha) = \frac{1}{\sqrt{1+b^2}}$ (b as defined in equation6.1) converts it into the distance at the z coordinate of the wire itself, which is what is needed for the fit.[6] In the inner tracker, this factor is unity. σ_i is the hit position uncertainty. For the Inner Tracker, the term including r_{drift} is omitted. For the Outer Tracker, the sign of the term including r_{drift} is not determined a priori because it is not known on which side of the wire the track passes. During the initial iteration of the fit, these *left-right ambiguities* in the OT are resolved using so-called *pitch residuals* (see below). For subsequent iterations, the sign is chosen such that the smaller contribution to χ^2 is taken because the track parameters are sufficiently close to the true position of the particle that the correct solution is chosen most of the time.

If the changes in parameters fall below the following values, the iterations are stopped early:

$$\begin{array}{ll} |\Delta a| < 5\cdot 10^{-3} & |\Delta a_y| < 5\cdot 10^{-2} \\ |\Delta b| < 5\cdot 10^{-6} & |\Delta b_y| < 5\cdot 10^{-5} \\ |\Delta c| < 5\cdot 10^{-9} & \end{array}$$

After the iterations, a track χ^2 is recalculated. If the χ^2 contribution of the worst fitting hit is above a threshold for outlier removal, it is removed from the fit, and the fit is restarted from scratch. If the number of layers with hits falls below a minimum, the fit returns, indicating an error, and the track candidate is discarded by the pattern recognition algorithm. Each track is assigned a track χ^2 which is the χ^2 just calculated, divided by the number of degrees of freedom.

Outlier cuts and minimum number of hits are set to different values in different places of the algorithm; the values used will be described later.

To compare the two alternative fit methods (separate $x-z$ and $y-z$ projection fits versus a single simultaneous fit), track-hit distances (i.e. biased hit residuals) can be compared for both alternatives. To make sure that purely the effect of the two different fitting methods is studied (e.g. the way left/right ambiguities are resolved in the Outer Tracker could also play a role), the check is done with Monte Carlo truth matched tracks which have only hits in the Inner Tracker. Figure 6.5 shows the residual distributions for both alternatives; the details of the triple Gaussian fits performed have been explained before, and are also given in Appendix A. A slightly narrower distribution is obtained with the simultaneous fit; it also finds more hits as can be seen from the number of entries in the histogram. Therefore, PatSeeding uses the simultaneous fit procedure by default.

Resolving left-right ambiguities

In the Outer Tracker, the track reconstruction has to decide if a track going through the detector has passed to the left or the right of the wire that collects the ionisation charge

of the hit is available.

[6] Strictly speaking, $\cos(\alpha)$ is a function of z, i.e. $\cos(\alpha) = \left.\frac{1}{\sqrt{1+(d/dt\,x(t))^2}}\right|_z$ with $x(t)$ defined as in equation

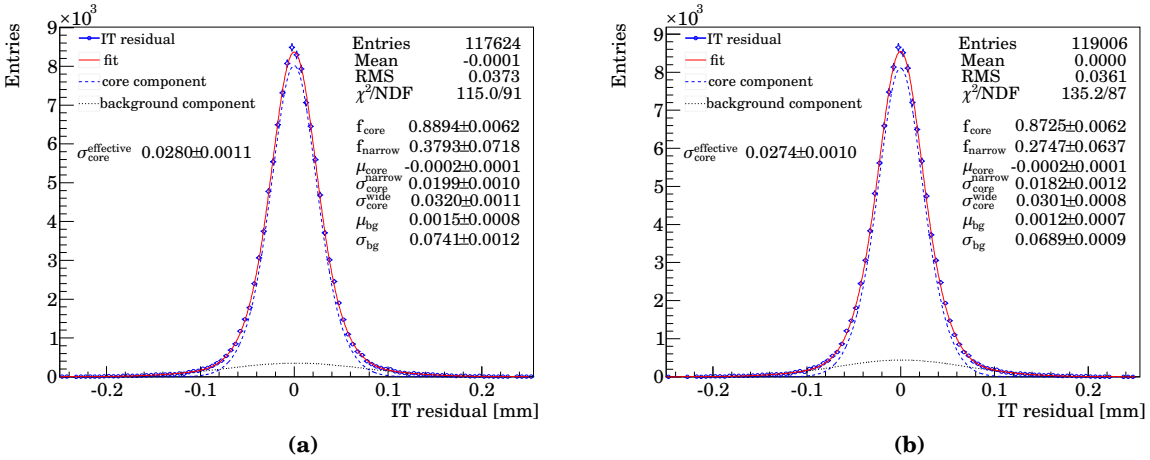


Figure 6.5.: *(Biased) track-hit residuals for Inner Tracker only truth-matched tracks. In (a), the approach of separate fits to the $x-z$ and $y-z$ projections was used, in (b), a simultaneous fit to both projections is employed. In (b), the both the tails and the core of the distribution are narrower, a fact which is also reflected in the slightly smaller RMS in (b). Moreover, the number of hits found has increased by about 1.2% with the simultaneous fit.*

in a straw. This is especially important for obtaining accurate track parameters in the fit described in the previous section.

Different approaches to solve that ambiguity typically fall into two categories: Either an algorithm can try several solutions (typically by exhaustively trying out all possible combination on a subset of hits, e.g. those which have the largest lever arm on the track parameters) and chose the one which gives the best track χ^2, or it is possible to resolve ambiguities by exploiting (geometrical) properties of the way the drift chamber is built.

For PatSeeding, one approach of each category was tried:

- The (historically) ”old” approach works by selecting the x (stereo) hits with the largest drift radius in the x (stereo) layers of each station, because it is these hits that will have the largest impact on track parameters. For the x layers, all eight possibilities to resolve the ambiguities are tried, and the algorithm chooses the one for which the mean squared distance of all other x hits to a parabola through the three initially chosen x hits is smallest. For the stereo layers, the two hits with the largest drift distances among the three preselected stereo hits hits are used to define a straight line, and in a similar manner to the x hits above, all four combinations are tried, and the one which best fits the stereo hits (in the sense defined above) is retained.

- The second approach exploits the geometrical properties of the Outer Tracker modules by using so-called *pitch residuals* (see below).

To judge which approach works better, we have a look at the track-hit distance (i.e. the

6.1; b is used as an approximation instead of the derivative for to save some CPU time.

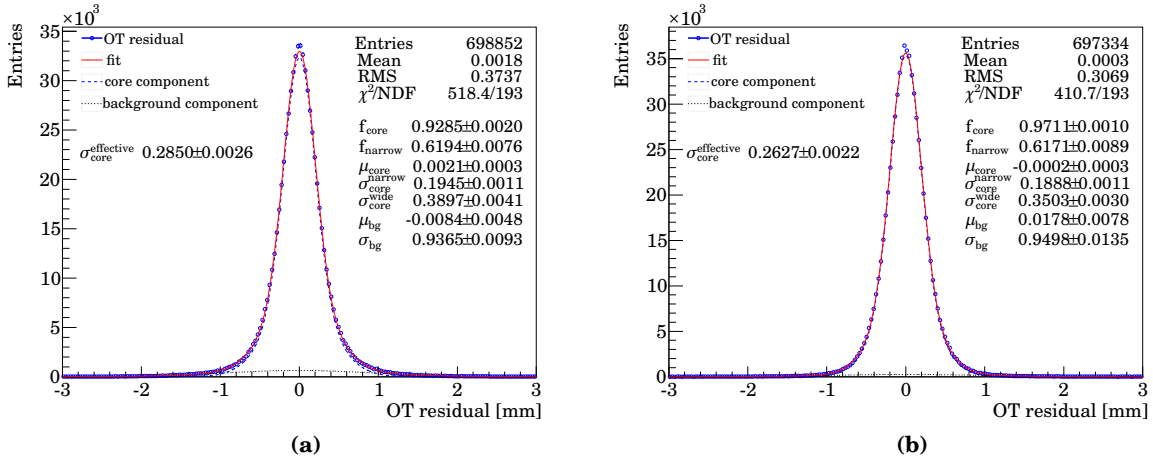


Figure 6.6.: *(Biased) track-hit residuals for Outer Tracker only truth-matched tracks. In (a), the "old" approach of resolving ambiguities was used, in (b), ambiguities are resolved using pitch residuals. In (b), the tails of the distribution are much less pronounced, a fact which is also reflected in the smaller RMS. In both cases, the effective width is wider than the 0.2 mm which one would expect from the settings that were put into the simulation. This is due the fact that the plots above still contain outliers, and the time-of-flight correction for the Outer Tracker drift time measurements is incorrect for tracks with low very momenta ($\beta = v/c < 1$), leading to a bias drift radii towards larger values.*

(biased) residual) after the full pattern recognition fit has been performed. Fig. 6.6 shows the hit residual distributions for both alternatives for reconstructed tracks which have been matched to Monte Carlo truth (we select tracks which have only Outer Tracker hits to avoid potential effects due to Inner Tracker hits). The tails in the distribution for ambiguity resolution using pitch residuals are much less pronounced, and the RMS is smaller. Since both variants reconstruct essentially the same number of truth-matched tracks and the ghost rate is slightly lower for the pitch residual case, it is clear that the pitch residual based option is preferable and has therefore been made the default. There is an additional benefit: Using the pitch residual based method resulted in an average reconstruction time of 9.6 ms per event instead of 11.5 ms per event using the "old" approach for ambiguity resolution.

To understand how pitch residuals work, the key observation is that a track usually leaves two hits with highly correlated drift radii in the two staggered monolayers of straws inside an OT module (Fig. 6.7a). In the simplest case where the direction of the particle is perpendicular to the module, the sum of the drift radii must be equal to the straw pitch (amount of staggering between the two monolayers in the module) within the uncertainties of the drift radius measurement. The difference between the sum of drift radii and the nominal pitch is called *pitch residual*. In this simple case of a track perpendicular to an Outer Tracker module, it is also clear how to resolve the ambiguities: The correct solution is the solution for which the track passes between both wires.

This idea is easily generalised to tracks not perpendicular to a module. First, the

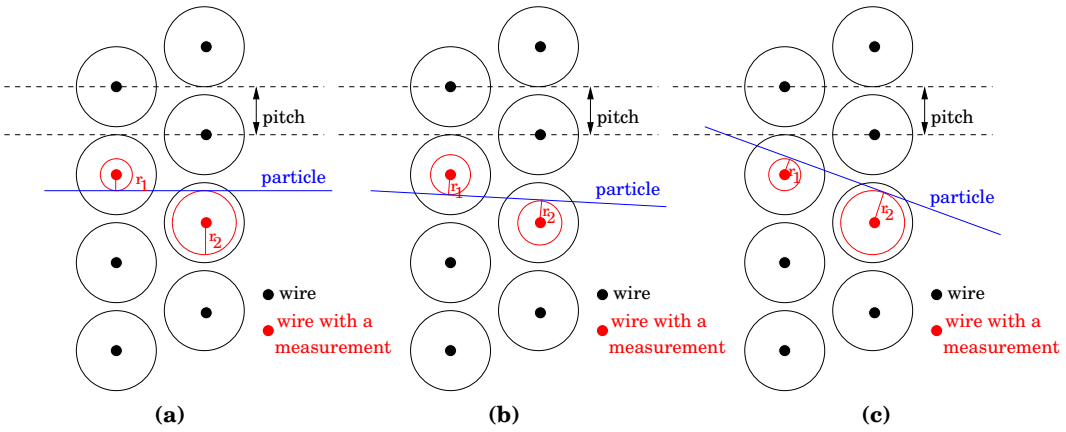


Figure 6.7.: *In the Outer Tracker, a particle typically produces two hits in adjacent monolayers of a module. In the simple case where the direction of the particle is perpendicular to the module (a), the sum of the drift radii must be equal to the pitch d (amount of staggering between the two monolayers in the module) within the uncertainties of the drift radius measurement. For non-perpendicular tracks, there are two possibilities: either the track passes between the two wires (b), or it does not (c).*

pitch d has to be replaced by an effective pitch $d_{eff} = |\vec{a} - (\vec{a} \cdot \vec{s}) \cdot \vec{s}|$ with the vectors $\vec{a}$ and $\vec{s}$ as defined in Fig. 6.8. The second step consists in realising that there are in fact two different topologies possible for non-perpendicular tracks: either the track passes between the two wires or it does not (Figs. 6.7b, 6.7c). In the former case, the sum of the two drift radii is equal to the effective pitch (within the uncertainties), in the latter case, the absolute value of the drift radius difference plays the same role. To decide if the track passes between the wires or not, one can check if $p_+ = |r_1 + r_2| - d_{eff}$ (between the wires) or $p_- = |r_1 - r_2| - d_{eff}$ is smaller (r_1 and r_2 are the two drift radii). The final definition of the pitch residual p reads:

$$p = \begin{cases} p_+ & \text{if } |p_+| < |p_-| \\ p_- & \text{otherwise} \end{cases}$$

Figure 6.9 shows the distribution of pitch residuals for reconstructed Monte Carlo truth-matched tracks. It can be seen that the effective width of the core (as defined in Appendix A), $\sigma_{core}^{effective} = 0.299\,\text{mm}$ is quite close to what one would expect from the Outer Tracker resolution put into the Monte Carlo, i.e. $\sigma_{pitchres.}^{MC} = \sqrt{2}\sigma_{OT} = \sqrt{2} \cdot 0.2\,\text{mm} = 0.283\,\text{mm}$. This is interesting also to measure the resolution of the Outer Tracker while remaining largely independent of misalignments: any misalignment of straws inside an Outer Tracker module is known to be tiny compared to the Outer Tracker resolution from module construction, and the slopes are not very sensitive to small misalignments due to the large lever arm in z.

In case the track passes between both wires, it is easy to resolve the ambiguities. Otherwise, the most straightforward thing to do is to check which of the two remaining

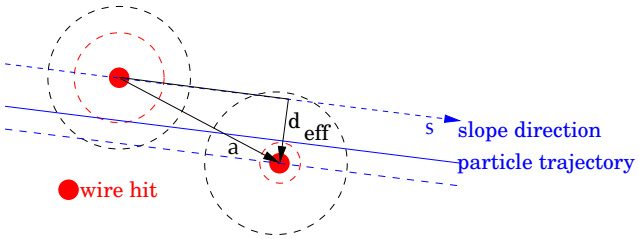


Figure 6.8.: *For non-perpendicular tracks, the pitch d has to be generalised to an effective pitch d_{eff} which can be obtained by removing from the vector $\vec{a}$ joining the two wires the component in direction of the track ($\vec{s}$ is a unit vector in track direction).*

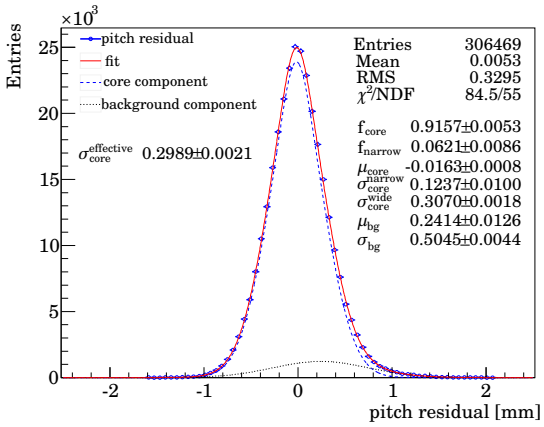


Figure 6.9: *Pitch residuals in reconstructed Monte Carlo truth-matched tracks. It can be seen that the effective width of the core, $\sigma_{core}^{effective} = 0.299\,\mathrm{mm}$ is quite close to what one would expect from the Outer Tracker resolution put into the Monte Carlo, i.e. $\sigma_{pitchres.}^{MC} = \sqrt{2}\sigma_{OT} = \sqrt{2}0.2\,\mathrm{mm} = 0.283\,\mathrm{mm}$.*

possibilities — the track is either to the left or to the right of both straws — is more consistent with the measured slope of the track which is known to very good precision without resolving ambiguities because of the long lever arm of the remaining hits in the main tracker.

If a track leaves only a single hit in either of the two monolayers of a module, the ambiguity cannot be resolved for that hit, and the problem is left to subsequent iterations of the fit which resolve the ambiguity towards the track as described above.

Technical implementation of the fit

To solve linear systems of equations arising from (linearised) fits, some form of matrix decomposition is needed to solve for the fit parameters iteratively. Assuming a good guess of fit parameters $\vec{x}_i$ from a previous iteration i which is close to the true solution $\vec{x}_{true}$, it is possible to expand the expression for χ^2 around the current best guess for the fit parameters $\vec{x}_i$ to obtain a fit parameter update vector $\delta\vec{x}$ up to quadratic order in $\delta\vec{x}$; since $\vec{x}_i$ is supposed to be sufficiently close to the minimum, higher order terms can be neglected, and the updated set of fit parameters thus reads $\vec{x}_{i+1} = \vec{x}_i + \delta\vec{x}$:

$$\chi^2(\vec{x}_i + \delta\vec{x}) = \chi^2(\vec{x}_i) + \nabla\chi^2\big|_{\vec{x}_i} \cdot \delta\vec{x} + \sum_{k,l} \frac{1}{2}(\delta\vec{x})_k \left.\frac{\partial^2\chi^2}{\partial x_k \partial x_l}\right|_{\vec{x}_i} (\delta\vec{x})_l$$

Forming the derivative with respect to $\delta\vec{x}$ and putting it to zero, we obtain an equation for $\delta\vec{x}$:

$$\left(-\nabla\chi^2\big|_{\vec{x}_i}\right)_k = \left.\frac{\partial^2\chi^2}{\partial x_k \partial x_l}\right|_{\vec{x}_i} (\delta\vec{x})_l$$

From this, it is immediately obvious that the matrix $M = \left.\frac{\partial^2\chi^2}{\partial x_k \partial x_l}\right|_{\vec{x}_i}$ which turns up in the problem is symmetric and has only positive Eigenvalues (because we are sufficiently close to a minimum of χ^2). We can thus decompose the matrix into a product of a lower triangular matrix L with real entries and its transpose:

$$M = LL^T \qquad \text{with} L = \begin{pmatrix} * & 0 & \cdots & 0 \\ * & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ * & \cdots & \cdots & * \end{pmatrix}$$

This matrix decomposition is known as Cholesky decomposition (more details are given Appendix D). The solution of a linear system $Mx = r$ is then straightforward: First, solve $Ly = r$ for y using substitution (L is a lower triangular matrix), then the solution x can be obtained by another substitution step on $L^T x = y$. It should be noted that the numerical stability of Cholesky decomposition, one of its many good properties, is important to reliably do simultaneous fits to both x and stereo hits which would difficult to achieve in a reasonable amount of time otherwise.

If we call our parameter vector $\delta\vec{x} = (\delta a, \delta b, \delta c, \delta a_y, \delta b_y)^T$, with δa, δb, δc, δa_y, δb_y being corrections to a previous reference trajectory, and assume we fit simultaneously for all parameters, we can write down the full form of the linear system $Mx = r$ with the abbreviations $dz_i = z_i - \texttt{zReference}$ (i is the index which runs over hits), $\eta_i = dz_i^2(1 + dz_i \cdot \texttt{dRatio})$, $\zeta_i = (\frac{dx}{dy})_i$, and dx_i the hit-track distance in x direction, or, more precisely, the numerator of the fraction that is squared in the χ^2 calculation in equation 6.4, using the convention $\langle q \rangle = \sum_i \frac{1}{\sigma_i^2/\cos^2(\alpha_i)} q_i$ for some hit-based quantity q:

$$M = \begin{pmatrix} \langle 1 \rangle & \langle dz \rangle & \langle \eta \rangle & \langle -\zeta \rangle & \langle -\zeta dz \rangle \\ \langle dz \rangle & \langle dz^2 \rangle & \langle \eta dz \rangle & \langle -\zeta dz \rangle & \langle -\zeta dz^2 \rangle \\ \langle \eta \rangle & \langle \eta dz \rangle & \langle \eta^2 \rangle & \langle -\zeta\eta \rangle & \langle -\zeta\eta dz \rangle \\ \langle -\zeta \rangle & \langle -\zeta dz \rangle & \langle -\zeta\eta \rangle & \langle \zeta^2 \rangle & \langle \zeta^2 dz \rangle \\ \langle -\zeta dz \rangle & \langle -\zeta dz^2 \rangle & \langle -\zeta\eta dz \rangle & \langle \zeta^2 dz \rangle & \langle \zeta^2 dz^2 \rangle \end{pmatrix} \qquad r = \begin{pmatrix} \langle dx \rangle \\ \langle dx dz \rangle \\ \langle dx\eta \rangle \\ \langle -dx\zeta \rangle \\ \langle -dx\zeta dz \rangle \end{pmatrix} \tag{6.5}$$

With a view towards outlier removal, it is also interesting to note that this formulation allows adding and removing the contribution of hits from M and r by element wise addition and subtraction of the appropriate hit quantities as long as the reference trajectory is not changed. This is not used in the present algorithm because removing a hit might change the way ambiguities are resolved in the Outer Tracker which can lead to an appreciable shift in the track position in the layer where the hit was removed, a fact which makes a track refit seem the prudent course of action. For pattern recognition algorithms which do not have to deal with such complications (e.g. an algorithms for a homogeneous silicon detector), this property might be very interesting.

6.4. Track search per detector region

After treating the details of the track model and track fit used in PatSeeding, the process of searching a set of hits from which to form a track must be described. The track search per detector region described in this section consists of two phases: First, track candidates are searched in $x-z$ projection. Then, stereo hits are collected, and a final track selection is made. The resulting track is fitted using a weighted least squares method as described above.

6.4.1. Track search in $x-z$ projection

From each hit in one of the x layers of station 0 and each suitable x hit in station 2, a straight line is constructed. Suitable means that the combination of hits must satisfy both of the following constraints:

- The hit in station 2 must be inside a window formed by lines joining the first hit with $(x = \pm x^{\max}_{\text{impact}}, z = 0)$, where

  $$x^{\max}_{\text{impact}} = \frac{1}{\texttt{MinMomentum}}\left(\frac{1\text{mm}^2}{\texttt{MomentumScale}\times\texttt{InitialArrow}} + 210\text{MeV}\times 2000\text{mm}\right)$$

  The first term translates a maximum curvature (due to a minimal track momentum `MinMomentum` = 500 MeV) and the hypothesis of the track coming from the primary vertex into an impact parameter estimate according to equation 6.3; `MomentumScale` = 35.31328 is the constant of proportionality linking track curvature c to track momentum with $p/\text{MeV} = \frac{\texttt{MomentumScale}}{c/\text{mm}^{-2}}$. The second term accounts for an additional displacement at the origin due to a potential K_S decay. Both contribution are proportional to the inverse minimum momentum.

- The lines joining the hit in station 0 with the points $(x = \pm\texttt{xMagTol}, \texttt{zMagnet})$ form an additional search window in station 2 which restricts the search region roughly to tracks which travel inside the LHCb dipole magnet ("centre of magnet compatibility").

Hits in station 2 have to be inside the search windows defined by the last two items; the situation is sketched in Figure 6.10.

In case of Outer Tracker hits, the hit positions are taken to be the wire positions. In the Inner Tracker, hit positions reflect silicon strip or cluster positions if several strips were combined into a cluster by the IT readout frontend.

If the hit manager indicates that an Outer Tracker hit has a successor with which it can be combined to form a cluster, a weighted mean of the two hit positions is used. The weight of an Outer Tracker hit is $w = \frac{1}{\sqrt{r^2+\sigma^2}}$ where r is the drift distance and σ is the detector resolution. The idea behind this is that hits with small drift radii contain more information about the true particle position than those with larger drift time. They should therefore give the greater contribution. As the measured drift radii can also occasionally be negative due to resolution effects (which would obviously spoil the weights we want to calculate), the modulus of the drift radius is used. This is done by squaring and taking the square root afterwards. The reason for introducing the addition

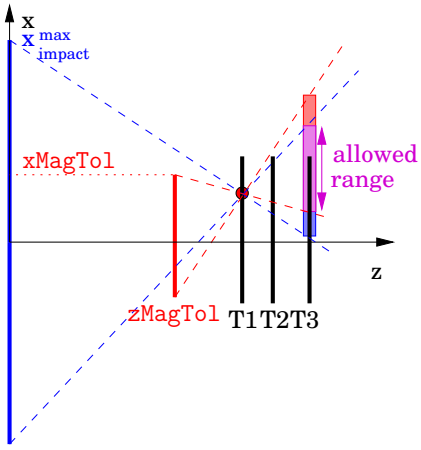


Figure 6.10: *Sketch of search windows in T3 during per region track search given a hit in T1 (red dot); the value of* $x^{\text{max}}_{\text{impact}}$ *used in the sketch corresponds to a* `MinMomentum` *of roughly 1 GeV to conserve space on the page.*

of the squared hit resolution under the square root is that one wants to avoid divisions by zero in case of one of the drift radii being zero. This procedure also has the nice property that it degenerates into the arithmetic mean if drift information is not used or not present (in this case, the drift radii for all hits read zero). Note that a similar procedure was also used in [57].

Then, a hit in station 1 is selected; a track curvature corrected prediction for the hit position can be obtained with the method outlined in equation 6.3 and Figures 6.3 and 6.4. The distance of the hit to this prediction has to be less than the curvature tolerance, `CurveTol` which has a default of 5 mm[7]. From these three points in (x,z)-space, a parabola is formed. This approach has the advantage that the selection of three initial hits is quite fast because a lot of the combinatorics involved in searching for a suitable hit in T2 is suppressed by using a hit position prediction which accounts for track curvature.

Both Outer and Inner Tracker have an inefficient area near $y = 0$. In case of the Inner Tracker, this is caused by a dead area in between two silicon ladders. For the Outer Tracker, modules are electrically divided in the middle to permit separate readout of top and bottom half. The resulting dead area of a monolayer in an Outer Tracker module has been staggered, resulting merely in decreased efficiency for the whole Outer Tracker module near $y = 0$.

To save some time, in the Outer Tracker, only combinations of three hits in either the first x layer of each station or the last x layer of each station are considered at this stage, because layers are sufficiently separated in z that a track passing trough the inefficient region near $y = 0$ are unlikely to go through this region in all layers of a station, allowing the algorithm to gain some time during execution. For the Inner Tracker, the separation in z of layers in the same station is much smaller, so all combinations of x layers are considered to reduce the risk of losing the track.

[7] Due to the tilt of the main tracker coordinate frame with respect to the LHCb one, it is necessary to slightly widen narrow search windows. This widening depends on the distance from the last known point in z, the most extreme slopes t_y expected in the detector region, and known rotations of the detector layer in question. Known rotations around the other axes are treated similarly. Since this geometric effect is not very interesting or difficult in itself but extremely tedious to explain in detail and probably not beneficial in terms of clarity, I will not describe it in places where search windows are used. Instead, such corrections are silently applied in the code where appropriate, and details may be found there.

Then, hits from all other x layers are added, provided that their distance to the parabola is less than the collection distance, `TolCollectIT` = 0.3 mm or `TolCollectOT` = 3mm, respectively. The values of `TolCollectIT` and `TolCollectOT` are chosen depending on the granularity of the subdetector in question and the typical uncertainty of the prediction of hit positions given by the initial parabola. A small additional safety margin is added to obtain some robustness against misalignments.

From these hits, a track candidate is constructed, if

- the resulting candidate does not miss too many layers; it has to have hits in all but `MaxMisses` x layers (the number of missed layers can be evaluated during collection, allowing hit collection to be terminated early if it is clear that the requirement cannot be fulfilled for the current candidate)
- the track has hits in at least `MinXPlanes` x layers after the weighted least squares fit in $x-z$ projection (inside the fit, a cut on outliers is placed at a per-hit contribution to the track χ^2 of `MaxChi2HitIT` or `MaxChi2HitOT`, respectively)
- the fit converged
- the track χ^2 (i.e. χ^2/NDF of the track fit) is below `MaxTrackChi2`
- in case of IT-only tracks, at least 3 hits must have enough charge found in the IT to be above the IT readout high charge threshold — the fraction of hits above the high charge threshold on the final truth-matched tracks (reconstructed without this cut applied) is well above 95%, so this is not a strong requirement; however, it will help to reject "spillover hits" which arise from the tail of slowly drifting charges in the silicon from a previous bunch crossing which sometimes happens to rise over the charge threshold in the current bunch crossing (this does not happen for 2010 data and is not simulated for the 2010 Monte Carlo samples but will become more common when the LHC moves to a bunch spacing of 25 ns)
- in case of an OT-only low multiplicity track (i.e. with less than 8 hits), the following additional requirements have to be met:
  - the outlier cut is tightened to a per-hit χ^2 contribution of `MaxFinalChi2`
  - the track χ^2 is below `MaxTrackChi2LowMult`
  - none of the potential OT clusters may have a hit removed by the outlier removal

The resulting track candidate is then stored for further inspection. To avoid generating too many clones at this early stage, each new track candidate which is about to be stored has its x position in the middle of the main tracker compared with all previously stored candidates. If the x distance between both tracks in the middle of the main tracker is above `CloneMaxXDistIT` = 3 mm (or `CloneMaxXDistOT` = 7 mm, respectively), the new track candidate is stored as is. Otherwise, both candidates are compared for hit content. If the fraction of common hits is below `CommonXFraction` = 0.7, the new candidate is stored as well, otherwise only the candidate with the higher track quality is kept, while the lower quality one is discarded. Here, track quality Q is defined as

$$Q = \texttt{QualityWeights[0]} \times n_{Hits} + \texttt{QualityWeights[1]} \times \chi^2$$

`QualityWeights` is `{ 1.0, -0.2 }` per default; these weights have been chosen after some experimentation because they give good results over a wide range of scenarios; using these weights in practice means that a track which misses a hit is penalised by the same amount as a track with a χ^2/NDF of 5.

6.4.2. Track search in stereo layers

For each track candidate in $x-z$ projection, the algorithm searches for hits in the stereo layers of the corresponding region. The algorithm uses a Hough transform to find the track in $y-z$ projection among all compatible stereo hits.

Hits are considered further if they satisfy the following requirements:

- $y_{min}\tan(\theta) - \texttt{TolCollect} < x_{hit} - x_{track} < y_{max}\tan(\theta) + \texttt{TolCollect}$
  In this expression, θ is the stereo angle, y_{min} and y_{max} describe the size of the sensitive area of the stereo layer, x_{hit} is the coordinate of the hit in the stereo layer at $y = 0$, and x_{track} is the x coordinate of the track candidate at the z of the stereo layer. Depending on the subdetector considered, you have to substitute `TolCollectIT` or `TolCollectOT` for `TolCollect` in the expression above.

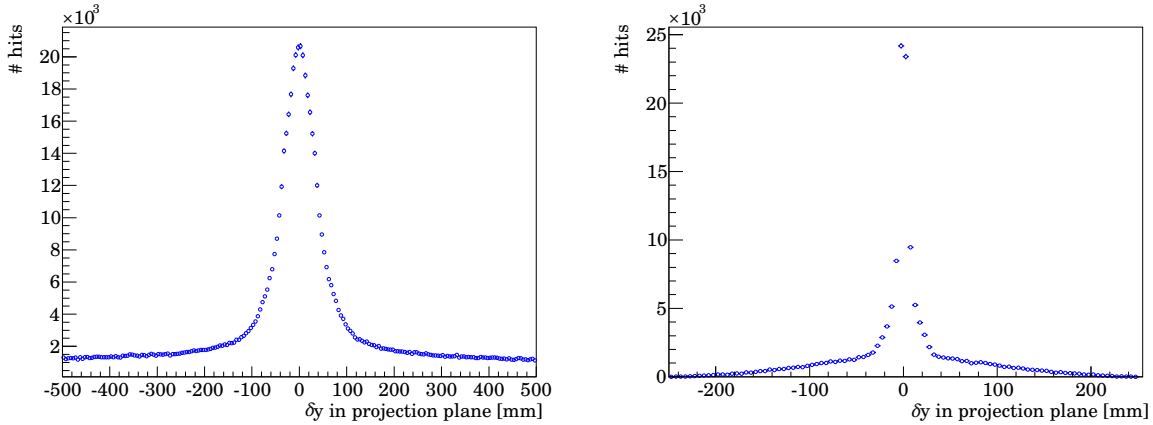
- If the candidate in $x-z$ projection is entirely in the Outer Tracker and has less than 7 hits, the algorithm assumes that the track goes through the detector at small $|y|$, where only one of the two monolayers is efficient. Therefore, the algorithm looks for stereo hits in both the upper and lower half of the Outer Tracker, and it restricts the sensitive region of both halves to the range $-5\text{cm} < y < 5\text{cm}$.

- For safety, the y coordinate calculated for the hit must not leave the sensitive area of the straw or silicon strip by more than `YTolSensArea` along y direction, the default value is 4 cm. This cut is not the equivalent to the one in the first item, because a layer may contain detection elements of different length. A prominent example is the Outer Tracker which has shorter modules in the middle to leave space for the Inner Tracker and the beam pipe.

As this is just a pre-selection of possible stereo hits, more work is required. The algorithm therefore projects the y coordinate of all hits to `zForYMatch`= 9 m (assuming a straight line in y from the origin through the hit). This plane forms the Hough space in which the algorithm searches for a cluster of hits. The spread of such a cluster must be below `MaxRangeIT` = 15 mm (or `MaxRangeOT` = 150 mm in case of the Outer Tracker). The distribution of hits in this Hough space can be seen in Figure 6.11. For illustrational purposes, the y coordinates of these hits in the projection plane have been shifted along y by the y coordinate of the found track in the projection plane, effectively moving hit clusters caused by tracks to $y = 0$, so that contributions from all tracks add up coherently. The width of the peak in the middle allows to determine suitable values for `MaxRangeIT` and `MaxRangeOT`. The wide and almost flat background is caused by hits which do not belong to the track.

A cluster of stereo hits is combined with the track candidate in $x-z$ to form a track, if

- the cluster of stereo hits together with the track candidate would form a track in which at least `MinTotalPlanes` layers have hits



(a) *OT* **(b)** *IT*

Figure 6.11.: *y projection of hits on truth-matched tracks found by PatSeeding. All hits have been projected to the projection plane at* `zForYMatch` *at 9 m; the y coordinates of these hits in the projection plane have been shifted along y by the y coordinate of the found track in the projection plane, effectively moving hit clusters caused by tracks to* $y = 0$*, so that contributions from all tracks add up coherently. Plots were done separately for Outer Tracker only tracks (a) and Inner Tracker only tracks (b). The wide and almost flat background is caused by hits which do not belong to a track. Cuts are placed at* $\pm 150\,\text{mm}$ *in the Outer Tracker and* $\pm 15\,\text{mm}$ *for the Inner Tracker.*

- if several such clusters exist, the algorithm considers only those in which the number of stereo layers with hits is maximal

- if there is still more than one such cluster, the one with the smallest spread in the projection plane is selected

The algorithm continues to add stereo hits at both sides of the cluster which are less than $\frac{\texttt{TolCollectIT}}{|\tan(\theta)|}$ (or $\frac{\texttt{TolCollectOT}}{|\tan(\theta)|}$, respectively) away from the cluster in the projection plane to catch the tails of the distributions in Figure 6.11. This step is skipped if the cluster already contains hits from six stereo layers or if it overlaps with the interval $[-5\text{cm}, 5\text{cm}]$ in the projection plane. The reason for not adding stereo hits if the cluster of hits overlaps with the range given above is that the track is likely to have passed through the inefficient area of IT or OT near $y = 0$ which was mentioned earlier. A track passing through that area is expected to have fewer hits, so the algorithm does not forcibly try to add more hits.

The resulting combination of hits must satisfy the following conditions to make it to the algorithm's output:

- After the initial fit in $y - z$ projection, a corrected y at $z = 0$ is calculated; this correction is due to the magnetic field having a tiny effect on the slope in $y - z$ projection

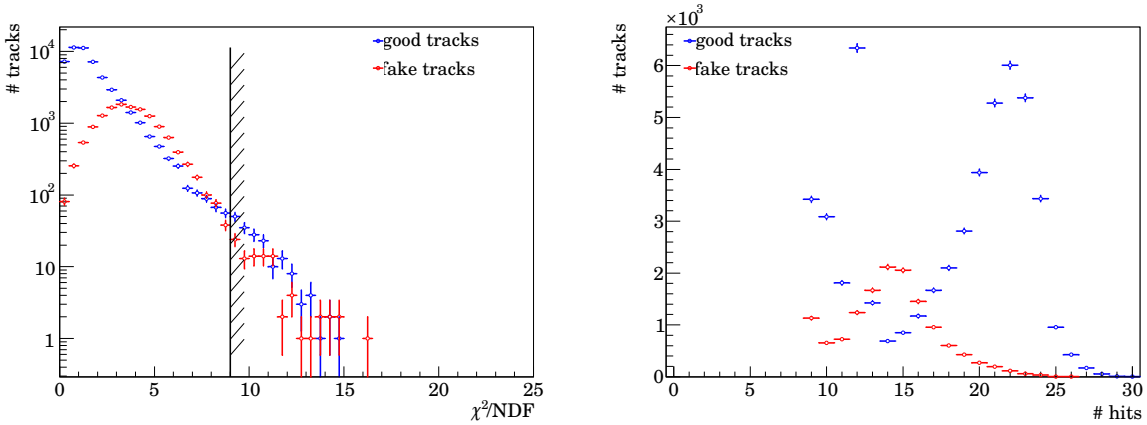


Figure 6.12.: *Distribution of track χ^2/NDF (a) and number of hits per track (b) for the per-region track search just before the cut on χ^2/NDF for truth-matched ("good") and fake tracks.*

of the track. Specifically, the algorithm demands $|y_{corr}(0)| <$ `maxYAtOrigin` where

$$y_{corr}(0) = a_y \pm \frac{b_y^2 c^2}{\texttt{yCorrection}}$$

The sign is positive for $b_y < 0$, negative otherwise. Outer Tracker tracks failing this criterion (`maxYAtOrigin` $\leq |y_{corr}(0)| \leq$ `maxYAtOriginLowQual`) are saved for further potential use during the last stage of the algorithm. The numerical value of `yCorrection` is determined from studies with simulated events.

- for Outer Tracker tracks, demand at least 6 stereo hits in total (unless $|y| < 5$cm in the projection plane from above)
- for Outer Tracker tracks with fewer than `OTNHitsLowThresh` hits, demand that for each hit marked as part of a potential cluster, the other hit in the potential cluster is also part of the track
- a refit of the track is made; the outlier cut is set to `MaxFinalChi2`, and `MinTotalPlanes` planes are required
- if the fit converged, the criterion for Outer Tracker tracks with less than `OTNHitsLowThresh` hits from above still holds and the track χ^2 is below `MaxFinalTrackChi2`, the track is stored for a final selection pass

To give a feeling what comes out of this step of the pattern recognition, Figure 6.12 shows the distribution of track χ^2/NDF just before the cut on χ^2/NDF for truth-matched ("good") and fake tracks.

The resulting tracks are passed on to a final track competition stage (cf. Section 6.6).

6.5. Additional track search stages

Having treated the "easy" track candidates which stay within their region, have enough hits and point back to the origin in $y-z$ projection in the last section, this section serves to introduce track search strategies which deal with tracks in the main tracker which are not caught by the per-region track search.

6.5.1. Track search for tracks migrating from Inner to Outer Tracker

In this stage, tracks overlapping regions are constructed from the remaining unused hits. The strategy starts with hits in the Inner Tracker. The idea is to form very short tracks which will also be called "stubs" or space-points with direction from hits in all four layers of a station. These stubs can then be extrapolated into other stations where additional hits can be picked up.

In each station of each IT region, we search for four hits satisfying the following constraints:

- two x hits are in different x layers and satisfy the centre of magnet constraint
- a stereo hit in the first stereo layer that is compatible with the x part of the track found (i.e. within $\pm$`TolCollectITOT` of the x position predicted from the straight line between the two x hits)
- a second stereo hit in the second stereo layer compatible with the previous three hits (the x coordinate is predicted using the straight line segment in x and line joining the first stereo hit with the origin; a hit must be within ± 1.5`TolCollectITOT` to be accepted
- For safety, the y coordinate calculated for the stereo hits must not leave the sensitive area of the silicon strips in question by more than `YTolSensArea` along y direction.

From these four hits, parameters for the full track model are estimated, assuming that it came from the primary vertex. In $x-z$ projection, one has thus two points, so position and slope in $x-z$ projection can be calculated. A curvature estimate is obtained by using equation 6.3. In $y-z$ projection, an average y is calculated, and the slope is estimated under the primary vertex assumption. This first determination is a rough one, so track parameters are not fitted. Such a short track segment is also called a *stub*.

This produces a stub to which neighbouring hits in the same station and region are added, provided that they are within `TolCollectITOT` of the hits already found.

Next, χ^2/NDF for the four hit stub is calculated. Since we have four hits or more and only three free parameters (x and y position, slope in $x-z$ projection), NDF is not zero, so there is some redundancy. If the calculated value is larger than `ITStubLooseChi2`, the stub is discarded; the default value is at 1500 because stub track parameters have only been determined in the rough. The aim of the cut is that implausible stubs are killed early in the process before more time is spent to fit them.

For the surviving stubs, a simultaneous fit to track parameters a, b and b_y is done with parameters as defined in Equations 6.1 and 6.2; a_y is kept fixed to zero (i.e. the

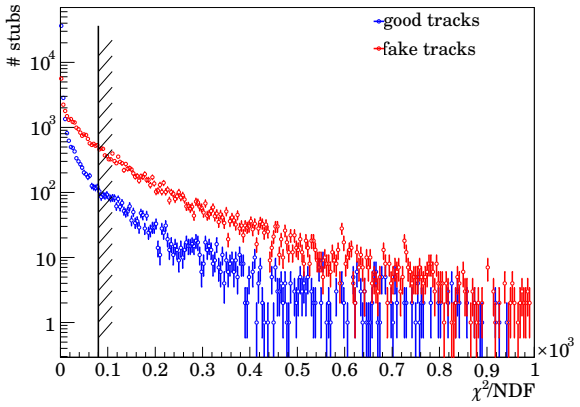


Figure 6.13: *Stub χ^2/NDF for fake and truth-matched stubs after the simultaneous fit to determine stub track parameters. Note the logarithmic scale.*

track comes from the origin in $y-z$ projection), while c is again estimated from the stub x intercept at $z=0$ using the method described with equation 6.3. The χ^2/NDF of the stub is recalculated, and stubs with χ^2/NDF below `ITStubTightChi2` are retained for further use; the default value for the cut is at a χ^2/NDF of 80. Figure 6.13 shows the distribution of χ^2/NDF for truth-associated and fake stubs. For fake stubs, there is a long tail going out to high values of χ^2/NDF. For truth-associated stubs, a much less pronounced tail exists. It is due to the fact that a particle may form a stub in more than a single main tracker station. If such a particle enters or leaves a station without leaving hits in all four silicon layers, the algorithm forcibly takes a nearby hit to form a stub; therefore, these stubs typically have a bad χ^2/NDF. Since the aim is to search for additional hits in other stations based on the prediction from a stub, it is OK to discard these "bad" stubs with the cut on χ^2/NDF because there is a high probability that there is a good stub for the same particle in another station of the main tracker.

Unused Outer Tracker hits are added to a stub to form a track candidate, if

- they are in a different station than the four Inner Tracker hits
- they are in the upper half of the Outer tracker for stereo hits in the Inner Tracker with $y>0$ and vice-versa
- they are within `TolExtrapolate` of the position predicted from the estimated track parameters
- for safety, the y coordinate calculated for stereo hits must not leave the sensitive area of the straws or silicon strips in question by more than `YTolSensArea` along y direction.

For every station added to the track with hits in three or more layers, there is a refit. An outlier cut is placed at `MaxChi2HitOT`. A track is abandoned if it lacks hits in more than one layer among the layers in stations contributing hits to the track.

If the resulting candidate still does not have hits in all three stations, the same procedure is applied to the Inner Tracker, looking for hits there.

If the fit converges, the track candidate has hits in all three stations and the track χ^2/NDF is below `MaxTrackChi2LowMult`, the track is put into the output container (after

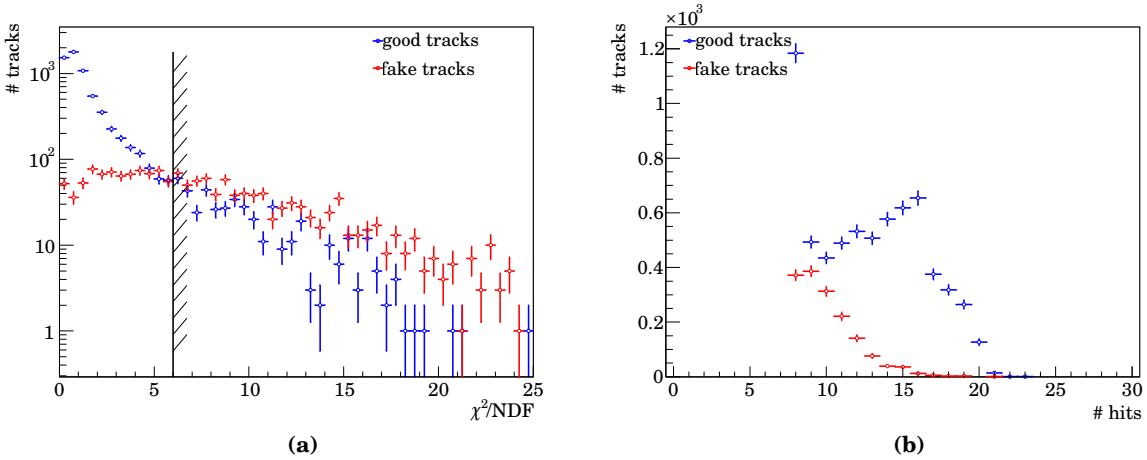


Figure 6.14.: *Distribution of track χ^2/NDF (a) and the number of hits per track (b) for the search of tracks migrating between regions just before the cut on χ^2/NDF for truth-matched ("good") and fake tracks.*

letting the resulting tracks compete as described in Section 6.6). Figure 6.14 shows the distribution of track χ^2/NDF for the search of tracks migrating between regions just before the cut on χ^2/NDF for truth-matched ("good") and fake tracks.

For tracks passing entirely inside the Inner Tracker, tracks which pass the insensitive Inner Tracker central $|y|$ range or tracks touching the hole for the beam pipe in one of the three stations, the track is also accepted if it has hits in only two of the three stations. This is done in order to avoid losing "difficult" tracks tracks which either pass through an insensitive area of the detector or which enter or leave Inner Tracker early but do not continue in the Outer Tracker.

This stage is actually repeated a second time, after hits found by the first first iteration have been tagged used. For the second iteration, all Inner and Outer Tracker regions are considered once a track segment in the Inner Tracker has been found. This enables the algorithm to also find tracks that migrate between different regions in the Inner Tracker itself.

6.5.2. Track search in the Outer Tracker at high $|y|$

As explained in 6.3.1, tracks at high $|y|$ in the Outer Tracker are much less curved than in the rest of the detector. Therefore, a second iteration of the track search per region is performed on the unused hits in the Outer Tracker. For this iteration, `InitialArrow` is temporarily set to zero to make the algorithm expect zero curvature for all tracks found during this stage, and `MaxUsedFractPerRegion` is temporarily decreased from its default value of 0.30 to 0.05, so that only tracks are found that share very few hits among each other. As an example, for a track with 20 hits, up to one hit may be shared with other tracks for the new setting.

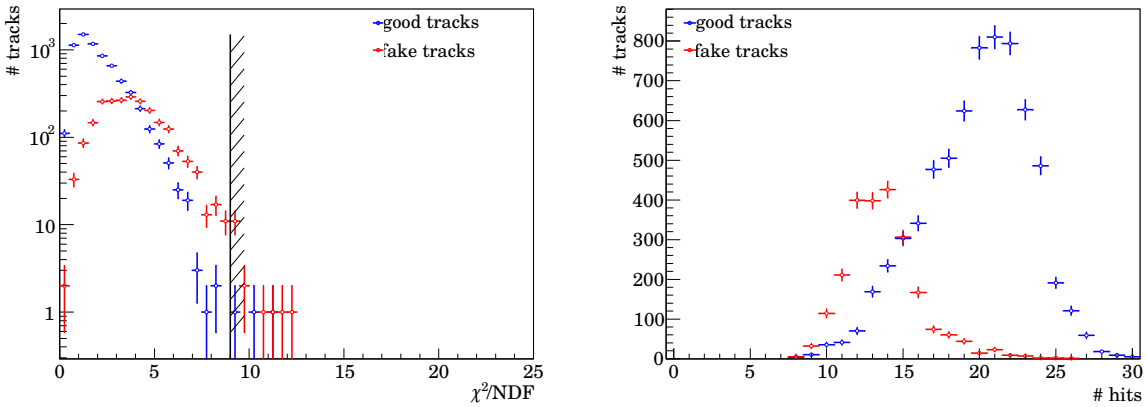


Figure 6.15.: *Distribution of track χ^2/NDF (a) and number of hits per track (b) for the search among low quality candidates just before the cut on χ^2/NDF for truth-matched ("good") and fake tracks.*

6.5.3. Track search among low quality candidates

This step builds upon work done during the per region track search by using low quality Outer Tracker tracks if they have enough unused hits. Essentially, these tracks are processed in the same way as those that were not flagged low quality tracks during the per region track search, but there are a few modifications. For brevity, only the modifications are listed below:

- The fraction of hits on a track used by previous stages must be below `MaxUsedFractLowQual`. Up to 3 used hits are removed before we start counting. The algorithm removes the used hits it encounters first (hits on track candidates are kept sorted by increasing z, so hits in the most densely populated station T1 are dropped first). This removing procedure is done to avoid losing tracks which have a used hit in very few layers.

- After removing hits, the initial straight line fit in $y-z$ projection is redone, and we require
  $|y(z=0)| <$ `MaxYAtOriginLowQual`.

- When the refit is performed, demand at least `MinTotalPlanes - 1` layers with hits, with the outlier cut set to `MaxFinalChi2` and the maximum track χ^2 smaller than `MaxFinalTrackChi2`.

Figure 6.15 shows the distribution of track χ^2/NDF for the search among low quality candidates just before the cut on χ^2/NDF for truth-matched ("good") and fake tracks.

Again, the resulting tracks are passed on to a final track competition stage (cf. Section 6.6).

6.6. Final track competition

After having passed the quality cuts of the different stages of the reconstruction described above, tracks are sorted by decreasing number of hits; if two tracks have the same number of hits, the track with the smaller χ^2/NDF comes first. Starting with the tracks with most hits (and lowest χ^2/NDF), tracks are examined in turn to see if they satisfy the following cuts:

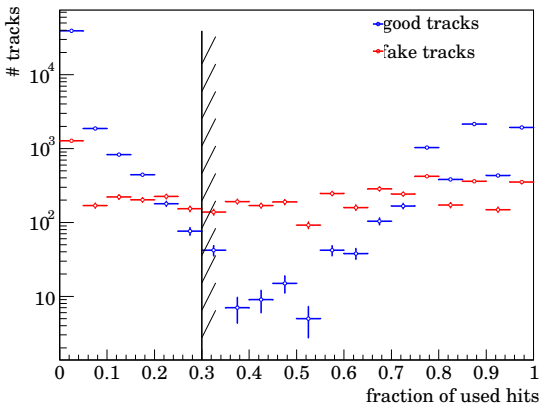
- The fraction of hits on the track which have been tagged as used must not be larger than the fraction specified for the different stages. Specifically, the job options are `MaxUsedFractPerRegion` for the per-region search (default value is 0.3), `MaxUsedFractITOT` for tracks migrating between regions (default value is 0.15) or `MaxUsedFractLowQual` for the search among low quality candidates (with a default of 0.05). Figure 6.16 shows the fraction of used hits per tracks for truth-matched and fake tracks.

- For tracks with Outer Tracker hits the "monolayer asymmetry" $|n_1 - n_2|$ is required not to be larger than four; here, n_1 counts the number of hits on the track in all first monolayers of straws in OT modules, n_2 counts the same number for hits in the second monolayers of contributing modules. If the track does not pass this cut, but satisfies a cut on the fractional monolayer asymmetry $\frac{|n_1-n_2|}{n_{OT}} < 0.25$, it it kept; n_{OT} is the total number of OT hits on the track. Figure 6.17 shows the corresponding distributions before any cuts have been applied.

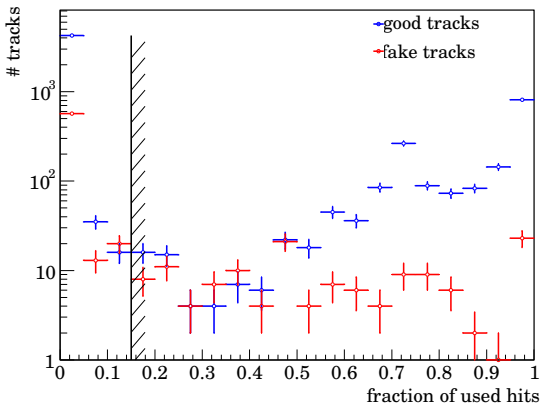
- The number minimum number of layers per (non-empty) stations has to be at least `MinPlanesPerStation` (the default value is 1, so in the default configuration, this cut is not applied). If tightened, the main aim of this cut is to ensure the quality of short tracks which are produced during the search for tracks migrating between regions. Figure 6.18 shows the corresponding distributions.

- The number of "holes" in a track has to be at most `MaxHoles`; the default value is 2. Here, a "hole" in the track is a consecutive set of detector layers without hits, surrounded by layers which have hits. Figure 6.18 shows the corresponding distributions.

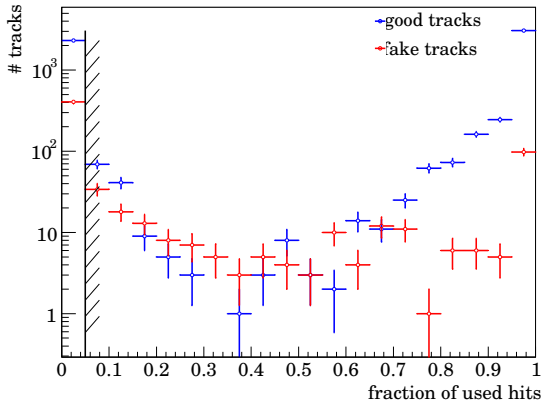
Tracks which do not pass these cuts are discarded. All other tracks have their hits tagged as used, and a complete state estimate is produced (with the momentum part calculated by the FastMomentumEstimate tool as described in Section 4.2.7) before the track is added to the output container. This process continues until all tracks sorted above have either been discarded or added to the output container. The χ^2/NDF distribution for these tracks is shown in Figure 6.19; the shape of the distribution is as expected: truth matched tracks peak around $\chi^2/NDF \approx 1$, while the distribution for fake tracks is considerably broader and shifted to the right.



(a) *per region search*



(b) *search for tracks migrating between regions*



(c) *search for low quality tracks*

Figure 6.16.: *Fraction of used hits on track in final track competition stage for truth matched ("good") and fake tracks. Separate plots have been produced for the three track search strategies. Generally, fake tracks have a higher fraction of used hits. However, for high fractions of used hits, the contribution from good tracks rises again. This is due to the PatSeeding finding the same track more than once, discarding the clones just before they end up in the output.*

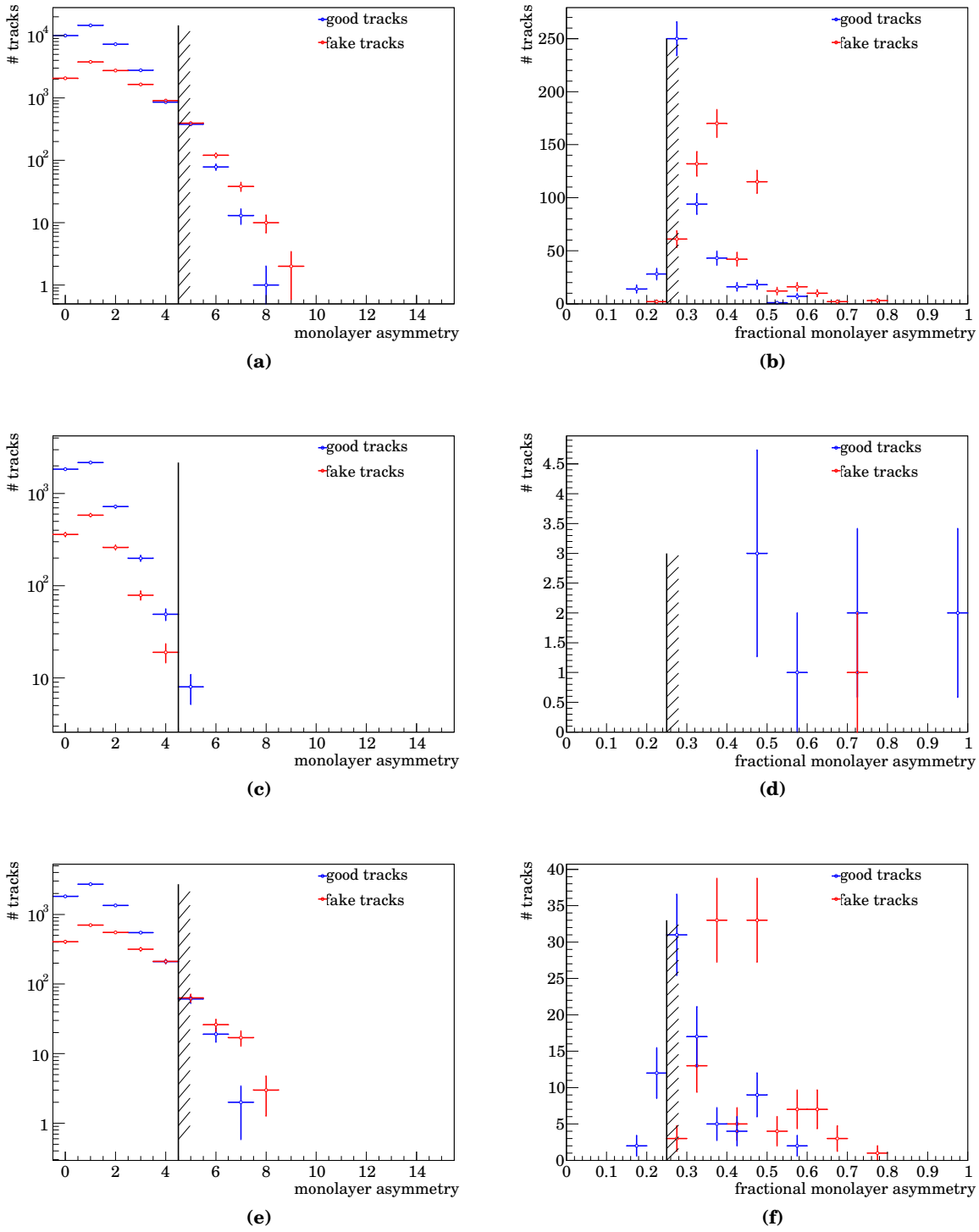


Figure 6.17.: *Monolayer asymmetry for tracks with hits in the Outer tracker. Top row is for per region track search, middle row for tracks migrating between regions, bottom row for low quality tracks. The left column shows the monolayer asymmetry itself, tracks left of the cut are accepted. For tracks which do pass the cut, the fractional monolayer asymmetry (as defined in the text) is plotted, again, tracks to the left of the cut are accepted.*

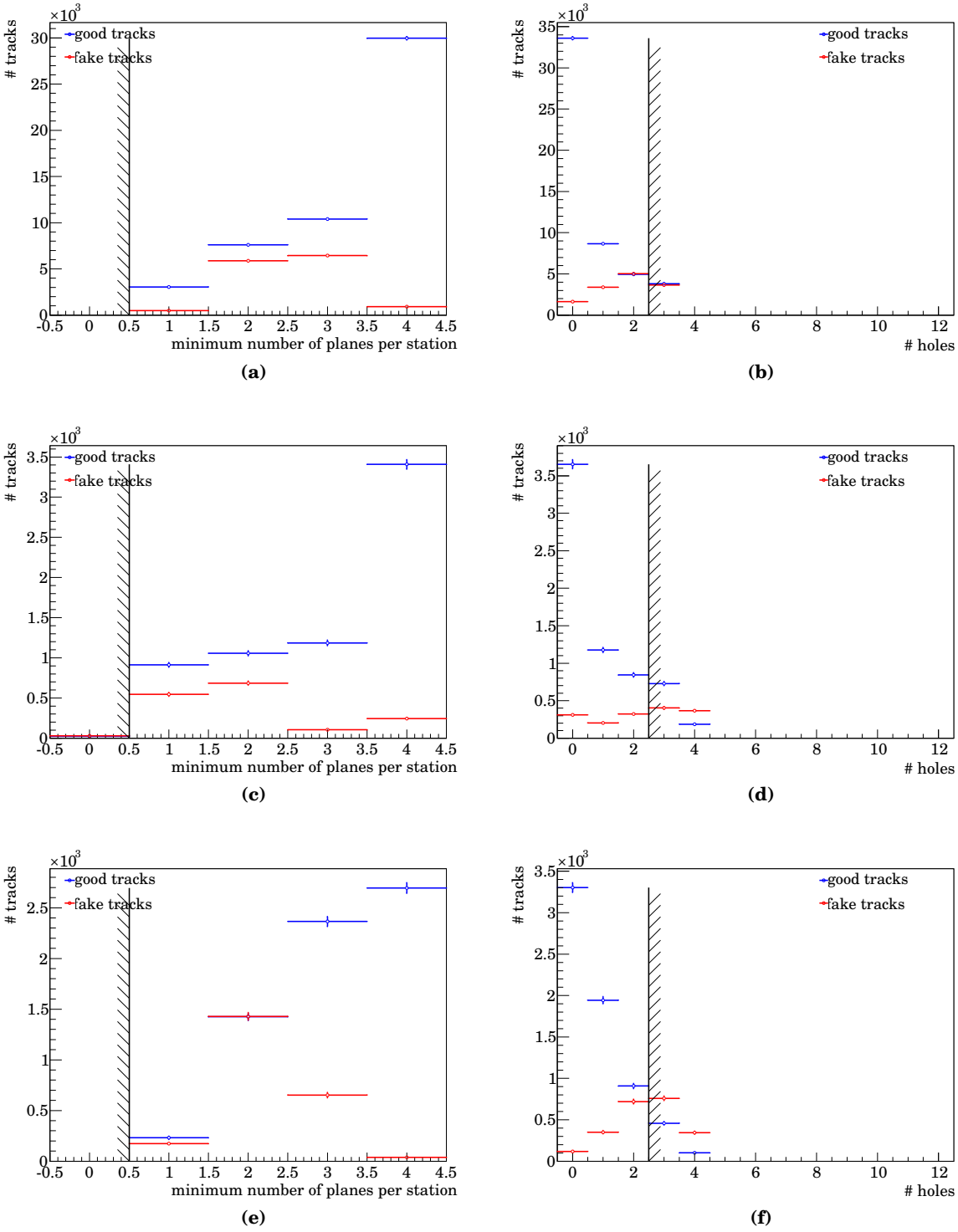


Figure 6.18.: *Distribution of minimum number of layers per station and track and number of holes per track. Top row is for per region track search, middle row for tracks migrating between regions, bottom row for low quality tracks. The plots on the left show the minimum number of layers per station and track, the plots on the right depict the number of holes per track.*

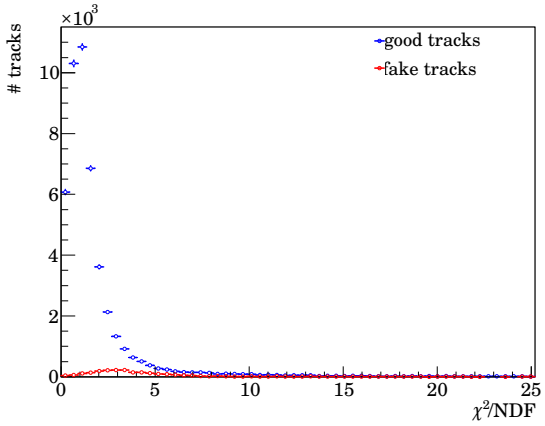


Figure 6.19: *Track χ^2/NDF distribution for truth-matched ("good") and fake tracks at the end of PatSeeding.*

7. PatSeeding: Special applications, special tunings

One of the distinguishing features of PatSeeding is that is is flexible enough to be useful outside the standard offline reconstruction sequence:

- PatSeeding has been used from the beginning of data taking in the software trigger to confirm hardware trigger candidates.
- During the commissioning phase of LHCb, it was the only algorithm to reliably reconstruct cosmic muons in the Outer Tracker, a feature which is still interesting to recommission the detector after a shutdown and for alignment purposes.

This chapter describes the use of PatSeeding in these and some other special applications, and it also discusses special tunings which have been made available for it. A few less important studies regarding special tunings or applications are also presented in Appendix C.

Unless specifically noted otherwise, any figures regarding performance (i.e. reconstruction efficiency, fake track fraction) have been measured on the same samples as in Section 5. For figures regarding algorithm execution time, care was taken to run the algorithm on the same computer as was used in Section 5 to obtain comparable results. As an exception to this rule, timing measurements regarding the application of PatSeeding in the trigger have been done on a cluster of contemporary computers, however, the effective speed of the machines may vary by a factor of 0.8 to 1.2 with respect to the benchmark machine used in the rest of this thesis (for the software trigger computing farm, the situation is similar).

7.1. Modification for application in the trigger

PatSeeding can be used in the first stage of the software trigger to quickly confirm or reject the trigger decision made by the hardware trigger, also called Level 0 or L0 trigger. The idea is to take a L0 candidate in the muon stations or the hadronic or electromagnetic calorimeters (i.e. the collection of muon hits or calorimeter deposits which caused L0 to trigger) and check the main tracker for the presence of a track. If none is present, the event must be discarded quickly to enable the CPU in question to process the next L0-triggered event. If there is a track, the L0 decision was probably correct, and the event deserves closer inspection within the software trigger to see if it contains interesting physics. Typically, the next step after a L0 candidate has been confirmed is to reconstruct tracks in the vertex detector to obtain vertices. With this information, one can either look for displaced tracks and keep relatively low thresholds for p and p_T of the confirmed track, or one can use the information in the vertex detector

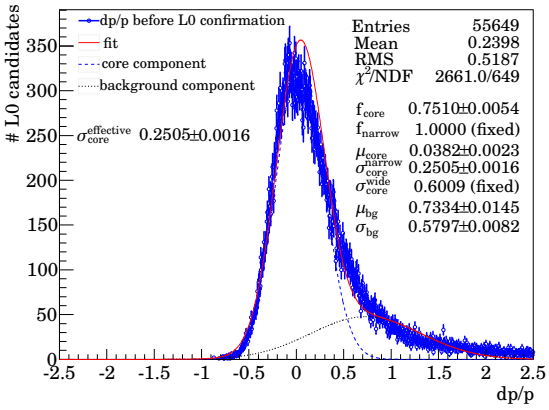


Figure 7.1: *Relative momentum resolution* $dp/p = p_{\text{reconstructed}}/p_{\text{true}} - 1$ *of (unconfirmed) L0 muon candidates. With about 25%, the momentum resolution is relatively poor, and L0 (momentum) thresholds and true muon momentum only have a relatively lose correspondence. The fit parameters are explained in Appendix A; effectively, the fit uses a single Gaussian to describe the core* ($f_{\text{narrow}} = 1$).

to improve upon the momentum resolution of the confirmed L0 candidate and raise the p and/or p_T thresholds to get a sample without (implicit) cuts on lifetime at the software trigger level. The trigger then proceeds to select specific decay modes or certain decay topologies (e.g. displaced three-prong vertices with high reconstructed invariant mass).

For reasons of clarity, the focus in this section will be on explaining the confirmation of L0 muon candidates, but confirmation of calorimeter candidates works in much the same way.

The trigger hardware obtains its (muon) candidates by looking for hits in the muon stations behind the calorimeters where the occupancy is low. These hits must be compatible with a straight line track under the assumption the track comes from the origin. The muon stations have been built in a way that exploits this "projectivity" to make the job of the L0 trigger hardware easier. Combining this track with a compatible hit in the (much more finely segmented) muon station in front of the calorimeter, it is possible to obtain a rough estimate of the muon momentum by using the track position and direction and relating it to the kink such a track would get in the bending plane of the magnet (assuming it comes from the origin). The momentum resolution is quite poor as can be seen in Figure 7.1, at around 25% (the fit is far from perfect, but for such a poor momentum resolution, it does not matter that much). Since the position of the muon candidate is known, this is straightforwardly converted into a measurement of transverse momentum p_T. Due to the bad resolution, L0 muon p_T thresholds must not be set too tight because a lot of muons which have their (true) transverse momentum over threshold would be lost otherwise.

Given the 1.1 MHz rate of L0-accept trigger decisions and a software trigger farm with about 2,000 cores, it is clear that the average software trigger decision must not take more than about 2 ms on average, otherwise dead time is introduced in the system. This means that L0 candidates need to be rejected fast (i.e. faster than 2 ms), while interesting events (which are a lot rarer) may take longer.

Since the pattern recognition of the full detector takes too long, only a part of the main tracker is reconstructed, the part which must contain the track corresponding to the L0 candidate if such a track exists. This region of interest around the L0 candidate is parametrised according to the detector in question (muon stations, electromagnetic or

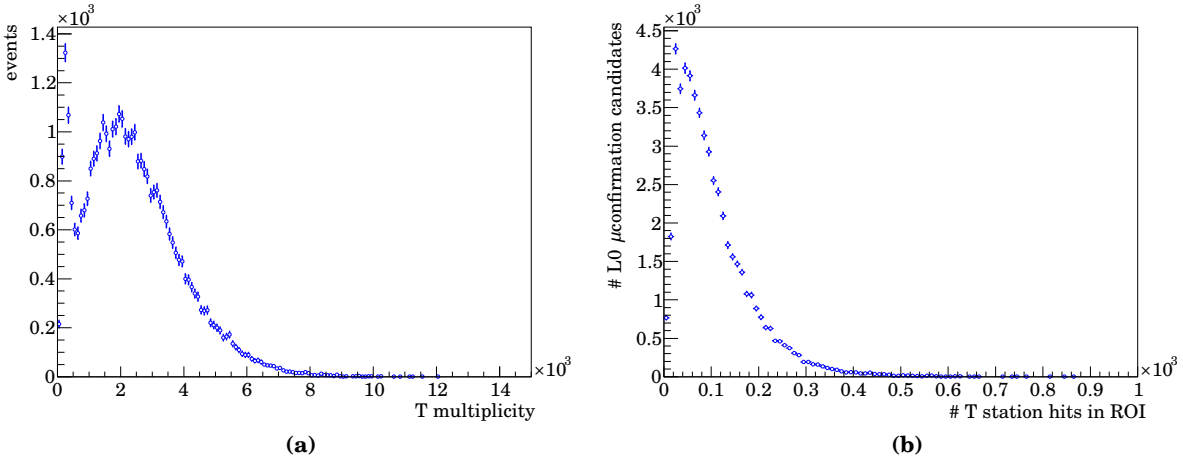


Figure 7.2.: *Number of main tracker hits, (a) for simulated L0 muon triggered events from a minimum bias sample, (b) number of main tracker hits in the region of interest (ROI) around the L0 muon candidates which cause L0 to trigger in that sample. The mean of the distribution in (a) is about 2400, while mean for the distribution in (b) is only about 100, a gain of a factor of about 24. The peak at low main tracker multiplicities in (a) is caused by events with very little activity in the detector, a common occurrence for minimum bias events.*

hadronic calorimeter) and its granularity in the region where the L0 candidate triggered (for example, the calorimeters go from coarse to fine granularity when moving from the edges of the calorimeter towards the beam pipe). Details on how these parametrisations are derived can be found in [58], [59] and [60].

In fact, life is a little more complicated: For calorimeter L0 candidates, only a position estimate and the approximate momentum are known, so one has to take into account two regions of interest for every L0 candidate because the charge of a track potentially confirming the L0 candidate is not known. Muon L0 candidates have both position and direction estimates in the muon stations, so charge and momentum can be inferred under the assumption that the track came from the origin. While it is important to keep this in mind, this complication will not be mentioned in the rest of explanation for the sake of simplicity.

With such a region of interest around the L0 candidate defined, it becomes possible to decode only the “interesting” parts of the main tracker. Technically, this is realised by only decoding the data sent from readout front end chips which are responsible for channels which are (at least in part) inside the region of interest (one cannot decode hits with a finer granularity due to the way raw data is taken by LHCb). Figure 7.2 shows the distribution of the number of all main tracker hits and the number of main tracker hits in a region of interest around a L0 muon candidate on a minimum bias sample[1]; the average number of hits in the region of interest is reduced by a factor of about 20 with respect to all hits in the main tracker.

[1] See Section 7.1.1 for details on the sample used

Since the front end chips in both IT and OT are responsible for a rather large region (e.g. 33 cm × 250 cm for a single front end chip in the OT), PatSeeding is not applied to all decoded hits, but only to those which are actually inside the region of interest. Reconstruction stages which are not needed for a given region of interest are skipped; for example, if the region of interest is entirely in the upper half of the Outer Tracker, the lower half of the Outer Tracker and the entire Inner Tracker reconstruction steps can be skipped. Moreover, PatSeeding will count the number of hits in the region of interest before attempting to form track candidates. If the number of hits is too low, the algorithm can stop early. These last two optimisations are important to be fast for L0 candidates which cannot be confirmed (i.e. "wrong" L0 decisions which caused a trigger but cannot be confirmed with a track), and also for the region of interest with the wrong charge assumption in case there is a track to confirm the L0 candidate.

The technical implementation of how such a region of interest is treated in the pattern recognition is postponed to subsection 7.1.2; the performance of L0 confirmation is investigated first.

7.1.1. Performance of L0 confirmation

To judge the performance of the L0 muon confirmation, two samples of simulated events are studied[2]: First, a minimum bias sample[3] containing about 2.75 million events is used to study timing behaviour and L0 muon confirmation retention (i.e. how many L0 triggered events the L0 muon confirmations retains for further processing). Second, an $B_s \longrightarrow J/\psi(\mu\mu)\phi$ sample[4] with about 545 thousand events is used to measure the efficiency of the L0 muon configuration with respect to offline reconstructed events. Since the $b\bar{b}$ production cross section is quite small compared to the total pp interaction cross section, this cannot be done on the minimum bias sample due to lack of statistics. Both hardware and software trigger emulation were applied to these simulated events; trigger settings have been chosen to correspond to the end of the 2010 data taking period[5].

To make a start with something, Figure 7.3 shows the momentum resolution of confirmed L0 muon candidates. The track in the main tracker improves dp/p from about 25% to about 2.1%. This improvement of a factor of roughly 10 allows the software to cut much more selectively on p or p_T than is possible in hardware.

The next important point is the time that is needed for decoding main tracker hits on demand and trying to reconstruct a track in the decoded hits. Table 7.1 lists average per-L0 candidate times for both decoding and pattern recognition using PatSeeding along with the total time spent. In fact, these figures are listed for both the minimum bias sample and the sample of signal events. The numbers are quite similar, although it seems that the decoding during L0 confirmation takes a little longer in the minimum bias sample. This is due to the fact that the signal has 3.93 L0 muon candidates per event with L0 trigger while there are only 1.53 L0 muon candidates in minimum bias events, giving the decoding on demand the possibility to profit from being able to reuse

[2] Software versions used: BRUNEL v37r8, DAVINCI v26r2p1.

[3] `2010-Beam3500GeV-VeloClosed-MagDown-Nu1-2010-Sim03Reco03-withTruth-3000000-minbias`

[4] `2010-Beam3500GeV-VeloClosed-MagDown-Nu1-2010-Sim03Reco03-withTruth-1314406--Bs_Jpsiphi,mm=LargeCPV,DecProdCut`

[5] The trigger configuration key (TCK) was `0x002a002c`.

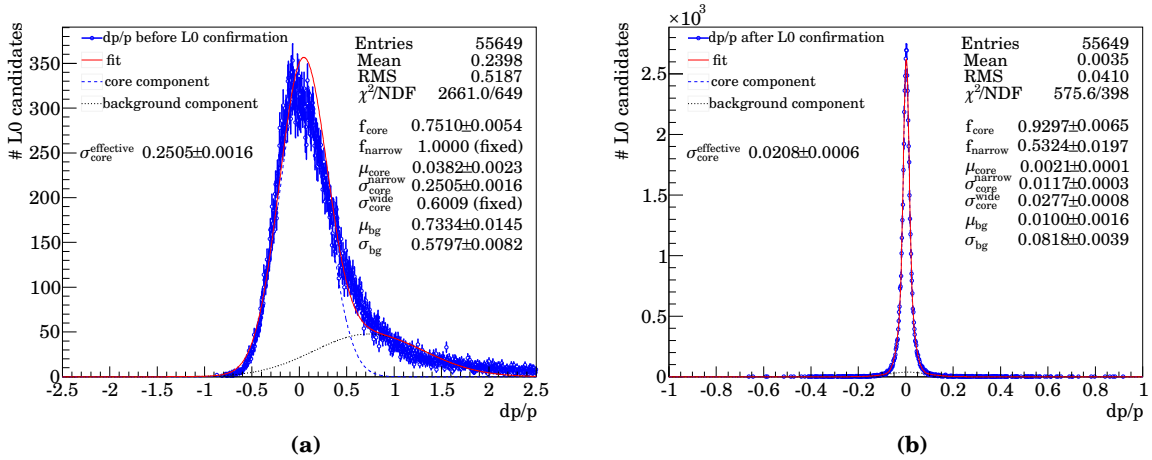


Figure 7.3.: *Relative momentum resolution of unconfirmed (a) and confirmed (b) L0 muon candidates. dp/p improves from about 25% to 2.1% when confirming the candidate, because of the much better spatial resolution of the main tracker. The fit parameters are explained in Appendix A; effectively, the fit in (a) uses a single Gaussian to describe the core ($f_{\text{narrow}} = 1$).*

	minimum bias	signal
total time [ms]	0.29	0.22
decoding time [ms]	0.17	0.08
pattern recognition time [ms]	0.12	0.13

Table 7.1.: *Per-L0 candidate average times for decoding hits on demand and pattern recognition using PatSeeding, together with the total time spent in confirming L0 muon candidates. These figures are given for minimum bias events (almost everything that enters the trigger) and signal events.*

the per-event initialisation of certain data structures in the signal sample. Moreover, there is a good chance that more than one triggering track in the signal sample is in one of the main tracker modules near the beam pipe where the track density is highest, so the decoding on demand may find that the module needed has already been requested by a previous track in the event. The total time per event spent in decoding main tracker hits on demand in minimum bias events is smaller than in signal events as one would expect (because minimum bias events are less busy).

It can also be seen that the time spent in PatSeeding to confirm a single L0 candidate is faster by a factor on the order of 75 compared to reconstructing the full event at $\nu = 1$, that is about a factor three more than one would expect just from the savings due to the reduction of hits when going from the full detector to hits in the region of interest. This additional factor of three is gained inside PatSeeding itself by exploiting the fact that the L0 candidate provides a rough prediction of the track which can be used to skip some stages of the pattern recognition (e.g. omit the Inner Tracker if the lower half of the

Outer Tracker if the L0 candidate must have passed through the upper half of the OT).

For completeness, Figure 7.4 shows the distributions for the time spent in decoding hits, tracking (i.e. PatSeeding and the total time per-L0 candidate).

Another important quantity is the minimum bias retention of the trigger; the desire is to suppress minimum bias as much as possible to have bandwidth to spare for interesting physics. On the minimum bias sample investigated, the combination of L0 muon trigger in hardware and L0 muon confirmation in the software trigger retained about 0.08% of all minimum bias events in the sample, so only one in about 1250 minimum bias events accidentally passes the L0 muon confirmation. Of course, this figure alone does not allow to predict the total minimum bias retention since there are other trigger lines in the L0 trigger, and the muon confirmation is but the first step of the software trigger. Nevertheless, the figure shows that the L0 muon confirmation is a highly selective process.

Now that the efficiency of L0 muon confirmation in rejecting minimum bias events has been established, it is interesting to ask what it does to signal events. To judge the efficiency of the L0 confirmation process, one can measure (track) reconstruction efficiency with respect to true muons which have been reconstructed by the offline reconstruction in events which have passed the LHCb standard selection for the signal channel $B_s \longrightarrow J/\psi\phi$; the selection used can be found in Table 7.2. The reasoning behind the measurement with respect to "offline" is that it does not make sense to optimise the trigger for events in which the signal tracks cannot be reconstructed or are discarded by the event selection for the signal channel in the subsequent offline processing and analysis of the data. Figure 7.5 shows the reconstruction efficiency for muon tracks during L0 confirmation for offline reconstructed muons from selected signal decays provided these muon tracks pass the software trigger cuts. The majority (97.5%) of events with a confirmed muon pass the single muon trigger line with a p_T cut at 5 GeV which explains the rise in the efficiency plot as function of p_T. Some of the muons with smaller p_T are recovered by one of the Dimuon lines which have lower effective p_T cuts because they cut on the invariant mass of the dimuon vertex. The shapes of the efficiencies as function of p and η essentially follow from the p_T shape and detector acceptance.

The reconstruction efficiency reaches 90% around a p_T of 5 GeV, and plateaus above 95% efficiency at slightly higher p_T.

This should make clear that PatSeeding is not only an excellent choice to reconstruct the entire main tracker in an event, it is also flexible and fast enough to be useful in the time-critical first stage of the software trigger in which the decision of the hardware trigger must be confirmed.

7.1.2. Technical implementation

As explained earlier, the standard PatSeeding settings are not fast enough to be run in trigger applications to confirm L0 candidates. Therefore, one uses position information from the L0 candidate to restrict the algorithm to look at a small region of interest, see below. It is also possible to ignore hits which have been tagged used by previous runs of PatSeeding (if there are several L0 candidates, for example) or other algorithms, see 6.2 for details.

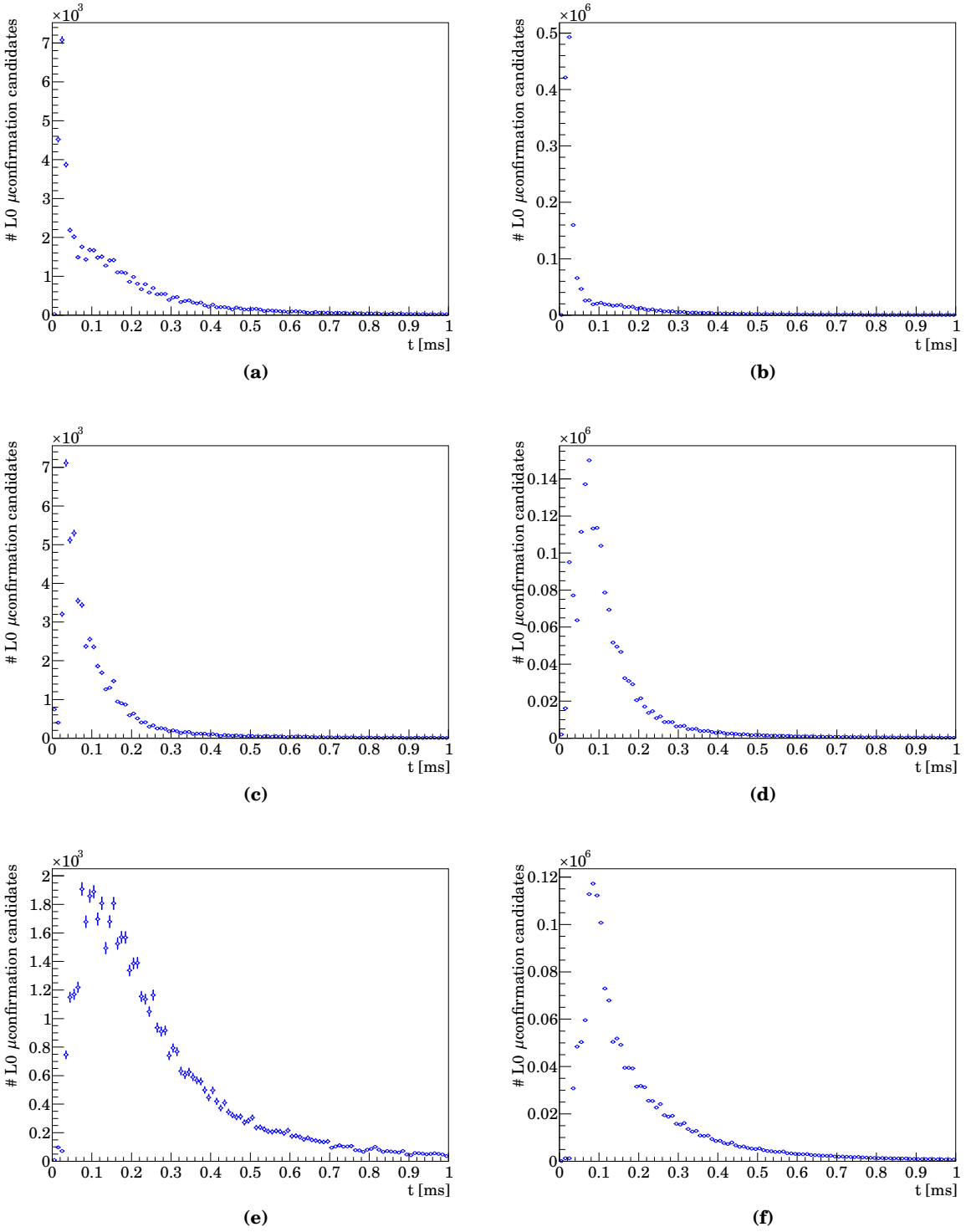


Figure 7.4.: *Distributions for the time spent in L0 confirmation in decoding hits, tracking (i.e. PatSeeding) and the total time per-L0 candidate. The plots on the left are for minimum bias events, while those on the right are for the signal sample. The top row is the decoding time, tracking time is in the middle while the total time per L0 muon candidate is given in the bottom row.*

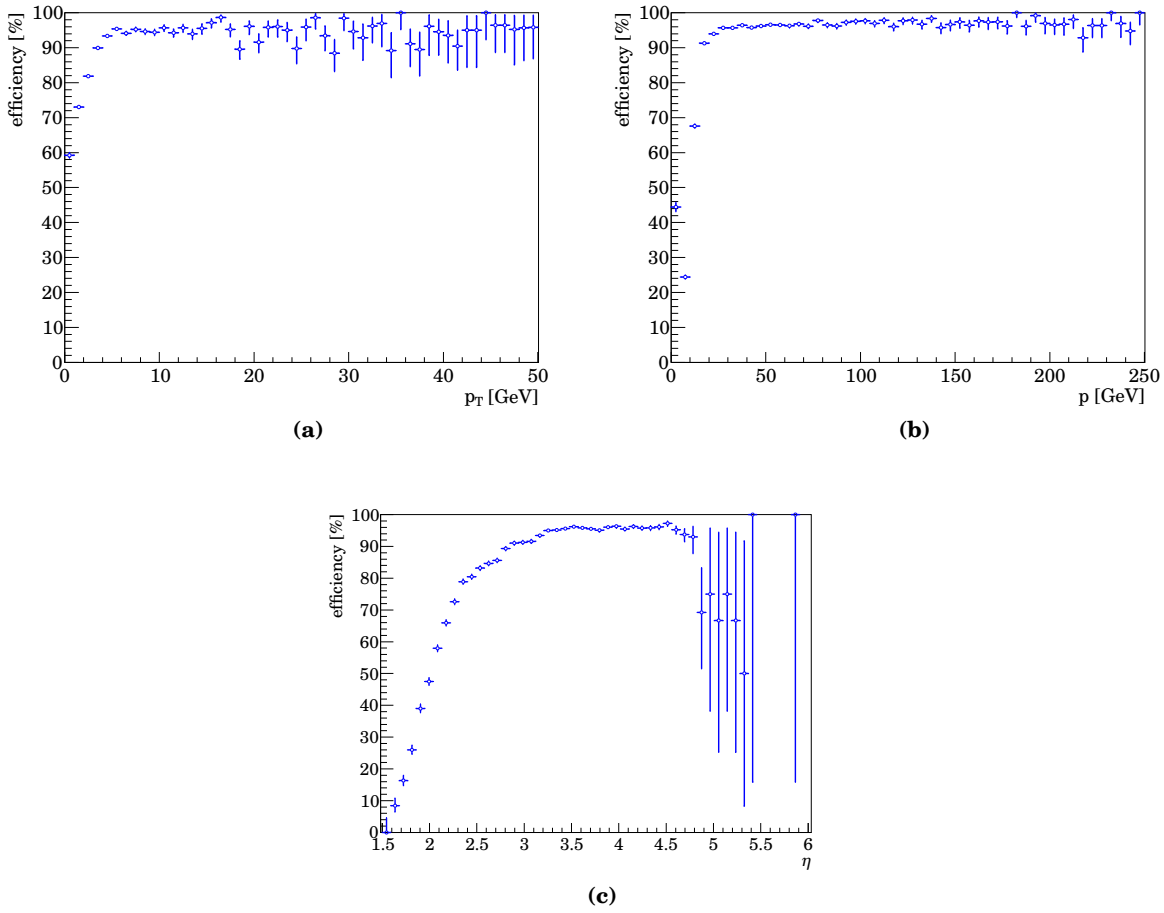


Figure 7.5.: *Muon track reconstruction efficiency as function of p_T, p, η for muons from offline selected $B_s \longrightarrow J/\psi(\mu\mu)\phi$ decays after software trigger cuts. The shape of the efficiency vs. p_T can essentially be understood from the software trigger cuts applied in the single muon trigger line which cuts at $p_T = 5$ GeV. The shapes of the efficiencies as function of p and η essentially follow from the p_T shape and detector acceptance.*

particle	quantity	cut
$\mu^{\pm}$	track χ^2/NDF	< 4
	p_T	> 500 MeV
	DLL(μ,π)	> 0
J/ψ	$\lvert m_{\mu\mu} - m^{\mathrm{PDG}}_{J/\psi}\rvert$	< 48 MeV
	decay vertex χ^2/NDF	< 4
$K^{\pm}$	track χ^2/NDF	< 10
	DLL(K,π)	> 0
ϕ	$\lvert m_{KK} - m^{\mathrm{PDG}}_{\phi}\rvert$	< 12 MeV
	p_T	> 1 GeV
	decay vertex χ^2/NDF	< 20
B_s	decay vertex χ^2/NDF	< 5
	min. $(\mathrm{IP}/\sigma_{\mathrm{IP}})^2$	> 25
	$\lvert m_{J/\psi\phi} - m^{\mathrm{PDG}}_{B_s}\rvert$	< 300 MeV

Table 7.2.: *Selection criteria for $B_s \longrightarrow J/\psi(\mu\mu)\phi$. DLL stands for "delta log likelihood" and is the logarithm of the likelihood ratio for the given particles as returned by the PID algorithms, i.e. $DLL(K,\pi)$ is the logarithm of the ratio of the likelihood of the track being a kaon divided by the likelihood of the track being a pion. IP stands for the impact parameter of a particle with respect to a primary vertex, so "min.* $\mathrm{IP}/\sigma_{\mathrm{IP}}$*" is the smallest value that* $\mathrm{IP}/\sigma_{\mathrm{IP}}$ *can reach for any primary vertex in the event.*

This subsection only discusses how the algorithm deals with such a region of interest on a technical level, examples of its performance and time requirements have been given above.

One can specify a region of interest to PatSeeding by giving a state and a covariance matrix at some reference z_{ref}. Track curvature is neglected.

Let $\vec{S}(z_{ref})$ be the state and $C(z_{ref})$ be the covariance matrix at z_{ref}:

$$\vec{S}(z_{ref}) = \begin{pmatrix} x \\ y \\ t_x \\ t_y \end{pmatrix}(z_{ref}) \qquad C(z_{ref}) = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xt_x} & \sigma_{xt_y} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yt_x} & \sigma_{yt_y} \\ \sigma_{xt_x} & \sigma_{yt_x} & \sigma_{t_x}^2 & \sigma_{t_xt_y} \\ \sigma_{xt_y} & \sigma_{yt_y} & \sigma_{t_xt_y} & \sigma_{t_y}^2 \end{pmatrix}(z_{ref}) \tag{7.1}$$

Then, centre and extension of the region of interest in the main tracker is then calculated using the following formulae:

$$\vec{S}(z) = P(z - z_ref)\vec{S}(z_{ref}) \tag{7.2}$$

$$C(z) = P(z - z_{ref})C(z_{ref})P^T(z - z_{ref}) \tag{7.3}$$

where $P(dz)$ is the matrix that transports the state over a distance dz, i.e.

$$P(dz) = \begin{pmatrix} 1 & 0 & dz & 0 \\ 0 & 1 & 0 & dz \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{7.4}$$

The extension of the region of interest around the centre is taken to be given by the square root of the elements on the diagonal of the covariance matrix at z.

Hits are only picked up in the spatial region given by above formulae, and at every step, the slopes resulting from the combination of hits are checked for consistence with the limits specified.

One can argue that restricting the region of interest to a window of one sigma is too tight a cut, but one can always scale the covariance matrix before passing it to PatSeeding to obtain a wider cut.

7.2. Tuning for early data taking conditions

During the initial stages of the LHCb experiment[6], (software) detector alignment has not yet reached its ultimate precision, and consequently, track reconstruction is more difficult because neither true position and orientation of detector elements is precisely known, nor are the uncertainties on these quantities very well understood. Technically, this is manifest in hit residuals (i.e. the track-hit distances) which are larger than for optimal alignment because the hit position and its uncertainty are not precisely known. This increased track-hit distance leads to lower quality tracks (worse track χ^2) and the need for wider search windows to maintain reconstruction efficiency, and usually results in higher fake track fractions. For this reason it is important to have a tuning which is tailored to these circumstances. Such a tuning for PatSeeding is demonstrated on (perfectly aligned) simulated events in this section. A cross-check with real data can be found in Section 5.2.2.

While it might be argued that testing such a tuning on simulated events is a waste of time, such a test gives a valuable cross check which helps study the effects of such a tuning, both in terms of efficiency and fake track fraction. First, track reconstruction efficiency in real data with a detector with unknown misalignments is not likely to be better that what one obtains on simulated events, so an estimate for an upper efficiency bound can be obtained. Second, the fake track fraction should not depend on the actual value of unknown misalignments (because fake tracks arise from random hit combinations), so this method gives an estimate of the fake track fraction provided the simulation models detector occupancy and detector parameters such as noise and crosstalk correctly.

Table 7.3 lists the changes in algorithm options with respect to PatSeeding's defaults. These changes are characterised by the opening of search windows and less stringent track quality requirements, both in terms of the number of hits required and the maximal contribution of individual hits to the total track χ^2.

Tables 7.4 and 7.5 show the performance of the tuning for early data taking conditions at $\nu = 1$ and $\nu = 3$, respectively. The corresponding plots with efficiency as function of p, p_T, η and the number of visible interactions can be found in Figures 7.6, C.11 and C.12 for all particles, good B daughters and K^0_S/Λ daughters. Plots at $\nu = 3$ can be found in Figures 7.7, C.13 and C.14.

While the tuning leaves reconstruction efficiency almost unaffected (not surprising

[6] . . . and sometimes also for later stages — the "early data" tuning was the default for the 2010 run, and remains the default until alignment is better understood. . .

option name	value	default value
MaxRangeIT	30.0 [mm]	15.0 [mm]
TolCollectOT	4.0 [mm]	3.0 [mm]
TolCollectIT	0.6 [mm]	0.3 [mm]
MaxChi2HitOT	56.25	30.0
MaxChi2HitIT	36.0	10.0
MinXPlanes	4	5
MinTotalPlanes	8	9
OTNHitsLowThresh	15	17
MaxMisses	2	1
MaxHoles	4	2

Table 7.3.: *Job options needed to tune PatSeeding to early data taking conditions (second column with values of tuning for early data conditions, third column with default values for comparisons).*

	long reconstructible		T reconstructible	
	eff. (clone) [%]	hit pur./eff. [%]	eff. (clone) [%]	hit pur./eff. [%]
all tracks	96.1 (0.3)	97.7 / 94.0	82.8 (0.3)	97.5 / 92.8
— $p > 5$GeV	96.8 (0.5)	97.8 / 94.5	96.5 (0.5)	97.9 / 94.1
B daughters	96.5 (0.2)	97.9 / 94.7	87.4 (0.2)	97.8 / 93.9
— $p > 5$ GeV	97.1 (0.3)	98.0 / 95.4	96.8 (0.3)	98.0 / 95.1
good B daughters	96.9 (0.2)	97.9 / 95.0	95.2 (0.3)	97.9 / 94.8
— $p > 5$ GeV	97.0 (0.2)	97.9 / 95.4	97.0 (0.2)	97.9 / 95.3
K_S^0/Λ daughters	95.8 (0.2)	97.7 / 93.5	80.8 (0.3)	97.4 / 92.3
— $p > 5$GeV	96.2 (0.3)	97.7 / 94.1	96.3 (0.5)	97.9 / 93.7
	tr. avg.	ev. avg.	tr. avg.	ev. avg.
ghost fraction	8.4	5.6	8.4	5.6

Table 7.4.: *Tracking efficiency for both long reconstructible tracks and tracks reconstructible in the main tracker only for tracks found by PatSeeding at $\nu = 1$ when tuned for early data taking conditions. Specifically, the subsamples of b daughters, good b daughters (fully reconstructible b, i.e. those b decays which have all decay products inside the LHCb acceptance) and K_S^0/Λ daughters are investigated in two momentum ranges, once for the entire spectrum, and once for tracks with $p > 5$ GeV. Clone fractions, hit efficiencies and hit purities are quoted for these categories, and the track and event averaged ghost fraction is given for the overall output.*

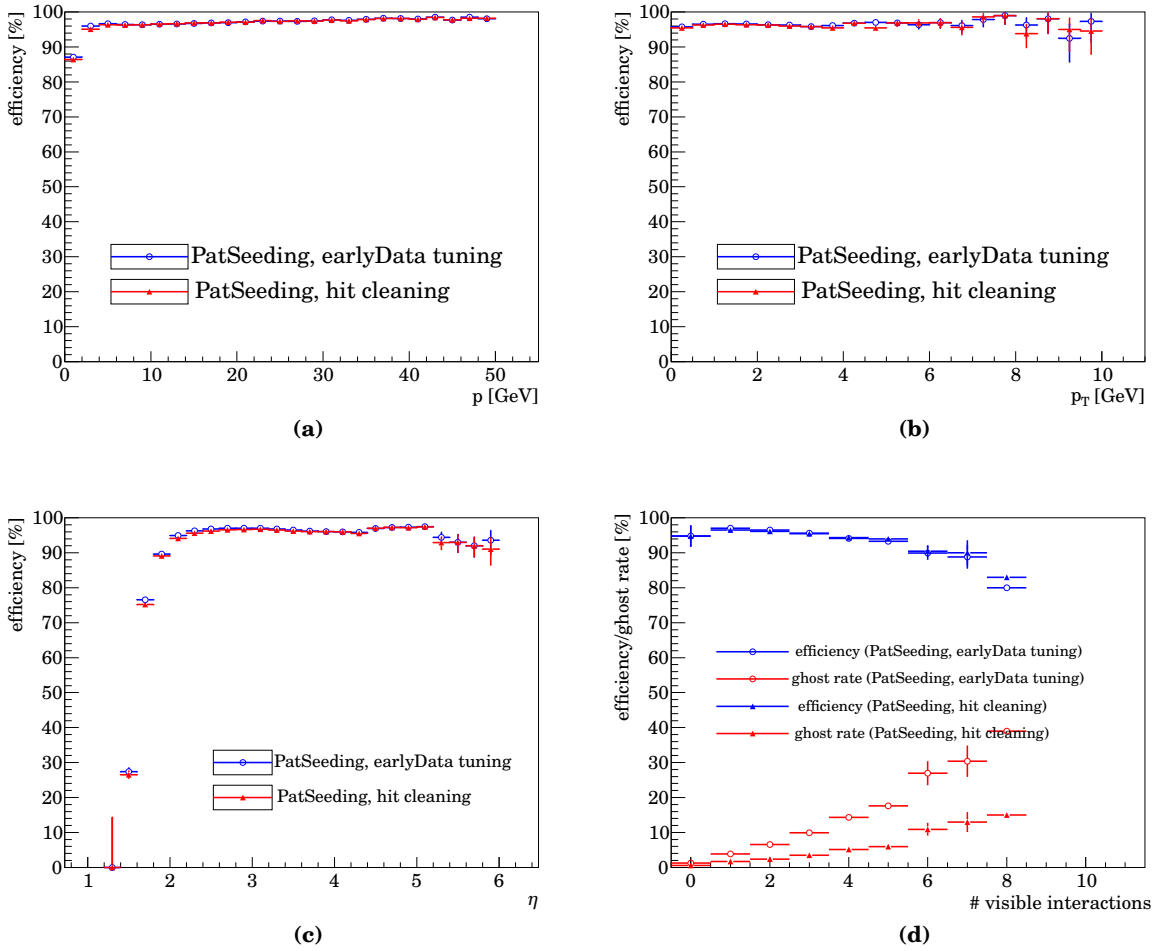


Figure 7.6.: *Efficiency versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for tracks found by PatSeeding at $\nu = 1$ using a tuning for early data taking conditions (circles, for comparison, the standard tuning is shown with triangles).*

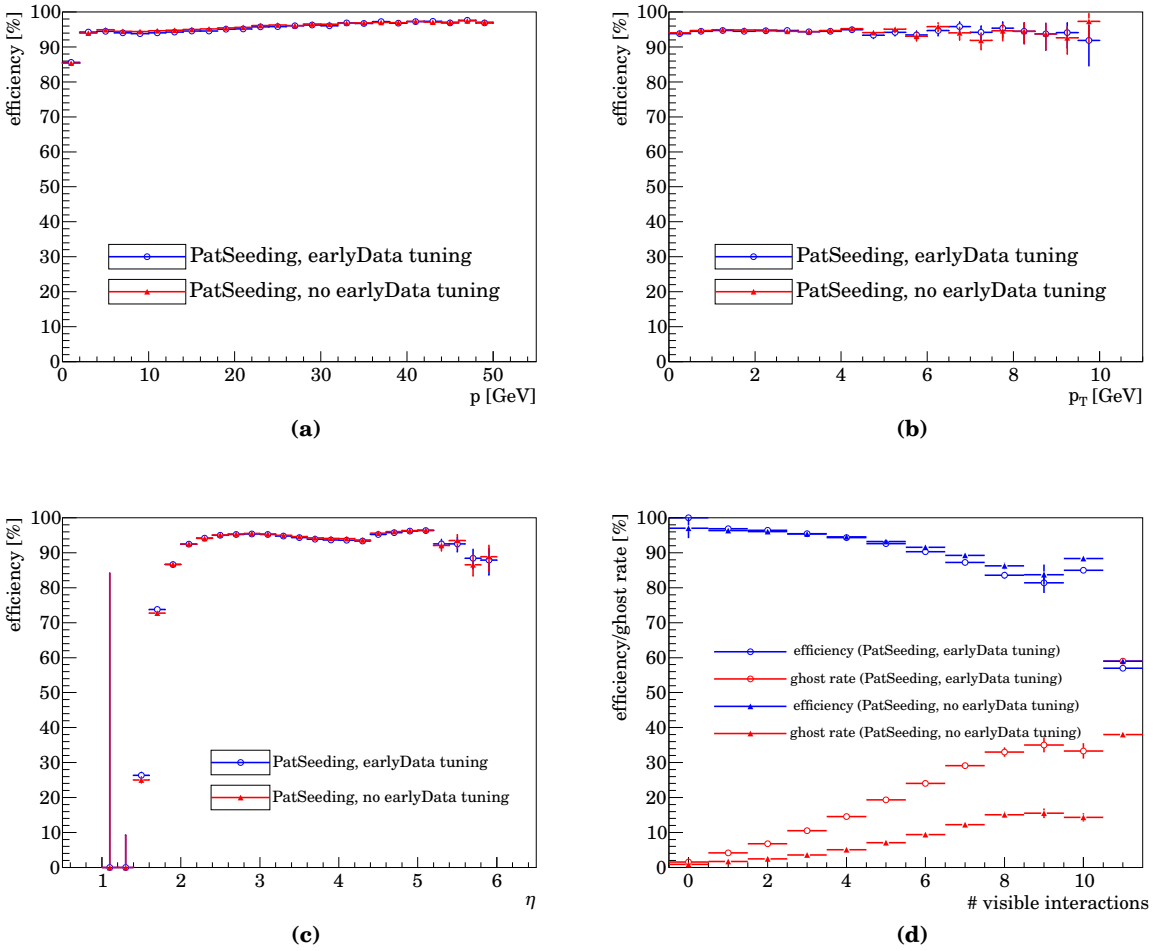


Figure 7.7.: *Efficiency versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for tracks found by PatSeeding at $\nu = 3$ using a tuning for early data taking conditions (circles, for comparison, the standard tuning is shown with triangles).*

	long reconstructible		T reconstructible	
	eff. (clone) [%]	hit pur./eff. [%]	eff. (clone) [%]	hit pur./eff. [%]
all tracks	94.1 (0.3)	96.9 / 92.9	80.6 (0.3)	96.7 / 91.7
— $p > 5$GeV	94.6 (0.5)	97.0 / 93.4	94.4 (0.5)	97.2 / 93.1
B daughters	95.1 (0.2)	97.2 / 93.9	86.0 (0.2)	97.0 / 93.1
— $p > 5$ GeV	95.6 (0.3)	97.2 / 94.6	95.4 (0.4)	97.3 / 94.3
good B daughters	95.4 (0.2)	97.1 / 94.4	94.1 (0.2)	97.2 / 94.2
— $p > 5$ GeV	95.4 (0.1)	97.1 / 94.8	95.3 (0.1)	97.3 / 94.6
K^0_S/Λ daughters	94.0 (0.2)	96.8 / 92.5	78.8 (0.3)	96.7 / 91.2
— $p > 5$GeV	94.1 (0.4)	96.8 / 92.9	94.3 (0.6)	97.2 / 92.6
	tr. avg.	ev. avg.	tr. avg.	ev. avg.
ghost fraction	15.3	10.3	15.3	10.3

Table 7.5.: *Tracking efficiency for both long reconstructible tracks and tracks reconstructible in the main tracker only for tracks found by PatSeeding at $\nu = 3$ when tuned for early data taking conditions. Specifically, the subsamples of b daughters, good b daughters (fully reconstructible b, i.e. those b decays which have all decay products inside the LHCb acceptance) and K^0_S/Λ daughters are investigated in two momentum ranges, once for the entire spectrum, and once for tracks with $p > 5$ GeV. Clone fractions, hit efficiencies and hit purities are quoted for these categories, and the track and event averaged ghost fraction is given for the overall output.*

with a perfectly aligned detector in the simulation), the fake track fraction is substantially higher, by a factor of about three, both for the event-averaged or track-averaged fake track fraction — a price one has to pay for some tolerance with respect to misalignment.

Table 7.6 shows average and maximum time spent per event; again, it is not surprising that the algorithm takes longer when search windows are wider. Figures 7.8 and 7.9 show timing behaviour as function of the number of main tracker hits and the number of visible interactions at $\nu = 1$ and $\nu = 3$. These parametrisations were fitted at $\nu = 1$:

$$\begin{aligned} t(n_{\text{hits}}) &= (0.2137 \pm 0.0006) \cdot 10^{-9} \cdot n^3_{\text{hits}} + (0.5439 \pm 0.0067) \cdot 10^{-6} \cdot n^2_{\text{hits}} + \\ &\quad (-0.0468 \pm 0.0230) \cdot 10^{-3} \cdot n_{\text{hits}} + (1.4887 \pm 0.0224) \end{aligned}$$

$$\begin{aligned} t(n_{\text{vis.}}) &= (0.2799 \pm 0.0031) \cdot n^3_{\text{vis.}} + (0.3273 \pm 0.0251) \cdot n^2_{\text{vis.}} + \\ &\quad (4.7018 \pm 0.0601) \cdot n_{\text{vis.}} + (1.2571 \pm 0.0408) \end{aligned}$$

At $\nu = 3$, the following results were obtained:

$$\begin{aligned} t(n_{\text{hits}}) &= (0.1482 \pm 0.0003) \cdot 10^{-9} \cdot n^3_{\text{hits}} + (1.6271 \pm 0.0043) \cdot 10^{-6} \cdot n^2_{\text{hits}} + \\ &\quad (-4.9168 \pm 0.0197) \cdot 10^{-3} \cdot n_{\text{hits}} + (6.8756 \pm 0.0264) \end{aligned}$$

$$\begin{aligned} t(n_{\text{vis.}}) &= (0.0239 \pm 0.0007) \cdot n^3_{\text{vis.}} + (2.6894 \pm 0.0087) \cdot n^2_{\text{vis.}} + \\ &\quad (-1.0652 \pm 0.0302) \cdot n_{\text{vis.}} + (5.4172 \pm 0.0307) \end{aligned}$$

ν	average time per event [ms]	maximum time per event [ms]
1	12.00	437.9
3	29.71	582.1

Table 7.6.: *Average and maximum time needed by PatSeeding for the reconstruction of a single event at $\nu = 1$ and $\nu = 3$, using a tuning for early data taking conditions.*

	eff. (clone) [%]	hit pur./eff. [%]
all tracks	90.1 (1.2)	98.5 / 94.2
— $p > 5$GeV	93.9 (0.9)	98.6 / 96.0
B daughters	90.3 (1.2)	98.6 / 94.2
— $p > 5$ GeV	94.2 (1.0)	98.7 / 95.8
good B daughters	93.3 (1.3)	98.7 / 95.1
— $p > 5$ GeV	95.1 (0.9)	98.7 / 96.1
K_S^0/Λ daughters	81.8 (2.4)	98.1 / 91.6
— $p > 5$GeV	88.0 (1.4)	98.3 / 94.8
	tr. avg.	ev. avg.
ghost fraction	18.2	14.6

(a) $\nu = 1$

	eff. (clone) [%]	hit pur./eff. [%]
all tracks	87.9 (1.2)	97.9 / 93.8
— $p > 5$GeV	92.1 (0.9)	98.1 / 95.6
B daughters	89.0 (1.3)	98.1 / 93.8
— $p > 5$ GeV	93.1 (1.1)	98.2 / 95.4
good B daughters	92.6 (1.3)	98.1 / 94.6
— $p > 5$ GeV	94.4 (1.0)	98.2 / 95.7
K_S^0/Λ daughters	77.8 (1.9)	97.5 / 91.2
— $p > 5$GeV	84.7 (1.2)	97.7 / 94.4
	tr. avg.	ev. avg.
ghost fraction	25.1	20.1

(b) $\nu = 3$

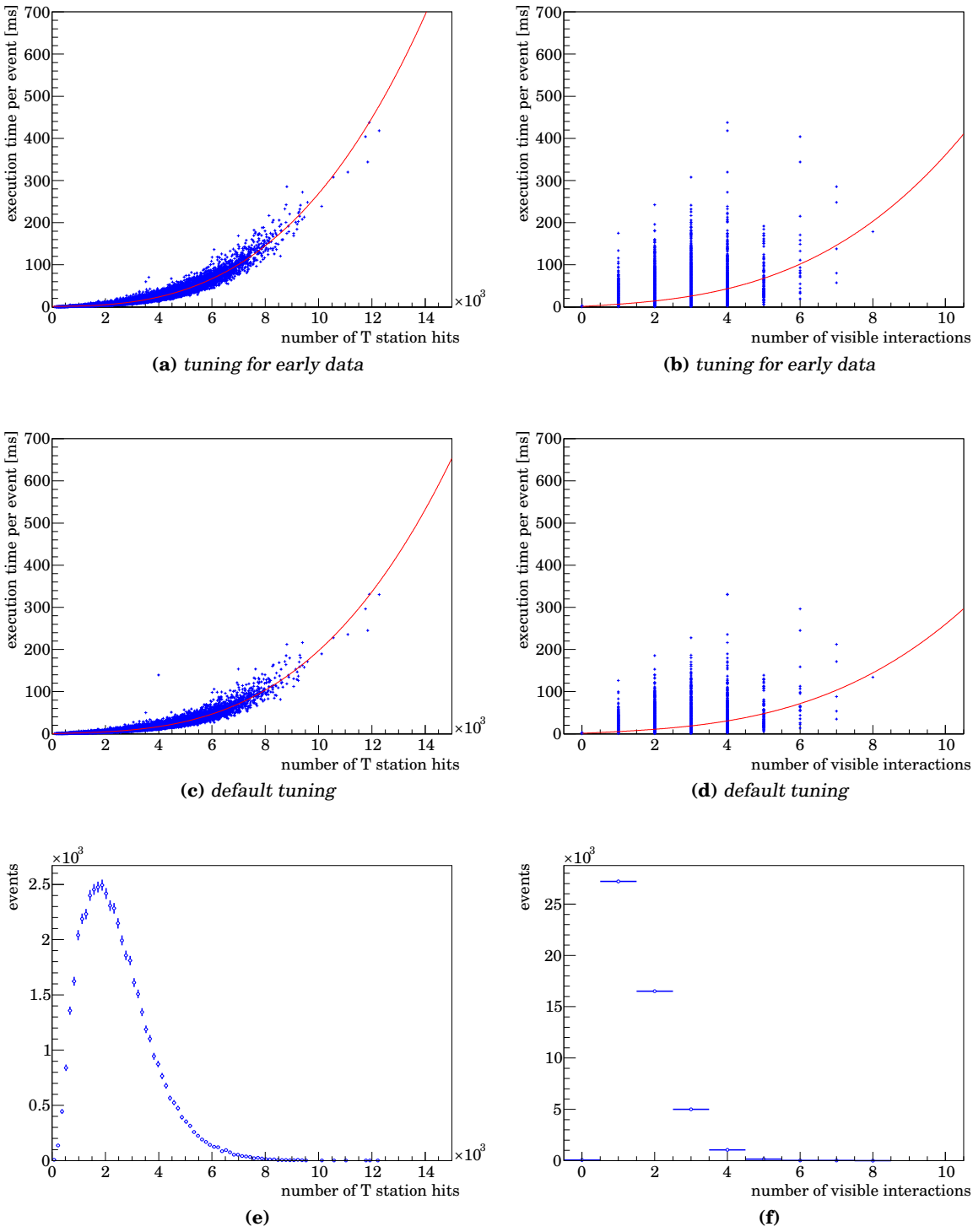
Table 7.7.: *Efficiency, ghost and clone fraction at $\nu = 1$ (a) and $\nu = 3$ (b) for long-reconstructible tracks of the algorithm evaluated using 50,000 inclusive b events, reusing tracks already found by PatForward.*

7.3. Working in tandem with PatForward

The algorithm can work reusing information about tracks found by PatForward (see [43] or [44]), a pattern recognition algorithm that starts from seeds in the vertex detector to reconstruct tracks in the T stations. Obviously, it is not necessary to find tracks already found by PatForward a second time. In time-constrained situations such as the software trigger, PatSeeding can be configured to save some time in the pattern recognition by exploiting this fact. Such a strategy can help to quickly reconstruct K_S^0 or Λ particles which decay after the Velo.

Technically, PatSeeding ignores hits used for PatForward tracks (cf. 6.2). At $\nu = 1$, the average execution time per event drops to 6.46 ms in that case (compared to 9.09 ms if the information of PatForward is not used). A full summary on the timing behaviour for $\nu = 1$ and $\nu = 3$ can be found in Table 7.10.

A summary of reconstruction efficiencies at $\nu = 1$ and $\nu = 3$ for long tracks at the end of the entire track reconstruction stage can be found in Table 7.7 (these figures are to be compared with those in Tables 5.3 (a) and 5.7 (a). The loss in reconstruction efficiency with respect to PatSeeding in its default configuration is typically below 1% (with roughly one third of the total execution time of PatSeeding being saved).



(a) *tuning for early data*

(b) *tuning for early data*

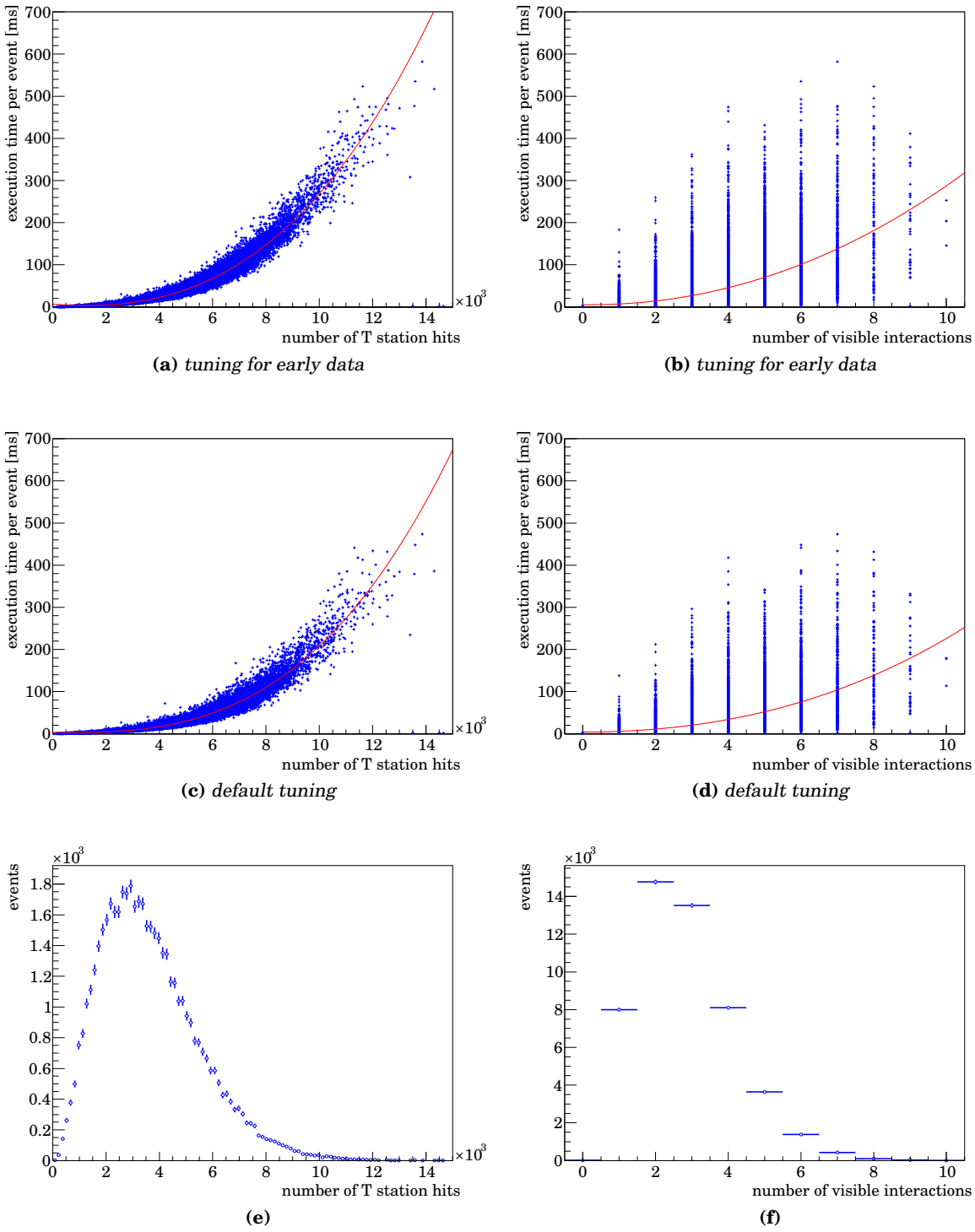
(c) *default tuning*

(d) *default tuning*

(e)

(f)

Figure 7.8.: *Execution time at $\nu = 1$ of PatSeeding tuned for early data taking conditions as function of the number of main tracker hits (a, c) and the number of visible interactions (b, d). A third order polynomial has been used to fit the graphs; (a, b) contain the plots for the tuning for early data, while (c, d) contain the default tuning for comparison. (e) and (f) contain the distribution of the number of main tracker hits and the number of visible interactions in the events which entered the plots.*



(a) *tuning for early data*

(b) *tuning for early data*

(c) *default tuning*

(d) *default tuning*

(e)

(f)

Figure 7.9.: *Execution time at $\nu = 3$ of PatSeeding tuned for early data taking conditions as function of the number of main tracker hits (a, c) and the number of visible interactions (b, d). A third order polynomial has been used to fit the graphs; (a, b) contain the plots for the tuning for early data, while (c, d) contain the default tuning for comparison. (e) and (f) contain the distribution of the number of main tracker hits and the number of visible interactions in the events which entered the plots.*

	long reconstructible		T reconstructible	
	eff. (clone) [%]	hit pur./eff. [%]	eff. (clone) [%]	hit pur./eff. [%]
all tracks	97.5 (0.2)	97.6 / 92.9	82.6 (0.2)	97.7 / 92.1
— $p > 5$GeV	98.5 (0.2)	97.8 / 95.5	97.2 (0.2)	98.0 / 94.7
B daughters	97.7 (0.2)	97.9 / 92.8	87.5 (0.2)	97.9 / 92.4
— $p > 5$ GeV	98.7 (0.1)	98.0 / 95.1	97.7 (0.2)	98.1 / 94.8
good B daughters	98.5 (0.1)	97.9 / 94.0	96.3 (0.1)	98.0 / 94.1
— $p > 5$ GeV	98.8 (0.1)	97.9 / 95.4	98.3 (0.1)	98.0 / 95.4
K^0_S/Λ daughters	96.9 (0.3)	97.5 / 91.4	79.4 (0.2)	97.8 / 91.4
— $p > 5$GeV	98.0 (0.2)	97.7 / 94.9	95.4 (0.3)	98.2 / 93.6
	tr. avg.	ev. avg.	tr. avg.	ev. avg.
ghost fraction	2.7	2.0	2.7	2.0

Table 7.8.: *Efficiency, ghost and clone fraction at $\nu = 1$ of the algorithm evaluated using 50,000 inclusive $b\bar{b}$ events, reusing tracks already found by PatForward.*

With this modification, PatSeeding only reconstructs tracks which were not found by PatForward, so the efficiency to reconstruct T tracks of PatSeeding alone is poor. To measure combined efficiencies for T tracks, it is possible to extract the main tracker segments of the tracks found by PatForward, optionally killing clones among these segments (which arise because PatForward is tuned to most efficiently match main tracker hits to vertex detector seeds, so it sometimes splits the main tracker part of a singe track to match two seeds in the vertex detector). With this track segment extraction, it becomes possible to compare the reconstruction efficiencies in the main tracker to PatSeeding in its default configuration. `UseForwardTracks` can be set to `true` to enable this track extraction. The clone merging procedure is activated by setting `ForwardCloneMergeSeg` to `true`. When activated, tracks with track parameters differing by less than `ForwardCloneMaxXDist`, `ForwardCloneMaxYDist` and `ForwardCloneMaxTXDist` in x, y, and slope in x direction, t_x, respectively, are considered clones (track parameters are evaluated at $z =$ `zReference`).

For tracks sharing more than a fraction of `ForwardCloneMaxShared` of their hits, the track with more hits is kept, otherwise, the hits of both tracks are merged to produce a single track.

The combined efficiency to find tracks in the main tracker is shown in Tables 7.8 and 7.9. The corresponding figures show the efficiency as function of p, p_T, η and the number of visible interactions for all long-reconstructible tracks, long-reconstructible b and K^0_S/Λ daughters (Figures 7.10, C.7 and C.8 are for $\nu = 1$, the plots for $\nu = 3$ are in Figures 7.11, C.9 and C.10). Without the clone killing among main tracker segments inherited from PatForward, the clone fraction is on the order of 5%. One should also note that ghost tracks found by PatForward can cause some inefficiency in PatSeeding because the hits are unavailable for PatSeeding when running in this mode.

Running the track extraction and clone killing stages adds 0.2 to 0.4 ms per event of execution time on average, bringing the average execution time up to 6.17 ms per event at $\nu = 1$. Execution time as function of the number of main tracker hits n_{hits} and number of visible interactions $n_{\text{vis.}}$ is shown in Figures 7.12 and 7.13, the fitted parametrisations

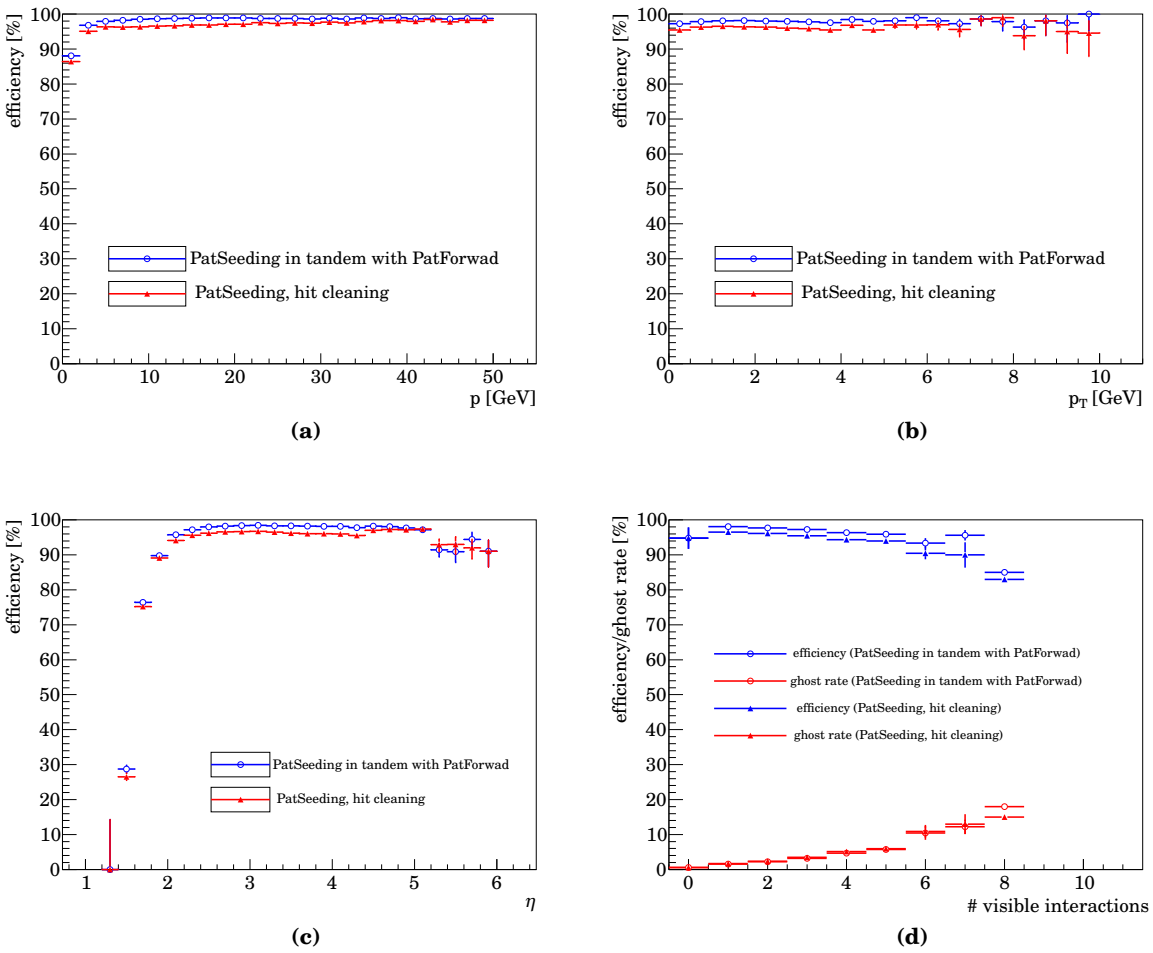


Figure 7.10.: *Efficiency versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for tracks found by PatSeeding when running in tandem with PatForward (circles) and it its default configuration (triangles).*

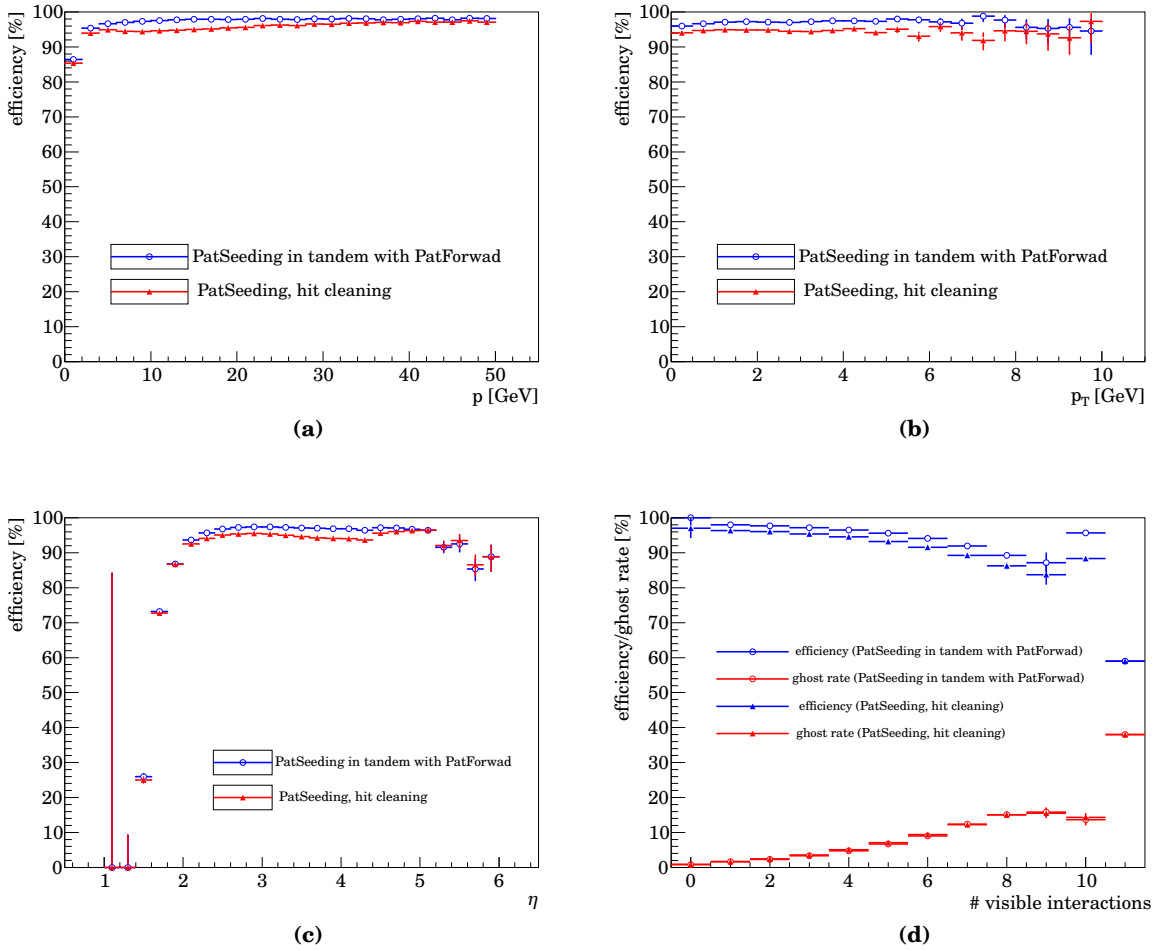


Figure 7.11.: *Efficiency (at $\nu = 3$) versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for tracks found by PatSeeding when running in tandem with PatForward (circles) and in its default configuration (triangles).*

	long reconstructible		T reconstructible	
	eff. (clone) [%]	hit pur./eff. [%]	eff. (clone) [%]	hit pur./eff. [%]
all tracks	96.3 (0.4)	96.7 / 92.3	80.8 (0.3)	96.8 / 91.3
— $p > 5$GeV	97.5 (0.3)	97.0 / 95.0	95.8 (0.3)	97.3 / 94.1
B daughters	97.0 (0.3)	97.1 / 92.4	86.5 (0.3)	97.1 / 91.8
— $p > 5$ GeV	98.0 (0.2)	97.2 / 94.7	96.9 (0.2)	97.3 / 94.2
good B daughters	97.9 (0.2)	97.0 / 93.5	96.1 (0.3)	97.1 / 93.5
— $p > 5$ GeV	98.2 (0.2)	97.1 / 95.0	97.7 (0.2)	97.2 / 94.9
K^0_S/Λ daughters	95.5 (0.5)	96.6 / 90.8	77.4 (0.3)	97.0 / 90.4
— $p > 5$GeV	96.6 (0.3)	96.8 / 94.2	93.3 (0.4)	97.5 / 92.6
	tr. avg.	ev. avg.	tr. avg.	ev. avg.
ghost fraction	5.0	3.5	5.0	3.5

Table 7.9.: *Efficiency, ghost and clone fraction at $v = 3$ of the algorithm evaluated using 50,000 inclusive $b\bar{b}$ events, reusing tracks already found by PatForward.*

(for the case where the track extraction and clone killing stages are active) are given by

$$\begin{aligned} t(n_{\text{hits}}) &= (0.1345 \pm 0.0006) \cdot 10^{-9} \cdot n_{\text{hits}}^3 + (-0.3373 \pm 0.0067) \cdot 10^{-6} \cdot n_{\text{hits}}^2 + \\ &\quad (2.1082 \pm 0.0230) \cdot 10^{-3} \cdot n_{\text{hits}} + (-0.6080 \pm 0.0224) \end{aligned}$$

$$\begin{aligned} t(n_{\text{vis.}}) &= (0.0789 \pm 0.0031) \cdot n_{\text{vis.}}^3 + (0.2367 \pm 0.0251) \cdot n_{\text{vis.}}^2 + \\ &\quad (2.1407 \pm 0.0601) \cdot n_{\text{vis.}} + (1.1713 \pm 0.0408) \end{aligned}$$

for $v = 1$ and

$$\begin{aligned} t(n_{\text{hits}}) &= (0.1713 \pm 0.0003) \cdot 10^{-9} \cdot n_{\text{hits}}^3 + (-1.0216 \pm 0.0043) \cdot 10^{-6} \cdot n_{\text{hits}}^2 + \\ &\quad (4.9976 \pm 0.0197) \cdot 10^{-3} \cdot n_{\text{hits}} + (-3.7119 \pm 0.0264) \end{aligned}$$

$$\begin{aligned} t(n_{\text{vis.}}) &= (0.0722 \pm 0.0007) \cdot n_{\text{vis.}}^3 + (0.2848 \pm 0.0087) \cdot n_{\text{vis.}}^2 + \\ &\quad (2.2019 \pm 0.0302) \cdot n_{\text{vis.}} + (1.1908 \pm 0.0307) \end{aligned}$$

for $v = 3$.

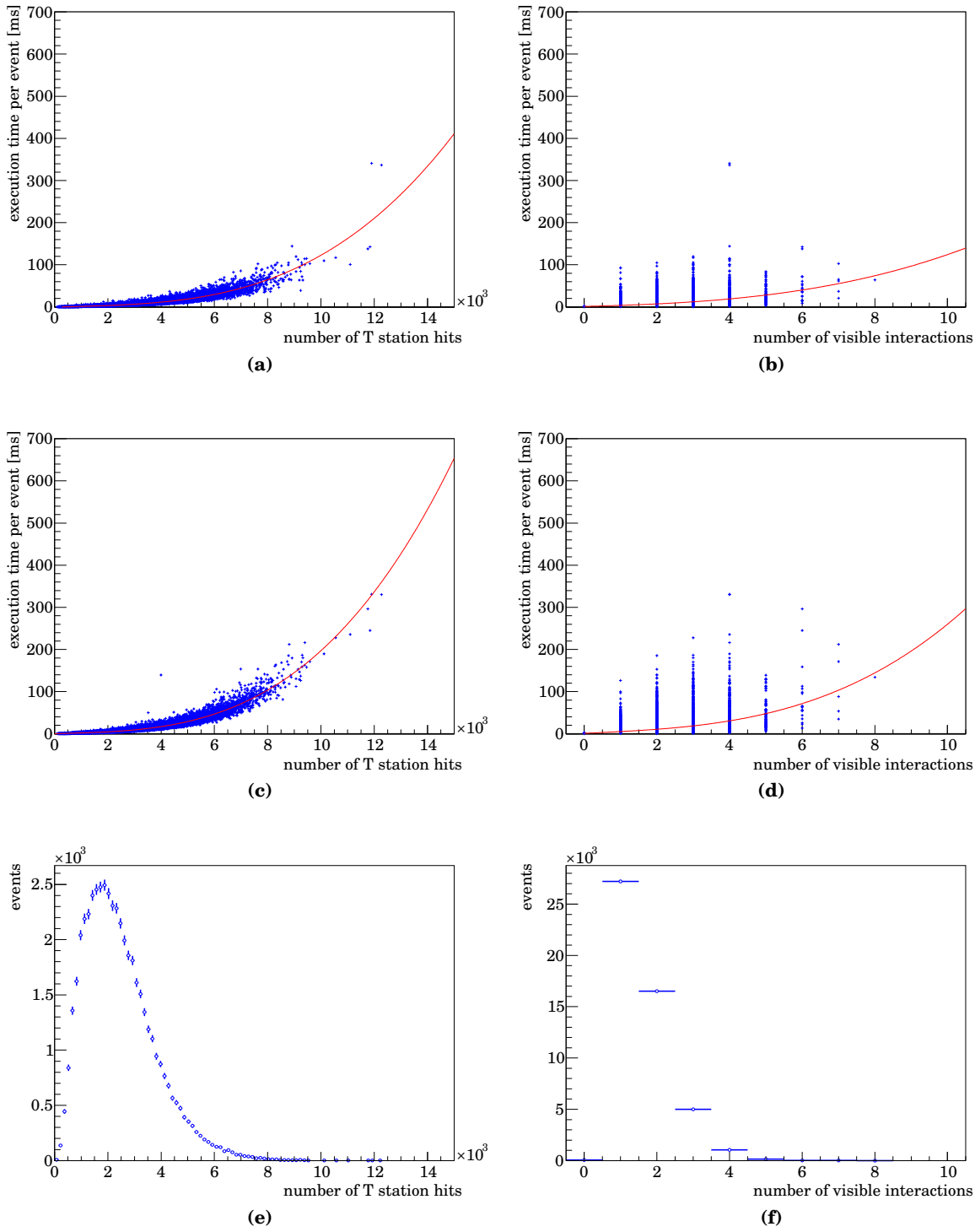


Figure 7.12.: *Execution time at $\nu = 1$ of PatSeeding working in tandem with* `PatForward` *as function of the number of main tracker hits (a, c) and the number of visible interactions (b, d). A third order polynomial has been used to fit the graphs (a, b), while (c, d) contain the corresponding plots for the default configuration. (e) and (f) contain the distribution of the number of main tracker hits and the number of visible interactions in the events which entered the plots.*

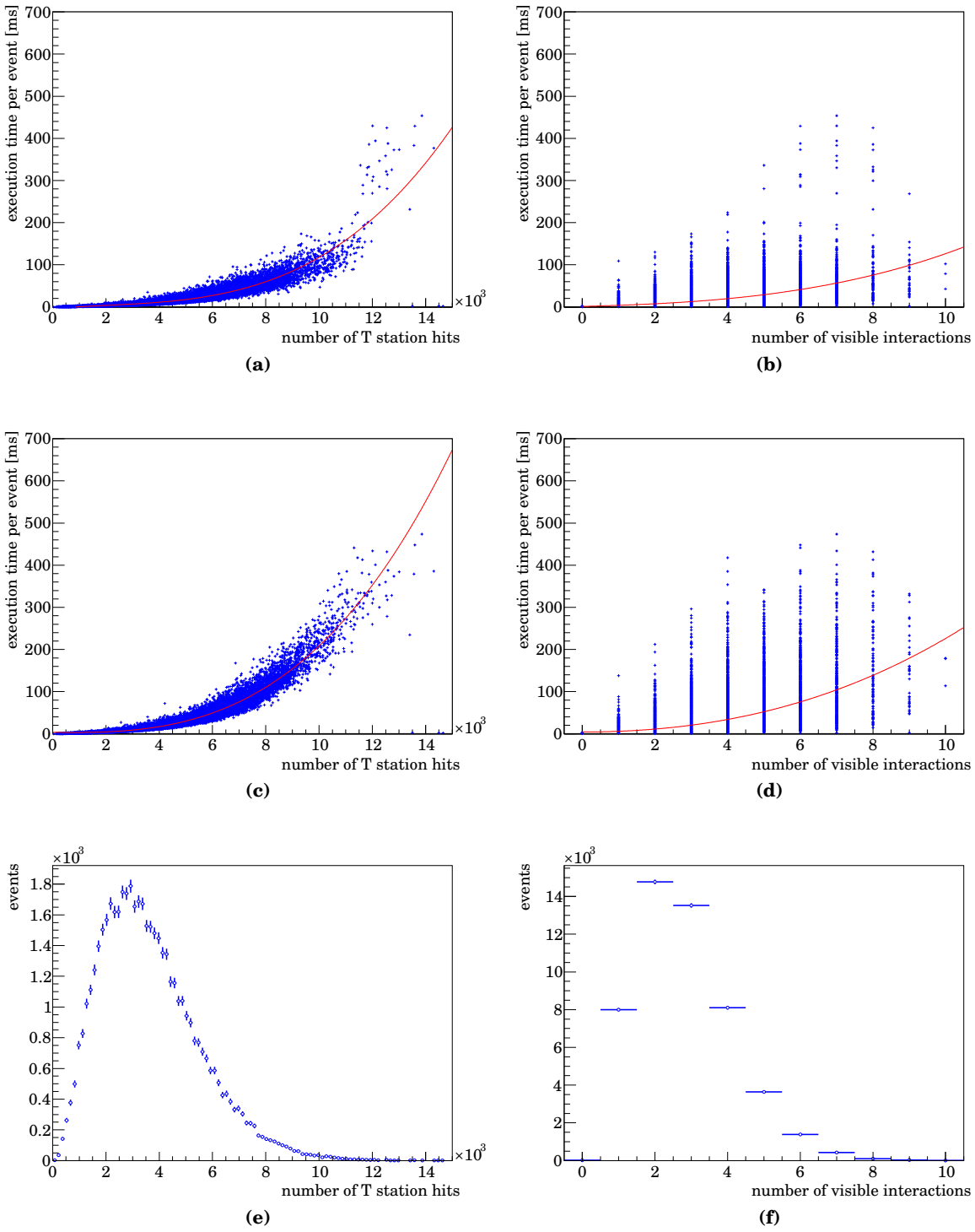


Figure 7.13.: *Execution time at $v = 3$ of PatSeeding working in tandem with* `PatForward` *as function of the number of main tracker hits (a, c) and the number of visible interactions (b, d). A third order polynomial has been used to fit the graphs (a, b), while (c, d) contain the corresponding plots for the default configuration. (e) and (f) contain the distribution of the number of main tracker hits and the number of visible interactions in the events which entered the plots.*

v		tandem [ms]	tandem + clone killing [ms]	PatForward [ms]	plain PatSeeding [ms]
1	average	5.97	6.17	12.44	9.09
	maximum	342.0	340.6	391.6	331.0
3	average	13.08	13.45	29.75	22.50
	maximum	442.2	453.8	1130.7	473.9

Table 7.10.: *Execution time of PatSeeding working in tandem with PatForward; average and maximum execution time are given for both algorithms at $v = 1$ and $v = 3$. If the optional clone killing stage for the input tracks found by PatForward at the beginning of PatSeeding is switched on, the time demands increase to what is shown in "tandem + clone killing" column. For completeness, the time spent in PatForward alone is given in the "PatForward" column, while the last column contains the execution time of plain PatSeeding, i.e. when not working in tandem with PatForward.*

option	value
xMagTol	400 mm
zMagnet	0 mm
FieldOff	true
MinMomentum	50000 MeV

Table 7.11: *Options for running without magnetic field.*

7.4. Tracking without magnetic field

While the LHCb experiment is usually run at nominal strength of its magnetic field, during 2008 and early 2009, it was important to have the possibility to get tracks in the main tracker for the then ongoing commissioning efforts even before the magnet was first ramped up. To provide this functionality, PatSeeding provides a tuning for these circumstances.

Tracking with and without magnetic field is not substantially different from the point of view of this algorithm, apart from the fact that one expects straight tracks instead of curved ones. To account for this fact, the set of options in table 7.11 has been prepared.

Effectively, it demands straight tracks by turning the centre of magnet compatibility cut into a pointing constraint at $z = 0$ and enforces relatively straight tracks by artificially introducing a cut on minimal momentum; this will translate into search windows which wrap more tightly around a straight line approximation in $x - z$ projection inside the algorithm. It also applies an additional cut on the track curvature c itself:

$$\frac{\texttt{TolCollect}}{dz^2} > |c|$$

where dz is half the distance between the first and the last hit in z and `TolCollect` has to be substituted by either `TolCollectIT` or `TolCollectOT`, depending on the subdetector.

7.4.1. Performance

Simulated Events without magnetic field were most important during the initial commissioning efforts in 2008 and 2009. Since then a lot of progress has been made, and both real data and simulated event have non-zero magnetic field nowadays, and samples of simulated events without magnetic field are no longer available. Therefore, the performance of this tuning will be evaluated using samples of simulated events and a software stack[7] which date back to that period. The performance of this tuning has been evaluated on a simulated minimum bias sample with 10,000 events prepared for alignment purposes (see [61]). The beam energy was 7 TeV, and the events were generated without spillover or magnetic field.

The results are summarised in Table 7.12 and Fig. 7.14. Efficiency for long tracks is lower (back then, the long-reconstructible track reconstruction efficiency without magnetic field was 89.1%; at full field, it was at 94.9%) than what was seen with magnetic field, and significantly lower when considering all reconstructible tracks (35.6% without field has to be compared to 70.3% with field for data from that period). This has two reasons: First, the magnetic field usually deflects tracks with very low momentum

[7] BRUNEL v32r8, with PatSeeding backported from BRUNEL v33r0.

	long-reconstructible eff. (clone) [%]	all eff. [%]
all tracks	89.1 (0.1)	35.6
— $p > 5$ GeV	90.1	
	tr. avg.	ev. avg.
ghost fraction	16.0	5.8

Table 7.12.: *Efficiency, ghost and clone fraction of the algorithm evaluated using 10,000 minimum bias events without magnetic field.*

away from the main tracker. These tracks are more difficult to reconstruct because of the increased importance of multiple scattering. Second, the total T station multiplicity increases as more and more low momentum tracks make it through the magnet. These factors both add to the difficulties encountered by the algorithm.

The dip in reconstruction efficiency around $\eta \approx 3.9$ is due the difference in granularity when going from Outer to Inner Tracker and the high main tracker occupancies when running without magnetic field. With high occupancies due to missing magnetic field, the Outer Tracker reconstruction becomes more and more difficult as η increases because the track density increases. The Inner Tracker left and right boxes make this less important for tracks which are roughly in the $x-z$ plane because these boxes extend to relatively large x, and the granularity is much finer. For tracks which are roughly in the $y-z$ plane, this is a different story, though, because tracks generate a relatively high occupancy in the central Outer Tracker modules, and the switch from the coarse-grained Outer Tracker to the Inner Tracker with its higher granularity happens at much higher η where the track density is bigger. Thus, tracks with small $|x|$ coordinates in the main tracker start entering the top or bottom boxes of the Inner Tracker around $\eta \approx 3.9$ and start to have hits in fewer than two full stations around $\eta \approx 4.4$. This explains the gradual decrease in efficiency when coming from low η towards $\eta \approx 3.9$ and also the subsequent rise when tracks move towards higher η. This general trend is "smeared" somewhat because the track slope in $x-z$ direction was not taken into account in the considerations above. This dip is absent when reconstructing tracks with magnetic field because the main tracker occupancy is lower and the magnetic field smears out the distribution of tracks along $x-z$ direction, leading to much less variation in the occupancy distribution along the x direction of the detector.

It should also be noted that the tuning is not particularly optimised. It was designed to quickly give acceptable performance for long tracks for alignment and calibration studies which can be done before the magnet is first ramped up.

7.4.2. Modification to reconstruct cosmic muons

To reconstruct cosmic muons taken during the commissioning effort, three minor modifications to the algorithm are required:

- Initially, neither the timing of the Outer Tracker with respect to the bunch clock

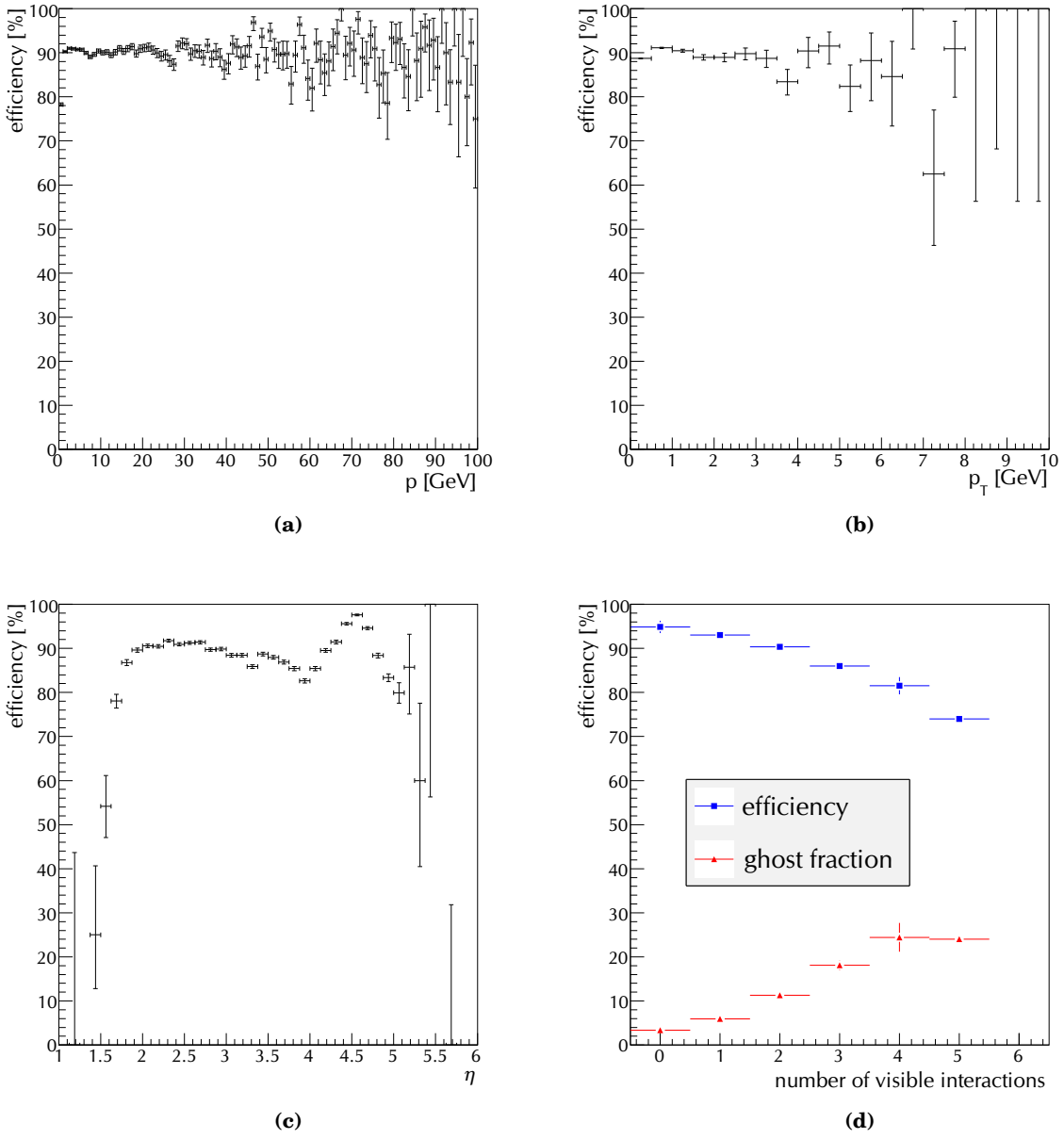


Figure 7.14.: *Efficiency versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions (all for minimum bias events without magnetic field). The dip around $\eta \approx 3.9$ is explained in the text.*

nor the relation between drift radius and drift time are known exactly. Moreover, the arrival time of cosmic muons has no relation whatsoever with the bunch clock from which all other clock signals in the experiment are derived. Therefore, the framework supplying the hits is instructed not to use drift time measurements, instead it returns zero drift distance in the Outer Tracker (with the resolution set to $\frac{5\text{mm}}{\sqrt{12}}$)[8].

- Cosmics do not point back to the primary vertex. PatSeeding in its normal configuration relies heavily on tracks pointing back to the primary vertex to find the pattern in the stereo layers during the per region track search. For cosmics, this is replaced by a combinatorial approach which counts the number of hits inside a window around a line connecting two of these hits. The combinations of two hits that gives most hits inside this window is taken.

- Because cosmics traverse the detector at much steeper slopes than tracks coming from the primary vertex, one cannot expect to have hits in more that one station of the Inner Tracker. Therefore, only the Outer Tracker is used to supply the three hits for the initial parabola. When collecting hits in a window around this parabola, all regions are searched for hits, if the track traverses them (including Inner Tracker regions).

Using these modifications, it is possible to reconstruct cosmic muon tracks. Usually, the calorimeter serves as trigger. An example is shown in Figure 7.15.

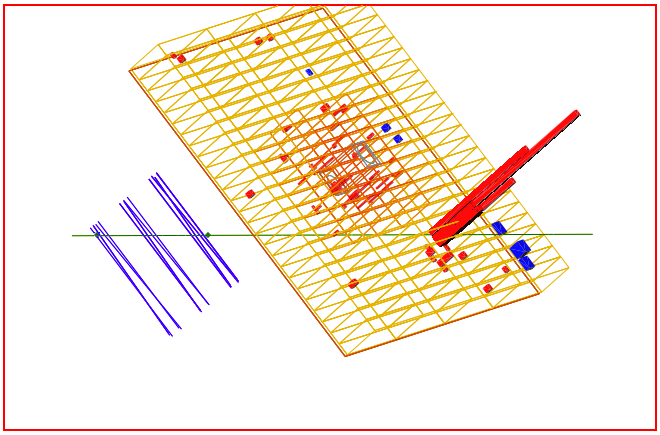


Figure 7.15.: *Cosmic track reconstructed from run 24941, event 138. Data taken July 1st, 2008, triggered on energy deposit in the calorimeters. In this run, one half of the Outer Tracker (C side) was read out together with the calorimeters. Reconstructed track in dark green, Outer Tracker straws on the track in blue, clusters in electromagnetic and hadronic calorimeters in red and blue, respectively.*

The job options used to reconstruct this event are shown in table 7.13. As there was no magnetic field, the options from 7.4.1 were used as well; in case of conflicting options,

[8]Note that if these tracks are to be fitted, the fitter has to be told as well that drift times are to be ignored.

those given here should take precedence.

option	value
PatSeeding.PatSeedingTool.Cosmics	true
PatSeeding.PatSeedingTool.MinXPlanes	4
PatSeeding.PatSeedingTool.MinTotalPlanes	8
PatSeeding.PatSeedingTool.OTNHitsLowThresh	9
PatSeeding.PatSeedingTool.MaxMisses	2
PatSeeding.PatSeedingTool.MaxYAtOrigin	400000
PatSeeding.PatSeedingTool.MaxYAtOriginLowQual	800000
PatSeeding.PatSeedingTool.xMagTol	400000
PatSeeding.PatSeedingTool.MinMomentum	1e-4
PatSeeding.PatSeedingTool.QualityWeights	1.0, 0.0
ToolSvc.OTHitCreator.NoDriftTimes	true

Table 7.13.: *Options used to reconstruct cosmics. The last entry puts to zero the Outer Tracker drift radii for all pattern recognition algorithms, as explained in the text.*

While cosmics data taking is an important ingredient to the calibration and alignment of the Outer Tracker[9], no formal evaluation of the performance of PatSeeding has been done. This is understandable: Given the fact that cosmic muons do not have a fixed relation to the front end electronics readout clock, triggering cosmics successfully is (to a certain extent) also a matter of timing. Putting the blame for an inefficiency on the pattern recognition alone does not seem quite fair. There are sufficient cosmics to ensure that inefficiencies due to either timing or pattern recognition do not pose serious problems, though.

The standard reconstruction sequence does a so-called pre-fit on tracks in the main tracker, ignoring drift times. Thus, options to switch off drift times in the fitter are best looked up there.

[9] Given the angular distribution of cosmic muons, it becomes apparent that other tracking subdetectors are either too small or too far away to get tracks at useful rates which would link these detectors to the Outer Tracker.

Part II.

Prompt K^0_S production in pp collisions at $\sqrt{s} = 900\,\mathrm{GeV}$

8. Introduction and analysis strategy

Production of strangeness provides sensitive tests of hadronic interactions since the mass of strange quark is of order Λ_{QCD}. Compared to up and down quarks, strangeness production is suppressed, but still contributes even in the non-perturbative regime. For this reason, it is interesting to study the production of strange particles (e.g. K^0_S or Λ) and compare the measurements to the predictions of hadronisation models.

K^0_S production has been studied in pp and $p\bar{p}$ collisions by several experiments at a variety of centre-of-mass energies (see [62], [63], [64], [65], [66],[67], [68]), and the most recent Tevatron measurements of K^0_S production measurements have shown deviations from expectations from hadronisation models ([67]).

The importance of a well described underlying event and good model of the fragmentation process in simulated events for most measurements which are planned at LHCb has already been discussed in Chapter 2. Besides this obvious physics motivation, the measurement of the K^0_S production cross section in pp collisions presented here serves as a test: With a well working detector and proper analysis procedures in place, a good understanding of the recorded data must be verified before more complicated analyses can go ahead. For a cross section measurement, the understanding of the various efficiency contributions and the knowledge of the luminosity are the most important inputs. In this thesis, the focus is on the determination of the tracking efficiency. Since it is taken from simulated events, good agreement between data and simulation is needed, and a lot of effort will be devoted to that.

This part of this thesis describes a measurement of the prompt K^0_S production cross section in pp collisions at $\sqrt{s} = 900\,\text{GeV}$ with the LHCb experiment using data from the 2009 pilot run. A sample corresponding to a luminosity of $\mathscr{L} = (6.8 \pm 1.0)\,\mu\text{b}^{-1}$ is used for the analysis. A K^0_S is defined to be prompt if it is produced directly in a pp collision, or if a non-weak decay of a resonance produced directly in a pp collision results in a K^0_S in the decay chain. The measurement is done in three bins of rapidity y (defined as $y = \frac{1}{2}\ln\left(\frac{E+p_z}{E-p_z}\right)$ in the range $2.5 < y < 4.0$ where $(E, \vec{p})$ is the K^0_S momentum in the pp centre-of-mass system) as function of the transverse momentum p_T of the K^0_S with respect to the beam (down to $p_T < 0.2\,\text{GeV}$). This region in y and p_T has never before been studied at this centre-of-mass energy, so the rapidity and transverse momentum regime is extended from the central region covered by the other LHC experiments into the forward direction down to very low transverse momenta.

8.1. Analysis strategy

The decay of K^0_S mesons is reconstructed in the channel $K^0_S \longrightarrow \pi^+\pi^-$ using calorimeter triggered events. Contributions from interactions with the detector material or decays of other long-lived particles such as Λ are suppressed by selecting K^0_S which point

back to the pp collision. This analysis does not attempt to disentangle diffractive and non-diffractive contributions to the K^0_S production cross section, because these contributions cannot be separated using the measured data itself. The two contributions could therefore only be estimated by using predictions from simulated events, thereby introducing a dependence on the diffraction model used in the Monte Carlo generator. Instead, some effort is made to show that trigger and reconstruction efficiencies do not differ significantly for different diffraction models, allowing any dependence on the diffraction model to be treated as a systematic effect which makes the result effectively independent of the diffraction model used, so comparisons of the result with present and future diffraction models is made a lot easier.

The analysis uses K^0_S yields $N_{K^0_S}$ (see Chapter 9), combined reconstruction and selection efficiencies ε_{rec} and trigger efficiencies $\varepsilon_{\text{trig}}$ on selected and reconstructed events in bins of the K^0_S phase space given in terms of the rapidity y and the transverse momentum p_T of the K^0_S. Given the integrated luminosity L_{int}, the K^0_S production cross section $\sigma_{pp\to K^0_S(\pi^+\pi^-)}(p_T, y)$ can be described as

$$\sigma_{pp\to K^0_S(\pi^+\pi^-)}(p_T, y) = \frac{N_{K^0_S}(p_T, y)}{\varepsilon_{\text{rec}}(p_T, y) \times \varepsilon_{\text{trig}}(p_T, y) \times L_{\text{int}}} \tag{8.1}$$

The combined reconstruction and selection efficiency will be determined in Chapter 11 from simulated events by measuring the K^0_S yield $N_{K^0_S}(p_T, y)$ in a bin of p_T and y and forming the ratio with the number of generated prompt K^0_S $N^{\text{gen}}_{K^0_S}(p_T, y)$ in that bin:

$$\varepsilon_{\text{rec}}(p_T, y) = \frac{N_{K^0_S}(p_T, y)}{N^{\text{gen}}_{K^0_S}(p_T, y)} \tag{8.2}$$

The trigger efficiency on selected events will also be taken from simulated events, see Chapter 12. With the trigger efficiency depending on event multiplicity (the higher the multiplicity, the more likely the event is to trigger), the simulated events have to be reweighted to reproduce the multiplicity distribution observed in data. Denoting the reweighted reconstructed and selected K^0_S yield $Y_{K^0_S}(p_T, y)$ and the corresponding reweighted yield on the subsample of simulated events passing the trigger $Y^{\text{trig}}_{K^0_S}(p_T, y)$, the trigger efficiency can be written as

$$\varepsilon_{\text{trig}}(p_T, y) = \frac{Y^{\text{trig}}_{K^0_S}(p_T, y)}{Y_{K^0_S}(p_T, y)} \tag{8.3}$$

The integrated luminosity L_{int} is obtained by a method which combines the beam currents measured by the LHC with beam profile measurements done with the LHCb vertex detector, allowing the luminosity to be directly calculated from first principles. This is taken from [3] and described more detail in Section 9.4.

Due to a lack of statistics, no measurement is possible in the lowest bin in p_T and y, i.e. $p_T < 200$ MeV and $2.5 < y < 3$.

Since the analysis relies heavily on simulated events to obtain figures for reconstruction/selection and trigger efficiencies, data/Monte Carlo agreement must be closely

monitored, and, where possible, verified using data-driven methods, see Chapter 10. Therefore, most of the effort done for this thesis has been spent working on this issue.

A significant fraction of K^0_S decay after leaving the LHCb vertex detector. Moreover, due to the large beam crossing angle at nominal magnetic field and larger beam sizes at $\sqrt{s} = 900\,\text{GeV}$ when compared to high energy collisions, the LHCb vertex detector was in a semi-open position with both Velo halves 15 mm away from their nominal data taking conditions to protect it against accidental beam induced damage. For the 2009 pilot run conditions, this results in more than half of the K^0_S daughter tracks in the LHCb acceptance to be only reconstructible as downstream tracks. Therefore, the analysis described in this thesis will use a combination of two oppositely charged downstream tracks to form K^0_S candidates, completely ignoring long track reconstruction. A similar analysis was done by another group on the same data set for combinations of two long tracks (see [69]) which confirms the measurements performed with downstream tracks. However, the analysis using downstream tracks presented here is more powerful (i.e. has smaller uncertainties) over most of the kinematical range investigated, so the final results which were published by LHCb in [3] are based on the downstream analysis presented in this thesis in 22 of the 24 bins in p_T and y.

The focus in this thesis is on the determination of the reconstruction and selection efficiency from simulated events, and on ensuring that these simulated events model the 2009 data as closely as possible. The luminosity measurement is taken from [3] as an external input, and the determination of the trigger efficiency based on the work done as part of this thesis was taken from [70].

8.2. Data samples used

A data sample of 13 runs taken during the 2009 pilot run with two slightly different trigger configurations was used (see Table 8.1). Since the the two trigger configurations are identical in the calorimeter settings (requiring one 2x2 cluster with more than 240 MeV transverse energy in the hadronic calorimeter or three or more hits in the 6016 cells of the Scintillating Pad Detector (SPD) at the calorimeter entrance) and the analysis uses only calorimeter triggered events, there is no need to distinguish between the two trigger settings for the purpose of yield extraction and efficiency determination, and it was checked that it is sufficient to only emulate one of the two settings for simulated events.

A sample of 10 million simulated minimum bias events was used for Monte Carlo studies[1]. Both real data and simulated events were reconstructed with Brunel v37r0 and analysed in DaVinci v25r2p3.

[1] `2009-Beam450GeV-VeloClosed15mm-MagDown-Fix1-GeoDec2009a-2009-Sim06Reco04-withTruth`, simulated with GAUSS v38r4 and digitised with BOOLE v21r1

Table 8.1: *List of runs which are used in the analysis. The first column contains the run number, the second one the number of physics events in these runs and the last column the trigger configuration key (TCK).*

run number	# physics events	TCK
63686	24391	0x1209
63687	15642	0x1209
63688	2169	0x1209
63690	20855	0x1209
63691	2074	0x1209
63713	14295	0x1209
63801	94112	0x1309
63807	75285	0x1309
63809	23465	0x1309
63813	71429	0x1309
63814	4629	0x1309
63815	11668	0x1309
63949	64179	0x1309

9. Signal selection, yield extraction and luminosity

9.1. Signal selection

The signal preselection and selection criteria are summarised in Tables 9.1 and 9.2 and exploit the long K^0_S lifetime, the displaced K^0_S decay vertex and the fact the the K^0_S must originate from the luminous region; a more detailed explanation of reasoning behind these cuts is given below. The preselection cuts are basically a loose version of the selection cuts; the sample of preselected events was written to ntuples to be able to more quickly perform the analysis, and the preselection cuts are mentioned here mainly to have them available for reference later.

Two oppositely charged downstream tracks are combined under the pion mass hypothesis to form a K^0_S candidate if they satisfy some minimum track quality requirements which include cuts on p and p_T of the tracks to suppress excessive combinatorial background from low momentum tracks. A peculiarity of the selection is that is does not use a potentially reconstructed primary vertex. The reasoning behind this is that by not using this information, the yield extraction process becomes practically independent of information in the vertex detector, which eliminates the Velo track reconstruction and vertex reconstruction efficiencies as a source of systematic errors. Instead, the z axis is used as reference. Hence, a track is considered displaced if its point of closest approach to the z axis has a minimum distance of 3 mm. Using this approach has the interesting property that it does not "distinguish" between diffractive and non-diffractive events; more traditional selections involving the reconstruction of a primary vertex are quite sensitive to the different properties of diffractive events through the dependence

cut	value
min. π p_T	25 MeV
max. track χ^2/NDF	35
max. K^0_S vertex χ^2/NDF	35
max. z of decay vertex	2200 mm
max. $\lvert z\rvert$ of point of closest approach to z axis	500 mm
min. cosine of angle between K^0_S momentum and line connecting decay vertex with $(0,0,z)$	0.995

Table 9.1.: *K^0_S preselection cuts. The first group of cuts defines the criteria for K^0_S daughter tracks, while the second group defines cuts that a combination of two oppositely charged pions have to satisfy.*

cut	value
min. π momentum	2 GeV
min. π p_T	50 MeV
max. track χ^2/NDF	25
min. π impact parameter to z axis	3mm
max. K^0_S vertex χ^2/NDF	25
max. z of decay vertex	2200 mm
max. $\|z\|$ of point of closest approach to z axis	150 mm
min. cosine of angle between K^0_S momentum and line connecting decay vertex with $(0,0,z)$	0.99995
min. K^0_S ct	5 mm

Table 9.2.: *K^0_S selection cuts. The first group of cuts defines the criteria for K^0_S daughter tracks, while the second group defines cuts that a combination of two oppositely charged pions have to satisfy.*

of the vertex reconstruction efficiency on quantities like track multiplicity and track momentum spectrum.

The resulting combination of two tracks must form a decent decay vertex which must be in front of the TT stations. Since no primary vertices are used, the K^0_S candidate is extrapolated to find the point $(0,0,z)$ at which the candidate is closest to the z axis, and this point is used instead of a primary vertex. With the first track position measurements in TT being at a z of a bit more than 2 metres, a more precise knowledge of the point of the pp interaction is not required because of the accumulating uncertainty in the extrapolation from TT. The point $(0,0,z)$ must have $|z| < 150\,\text{mm}$, i.e. it must be inside the luminous region, and the K^0_S momentum must point back to that point, thus suppressing contributions which do not come from the pp interaction itself. Moreover, the lifetime ct of the K^0_S candidate must be larger than 5 mm to further suppress combinatorial background from tracks which come directly from the interaction (the mean K^0_S lifetime is 2.6842 cm, see [7]).

9.2. Beam gas subtraction

When extracting K^0_S yields, there is an additional complication: K^0_S may also be produced when a proton from one of the two beams interacts with a residual gas molecule left inside the vertex detector vacuum. In principle, three possible configurations of non-empty events can be recorded, the most important being so-called *beam-beam* events which have proton filled bunches from both colliding beams. The remaining possibilities are given by *beam-empty* and *empty-beam* events, with the former having a filled bunch in beam 1 and an empty bunch in beam 2, while the roles are reversed for latter. Due to the direction of beam 2, empty-beam events produce practically all tracks in the negative z direction, so they do not contribute to calorimeter triggered events and can be neglected.

Since the amount of beam gas interactions depend only on bunch currents and the pressure of the residual gas inside the Velo vacuum (which is quite stable), it is possible

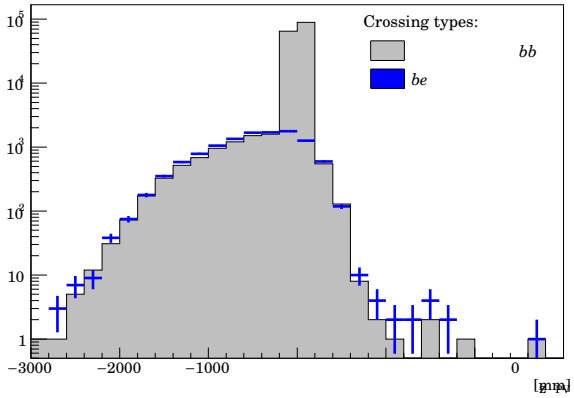


Figure 9.1: *z position of primary vertices in beam-beam (bb) and beam-empty (be) collisions. Figure from [69].*

to extract the contribution of K^0_S production from beam gas interactions in beam-beam events from beam-empty events; the correct normalisation can be obtained by comparing the number interactions outside the luminous region (which are known beam gas interactions for both event types) for the two remaining event types. Figure 9.1 shows the distribution of the z coordinates of primary vertices reconstructed in beam-beam and beam-empty events; the good agreement in shapes between beam-beam and beam-empty events outside the luminous region shows that a statistical subtraction of the beam gas contribution is feasible and can be done according to

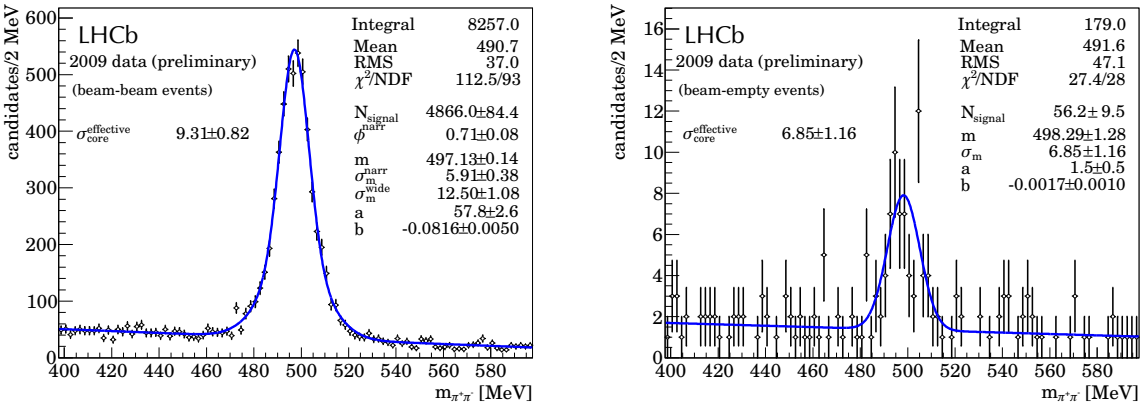
$$N = N^{beam-beam} - \beta \cdot N^{beam-empty} \tag{9.1}$$

where N is the (corrected) number of events from pp collisions, $N^{beam-beam}$ is the number of events in beam-beam configurations, $N^{beam-empty}$ is the number of events in beam-empty configurations and β is the factor which ensures correct normalisation.

The extraction of β was done on a run-by-run basis by comparing the beam gas interaction rates in beam-beam and beam-empty events outside the luminous region (see [69] for details), and a combined β for all considered runs was be calculated as $\beta = 0.9075 \pm 0.0148$. The beam gas subtraction is then performed by assigning per event weights to all quantities which are used in the remainder of this analysis; beam-beam events get a weight of 1, while beam-empty events get a weight of -0.9075.

The determination of the normalisation factor β does of course need the vertex detector, so, strictly speaking, it is no longer true that the yield extraction procedure is independent of the vertex detector. However, β can be measured without using simulated events, and there is no reason to believe that the tracking efficiency or vertex reconstruction efficiency differs for a given vertex position between beam-beam and beam-empty events. Therefore, systematic effects due to vertex detector tracking and vertexing performance will cancel in the determination of β.

Figure 9.2 shows the invariant mass distribution of the two pion combination for beam-beam and beam-empty events without the beam gas subtraction. In the remainder of this thesis, all plots will have the beam gas subtraction applied.



(a) *beam-beam events* **(b)** *beam-empty events*

Figure 9.2.: *Invariant mass distribution of the two pion system without beam gas subtraction. The fit model is explained in Section 9.3.1.*

9.3. Yield extraction

9.3.1. Fit model

The model used for the fit is a linear function for the background plus a double Gaussian for the signal (a single Gaussian may be used under certain circumstances, see Section 9.3.2 for details):

$$N(a,b,N_{\text{signal}},\phi_{\text{narr}},m,\sigma_m^{\text{narr}},\sigma_m^{\text{wide}};m_{\pi\pi}) = a+b\cdot m_{\pi\pi}+N_{\text{signal}}\cdot DG(\phi_{\text{narr}},m,\sigma_m^{\text{narr}},\sigma_m^{\text{wide}};m_{\pi\pi}) \tag{9.2}$$

Here, the terms a and b describe the linear part of the background, while N_{signal} denotes the number of signal events in the double Gaussian; $m_{\pi\pi}$ is the (measured) invariant mass of the two pion combination and m is the fit parameter representing the mass of the K^0_S. $DG(\phi_{\text{narr}},m,\sigma_m^{\text{narr}},\sigma_m^{\text{wide}};m_{\pi\pi})$ is defined as follows:

$$DG(\phi_{\text{narr}},m,\sigma_m^{\text{narr}},\sigma_m^{\text{wide}};m_{\pi\pi}) = \frac{\cos^2(\phi_{\text{narr}})}{\sqrt{2\pi(\sigma_m^{\text{narr}})^2}}\cdot\exp\left(-\left(\frac{m-m_{\pi\pi}}{2\sigma_m^{\text{narr}}}\right)^2\right)+\frac{\sin^2(\phi_{\text{narr}})}{\sqrt{2\pi(\sigma_m^{\text{wide}})^2}}\cdot\exp\left(-\left(\frac{m-m_{\pi\pi}}{2\sigma_m^{\text{wide}}}\right)^2\right) \tag{9.3}$$

σ_m^{narr} and σ_m^{wide} denote the narrow and wide contribution of the double Gaussian, while the "phase" ϕ_{narr} determines their relative importance. This description was chosen because it is more stable in low statistics fits: Due to the periodicity in ϕ_{narr} and $\cos^2(\phi_{\text{narr}})+\sin^2(\phi_{\text{narr}})=1$, it is not necessary to give bounds for the parameter ϕ_{narr}, it will stay inside the physically sensible bounds by construction.

9.3.2. Fit method

All fits to data and Monte Carlo are performed in a $\pm 100\,\text{MeV}$ window around the nominal mass of the K^0_S resonance, $m_{PDG} = (497.614 \pm 0.024)\,\text{MeV}$, as given in [7]. A MINUIT

χ^2 fit is used (see [71]); the estimated number of entries in a bin is obtained from the integral of the fit function over that bin. In low statistics regions of the plot, several bins may be combined to form an enlarged bin for the fit. This combining of bins is applied if the bin has less than ten entries, and no more than five consecutive bins are combined to form an enlarged bin. In this way, the fit becomes more stable in low statistics regions, while the fine binning in the more populated regions near the peak is retained. These combined bins are only used in the fit, they are not visible in the plots.

As explained in Section 9.3.1, a fit is performed to the two-pion invariant mass distribution using a linear background plus a double (single) Gaussian; the choice between double and single Gaussian is made according to the following criteria:

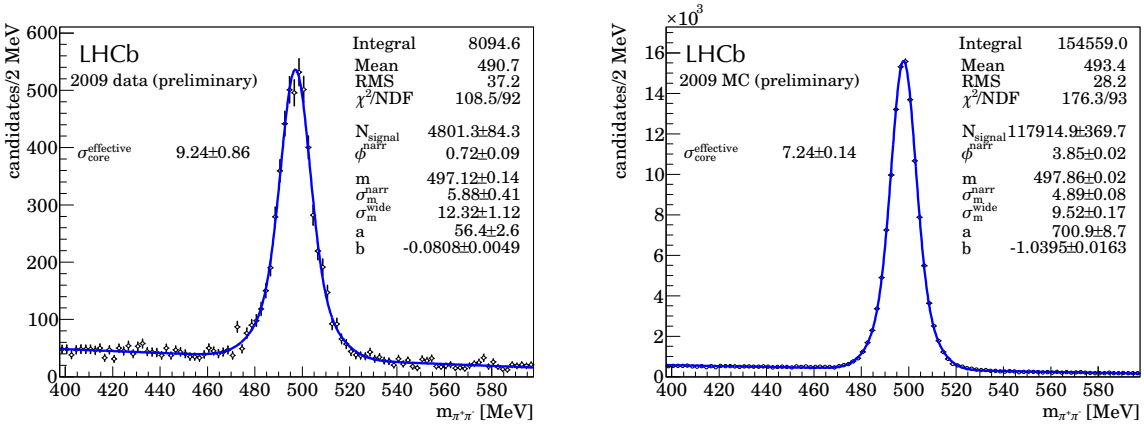
- Both single and double Gaussian fits are performed. If the single Gaussian fit finds less than 100 signal events, the result from the single Gaussian fit is quoted.
- If the fraction of events in the narrow or wide contribution of the double Gaussian reaches 0 or 1, the single Gaussian result is quoted.
- If the widths of narrow and wide contributions of the double Gaussian are very similar ($\Delta\sigma < 0.5\,\mathrm{MeV}$), the single Gaussian result is quoted.
- If the width of the wide contribution to the double Gaussian is more than 15 MeV, the single Gaussian result is quoted to avoid describing the background with a very wide Gaussian.
- If the width of the narrow contribution to the double Gaussian is less than 3 MeV, the result of the single Gaussian fit is quoted to avoid describing statistical fluctuations of the signal with a very narrow Gaussian.
- Otherwise, the result of the double Gaussian model is quoted.

From the list of parameters in the plot, it will be clear which of the two alternatives was used in the fit.

Figure 9.3 shows the fits to the full samples of data and simulated events, Table 9.3 summarises the fit results in tabular form. It is obvious that the width of the K^0_S peak in data is wider than in simulated events; alignment and calibration of the tracking detectors have not reached their final accuracy. Unfortunately, further improvement in this direction is unlikely because there is little interest in the collaboration to improve the understanding of the 2009 pilot run data — the data sample is small, and priorities have shifted towards newer data. The effects causing the broadening of the peak in data with respect to simulation also reduce the hit finding and tracking efficiencies; this will be discussed later.

9.3.3. Background from Λ decays

With selection criteria used, the main source of non-combinatorial background is $\Lambda \longrightarrow p\pi$ where the proton is assigned the pion mass hypothesis; the resulting shape can be found in Figure 9.4. It can be seen that this background does not peak and is roughly linear. It can therefore be absorbed in the (linear) background part of the fit model without disturbing the extraction of the K^0_S yield.



(a) *2009 data* **(b)** *Monte Carlo sample*

Figure 9.3.: *Invariant mass of K^0_S candidates for 2009 data and simulation. The difference in mass resolution between data and simulation is due to residual imperfections in detector calibration and alignment.*

Table 9.3: *Fit parameters to mass of K^0_S candidates in data and MC.*

	2009 data	simulation
N	4801± 84	117910± 370
σ_1 [MeV]	5.9± 0.4	4.9± 0.1
σ_2 [MeV]	12.3± 1.1	9.5± 0.2
m [MeV]	497.1± 0.1	497.9± 0.1
ϕ^{narr}	0.72±0.09	3.85±0.02
$f_{narr} = \cos^2(\phi^{narr})$	0.57	0.58
slope	-0.08±0.01	-1.04±0.02
a	56.3± 2.6	700.9± 8.7
σ_{av} [MeV]	9.2	7.2

9.3.4. Yields in bins of p_T and y

The following bins in p_T and y were chosen for this analysis as a compromise between sufficient statistics in each bin and sufficiently fine granularity to make the measurement interesting as a potential input for theory models:

- $0 < p_T < 200$, $200 < p_T < 400$, $400 < p_T < 600$, $600 < p_T < 800$, $800 < p_T < 1000$, $1000 < p_T < 1200$, $1200 < p_T < 1400$, $1400 < p_T < 1600$ MeV
- $2.5 < y < 3$, $3 < y < 3.5$, $3.5 < y < 4$

The distribution of K^0_S candidates in rapidity and transverse momentum is shown in Figure 9.5; the distribution of the selected K^0_S daughter pions is given in Figure 9.6.

The fits for each bin are shown in Figures 9.7, 9.8 and 9.9. Table 9.4 summarises the extracted yields and the quality of each fit. These yields will enter the cross section determination.

To cross check the results, Table 9.5 shows the yields extracted using sideband subtraction with a linear background model. The signal region is the region with

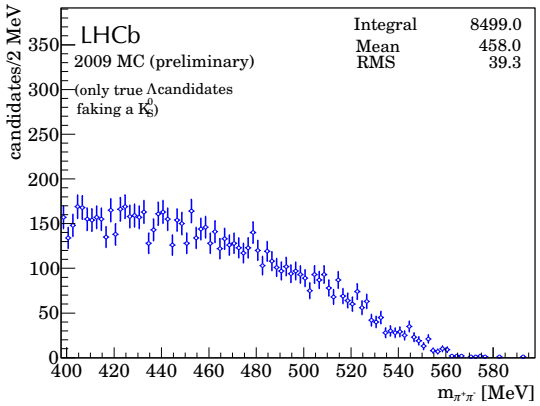
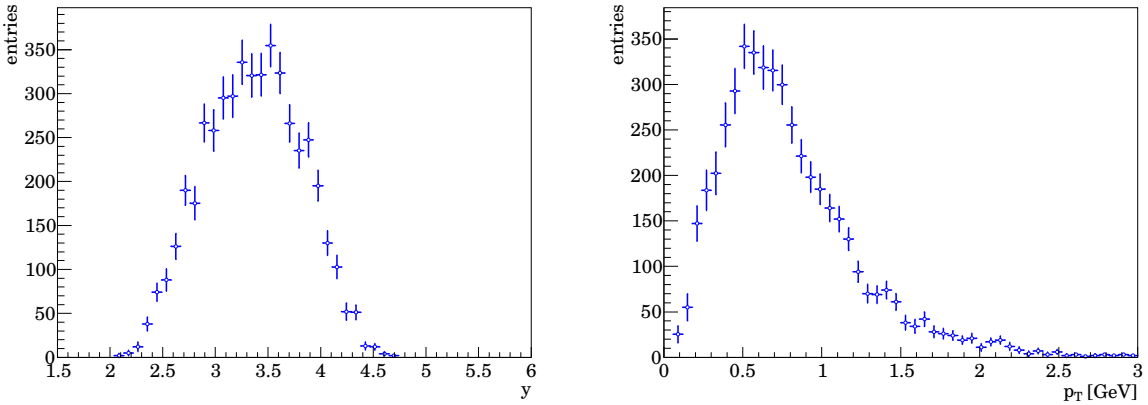
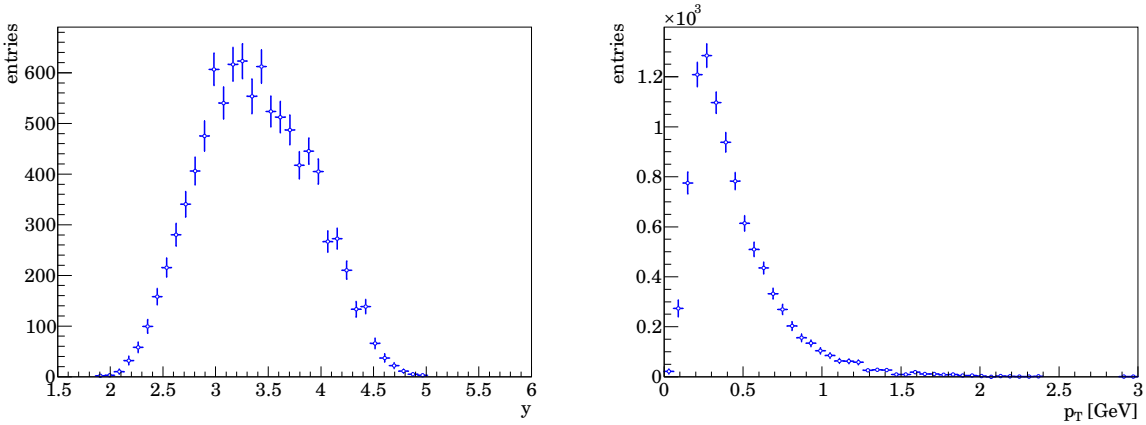


Figure 9.4: *Invariant mass distribution for Λ decays reconstructed with the ππ mass hypothesis.*



(a) *y of* K^0_S **(b)** p_T *of* K^0_S

Figure 9.5.: *Distribution of* K^0_S *candidates in rapidity (a) and transverse momentum (b) in data.*



(a) *y of* π *for* K^0_S *daughters*

(b) p_T *of* π *for* K^0_S *daughters*

Figure 9.6.: *Distribution of the charged daughter pions from* K^0_S *candidates in rapidity (a) and transverse momentum (b) in data.*

$|m_{\pi\pi} - m_{PDG}| < 50\,\text{MeV}$ (with $m_{\pi\pi}$ being the measured invariant mass of the two pion combination and m_{PDG} being the world-average K^0_S mass given in [7]. The sideband region is the portion of the interval $|m_{\pi\pi} - m_{PDG}| < 100\,\text{MeV}$ which does not include the signal region. Since the yields extracted with sideband subtraction agree with the results obtained with fitting within their respective uncertainties and no systematic pattern is visible, no systematic uncertainty will be assigned due to this difference. (Contributions from variation of the fit model are discussed later.)

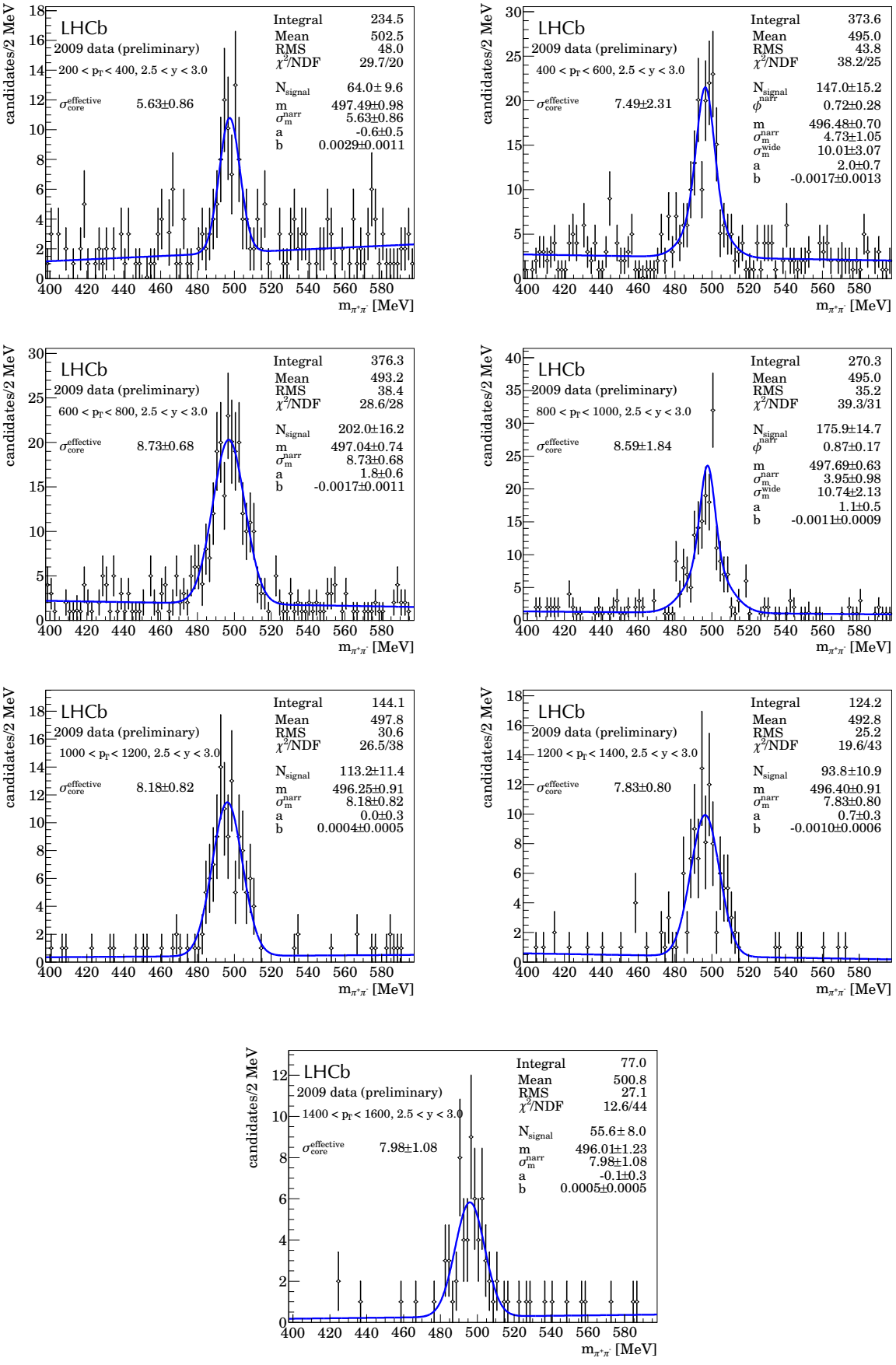


Figure 9.7.: *K^0_S candidate mass distribution for 2009 data in different p_T bins with* $2.5 < y < 3.0$.

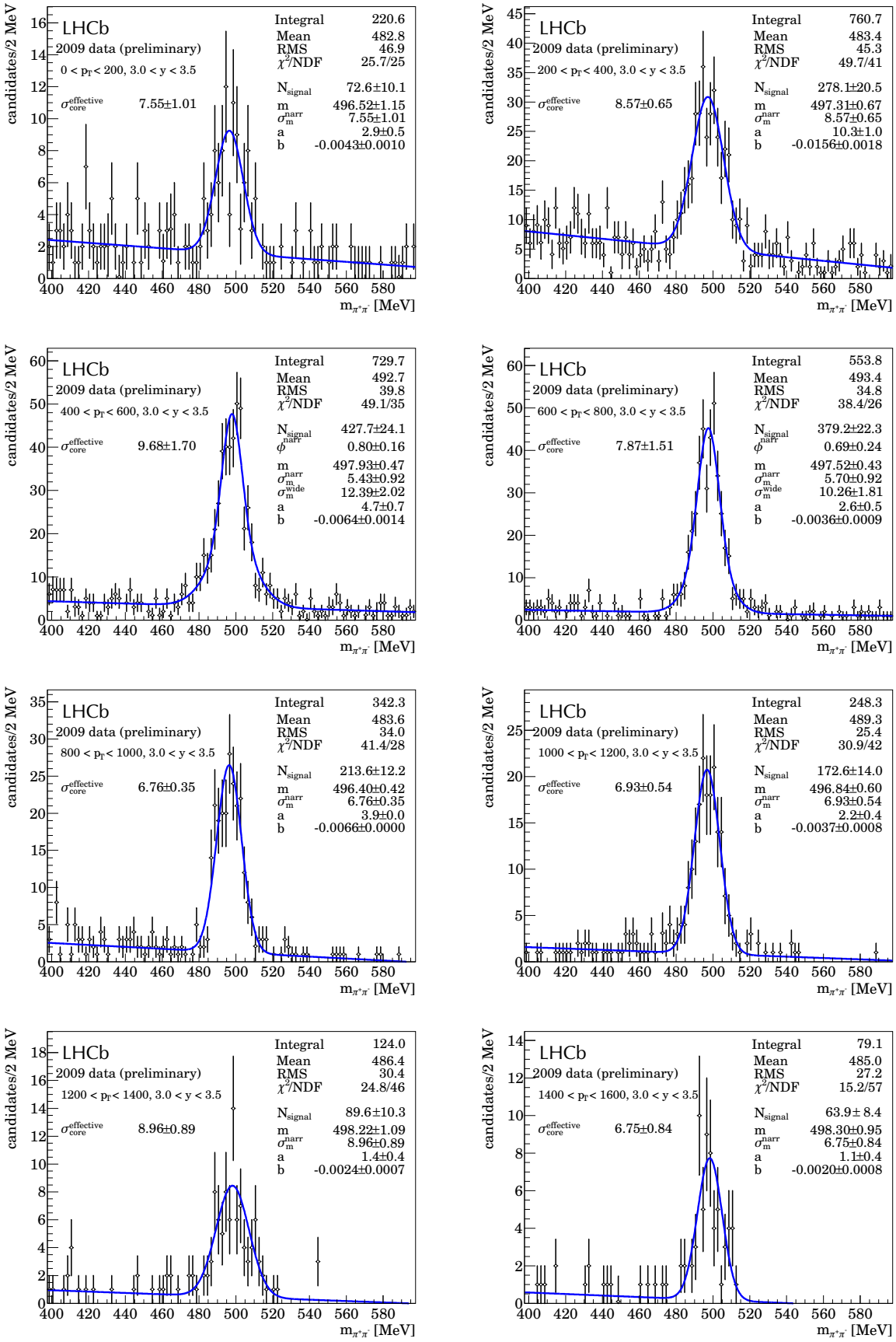


Figure 9.8.: *K^0_S candidate mass distribution for 2009 data in different p_T bins with $3.0 < y < 3.5$.*

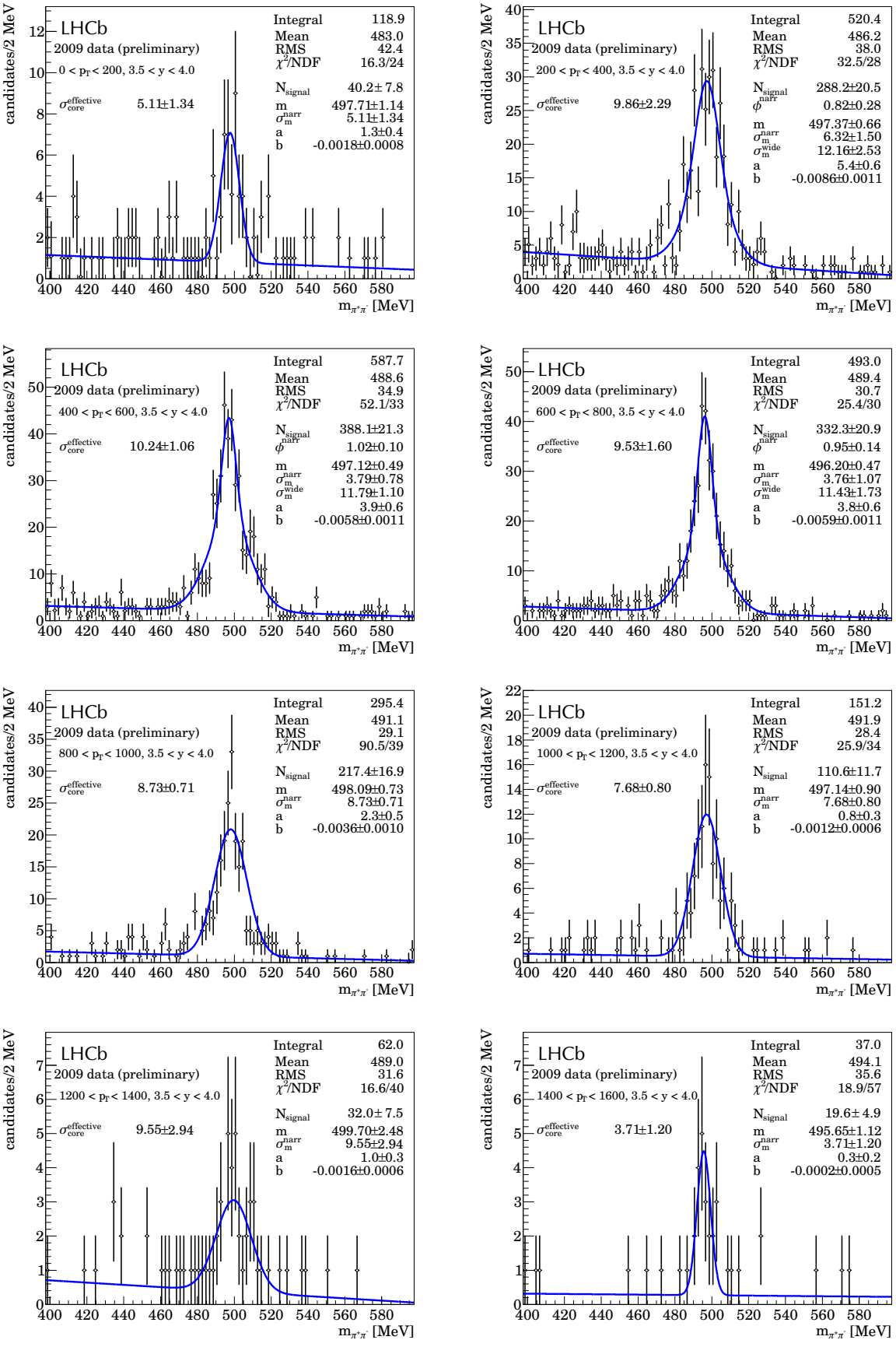


Figure 9.9.: *K^0_S candidate mass distribution for 2009 data in different p_T bins with* $3.5 < y < 4.0$.

$p_T[MeV]/y$	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	–	72.6±10.1	40.2± 7.8
200 – 400	64.0± 9.6	278.1±20.5	288.2±20.5
400 – 600	147.0±15.2	427.7±24.1	388.1±21.3
600 – 800	202.0±16.2	379.2±22.3	332.3±20.9
800 – 1000	175.9±14.7	213.0±16.1	217.4±16.9
1000 – 1200	113.2±11.4	172.6±14.0	110.6±11.7
1200 – 1400	93.8±10.9	89.6±10.3	32.0± 7.5
1400 – 1600	55.6± 8.0	63.9± 8.4	19.6± 4.9

(a)

$p_T[MeV]/y$	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	–	16 / 24	26 / 25
200 – 400	30 / 20	50 / 41	32 / 28
400 – 600	38 / 25	50 / 35	52 / 33
600 – 800	29 / 28	38 / 26	25 / 30
800 – 1000	39 / 31	41 / 28	91 / 39
1000 – 1200	27 / 38	31 / 42	26 / 34
1200 – 1400	20 / 43	25 / 46	17 / 40
1400 – 1600	13 / 44	15 / 57	19 / 57

(b)

Table 9.4.: *Number of signal candidates in 2009 data in p_T and y bins (a) and the corresponding fit quality χ^2/NDF (b). The different number of degrees of freedom is related to the chosen fit model (single or double Gaussian) and the combining of low statistics bins in the histogram.*

Table 9.5: *Number of signal candidates in 2009 data in p_T and y bins when using sideband subtraction to extract the signal yield.*

$p_T[MeV]/y$	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	-	66± 11	48± 15
200 – 400	66± 16	261± 28	285± 24
400 – 600	147± 20	408± 28	382± 25
600 – 800	210± 20	361± 24	346± 23
800 – 1000	178± 17	204± 19	229± 17
1000 – 1200	108± 12	208± 16	120± 12
1200 – 1400	102± 11	92± 11	42± 8
1400 – 1600	59± 9	57± 9	25± 6

9.4. Luminosity determination

The luminosity for the 2009 pilot run was determined by the LHCb collaboration and is used as an external input in this analysis. The principle of measurement and the result is briefly described here, details can be found in [3].

9.4.1. Principle

For head-on relativistic beams, the instantaneous luminosity of two colliding bunches can be written as

$$L = 2cn_1n_2f \int \rho_1(x,y,z,t)\rho_2(x,y,z,t)\,dx\,dy\,dz\,dt \tag{9.4}$$

Here n_1 and n_2 are the number of protons in the two colliding bunches, $f = 11.245\,\text{kHz}$ is the LHC revolution frequency and $\rho_1(x,y,z,t)$ and $\rho_2(x,y,z,t)$ are the densities of the bunches (each bunch $i = 1,2$ normalised such that $\int \rho_i(x,y,z,t)\,dx\,dy\,dz = 1$ at all times t). The overlap integral in equation 9.4 extends for the duration of a bunch crossing. If the beams are brought into collision with a beam crossing angle, the equation needs to be modified (θ is half the beam crossing angle):

$$L = 2cn_1n_2\cos^2(\theta)f \int \rho_1(x,y,z,t)\rho_2(x,y,z,t)\,dx\,dy\,dz\,dt \tag{9.5}$$

The idea is to measure the beam profiles of the bunches in both beams with the LHCb vertex detector by observing the distribution of vertices. Together with the bunch current measurements performed by the machine, the luminosity can be calculated by adding up the contributions from all pairs of colliding bunches.

The beam crossing angle is limited to the horizontal plane because the LHCb dipole magnet deflects mostly in direction of the x axis. Assuming that the bunch density profiles factorise, i.e. that $\rho(x,y,z,t) = \rho_x(x,t)\cdot\rho_y(y,t)\cdot\rho_z(z,t)$ holds, the beam profile can be extracted from distribution of vertices caused by interactions of protons from each bunch with beam gas molecules which are taken from the non-luminous regions for both beams ($-1000\,\text{mm} < z < -200\,\text{mm}$ or $200\,\text{mm} < z < 1000\,\text{mm}$, respectively). Figure 9.10 shows the vertex distributions obtained.

Using Gaussians to approximate the distributions observed and assuming the bunch density profile along z to be the same for both beams, equation 9.5 transforms into

$$L = \frac{n_1n_2f}{2\pi\sqrt{1+2(\theta\sigma_z)^2/(\sigma_{1x}^2+\sigma_{2x}^2)}} \prod_{j=x,y} \frac{1}{\sqrt{\sigma_{1j}^2+\sigma_{2j}^2}} \exp\left(-\frac{1}{2}\frac{(\mu_{1j}-\mu_{2j})^2}{\sigma_{1j}^2+\sigma_{2j}^2}\right) \tag{9.6}$$

where the means μ_{ij} and widths σ_{ij} ($i = 1,2$ and $j = x,y$) are extracted from the respective non-luminous region of Figure 9.10. The half crossing angle θ can also be extracted from there. σ_z is extracted from the longitudinal distribution of pp collision vertices.

To further constrain bunch observables, it is possible to exploit the higher statics in pp collisions by measuring the position μ_j and width σ_j ($j = x,y$) of the luminous region ($|z| < 150\,\text{mm}$) and relating them to the less precise measurements of beam widths and positions outside the luminous region:

$$\mu_j = \frac{\mu_{1j}\sigma_{2j}^2+\mu_{2j}\sigma_{1j}^2}{\sigma_{1j}^2+\sigma_{2j}^2} \qquad \text{and} \qquad \sigma_j^2 = \frac{\sigma_{1j}^2\sigma_{2j}^2}{\sigma_{1j}^2+\sigma_{2j}^2} \qquad (j = x,y) \tag{9.7}$$

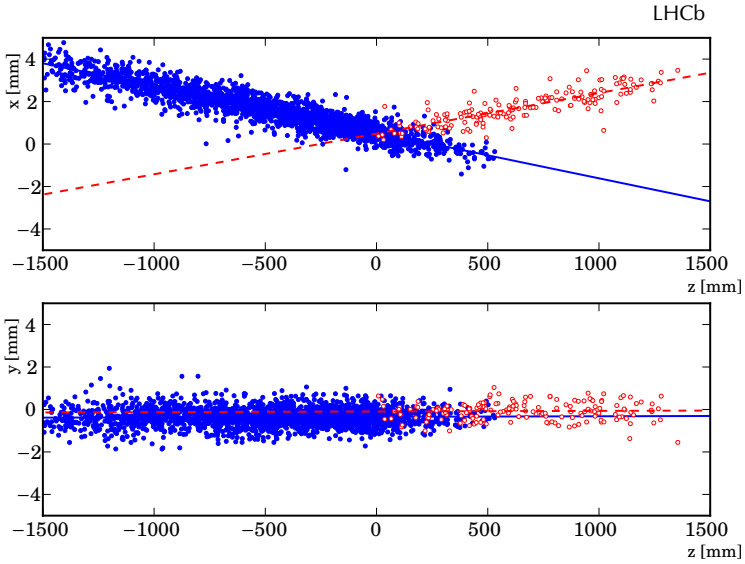


Figure 9.10.: *Vertex distributions in the horizontal (top) and vertical (bottom) planes for beam 1 (blue filled circles) and beam 2 (red open circles). The fits (solid blue for beam 1 and dashed red for beam 2) are straight line fits to the distribution. Plot from [3].*

Due to the much higher statistics of pp collisions compared to beam-gas collisions, these constraints provide the most significant input to the calculation of the overlap integrals.

Since the widths of the bunch profiles σ_{ij} ($i = 1,2$, $j = x,y$) are of the same order of magnitude as the vertex resolution, the influence of the vertex resolution must be corrected for by unfolding (see [3] for details) before they are substituted into equation 9.6. For this, the vertex resolution must be measured on data. This can be accomplished by splitting the set of tracks which make up a primary vertex in half and using each half to reconstruct two independent vertices. The distribution of their relative distance allows to draw conclusions about the vertex resolution.

9.4.2. Results

By analysing the 2009 data sample in the manner indicated above, one arrives at an integrated luminosity of $L_{int} = 6.8 \pm 1.0\,\mu\text{b}^{-1}$. The largest contribution to the total uncertainty comes from the relatively poor knowledge of the bunch currents (12%), to be followed by the uncertainty of beam widths (5%), relative beam positions (3%) and crossing angle (1%). The large uncertainty in the bunch currents is expected to improve with higher intensity beams.

The luminosity was measured for one benchmark fill comprising about 25% of the the

2009 pilot run data. The total luminosity can then be obtained by cross-calibrating the luminosity with the number of pp interaction vertices. This calibration is then used to obtain the integrated luminosity for the remainder of the data sample. Three shorter runs were used to cross-check the calibration, and they give consistent results.

10. Track reconstruction efficiency

For the determination of the K^0_S production cross section, the knowledge of the track reconstruction efficiency is of paramount importance, especially since it is the contribution to the total efficiency which differs most between data and simulation.

The combined K^0_S selection and reconstruction efficiency will be determined from simulated events in Chapter 11 and includes the track reconstruction efficiency. For this to work, the agreement between data and simulation must be good, and the systematic uncertainties associated with any remaining discrepancies must be estimated. This chapter presents a method to improve the agreement between data and simulation, and provides input to estimate the systematic uncertainties due to track reconstruction efficiency later on.

All simulated events used in this analysis will be treated with the method developed in this Chapter to ensure good data/Monte Carlo agreement. As such, it is the base for all efficiency determinations which enter the analysis, and the analysis could not have been done without it.

10.1. Preparation of simulated events

Data/Monte Carlo agreement for the 2009 pilot run is an issue; the alignment and calibration constants available for this data sample cause different hit finding efficiencies in data and simulated events. To bring data and simulated events to better agreement, simulated events are "post-processed" to account for these facts. While the data sample has been reconstructed with the best set of alignment constants available, residual misalignments still cause hits to be lost. Due to a flaw in the pattern recognition software in use at the time (TsaSeeding), some more hits are lost because the search windows in the pattern recognition do not account for known misalignments under all circumstances. The main aim of this section is to derive a method which is able to effectively mimic this behaviour on a sample with simulated events reconstructed with perfect alignment.

To save some processing time, most of the studies done in this chapter were done on subsample of the 10 million simulated events which was a bit larger than 1 million events; it will be stated explicitly for which studies the full sample was used.

10.1.1. Dead channels

For Inner Tracker and TT, a list of dead channels is already in the geometry description, therefore these dead channels are accounted for correctly already at the time simulated events are generated and digitised. For the Outer Tracker however, such a treatment was missing at the time the simulated events were generated. For this reason, a list of dead modules, readout chips and high voltage channels in the Outer Tracker during the 2009 pilot run was compiled. With this list, it is possible to introduce the corresponding

inefficiency also for simulated events. This is done by flagging affected hits with an "ignore" flag, causing the pattern recognition algorithms (and subsequent stages of processing) to treat these hits as non-existent. [1].

Compared to the sample as simulated and digitised, this "hit dropping" in the Outer Tracker reduces the K^0_S yield in simulated events from 13422 ± 126 to 12974 ± 124, a reduction by 3.3%.

As a cross check, the "hit dropping" was also applied once to the sample from the 2009 pilot run. The K^0_S yield did not change, indicating that the list of dead channels in the Outer Tracker does not accidentally include channels which were in fact working well in the real detector.

10.1.2. Remaining inefficiencies

The remaining hit inefficiencies present in data but not in the simulation can be due to one of several reasons:

- The detection efficiency of the detector in the simulation might not reflect the situation in data.
- Detector calibration in data might not be optimal.
- The alignment has not reached its final precision, causing hits to be absent from tracks in data.
- There might be artifacts of the pattern recognition which behave differently in data and simulation.

All of these phenomena result in hits being absent from tracks in data (or may cause the track to not be found at all) whereas one would expect to see the hit in simulated events. To be able to simulate this effect for simulated events, it is useful to define an *effective hit finding efficiency* to quantify these effects. More specifically, we define this effective hit efficiency for each layer of the tracking subdetectors (e.g. in layer 3 of the OT) as

$$\varepsilon^{\text{eff}}_{\text{hit}}(\text{OT layer 3}) = \frac{\text{\# tracks with an expected hit in OT layer 3 is actually found on track}}{\text{\# tracks with an expected hit in OT layer 3}} \tag{10.1}$$

A hit is expected in a layer of a detector if the track goes through the sensitive area of that detector layer. Using this definition, we can then determine a fraction ε_{rel} of hits to drop in each layer of each subdetector in our Monte Carlo sample by forming

$$\varepsilon_{\text{rel}} = \frac{\varepsilon^{\text{eff, data}}_{\text{hit}}}{\varepsilon^{\text{eff, MC}}_{\text{hit}}} \tag{10.2}$$

This is accomplished by randomly flagging the correct fraction of hits (using the same mechanism that was used to introduce dead modules in the Outer Tracker above) in each

[1] Technically, this hit dropping is accomplished using the `FlagHitsForPatternReco` algorithm. This means re-reconstructing the sample of simulated events with BRUNEL and re-running DAVINCI to run the preselection and write analysis NTuples. For completeness, the corresponding options are given in the

subdetector layer at the beginning of the reconstruction stage, again causing subsequent pattern recognition algorithms to not pick up these hits.

To avoid polluting the measurement with fake tracks, only K^0_S daughter tracks are used, and the background contribution is suppressed by using sideband subtraction.

The pattern recognition process introduces correlations among the effective hit finding efficiencies of different layers, therefore it is expected that applying such a correction is an iterative procedure. A single step is not sufficient because it will over- or undercorrect in some layers because the measurement of ε_{rel} does not take into account the correlations between layers. The final correction factor to be applied to each layer is given by the the product of values of $\varepsilon^{(i)}_{\text{rel}}$ in each iteration i (this study will use three iterations):

$$\varepsilon^{\text{final}}_{\text{rel}} = \varepsilon^{(1)}_{\text{rel}}\varepsilon^{(2)}_{\text{rel}}\varepsilon^{(3)}_{\text{rel}} \tag{10.3}$$

Such a correction is applied per layer and per subdetector, with the Inner Tracker split into two units consisting of left and right boxes together (called "A/C side boxes" from the access and cryo side of the detector) and top and bottom boxes together. The reason for this split is the observation of significant differences between A/C side and top/bottom boxes.

Figures 10.1, 10.2, 10.3 a and 10.4 show $\varepsilon^{\text{eff}}_{\text{hit}}$(subdetector, layer) for data and simulated events before any correction is applied, after the first, the second and the third iteration. The dead OT channels are already included in the plots before the first iteration. The correction factor applied in each iteration is obtained by applying the definition of ε_{rel} from above. Especially in the IT (cf. Figure 10.1), the agreement was poor with discrepancies of up about 25% in effective hit finding efficiencies in one layer. For the IT, the main reason for this large discrepancy is the fact that the pattern recognition in use at the time did not properly account for known misalignments (small rotations) in its search window determinations; the windows were widened as far as was possible, but the effect is still obvious. This was not found in simulations because the detector simulation could not properly handle misaligned geometries under all circumstances. This means that effective hit finding efficiencies are a valuable tool to catch such discrepancies before they can do harm in an analysis.

After three iterations, the data/Monte Carlo agreement was deemed satisfactory, and no further iterations were performed. In the OT (TT), about 98% (99%) of the hits were kept, in the IT the situation is much worse, and some layers had as much as 16% of their hits dropped (see Appendix E).

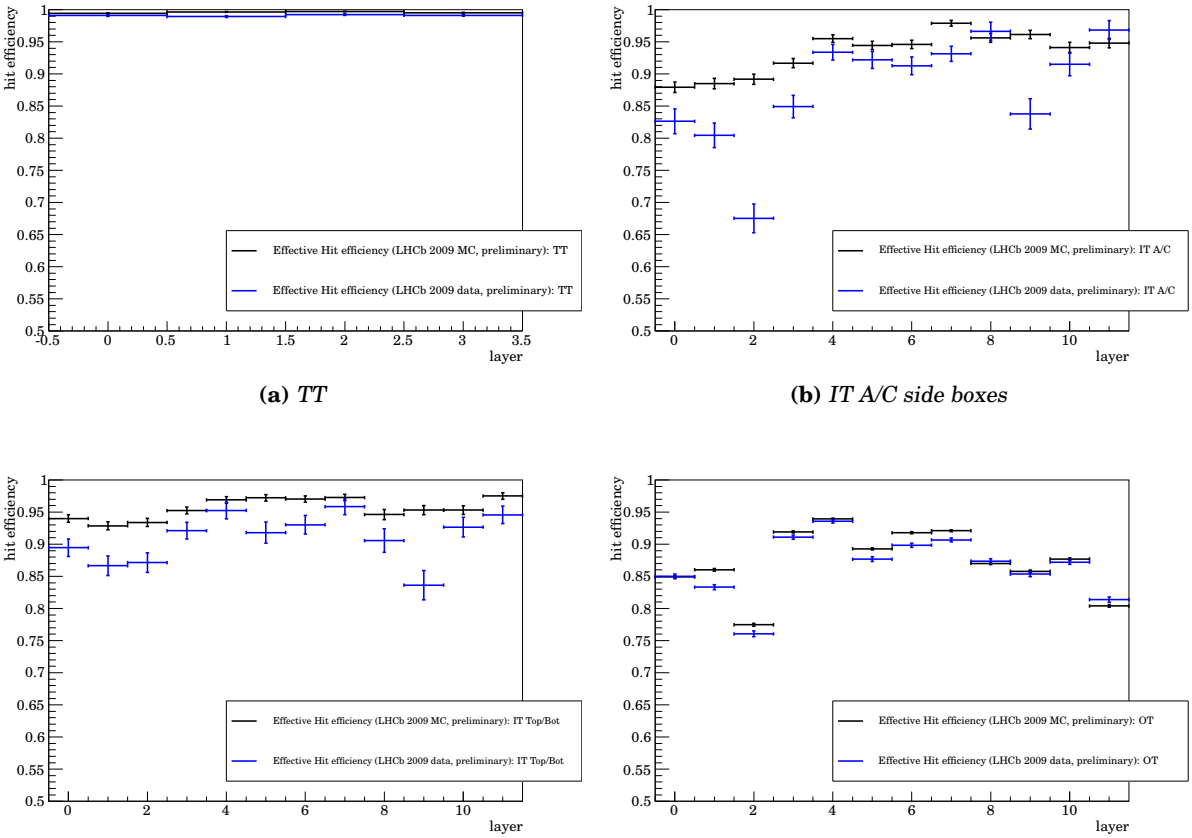
To illustrate that taking only the first million Monte Carlo events is justified, Figure 10.5 shows the effective hit efficiencies for the data and the full 10 million sample. The result hardly differs from the one obtained with only one million of Monte Carlo events.

10.2. Cross checks

10.2.1. Number of hits on a track

To check if the method causes problems, it is a good idea to look at track level quantities related to the hit content of a track other than the effective hit finding efficiency per

appendix E.



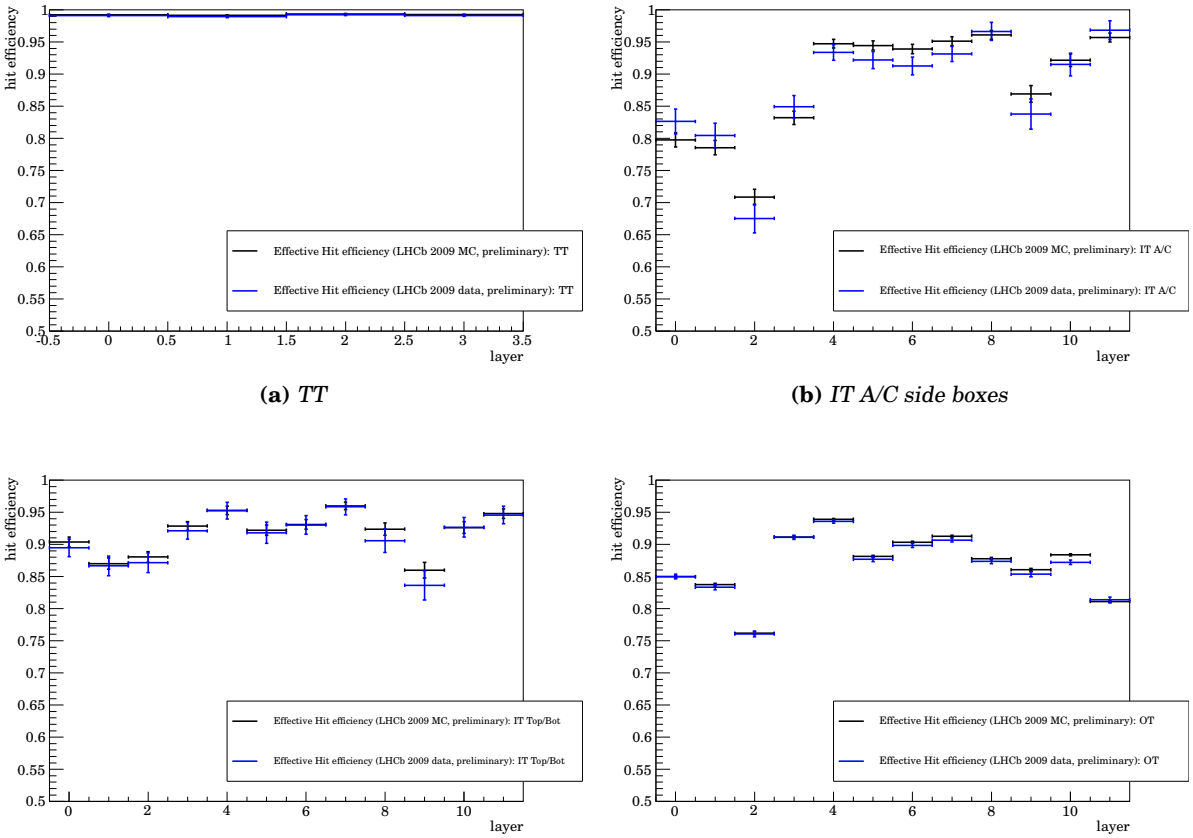
(a) *TT*

(b) *IT A/C side boxes*

(c) *IT top/bottom boxes*

(d) *OT*

Figure 10.1.: *Effective hit efficiencies per subdetector and layer for data and Monte Carlo before applying any correction. The ratio of the blue curve over the black one is the per-layer efficiency correction to apply in the first iteration. The large discrepancies (especially in the IT) are due to a flaw in the pattern recognition which renders it susceptible to misalignments in data (even if they are known and used during reconstruction).*



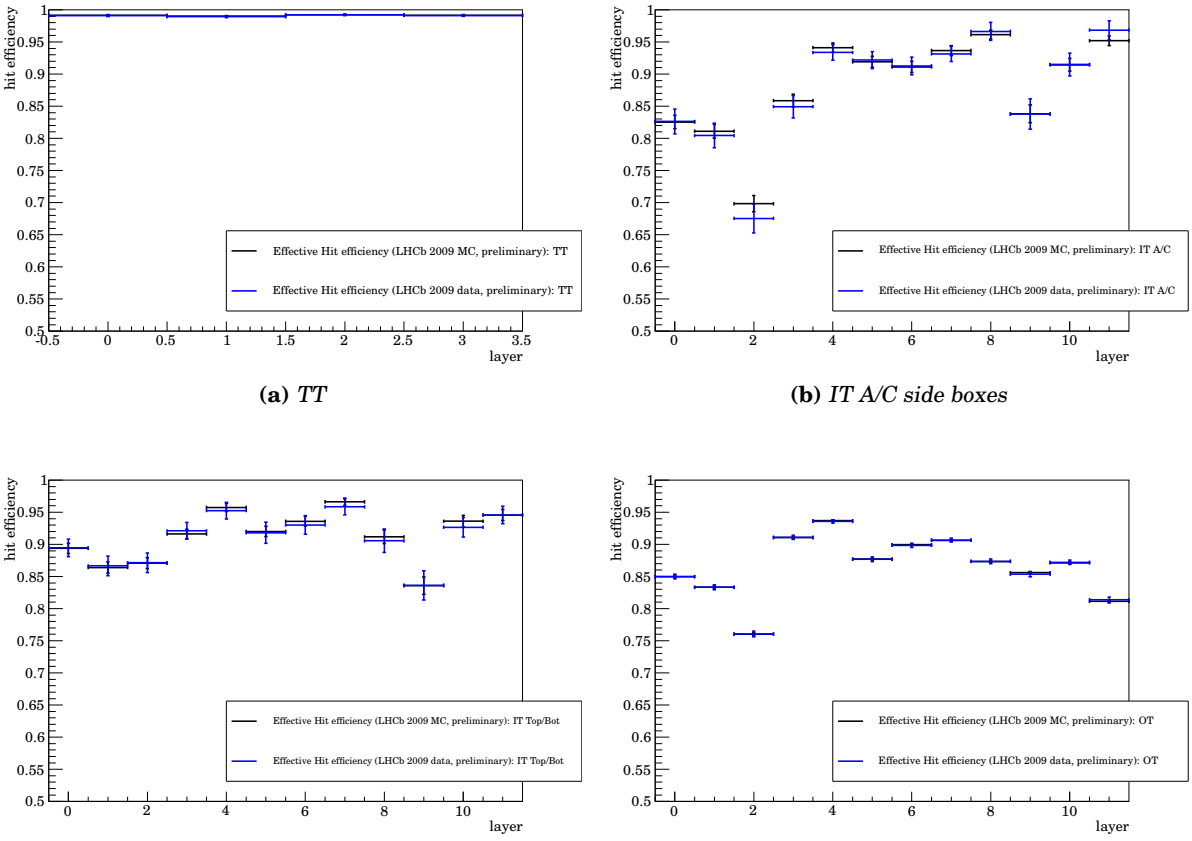
(a) *TT*

(b) *IT A/C side boxes*

(c) *IT top/bottom boxes*

(d) *OT*

Figure 10.2.: *Effective hit efficiencies per subdetector and layer for data and Monte Carlo after applying the correction of the first iteration. The ratio of the blue curve over the black one is the per-layer efficiency correction to apply in addition to the one obtained in the first iteration.*



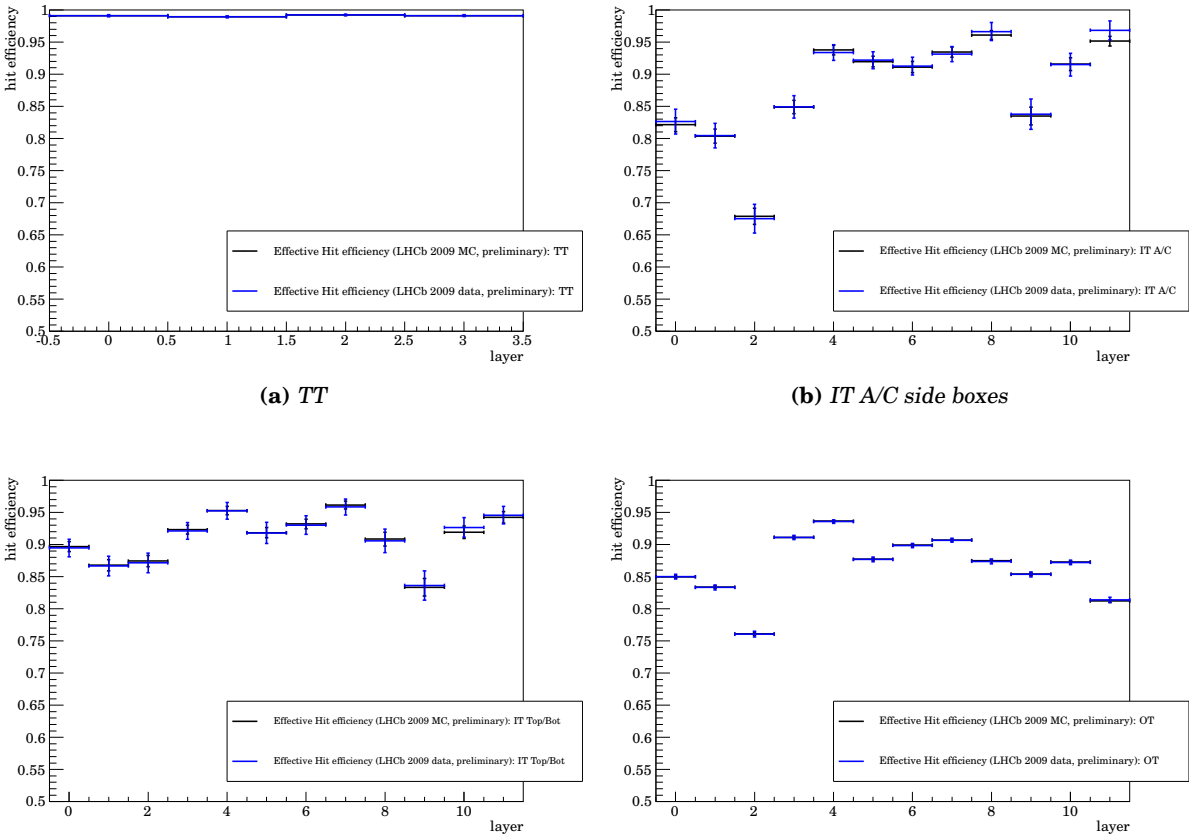
(a) *TT*

(b) *IT A/C side boxes*

(c) *IT top/bottom boxes*

(d) *OT*

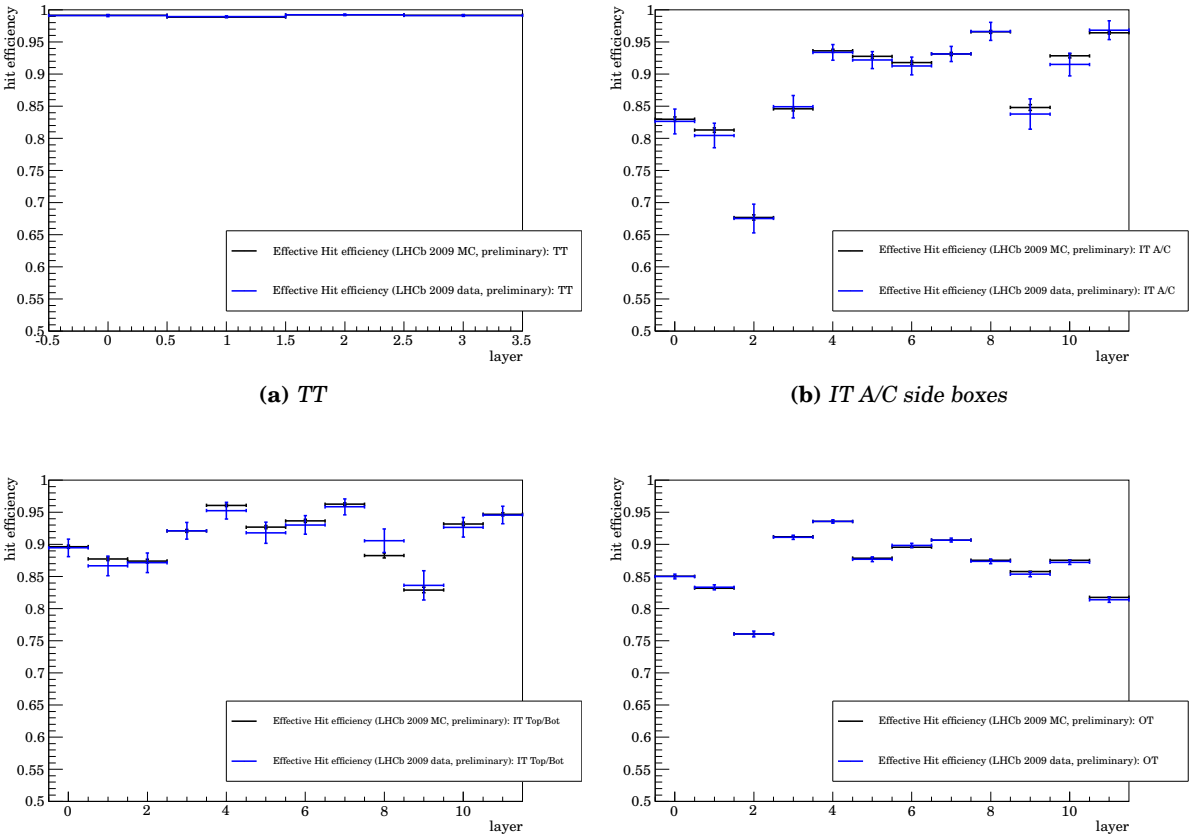
Figure 10.3.: *Effective hit efficiencies per subdetector and layer for data and Monte Carlo after applying the correction of the second iteration. The ratio of the blue curve over the black one is the per-layer efficiency correction to apply in addition to the ones obtained in the first and second iterations.*



(a) *TT* **(b)** *IT A/C side boxes*

(c) *IT top/bottom boxes* **(d)** *OT*

Figure 10.4.: *Effective hit efficiencies per subdetector and layer for data and MC after three iterations.*



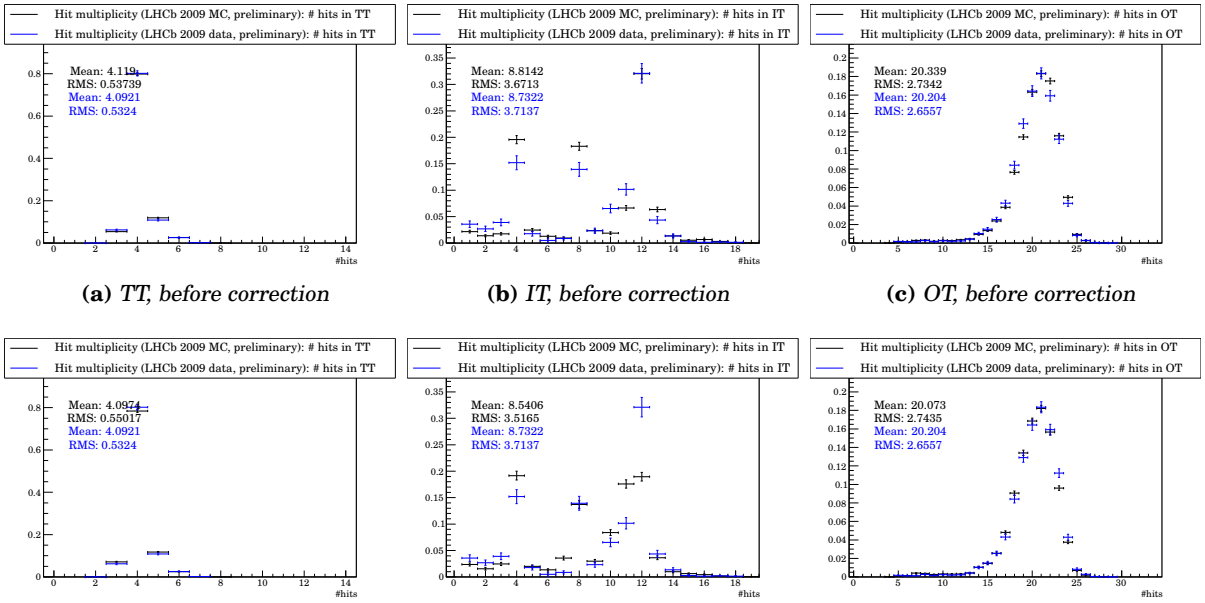
(a) *TT*

(b) *IT A/C side boxes*

(c) *IT top/bottom boxes*

(d) *OT*

Figure 10.5.: *Effective hit efficiencies per subdetector and layer for data and MC after three iterations; this plot was made using the full 10 million Monte Carlo events. There is no significant difference to the result obtained with only one million Monte Carlo events.*



(a) *TT, before correction* **(b)** *IT, before correction* **(c)** *OT, before correction*

(d) *TT, after correction* **(e)** *IT, after correction* **(f)** *OT, after correction*

Figure 10.6.: *Number of hits per track in each subdetector for data and Monte Carlo, before and after correction.*

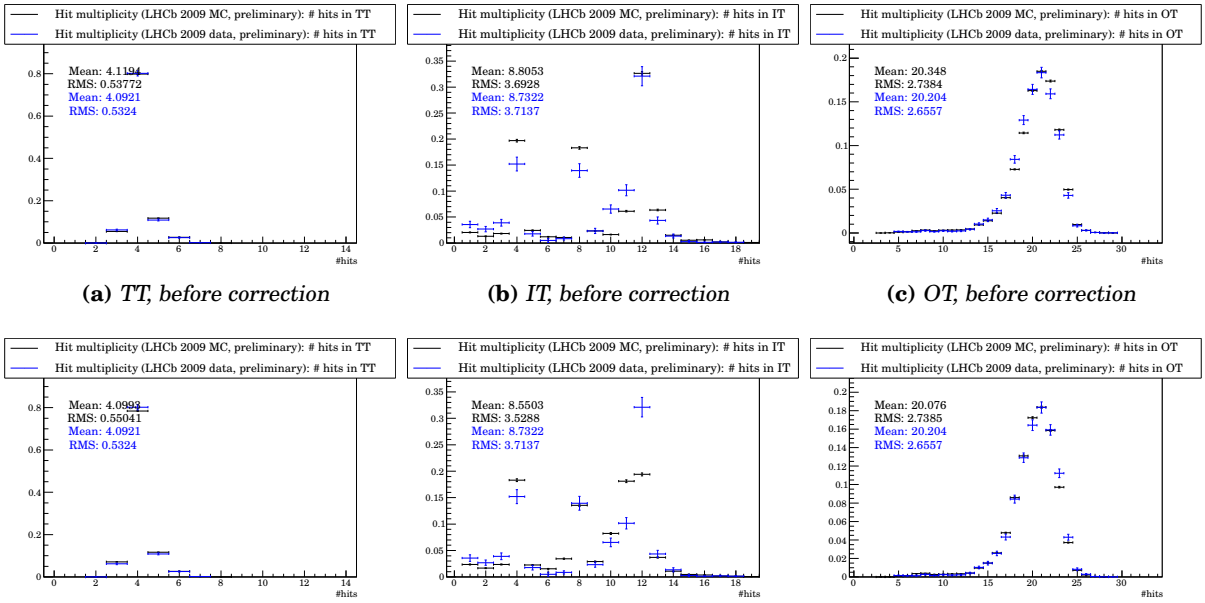
layer. One such quantity is the number of hits per track and subdetector. If the method works well, the agreement in the number of hits per tracks between data and simulation should not become worse.

The histograms in Figure 10.6 are filled with $K^0_S \to \pi^+\pi^-$ daughter tracks (i.e. each K^0_S candidate contributes two entries), and are properly background subtracted[2]. Since the focus is on downstream tracks in this analysis, the method is performed with downstream tracks. Despite the large difference of hit content in the individual layers of the subdetectors before the correction, the average number of hits agree rather well already before the corrections. The correction does not make the agreement significantly worse, so no apparent problem is spotted when looking at the subdetectors as a whole. Figure 10.7 shows that the situation on the full sample of 10 million simulated events is not substantially different from the results obtain using only the first million events.

10.2.2. Number of hits on a track in different regions of phase space

Of course, it is still possible that different regions of a subdetector have too many or too few hits dropped in a manner which leaves the average number of hits per track in a subdetector unchanged. To check against such an eventuality, one can check the ratio of the number of hits on a track in data over the corresponding number in simulated events in different regions of phase space. If there are no large variations over phase space, these ratios should come out close to 1.0.

[2]The signal region extends from $m_{PDG} - 50\,\text{MeV}$ to $m_{PDG} + 50\,\text{MeV}$ where m_{PDG} is the K^0_S mass given in [7], the sideband region is the remainder of the plot.



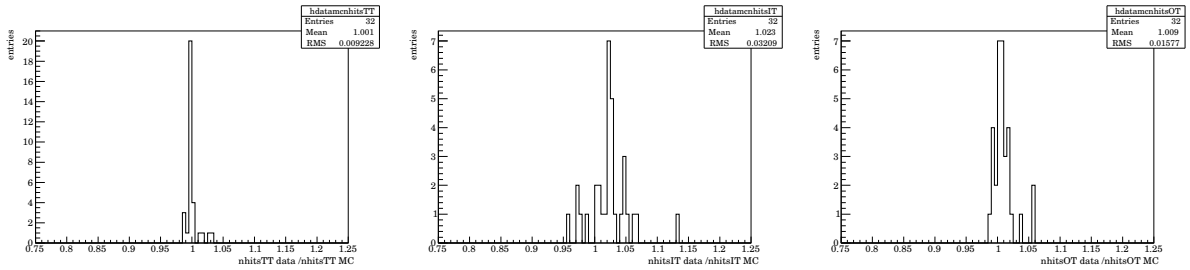
(a) *TT, before correction* **(b)** *IT, before correction* **(c)** *OT, before correction*

(d) *TT, after correction* **(e)** *IT, after correction* **(f)** *OT, after correction*

Figure 10.7.: *Number of hits per track in each subdetector for data and Monte Carlo, before and after correction. For these plots, the full sample of 10 million Monte Carlo events was used.*

These ratios are shown in Figure 10.8; similar checks were performed on various subsets of K^0_S candidates and daughter tracks. The following subsets have been studied, each contributing one entry to the histograms in the end:

- K^0_S momentum in two bins: 0 GeV $\leq p \leq$ 10 GeV and 10 GeV $\leq p$,
- daughter π momentum in two bins: 0 GeV $\leq p \leq$ 5 GeV and 5 GeV $\leq p$,
- per quarter (for the OT, a quarter is one of the four quadrants, in the IT, each of the four boxes makes up a quarter, and for TT a similar notion as for the OT is used, although the splitting into quarters is less clear-cut than in the OT case because of the layout of the silicon sensors in TT),
- 4 (uniformly spaced) bins in pseudo rapidity η of the K^0_S in the range from 2 to 6,
- 4 (uniformly spaced) bins in rapidity y of the K^0_S in the range from 2 to 5,
- 4 (uniformly spaced) bins in pseudo rapidity η of the daughter π in the range from 2 to 6,
- 4 (uniformly spaced) bins in rapidity y of the daughter π in the range from 2 to 5,
- 4 (uniformly spaced) bins in ϕ (angle of track in plane transverse to beam direction, i.e. $\tan(\phi) = p_x/p_y$ if p_x and p_y are the x and y components of the K^0_S momentum) of the K^0_S in the range from $-\pi$ to π,



(a) *TT, after correction* **(b)** *IT, after correction* **(c)** *OT, after correction*

Figure 10.8.: *Ratio of average number of hits per track for each subdetector in data and Monte Carlo; each entry corresponds to a different area of phase space (not necessarily disjoint).*

subdetector	TT	IT	OT
variation	±1%	±3.5%	±2%

Table 10.1: *Variation of hit efficiency corrections per subdetector for systematic studies.*

- 4 (uniformly spaced) bins in ϕ (defined analogously to the last item) of the daughter π in the range from $-\pi$ to π.

The choice of the bins above was dictated by the need to have sufficiently many entries in each bin to be able to accurately determine the data/Monte Carlo ratio of the number of hits per track in that bin. While the particular choice of quantities to look at and the bins in which to look are somewhat arbitrary, some form of cross check in this direction is clearly required, and it is therefore better to perform such a test (even is some aspects may seem arbitrary) than to omit the test.

The means in all three distributions in Figure 10.8 are close to 1.0, and the spread (measured with the RMS) is below 4% for all subdetectors.

This information can be used to derive an upper bound for the per-track reconstruction efficiency difference between data and simulation due to different effective hit finding efficiencies. While a direct measurement of the track reconstruction efficiency will be performed later, this can be used as an independent cross-check. The idea is to ascribe the observed disagreement in each subdetector to the effective hit finding efficiency correction factors being off. By varying this correction and observing the change in K^0_S yield in simulation, it is possible to obtain a (very pessimistic) estimate on how much the track reconstruction efficiency differs between data and simulation due to imperfections in the correction procedure. Table 10.1 shows by how much the correction factors for the effective hit finding efficiencies were varied in each subdetector. The sample which applies an upward variation will be called the “+RMS” sample, the sample which uses the downward variation will be called the “-RMS” sample.

Table 10.2 shows the resulting K^0_S yields for the completely uncorrected sample of simulated events, for the sample where dead Outer Tracker modules were included, the sample including the full correction, and the “+RMS” and “-RMS” samples.

From this table, it becomes clear that the correction is a significant effect which causes a drop of 10.3% in K^0_S yield with respect to the untreated sample. The “+RMS” and

sample	K^0_S yield	relative ratio
untreated Monte Carlo sample	13422±126	100.0 %
dead OT channel correction	12974±124	(96.7± 1.3) %
dead OT ch. + hit eff. corr.	12043±120	(89.7± 1.4) %
dead OT ch. + hit eff. corr., "+RMS"	12759±123	(95.1± 1.3) %
dead OT ch. + hit eff. corr., "-RMS"	10987±113	(81.9± 1.4) %

Table 10.2.: *K^0_S yields (number of signal candidates from fit) for Monte Carlo samples prepared in different ways. The uncertainties given for the yields are the statistical uncertainties given by the fit, the ones for the fraction have been obtained by Gaussian error propagation which (while not being entirely correct due to e.g. samples not being strictly uncorrelated) gives a conservative estimate of the uncertainties in the fractions. Clearly, the correction is a significant effect.*

"-RMS" sample cause a yield variation of +5.9% and -8.8% with respect to the sample with nominal correction factors, corresponding to a maximal data/simulation disagreement in reconstruction efficiency between +2.9% to -4.5% due to the method. The reason for these asymmetric bounds is that for several layers the upward variation of the effective hit efficiency by the amount specified in Table 10.1 would cause the numerical value of the hit efficiency to go above 100%. Clearly, this does not make sense, and the values are set to 100% in this case. This effectively causes the variation in K^0_S yield to be limited in one direction. (One would observe the same effect for varying downward if the effective hit efficiency was close to zero.)

It should be remembered that this method is not very sensitive and should not be used to actually measure the track reconstruction efficiency (this will be done in the next section). Still, it is a nice cross-check which is not too difficult to do and gives some idea as to what to expect in terms of data/MC agreement for track reconstruction efficiencies.

10.3. Tracking efficiency

To better assess the validity of the hit level efficiency correction applied to the sample of simulated events, data-driven methods to measure the tracking efficiency are investigated. The idea is to compare the tracking efficiency in data and simulation for different subdetectors and tracking algorithms. The resulting efficiency measurements should not be regarded as absolute measurements; they serve only to compare data and simulation in a consistent manner. All of the methods presented below perform the measurement in a specific region of phase space which is not necessarily the same as the phase space used in the K^0_S analysis. The presentation here focuses on obtaining figures for the difference in downstream tracking efficiency between data and Monte Carlo because tracking efficiencies for other track types are of little consequence for this analysis.

	data	ideal MC	massaged MC	+RMS	-RMS
Forward tracks	$95.9 \pm 0.6\%$	$97.8 \pm 0.3\%$	97.4%	97.6 %	96.6%
Match tracks	$96.6 \pm 0.6\%$	$98.0 \pm 0.3\%$	97.6%	98.0 %	96.9%
Forward tracks picking up hits in TT	$96.8 \pm 0.5\%$	$98.9 \pm 0.2\%$	98.9%	99.0%	99.0%
Match tracks picking up hits in TT	$96.2 \pm 0.6\%$	$98.7 \pm 0.2\%$	98.7%	98.8%	98.7%

Table 10.3.: *Comparison of TT hit finding efficiency for Forward and Match tracks in data and Monte Carlo. The term massaged MC refers to the method described in the previous section. The first two rows are checking for the presence of a reconstructed downstream track relative to the forward and match tracks. The 3rd (4th) row gives the efficiency of the Forward tracking (Track Matching) to pick up hits in the TT (number of tracks with hits in the TT over number of tracks in TT acceptance).*

10.3.1. Methods using K_S^0 daughter tracks

In this section, K_S^0 daughter tracks are used, and sideband subtraction is applied to statistically subtract two-pion combinations which do not form a K_S^0 (this will also subtract the effect of fake tracks). In most studies presented in this section, two tracks of different types (e.g. long and downstream tracks) will be considered associated to each other if the track with fewer hits shares 70% or more of its hits with the one with more hits.

Hit finding efficiency in TT

There is no standalone TT track reconstruction in LHCb since TT has very little redundancy, so all pattern recognition algorithms which use TT hits only try to add TT hits to tracks which have been found in a different subdetector. Therefore, this section will not measure a TT tracking efficiency, but a TT hit finding efficiency which gives the efficiency with which TT hits are added to an existing track in the main tracker to form a downstream track. This is possible by using long tracks, either long tracks found by the Forward tracking or the Track Matching, which do not require TT hits to be found. The TT hit finding efficiency denominator is defined by choosing long tracks which go through the TT acceptance and are linked to a T track. The corresponding numerator can be determined by checking how many of these are linked to a downstream track. The results can be found in Table 10.3, along with the efficiency to find TT hits on long tracks which go through the TT acceptance. The difference between data and the ideal (untreated) Monte Carlo sample is between 1.5% and 2%; this discrepancy becomes smaller when going to the hit finding efficiency corrected (i.e. the “massaged”) Monte Carlo sample, but does not cover the full effect observed. This is because of the pattern recognition algorithms which effectively demand three out of four possible hits in TT. This in turn introduces a bias towards high efficiencies, thus rendering the hit finding efficiency correction technique less effective. Therefore, the derived correction for TT is slightly too small to cover the full data/Monte Carlo difference seen.

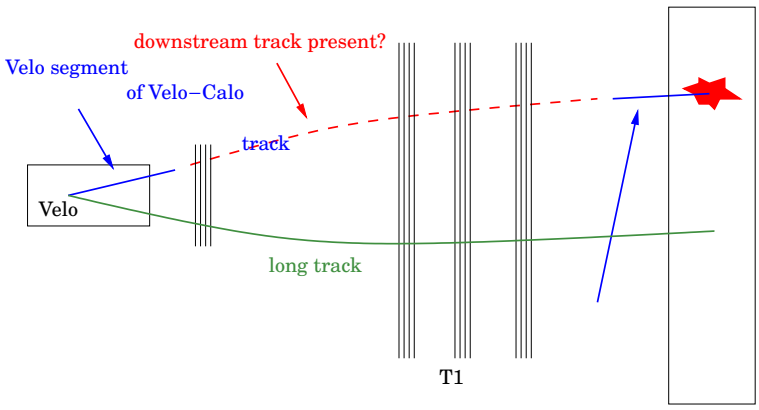


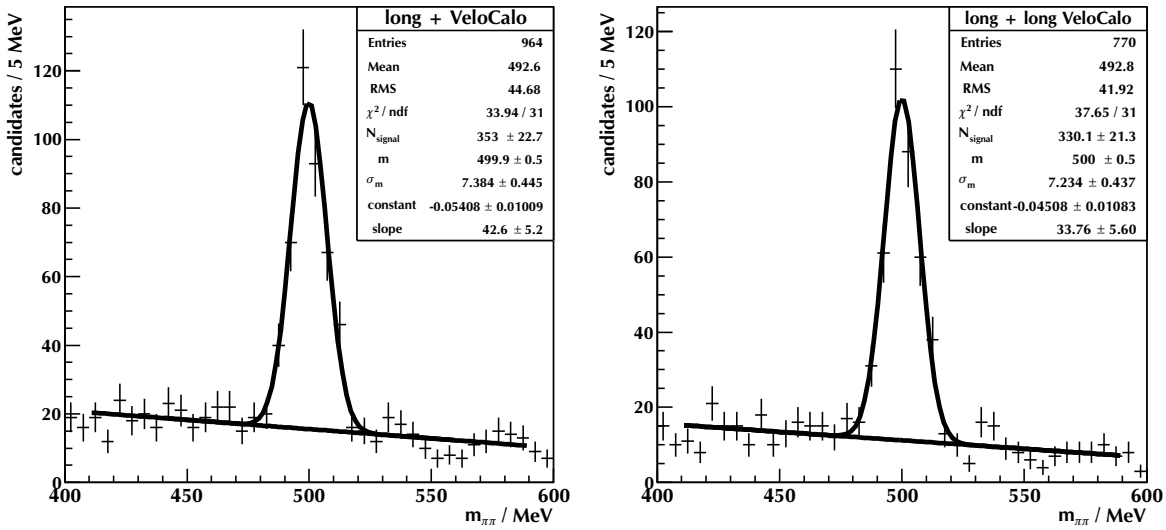
Figure 10.9.: *Principle of data-driven tracking efficiency measurement using Velo-Calo tracks: The downstream tracking is efficient if a downstream track can be found which can be matched to a Velo-Calo track constructed from a Velo and a calorimeter segment. To avoid pollution by fake matches of Velo and calorimeter segment, this analysis is performed on K_S^0 daughter tracks with one daughter being a Velo-Calo segment; fake Velo-Calo track pollution can then be handled with sideband subtraction.*

Track finding efficiency in the main tracker

To measure a tracking efficiency in the main tracker, it is necessary to form a track with some kind momentum information which is known to reach the main tracker acceptance without actually using the main tracker itself. This is accomplished by matching Velo tracks to calorimeter clusters, thus forming so-called *Velo-Calo* tracks using a method described in [72]. K_S^0 candidates can then be formed from a long track and a Velo-Calo track, see Figure 10.10a. For the Velo-Calo leg of the K_S^0, it is possible to check if the Velo portion of the Velo-Calo track is linked to a long track (cf. Figure 10.10). Since the corresponding long track can only exist if the tracking in the main tracker was successful, the main tracker reconstruction efficiency can be written as:

$$\varepsilon_{\text{T station}} = \frac{\#\text{VeloCalo tracks reconstructed as long track}}{\#\text{VeloCalo tracks}} \tag{10.4}$$

The fits in Figure 10.10 are for illustration only, the results given in Tables 10.4 and 10.5 were obtained with sideband subtraction, with the Velo hit content being used to link tracks to each other. For K_S^0 reconstructed from Downstream tracks, the figure obtained with Match Tracks is the most relevant, since both Downstream and Match tracks start from the same main tracker seed track. The data/Monte Carlo discrepancy starts out with 6% for the untreated ideal Monte Carlo and decreases to 2.7% for the Monte Carlo sample including the hit efficiency correction. Table 10.5 gives an overview of the behaviour of the main tracker tracking efficiency as function of track p_T for Match tracks; the main issue here is that the sample of Velo-Calo K_S^0 candidates is rather small and is further reduced by binning in track p_T, so the statistical uncertainties given in Table 10.5 are not completely trustworthy because they do not include the full extent of



(a) K^0_S *mass peak from long-Velo-Calo combinations*

(b) K^0_S *mass peak from long-Velo-Calo combinations where the Velo-Calo track is linked to a long track*

Figure 10.10.: *Fits to the Velo-Calo track - long track combined K^0_S candidates' mass distribution, not demanding (left) and demanding (right) the VELO-Calo track to be linked to a long track*

	data	ideal MC	massaged MC
match tracks	86.1 ± 2.6 %	92.1 ± 1.2 %	88.8 ± 1.3 %
forward tracks	93.4 ± 2.4 %	95.3 ± 1.0 %	94.1 ± 1.1 %

Table 10.4.: $\varepsilon_{\text{Tstation}}$*; detailed description of the definition is given in the text.*

fluctuations which are possible with low statistics; the lowest two p_T bins which contain the majority of tracks are trustworthy, though.

Summary of K^0_S driven studies

The per-track difference between data and ideal (untreated) Monte Carlo is -6% in the main tracker and -2% in TT, resulting in a combined difference of -8% for downstream tracks. This is reduced by the hit finding efficiency correction to -2.7% in the main tracker and -1% in TT, resulting in a combined difference of -3.7% for downstream tracks, a result well in agreement with the prediction of +2.9% to -4.5% from Section 10.2.2.

10.3.2. Track based method

In the last subsection, the tracking efficiency was determined with K^0_S daughter tracks, and the main issue was the limited statistics available on data. To alleviate this issue, a different method using all tracks in an event was studied in [73]. In this method, fake

p_T [MeV]	$0-200$	$200-400$	$400-600$	$600-800$
data	70 ± 8 %	90 ± 4 %	83 ± 6 %	83 ± 2 %
ideal MC	81 ± 3 %	97 ± 2 %	93 ± 2 %	97 ± 3 %
massaged MC	77 ± 3 %	92 ± 2 %	91 ± 2 %	93 ± 3 %

Table 10.5.: *Results of track efficiency measurements for Match tracks in bins of track p_T on various samples using the K^0_S based Velo-Calo method.*

p_T [MeV]	0-200	200-400	400-600	600-800	800-1000	1000-1600
data	52.7±2.4	86.7±2.4	87.3±3.0	90.2±4.1	83.8± 4.7	88.2± 5.0
ideal MC	99.6±5.2	88.1±1.6	90.5±2.0	88.9±2.7	88.2± 3.8	89.3± 4.2
massaged MC	84.0±2.9	87.8±1.6	84.4±1.8	88.6±2.7	84.1± 3.6	92.1± 5.0

Table 10.6.: *Results of track based efficiency measurements for downstream in bins of track p_T on various samples using the track based Velo-Calo method.*

tracks are suppressed by matching Velo tracks to calorimeter clusters and looking at the difference in y coordinate between the extrapolation of the Velo track and the calorimeter cluster, allowing fake tracks to be subtracted statistically. The result of this study for downstream tracks is given in Table 10.6 in bins of track p_T.

10.4. Summary of data/Monte Carlo tracking efficiency comparison

Since the two methods to measure the downstream tracking efficiency in a data-driven fashion cannot measure absolute efficiencies because they work on different track samples, their results cannot be compared directly. However, the "relative tracking efficiency", i.e. the tracking efficiency in data divided by that in the hit finding efficiency corrected ("massaged") Monte Carlo, should suffer much less from these considerations. These relative tracking efficiencies have been computed in bins of track p_T for the K^0_S based and the track based method, see Table 10.7.

Both methods show good agreement above a track $p_T > 200\,\text{MeV}$. The agreement between both methods in the lowest p_T bin is much worse, however the track-based method has problems in the lowest p_T bin because the background in this bin is large and tends to peak under the signal, so the result from the track based method in this

p_T [MeV]	0-200	200-400	400-600	600-800	800-1000	1000-1600
K^0_S based method [%]	91± 13	98± 5	91± 7	89± 4	–	–
track based method [%]	63± 4	99± 3	103± 4	102± 6	100± 7	96± 8

Table 10.7.: *Tracking efficiency ratio Data/Monte Carlo for downstream tracks measured using different methods. Uncertainties are statistical uncertainties only. Efficiencies larger than 100% mean that the overall efficiency correction applied through the hit dropping procedure is over-correcting a little bit.*

bin will be ignored. For all other bins, the track based method has smaller statistical uncertainties, so it is preferred over the K^0_S based method. For some bins in Table 10.7, the relative tracking efficiency is larger than 100% which indicates that the correction of effective hit efficiency might sometimes correct a little too much.

For the evaluation of systematic uncertainties due to reconstruction efficiency differences in data and simulated events, the following values will be used:

- For tracks with $p_T < 200\,\text{MeV}$, 91% relative efficiency is used; the track based method is ignored in this bin due to its large background
- For tracks with $200\,\text{MeV} \leq p_T \leq 400\,\text{MeV}$, use 99% relative efficiency (the mean of both methods in that bin).
- For tracks with $p_T > 400\,\text{MeV}$, use 101%. From the tracking efficiency studies in the first part of this thesis, we expect the tracking efficiency to form a plateau, so the weighted mean of the corresponding bins of the (more precise) track based method is taken as an estimate of the plateau value.

The contribution of this difference in reconstruction efficiency between data and Monte Carlo will be treated later.

11. Combined K^0_S selection and reconstruction efficiency

In this chapter, the combined selection and reconstruction efficiency is determined. It includes contributions from selection, tracking efficiency, decay or absorption of the K^0_S in flight and acceptance effects. For the sake of brevity, it will just be called "reconstruction efficiency". For all studies, the sample of simulated events which was corrected to have effective hit efficiencies comparable to data was used.

11.1. Determination of reconstruction efficiency

The reconstruction efficiency is determined on a sample of simulated events which would have caused a calorimeter hardware trigger by forming the ratio of the number of successfully reconstructed $K^0_S \longrightarrow \pi^+\pi^-$, $N_{K^0_S}(p_T, y)$ and the number of generated prompt[1] K^0_S, $N^{\text{gen}}_{K^0_S}(p_T, y)$, in a bin of p_T and y of the K^0_S:

$$\varepsilon_{\text{rec}}(p_T, y) = \frac{N_{K^0_S}(p_T, y)}{N^{\text{gen}}_{K^0_S}(p_T, y)} \tag{11.1}$$

Thus, its definition includes effects from detector acceptance, track reconstruction efficiency, selection efficiency and the branching ratio $K^0_S \longrightarrow \pi^+\pi^-$ (since we do not require the K^0_S to decay into two charged pions in the denominator).

Of course, there are several ways how $N_{K^0_S}(p_T, y)$ and $N^{\text{gen}}_{K^0_S}(p_T, y)$ can be defined, and each way provides information about different effects. These alternative definitions will later be used to derive estimates for the systematic uncertainties of the reconstruction efficiencies.

The default efficiency numerator, $N_{K^0_S}(p_T, y)$, is defined as the number of reconstructed K^0_S as extracted from the fit to the invariant mass distribution in bins of reconstructed p_T and y of the K^0_S. This definition is the one which most closely resembles how the K^0_S yield is extracted on data. For the denominator, the generator level quantities for p_T and y are used. Table 11.1 lists the results. The corresponding mass fits can be found in Figures 11.1, 11.2 and 11.3.

Note that the denominator does not include non-prompt K^0_S while the numerator does. Assuming that the fraction of non-prompt K^0_S is the same in data and simulation (it is found to be 0.6% in simulated events), this definition of reconstruction efficiency will actually correct for the pollution of non-prompt K^0_S in the measured K^0_S yields. (An estimate for the effect on the yields due to different non-prompt fractions in data and simulation is presented later.)

[1] Prompt K^0_S are K^0_S with a lifetime $ct(K^0_S) < 10^{-11}$ m.

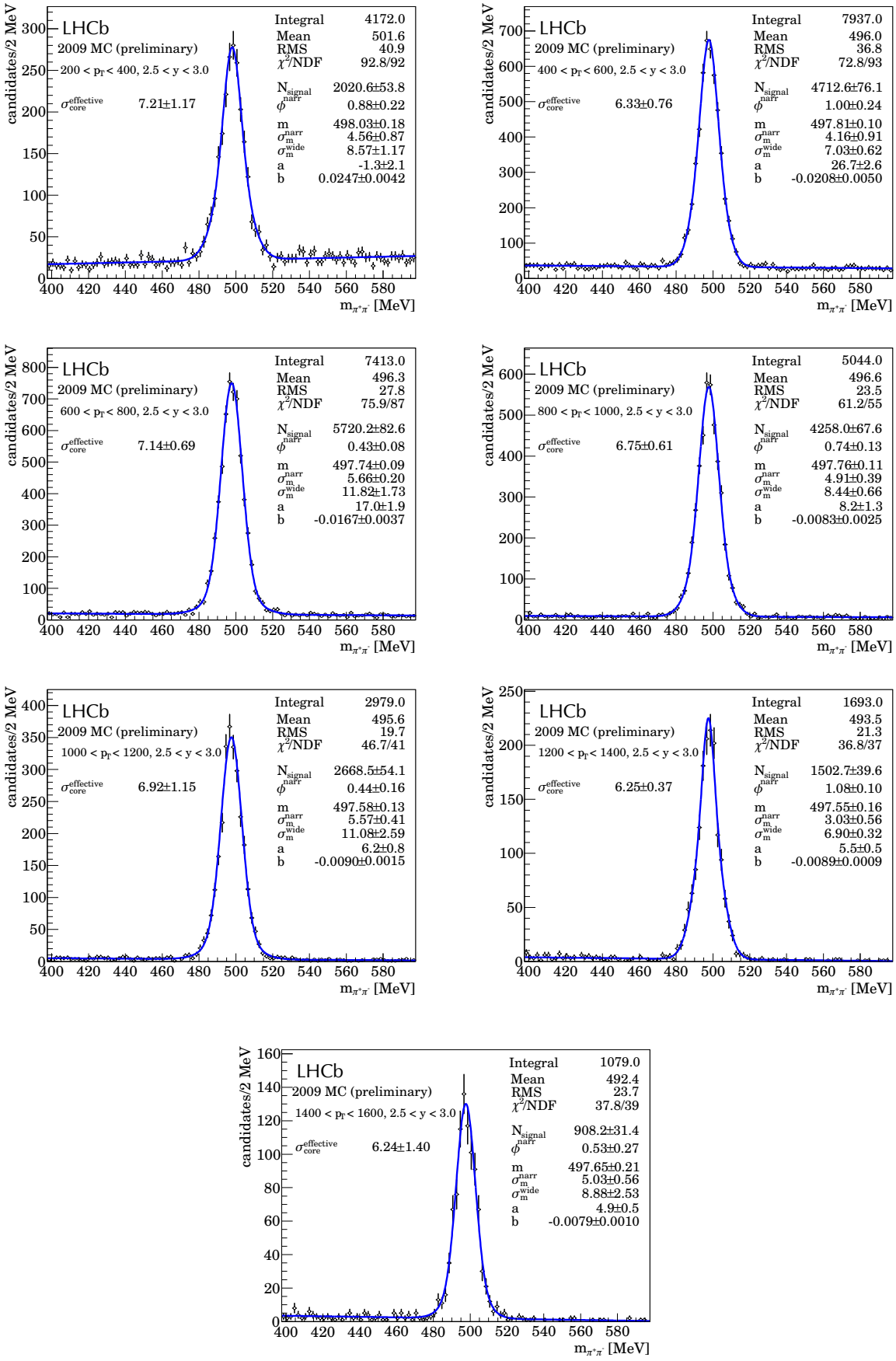


Figure 11.1.: *Fits to the K^0_S candidates mass distribution in Monte Carlo for different p_T bins with $2.5 < y < 3.0$.*

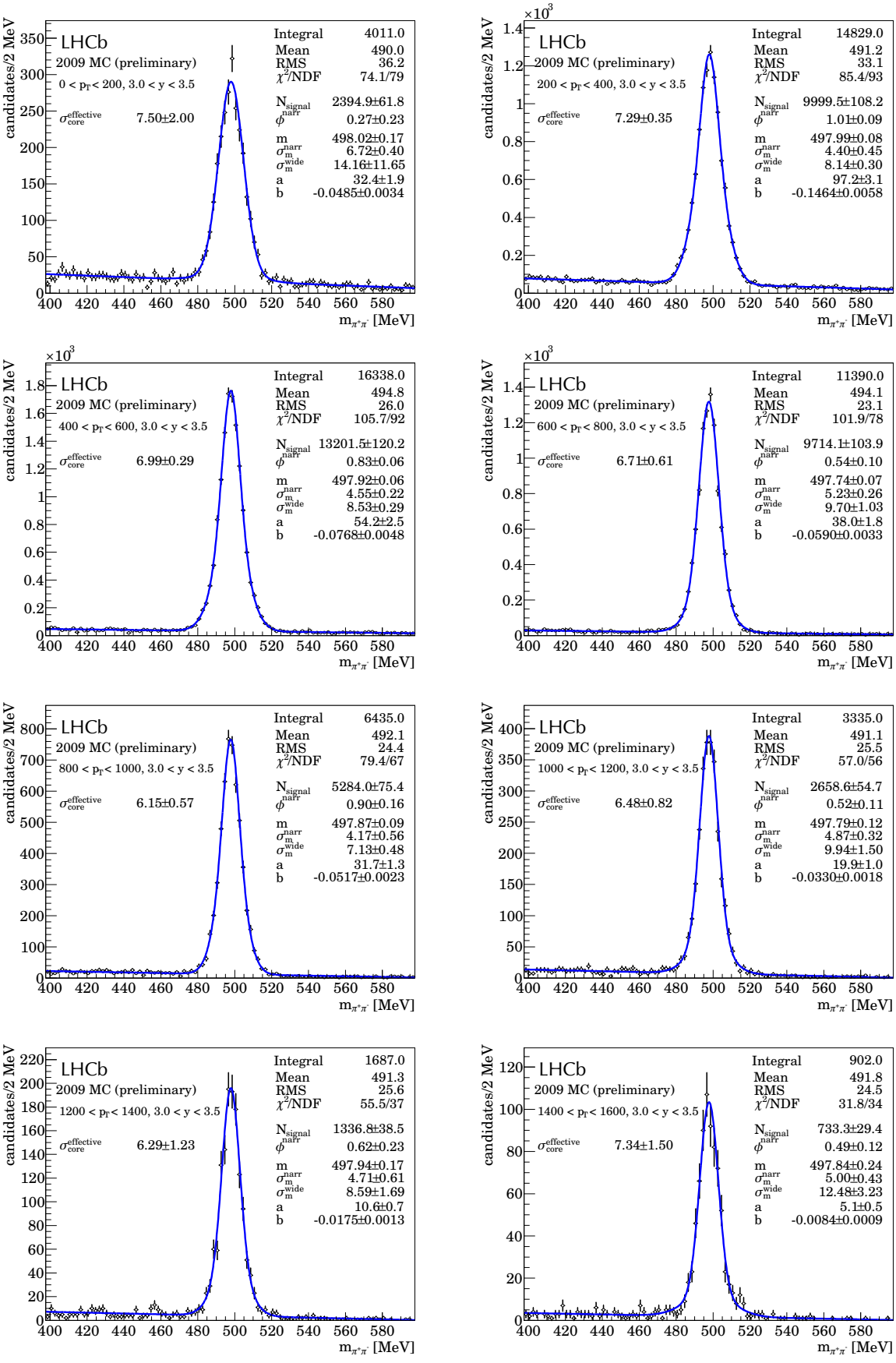


Figure 11.2.: *Fits to the K^0_S candidates mass distribution in Monte Carlo for different p_T bins with $3.0 < y < 3.5$.*

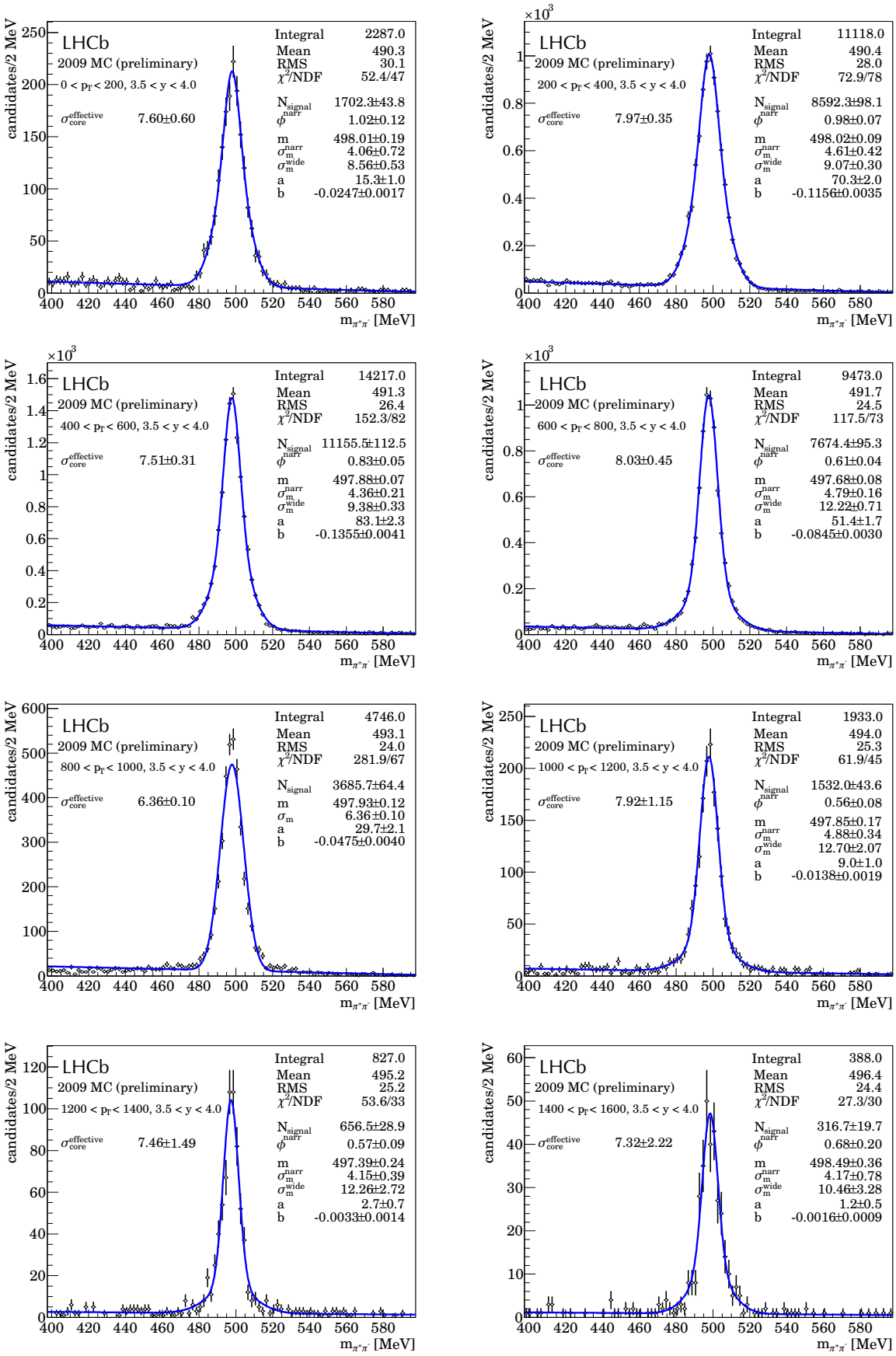


Figure 11.3.: *Fits to the K^0_S candidates mass distribution in Monte Carlo for different p_T y bins with $3.5 < y < 4.0$.*

$p_T[MeV]/y$	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	-	3.3±0.1	2.8±0.1
200 – 400	1.2±0.1	7.0±0.1	7.1±0.1
400 – 600	3.4±0.1	11.3±0.1	11.5±0.1
600 – 800	7.1±0.1	14.4±0.2	14.3±0.2
800 – 1000	10.7±0.2	16.4±0.2	15.0±0.3
1000 – 1200	13.6±0.3	17.7±0.4	14.4±0.4
1200 – 1400	15.3±0.4	18.1±0.5	12.9±0.6
1400 – 1600	16.8±0.6	18.1±0.7	11.9±0.7

Table 11.1.: *Default reconstruction efficiencies (in percent), using the fit to extract the number of reconstructed K^0_S candidates. Contributions from non-prompt K^0_S are per construction included in the numerator, however not in the denominator of the efficiency.*

11.2. Reconstructed versus generator level quantities

To estimate the effect of using the reconstructed p_T and y for the efficiency numerator and the corresponding generator level quantities for the denominator, the efficiency is determined twice, requiring the K^0_S candidates to be matched to Monte Carlo truth, using both variants for the efficiency numerator and forming the difference. The result can be seen in Table 11.2 and will be added as a contribution to the systematic uncertainties to give an estimate for the effect of bin-to-bin migration due to the finite resolution of the reconstructed p_T and y.

The reason for not defaulting to one of the two efficiency definitions used in this subsection is that the association to Monte Carlo truth is not 100% efficient, i.e. there are well reconstructed K^0_S candidates which fail the association requirements, for example because the hit purity of one of the daughter tracks is insufficient. Figure 11.4 illustrates the effect.

11.3. Extraction of efficiency numerator

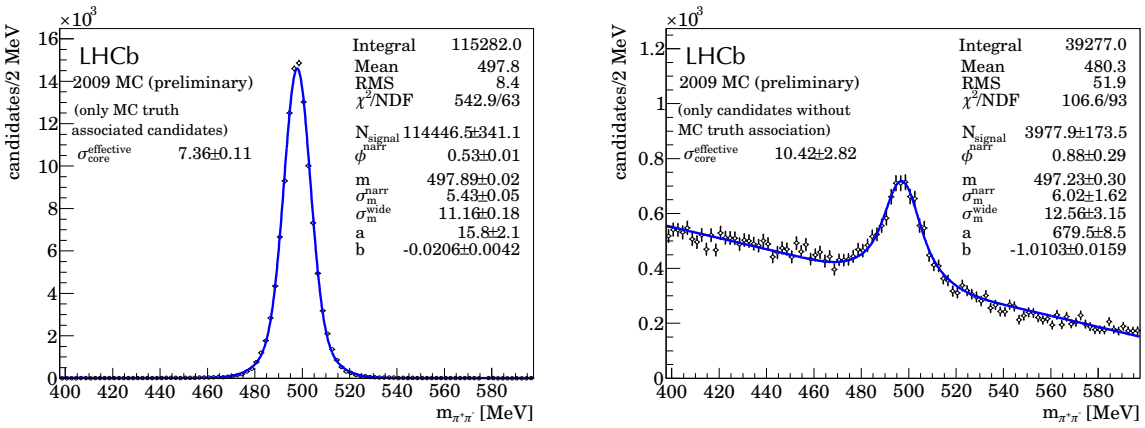
In the default definition for the reconstruction efficiency, the efficiency numerator was obtained from a fit to the invariant mass of the two pion combinations forming the K^0_S candidates. In this section, the difference of the default option to a counting method based on Monte Carlo truth matched candidates is investigated. In fact, two counting based methods are evaluated:

- The standard method of association to Monte Carlo truth, i.e. all K^0_S daughters must be associated (as defined in Section 4.3.1) to the corresponding daughter particle on generator level for the K^0_S itself to be associated to Monte Carlo truth. The method will be referred to as “counting”.

- The “counting” method can be augmented to recover some of the inefficiency by matching the reconstructed K^0_S candidate to the generator level one in phase space

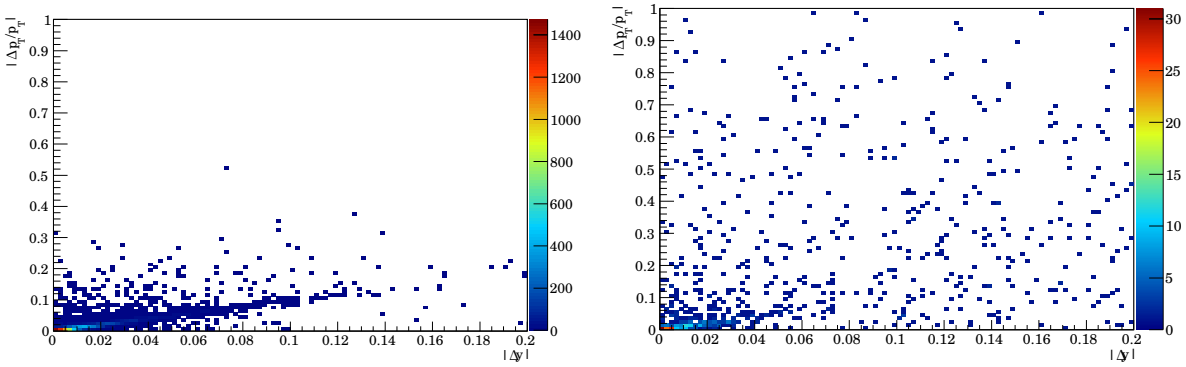
$p_T[MeV]/y$	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	-	0.1	< 0.1
200 – 400	< 0.1	< 0.1	< 0.1
400 – 600	< 0.1	< 0.1	0.1
600 – 800	< 0.1	< 0.1	< 0.1
800 – 1000	< 0.1	< 0.1	< 0.1
1000 – 1200	< 0.1	0.1	< 0.1
1200 – 1400	< 0.1	< 0.1	0.2
1400 – 1600	0.1	0.2	0.2

Table 11.2.: *Difference in reconstruction efficiencies (in percent) for using the reconstructed p_T and y or the generated p_T and y for the K^0_S candidates which enter the numerator of the efficiency correction. As these numbers are highly correlated, no uncertainties are given. The observed difference will be added to the systematic uncertainties.*



(a) **(b)**

Figure 11.4.: *K^0_S candidates in Monte Carlo. Candidates in (a) are associated to MC truth. K^0_S candidates in Monte Carlo which are not associated to MC truth (b); a clear peak is seen here as well.*



(a) *Monte Carlo truth matched* K^0_S *candidates*

(b) *unmatched* K^0_S *candidates*

Figure 11.5.: *$\Delta p_T/p_t$ and $\Delta\eta$ distributions between reconstructed and generated K^0_S candidates for Monte Carlo truth matched (a) and unmatched (b) candidates.*

if the association using hit level information from above fails. If the differences in p_T and y between reconstructed and generator level quantities are small enough, the reconstructed candidates is considered to be matched to Monte Carlo truth as well. This method will be referred to as "counting++". Figure 11.5 shows the $|\Delta p_T/p_T|$ versus $|\Delta y|$ distribution for truth-matched and non-truth-matched K^0_S in the sense of the "counting method". The permissible deviations for the "counting++" method are chosen to be $|\Delta p_T/p_T| < 0.1$ and $|\Delta y| < 0.04$ based on these plots.

The results of the study are shown in Table 11.3 with the difference between the counting++ method and the fit result denoted Δ. The relative deviation between both methods (column "Δ/counting++ does not show a systematic pattern of deviations and is comparable in size to the statistical uncertainties. Nevertheless, this deviation potentially affects both the measurement of reconstruction efficiency and the yield extraction on data and will therefore be used to estimate systematic uncertainties on a bin-by-bin basis for these two quantities.

11.4. Breakdown of reconstruction efficiency into its components

This section presents a breakdown of the total reconstruction efficiency into its contributing components. Due to the inefficiencies in the association of K^0_S candidates to Monte Carlo truth (which is needed to determine the values of the individual components), the information given in this section cannot be used to construct the total reconstruction efficiency by multiplying up its component efficiencies. Therefore, this section can only serve to give a qualitative insight into the relative importance of the individual contributions, it is not possible to derive quantitative results based on the information presented here. Since the results in this section are not used in the rest of the analysis, they were derived on a smaller sample of roughly 1 million simulated events. The total

p_T [MeV]; y	counting	counting++	single Gaussian	double Gaussian	Δ	Δ/counting++
200 – 400; 2.5 – 3.0	1900	2002	1984± 51	2021± 54	19	0.9 %
400 – 600; 2.5 – 3.0	4592	4845	4678± 74	4713± 76	-132	-2.7 %
600 – 800; 2.5 – 3.0	5410	5691	5605± 79	5720± 83	29	0.5 %
800 – 1000; 2.5 – 3.0	4076	4285	4232± 67	4258± 68	-27	-0.6 %
1000 – 1200; 2.5 – 3.0	2569	2689	2632± 53	2668± 54	-21	-0.8 %
1200 – 1400; 2.5 – 3.0	1457	1534	1503± 40	1503± 40	-31	-2.0 %
1400 – 1600; 2.5 – 3.0	886	936	901± 31	908± 31	-28	-3.0 %
0 – 200; 3.0 – 3.5	2336	2435	2395± 62	2364± 53	71	2.9 %
200 – 400; 3.0 – 3.5	9787	10223	9928±107	10000±108	-223	-2.2 %
400 – 600; 3.0 – 3.5	12898	13434	13109±120	13201±120	-233	-1.7 %
600 – 800; 3.0 – 3.5	9552	9922	9594±101	9714±104	-208	-2.1 %
800 – 1000; 3.0 – 3.5	5232	5442	5259± 75	5284± 75	-158	-2.9 %
1000 – 1200; 3.0 – 3.5	2567	2678	2608± 53	2659± 55	-19	-0.7 %
1200 – 1400; 3.0 – 3.5	1278	1331	1319± 38	1337± 39	6	0.4 %
1400 – 1600; 3.0 – 3.5	678	704	713± 28	733± 29	29	4.2%
0 – 200; 3.5 – 4.0	1691	1750	1702± 44	1694± 44	56	3.2 %
200 – 400; 3.5 – 4.0	8564	8886	8541± 98	8592± 98	-294	-3.3 %
400 – 600; 3.5 – 4.0	11031	11451	11028±111	11156±112	-295	-2.6 %
600 – 800; 3.5 – 4.0	7249	7504	7418± 92	7674± 95	170	2.3 %
800 – 1000; 3.5 – 4.0	3549	3672	3686± 64	–	14	0.4 %
1000 – 1200; 3.5 – 4.0	1429	1483	1473± 41	1532± 44	49	3.3 %
1200 – 1400; 3.5 – 4.0	627	647	628± 27	657± 29	10	1.5 %
1400 – 1600; 3.5 – 4.0	297	309	311± 19	317± 20	8	2.5 %

Table 11.3.: *Different methods to determine the efficiency numerator, for detailed explanation see the text.*

reconstruction efficiency is broken down into component contributions according to the following scheme:

$$\varepsilon_{rec} = \varepsilon_{decay} \cdot \varepsilon_{acceptance} \cdot \varepsilon_{tracking} \cdot \varepsilon_{selection} \tag{11.2}$$

The different contributions to the total reconstruction efficiency are defined as follows:

- $\varepsilon_{decay} = \frac{\text{generator level prompt } K^0_S \to \pi^+\pi^-}{\text{all generator level prompt } K^0_S}$; the difference of this efficiency to the PDG branching ratio gives the fraction of absorption or decay outside the simulated LHCb volume.
- $\varepsilon_{acceptance} = \frac{\text{generator level prompt } K^0_S \to \pi^+\pi^-, \text{ with both pions defined as reconstructible}}{\text{generator level prompt } K^0_S \to \pi^+\pi^-}$; the standard tracking definition for *reconstructible as downstream track* is used here: There is at least one measurement in each of the two TT stations and at least one measurement in the x and one in the stereo layers in each of the three T stations, which are associated to the particle.
- $\varepsilon_{tracking} = \frac{\text{generator level prompt } K^0_S \to \pi^+\pi^- \text{ with both pions reconstructible and reconstructed}}{\text{generator level prompt } K^0_S \to \pi^+\pi^-, \text{ with both pions reconstructible}}$; this is essentially the probability to successfully reconstruct both daughter tracks which is the reason behind the choice of name
- $\varepsilon_{selection} = \frac{\text{reconstructed\&selected } K^0_S \text{candidates, associated to prompt } K^0_S \text{ with reconstructible pions}}{\text{prompt generated } K^0_S \to \pi^+\pi^- \text{ with both pions reconstructible and reconstructed}}$;

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	-	68.8±1.0	68.2±1.0
200 – 400	69.3±0.6	68.5±0.7	68.2±0.7
400 – 600	69.2±0.7	68.9±0.7	68.0±0.8
600 – 800	69.2±0.9	68.6±1.0	69.2±1.1
800 – 1000	67.9±1.3	68.2±1.4	68.2±1.6
1000 – 1200	67.5±1.8	66.8±2.1	68.8±2.4
1200 – 1400	69.8±2.6	68.1±2.9	69.1±3.6
1400 – 1600	66.0±3.4	67.9±4.0	66.4±4.7

(a) ε_{decay} *in percent*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	-	27.8±0.8	28.8±0.9
200 – 400	10.0±0.3	30.3±0.6	30.1±0.7
400 – 600	17.6±0.5	38.0±0.8	40.5±0.9
600 – 800	26.9±0.8	47.6±1.2	48.0±1.3
800 – 1000	36.9±1.3	53.7±1.8	52.9±2.1
1000 – 1200	44.1±2.1	57.8±3.0	54.6±3.2
1200 – 1400	49.9±3.2	63.7±4.4	57.6±4.9
1400 – 1600	56.2±4.9	58.8±5.6	54.5±6.5

(b) $\varepsilon_{acceptance}$ *in percent*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	-	73.6±3.0	71.6± 3.1
200 – 400	55.9±2.7	71.1±2.0	68.5± 2.1
400 – 600	62.3±2.4	72.0±2.0	68.7± 2.0
600 – 800	72.3±2.8	73.4±2.3	70.3± 2.5
800 – 1000	75.5±3.5	74.5±3.2	73.7± 3.7
1000 – 1200	76.3±4.7	78.9±4.8	74.4± 5.4
1200 – 1400	74.7±6.0	80.4±6.5	77.7± 8.0
1400 – 1600	78.5±8.2	79.2±9.1	82.7±11.7

(c) $\varepsilon_{tracking}$ *in percent*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	-	19.5±1.5	16.8±1.5
200 – 400	21.0±1.9	41.4±1.6	40.6±1.8
400 – 600	34.8±2.0	48.9±1.8	52.1±2.0
600 – 800	42.4±2.3	49.6±2.1	54.8±2.5
800 – 1000	47.1±2.9	51.4±2.8	49.5±3.3
1000 – 1200	51.6±4.1	51.3±4.0	47.4±4.6
1200 – 1400	53.6±5.5	53.3±5.5	45.5±6.3
1400 – 1600	54.9±7.2	49.6±7.4	39.6±7.8

(d) $\varepsilon_{selection}$ *in percent*

Table 11.4.: *Different contributions to the total reconstruction efficiency; a detailed description of the individual contributions is given in the text.*

this is essentially the efficiency of the selection process on successfully reconstructed K^0_S

Table 11.4 lists the four contributions to the total reconstruction efficiency in bins of p_T and y. Table 11.5 lists the fraction of reconstructed K^0_S candidates in each bin for which at least one daughter is not reconstructible. This happens for about 3.4% of candidates; this is another reason why just multiplying the contributions to get the total reconstruction efficiency does not work.

11.5. Influence of event properties

Since this analysis relies on simulated events for efficiency determination, the influence of the event properties must be checked, and appropriate systematic uncertainties should be assigned. For this reason, diffractive and two types of non-diffractive events are checked. The influence of the tuning of the Monte Carlo event generator used (PYTHIA) is also studied by using two alternate tunes.

11.5.1. Reconstruction efficiency in diffractive events

To judge the behaviour of the reconstruction efficiency with respect to diffractive and non-diffractive events, the total reconstruction efficiency is determined on two subsamples of a sample of simulated events, the subsample of diffractive events and the subsample of

Table 11.5: *Fraction of reconstructed K^0_S with at least one daughter track defined as not reconstructible.*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	-	2.4±0.3	2.8±0.4
200 – 400	1.6±0.9	2.2±0.5	3.3±0.6
400 – 600	1.9±0.7	2.5±0.4	2.5±0.5
600 – 800	3.3±0.7	3.3±0.6	4.0±0.7
800 – 1000	1.4±0.6	1.6±0.5	3.3±0.9
1000 – 1200	3.8±1.2	1.2±0.7	2.4±1.2
1200 – 1400	2.5±1.2	6.0±2.0	3.6±2.1
1400 – 1600	0.0±0.0	2.7±1.9	7.1±6.3

Table 11.6: *Difference of reconstruction efficiencies (in percent) for K^0_S in Monte Carlo generated with* PYTHIA *6.4 between diffractive and non-diffractive events. Due to low statistics, no K^0_S from diffractive events were found in the highest (p_T, y) bin.*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	-	0.4±0.3	0.3±0.2
200 – 400	0.1±0.1	0.5±0.2	0.3±0.2
400 – 600	0.4±0.2	0.4±0.3	-0.2±0.3
600 – 800	1.0±0.4	0.4±0.5	1.7±0.5
800 – 1000	1.0±0.8	-0.2±0.9	0.8±0.9
1000 – 1200	1.2±1.7	1.9±2.2	0.1±1.7
1200 – 1400	1.9±4.2	-3.4±3.8	2.2±3.9
1400 – 1600	-1.3±8.7	4.1±9.3	–

non-diffractive events. The difference between the reconstruction efficiencies obtained in both subsamples can later be used to estimate the systematic uncertainty related to different diffraction models used in the simulation.

Diffractive and non-diffractive events in the standard LHCb Monte Carlo

The standard LHCb Monte Carlo uses PYTHIA 6.4 to generate simulated events[2]. This version of the generator software will only generate soft diffractive events. The fraction of diffractive events among events with a generated K^0_S in the range $2.5 \leq y \leq 4.0$ and $0\,\text{MeV} \leq p_T \leq 1600\,\text{MeV}$ is 11%. The difference in reconstruction efficiency between diffractive and non-diffractive events in the standard LHCb Monte Carlo sample is shown in Table 11.6. The observed difference is not statistically significant.

Hard diffractive events

Since the standard LHCb Monte Carlo does not include hard diffractive events, a sample containing hard diffractive events was generated with PYTHIA 8.1. This sample contains only diffractive events (both soft and hard diffractive events) and is compared to the non-diffractive events in the standard LHCb Monte Carlo. Table 11.7 shows the results. Again, no statistically significant difference was found.

[2]We use PYTHIA 6.421, and include process types 11–13, 28, 53, 68, 91–95, 421–439, 461–479 with non-default parameter values ckin(41)=3.0, mstp(2)=2, mstp(33)=3, mstp(128)=2, mstp(81)=21, mstp(82)=3, mstp(52)=2, mstp(51)=10042, parp(67)=1.0, parp(82)=4.28, parp(89)=14000, parp(90)=0.238, parp(85) =0.33, parp(86)=0.66, parp(91)=1.0, parp(149)=0.02, parp(150)=0.085, parj(11)=0.5, parj(12)=0.4, parj(13) =0.79, parj(14)=0.0, parj(15)=0.018, parj(16)=0.054, parj(17)=0.131, mstj(26)=0, parj(33)=0.4. The particle decay probabilities are computed using EvtGen

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	-	-0.1±0.3	-0.3±0.3
200 – 400	-0.1±0.2	<0.1±0.3	-0.4±0.4
400 – 600	-0.7±0.3	0.3±0.5	-0.4±0.5
600 – 800	0.4±0.6	<0.1±0.9	-0.3±0.9
800 – 1000	<0.1±1.1	-0.4±1.6	-0.6±1.8
1000 – 1200	0.8±2.1	-0.8±2.7	-1.0±2.9
1200 – 1400	-1.7±3.8	–	–
1400 – 1600	0.1±4.5	–	3.6±5.7

Table 11.7: *Difference of reconstruction efficiencies (in percent) for K^0_S in Monte Carlo between diffractive events generated by* PYTHIA *8.1 and non-diffractive events from the standard LHCb Monte Carlo. Due to low statistics, no K^0_S were found in some bins.*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	-	-0.1±0.2	-0.1±0.2
200 – 400	-0.1±0.1	<0.1±0.2	-0.2±0.2
400 – 600	-0.2±0.2	0.1±0.3	-0.8±0.4
600 – 800	-0.6±0.3	0.3±0.4	0.8±0.5
800 – 1000	-0.9±0.5	0.3±0.7	-0.2±0.8
1000 – 1200	-0.6±0.8	0.6±1.0	-0.7±1.1
1200 – 1400	-1.7±1.1	0.6±1.0	0.2±1.3
1400 – 1600	-2.4±1.5	1.7±1.6	0.5±1.6

(a) *Difference between Perugia-0 tune and standard LHCb tune*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	-	-0.1±0.2	0.1±0.2
200 – 400	-0.1±0.1	-0.2±0.2	-0.2±0.2
400 – 600	-0.1±0.2	-0.1±0.3	0.2±0.3
600 – 800	0.3±0.3	<0.1±0.4	0.3±0.5
800 – 1000	-0.3±0.5	-1.2±0.7	0.6±0.8
1000 – 1200	-0.1±0.8	-0.2±1.0	0.5±1.1
1200 – 1400	-0.3±1.1	1.1±1.4	0.1±1.4
1400 – 1600	0.6±1.5	0.1±1.8	0.2±1.8

(b) *Difference between Perugia-NOCR tune and standard LHCb tune*

Table 11.8.: *Difference of reconstruction efficiencies (in percent) for K^0_S in Monte Carlo between Perugia-0 tuning and standard LHCb tuning and Perugia-NOCR tuning and standard LHCb tuning.*

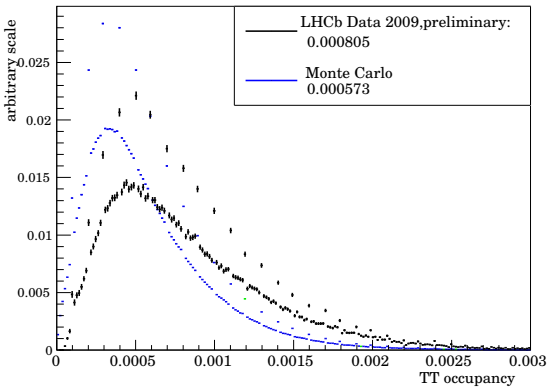
11.5.2. Reconstruction efficiency with different Monte Carlo tunings

To study the influence of the different tunings of the Monte Carlo generator, two more tunings, Perugia-0 and Perugia-NOCR, were compared to the standard LHCb tuning (see [74] for details on the Perugia tunes). The resulting difference in reconstruction efficiency can be found in Table 11.8. Again the differences are negligible. No systematic uncertainty will be assigned for the effect of different Monte Carlo generator tunes on reconstruction efficiency.

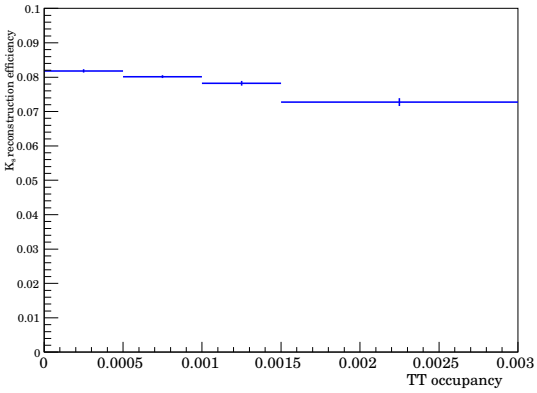
11.6. Reconstruction efficiency as function of occupancy

Detector occupancy in data is 10% to 40% higher in data than in simulated events. Therefore, it is necessary to check if the total reconstruction efficiency depends on the occupancy. Such a cross-check was performed in bins of subdetector occupancy on the full sample of 10 million simulated events, the result can be seen in Figure 11.6. For TT and IT, one can see a drop of about 0.2% when going from the mean occupancy in simulated events to the mean observed in data. With an average reconstruction efficiency of about 8%, this corresponds to a relative drop of about 2.5% of the average efficiency.

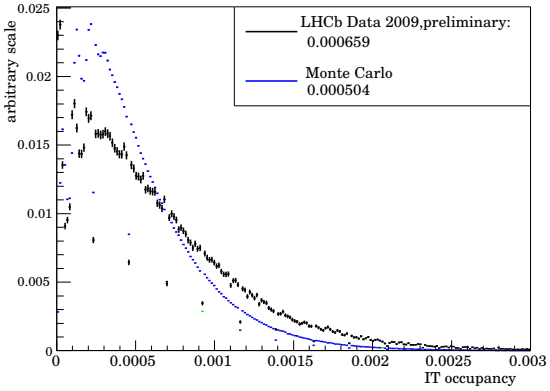
However, it turns out that this drop is not caused by the detector occupancy itself but by K^0_S candidates covering slightly different regions in phase space for low and high



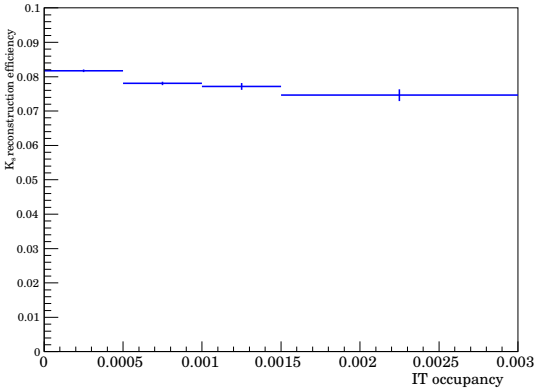
(a) *TT occupancy distribution*



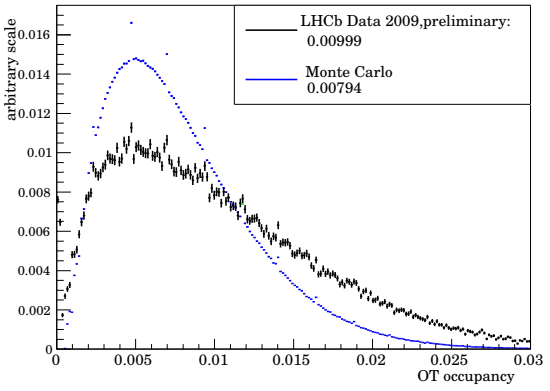
(b) K^0_S *reconstruction eff. vs TT occupancy*



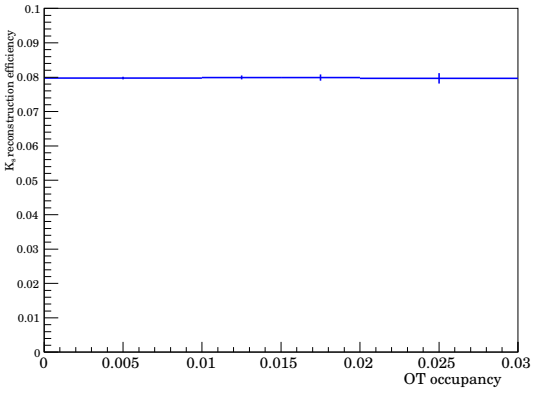
(c) *IT occupancy distribution*



(d) K^0_S *reconstruction eff. vs IT occupancy*



(e) *OT occupancy distribution*



(f) K^0_S *reconstruction eff. vs OT occupancy*

Figure 11.6.: K^0_S *reconstruction efficiency as function of TT, IT and OT occupancy. The distributions do not show large variations as function of the occupancy. The spikes which occur regularly in the occupancy distribution are caused by a combination of the binning chosen and the fact that the occupancy can only take discrete and equally-spaced values (because the number of hits per subdetector is discrete).*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	0.1±0.1	-0.4±0.2	<0.1±0.2
200 – 400	-0.2±0.1	-0.4±0.2	-0.3±0.2
400 – 600	-0.3±0.1	-0.8±0.2	-0.4±0.3
600 – 800	0.3±0.2	-0.9±0.4	-0.9±0.4
800 – 1000	0.1±0.4	-0.2±0.5	-0.2±0.6
1000 – 1200	0.2±0.6	-1.0±0.8	-0.9±0.9
1200 – 1400	0.3±0.9	1.3±1.1	-0.5±1.2
1400 – 1600	1.9±1.3	-2.4±1.5	-1.8±1.5

(a) *TT*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	<0.1±0.1	-0.2±0.2	-0.6±0.2
200 – 400	-0.1±0.1	-0.4±0.2	-0.4±0.2
400 – 600	<0.1±0.2	-0.2±0.3	-0.3±0.4
600 – 800	-0.4±0.3	-1.0±0.5	-1.3±0.5
800 – 1000	0.3±0.6	-0.1±0.7	-0.3±0.8
1000 – 1200	1.4±0.9	-0.7±1.1	0.1±1.2
1200 – 1400	-1.5±1.2	5.4±1.7	<0.1±0.7
1400 – 1600	-0.2±1.9	1.9±2.2	1.3±2.2

(b) *IT*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	0.1±0.1	<0.1±0.2	-0.1±0.2
200 – 400	<0.1±0.1	<0.1±0.2	<0.1±0.2
400 – 600	0.2±0.1	<0.1±0.2	-0.3±0.2
600 – 800	0.3±0.2	-0.2±0.3	-1.0±0.4
800 – 1000	0.9±0.3	-0.4±0.5	<0.1±0.5
1000 – 1200	0.8±0.5	-0.9±0.7	-0.4±0.8
1200 – 1400	1.0±0.8	1.0±1.0	-0.8±1.0
1400 – 1600	-0.5±1.2	-0.9±1.5	-2.7±1.4

(c) *OT*

Table 11.9.: *Difference in K^0_S reconstruction efficiency between events with a TT (a), IT (b) or OT (c) occupancy greater than 0.001 (0.01 for OT) and events with an occupancy smaller than 0.001 (0.01 for OT).*

occupancy events. Therefore, the reconstruction efficiency is studied in bins of p_T and y for high and low occupancy events; the difference in reconstruction efficiency between high and low occupancy events in TT, IT and OT can be found in Table 11.9. It can be seen that the observed difference in reconstruction efficiency is compatible with zero within statistical uncertainties. This leads to the conclusion that detector occupancy in 2009 data does not affect reconstruction efficiency; therefore, no systematic uncertainty will be assigned to detector occupancy.

11.7. Selection

In this section, the influence of the selection on the total reconstruction efficiency is studied. More specifically, one is interested in variations in total reconstruction efficiency because of data/Monte Carlo differences. For example, the average track χ^2/NDF on data is higher than in simulated events because detector alignment and calibration have not reached their final precision.

The preselection applied is very loose and therefore quite efficient ($(89.6 \pm 0.6)\%$). The relevant distributions are shown in Figures 11.7 and 11.8. It can be seen that the agreement between data and simulated events is not an issue in the region(s) where the preselection cuts are performed (cf. Table 9.1) because these cuts are either performed in the tails of the distributions where data and simulated events do not show different behaviour or are done on quantities where data and simulation do not show different

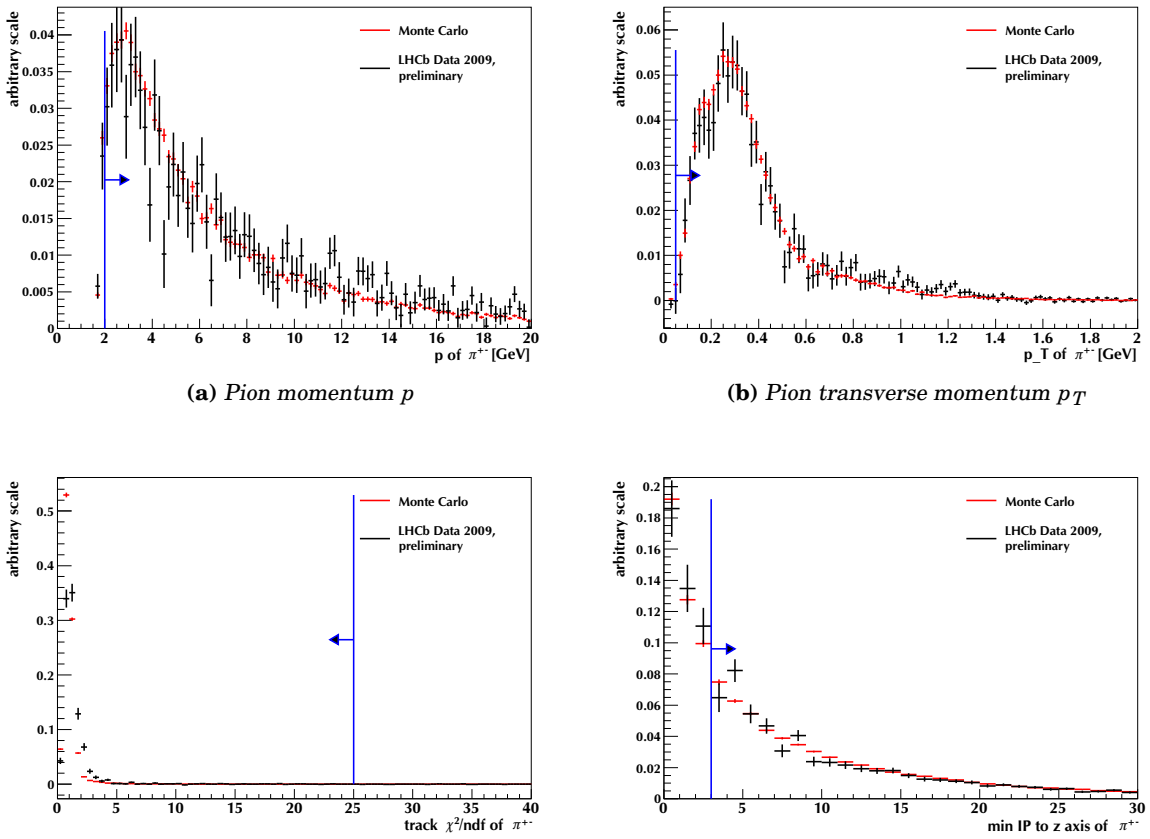
behaviour (e.g. pion transverse momentum, z of decay vertex, z of point of closest approach between z axis and K^0_S candidate). Therefore, the preselection is not assumed to have a different efficiency in data and simulated events. Hence, differences in the efficiency of the selection itself can be studies on a sample of preselected events.

For the cuts in the selection, the agreement is quite good for all quantities which do not involve the extrapolation of a trajectory of a particle over long distances. However, for the pion impact parameter with the z axis (IP), the cosine of the angle between K^0_S momentum and the vector connecting secondary with primary vertex the agreement is not satisfactory in the area where the cut is performed. The same conclusion can be drawn from the K^0_S lifetime as well. This is not surprising because these three quantities are the ones which are most sensitive to residual misalignments in TT, the effects of which are amplified by the long distance extrapolation (about 2 metres) from the first measurement in TT to the luminous region.

The effect of the differences in selection efficiency between data and simulated events is estimated by measuring the efficiency of the selection with respect to preselected events. Signal yields on preselected events are extracted using either a fit (with a double Gaussian for the signal and a quadratic background) or sideband subtraction. The corresponding invariant mass distributions for data and simulated events are shown in Figure 11.9. The results of this study are given in Table 11.10 for the extraction using the fit and the sideband subtraction.

The relative difference in selection efficiency is $\Delta\varepsilon = (\varepsilon_{\text{Data}} - \varepsilon_{\text{MC}})/\varepsilon_{\text{MC}} = (7 \pm 4)\%$. To check if this difference depends on p_T or y of the K^0_S candidate, Table 11.10 also contains corresponding results in two bins of p_T and y, the outcome in each bin is compatible with the $(7 \pm 4)\%$ difference observed integrated over all p_T and y, so that the difference appears to be independent of p_T and y within its uncertainties.

Since the difference in selection efficiency can be measured, the appropriate correction of 7% is made. Since the dominating uncertainty in the determination of $\Delta\varepsilon$ is the extraction of the K^0_S yield on the sample of preselected events in 2009 data, a 4% systematic uncertainty is assigned to the relative difference in selection efficiency between data and simulation.



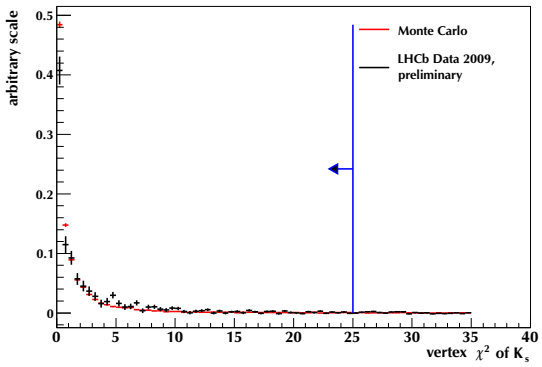
(a) *Pion momentum p*

(b) *Pion transverse momentum p_T*

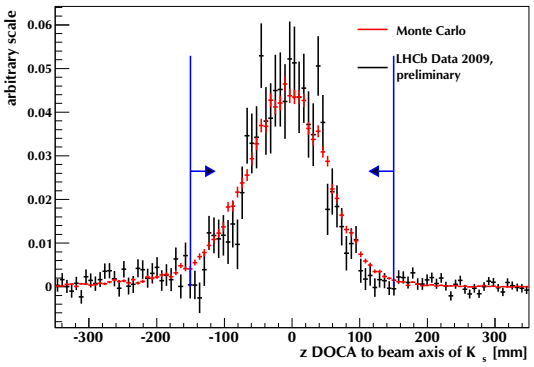
(c) *track χ^2/NDF*

(d) *Pion impact parameter with z axis (IP)*

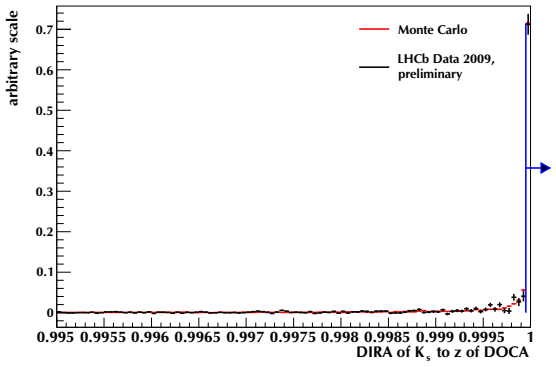
Figure 11.7.: *Cut variable distributions for signal K^0_S candidates in data (black) and Monte Carlo (red) after the preselection for track level quantities. Final selection cuts on all quantities except the one on the x axis of each plot have been applied, i.e. the plots are "$n-1$ plots" in the sense that they show the effect of all cuts but one. The blue lines indicate the cut values.*



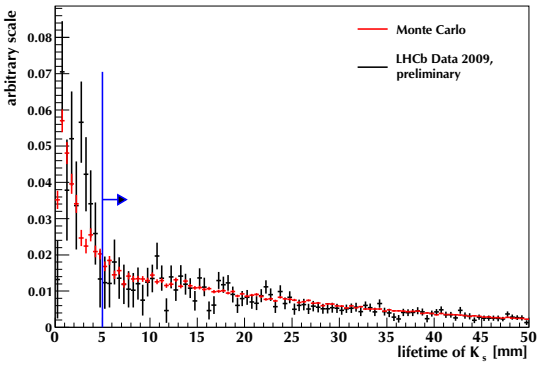
(a) K^0_S *vertex* χ^2/NDF



(b) *z coordinate of point of closest approach between* K^0_S *and z axis*

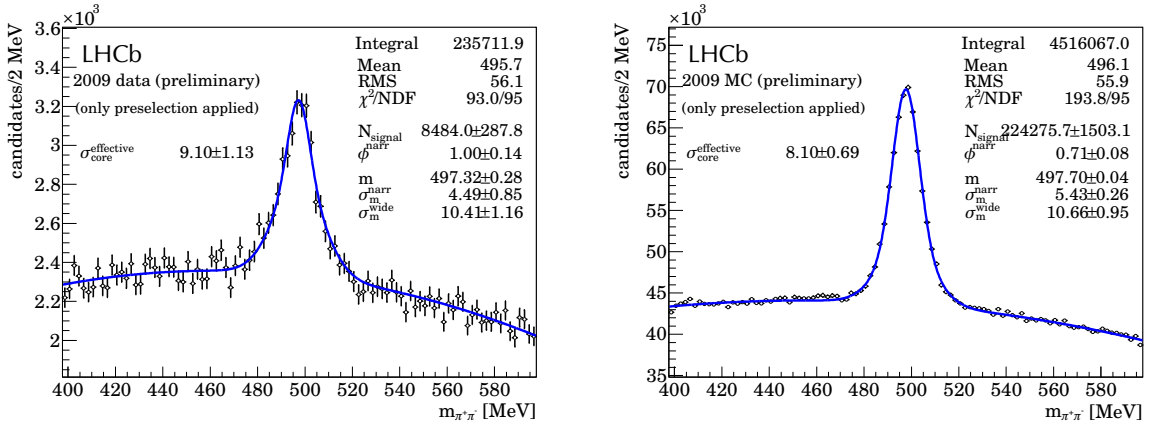


(c) *Cosine of angle between* K^0_S *momentum and vector connecting secondary with primary vertex*



(d) K^0_S *lifetime ct*

Figure 11.8.: *Cut variable distributions for signal* K^0_S *candidates in data (black) and Monte Carlo (red) after the preselection for* K^0_S *level quantities. Final selection cuts on all quantities except the one on the x axis of each plot have been applied, i.e. the plots are "*$n-1$ *plots" in the sense that they show the effect of all cuts but one. The blue lines indicate the cut values.*



(a) *2009 data* **(b)** *Monte Carlo*

Figure 11.9.: *K^0_S candidate mass distribution for 2009 data and simulated events after the preselection.*

bin		$\varepsilon_{fit.data}$	$\varepsilon_{fit.MC}$	$\Delta\varepsilon = (\varepsilon_{fit.data} - \varepsilon_{fit.MC})/\varepsilon_{fit.MC}$
whole sample		56 ± 2	53 ± 1	7 ± 4
$2.5 < y < 3.4$	$200 < p_T/\text{MeV} < 1600$	60 ± 7	53 ± 1	14 ± 13
$3.4 < y < 4.0$	$200 < p_T/\text{MeV} < 1600$	56 ± 7	58 ± 1	-4 ± 12
$2.5 < y < 4.0$	$200 < p_T/\text{MeV} < 600$	57 ± 4	52 ± 1	9 ± 7
$2.5 < y < 4.0$	$600 < p_T/\text{MeV} < 1600$	63 ± 3	58 ± 1	9 ± 5

(a) *efficiencies extracted using fit*

bin		$\varepsilon_{SB.data}$	$\varepsilon_{SB.MC}$	$\Delta\varepsilon = (\varepsilon_{SB.data} - \varepsilon_{SB.MC})/\varepsilon_{SB.MC}$
whole sample		58 ± 5	52 ± 1	11 ± 9
$2.5 < y < 3.4$	$200 < p_T/\text{MeV} < 1600$	64 ± 9	53 ± 1	22 ± 17
$3.4 < y < 4.0$	$200 < p_T/\text{MeV} < 1600$	55 ± 7	57 ± 1	-4 ± 12
$2.5 < y < 4.0$	$200 < p_T/\text{MeV} < 600$	57 ± 9	53 ± 1	7 ± 16
$2.5 < y < 4.0$	$600 < p_T/\text{MeV} < 1600$	63 ± 7	57 ± 1	11 ± 12

(b) *efficiencies extracted using sideband subtraction*

Table 11.10.: *Efficiency in percent of the selection cuts with respect to preselected K^0_S signal candidates on data and Monte Carlo. The efficiencies are obtained either by fitting a double Gaussian with a quadratic background or by using sideband subtraction. Bin boundaries have been chosen so as to ensure an equal amount of K^0_S candidates in each bin.*

12. Determination of trigger efficiency

In this chapter, the trigger efficiency is determined. It will be determined on a sample of simulated events, because the size of the 2009 data sample does not allow to extract the trigger efficiency simultaneously as function of two variables (p_T and y). Therefore, it is important to check the data/Monte Carlo agreement first. To this end, several methods to measure the trigger efficiency are evaluated, both data- and Monte Carlo-driven. This allows the determination of trigger efficiencies on simulated events to be cross-checked with data in a controlled way, despite different measurement methods being used. Of course, the modified sample of simulated events produced as a result of the studies done in Chapter 10 was used for all plots and figures to keep the data/Monte Carlo discrepancy to a minimum.

The treatment described here is based on the work done in [70]; plots and tables in this chapter were also taken from that study.

12.1. Data/Monte Carlo agreement

12.1.1. Method

The trigger efficiency is measured on selected events which contain a reconstructed K^0_S ("selected events"). Let the number of reconstructed and selected events be N_{rec}, among which N_{trig} events have triggered the calorimeter trigger. The trigger efficiency can then be written as:

$$\varepsilon_{trig} = \frac{N_{trig}}{N_{rec}} \tag{12.1}$$

Among the triggered events, there are two categories of events: An event can *trigger on signal* (TOS), i.e. the trigger is caused by one of daughters of the K^0_S, or an event can *trigger independent of signal* (TIS), so the trigger is caused by a track which does not have anything to do with the decay of the K^0_S. The two categories are not mutually exclusive; for example, an event can be triggered by both a calorimeter trigger and a muon trigger, the former being caused by a K^0_S daughter and the latter being independent of signal. Let the number of events which are TOS be called N_{TOS}, the number of events which are TIS N_{TIS} and the number of events which are both TIS and TOS N_{TISTOS}. Events is this category are also called TISTOS events. The relative frequencies with which these categories occur in data can be found in Table 12.1.

Clearly, the relation $N_{trig} = N_{TIS} + N_{TOS} - N_{TISTOS}$ holds. With this relation, the trigger efficiency can be rewritten in terms of the trigger efficiency for TIS events, the efficiency for TOS events and the efficiency for events which are both TIS and TOS:

$$\varepsilon_{trig} = \frac{N_{trig}}{N_{rec}} = \frac{N_{TIS} + N_{TOS} - N_{TISTOS}}{N_{rec}} = \varepsilon_{TIS} + \varepsilon_{TOS} - \varepsilon_{TISTOS} \tag{12.2}$$

N_{rec}	N_{TIS}	N_{TOS}	N_{TISTOS}	$N_{TIS} - N_{TISTOS}$	$N_{TOS} - N_{TISTOS}$
8438	8226	4588	4376	3850	212

Table 12.1.: *Number of K^0_S candidates in data in TIS and TOS events.*

Assuming that the probabilities for a triggered event to be TIS and to be TOS are independent of each other, this can be further rewritten:

$$\varepsilon_{trig} = \varepsilon_{TIS} + \varepsilon_{TOS} - \varepsilon_{TISTOS} = \varepsilon_{TIS} + \varepsilon_{TOS} - \varepsilon_{TIS} \cdot \varepsilon_{TOS} \tag{12.3}$$

By the exploiting the independence of TIS and TOS, the TIS and TOS efficiencies can also be expressed in terms of the overlap of the TIS and TOS categories as:

$$\varepsilon_{TIS} = \frac{N_{TISTOS}}{N_{TOS}} \qquad \varepsilon_{TOS} = \frac{N_{TISTOS}}{N_{TIS}} \tag{12.4}$$

The following uncertainties are assigned to ε_{TIS} and ε_{TOS}, using simple error propagation and taking into account the correlations between numerators and denominators of ε_{TIS} and ε_{TOS} as given in equation 12.4:[1]

$$\sigma_{\varepsilon_{TIS}} = \frac{1}{N_{TOS}} \sqrt{(1-2\varepsilon_{TIS})\sigma^2_{N_{TISTOS}} + \varepsilon^2_{TIS}\sigma^2_{N_{TOS}}} \tag{12.5}$$

$$\sigma_{\varepsilon_{TOS}} = \frac{1}{N_{TIS}} \sqrt{(1-2\varepsilon_{TOS})\sigma^2_{N_{TISTOS}} + \varepsilon^2_{TOS}\sigma^2_{N_{TIS}}} \tag{12.6}$$

To check that the assumption of TIS and TOS being independent, the TIS and TOS efficiencies are measured on simulated events as suggested in equation 12.4 and using the traditional method of using $\varepsilon_{TIS} = \frac{N_{TIS}}{N_{rec}}$ and $\varepsilon_{TOS} = \frac{N_{TOS}}{N_{rec}}$. Figure 12.1 shows the results for ε_{TIS} and ε_{TOS} determined using both methods. The results agree to much better than 1%, hence the assumption of independence of TIS and TOS is valid. A systematic uncertainty of 1% is thus assigned to the trigger efficiency to account for the agreement between the traditional method and the one in equation 12.4.

The trigger efficiency in equation 12.2 can then be rewritten as:

$$\begin{aligned} \varepsilon_{trig} &= \varepsilon_{TIS} + \varepsilon_{TOS} - \varepsilon_{TIS} \cdot \varepsilon_{TOS} = \frac{N_{TISTOS}}{N_{TOS}} + \frac{N_{TISTOS}}{N_{TIS}} - \frac{N^2_{TISTOS}}{N_{TIS}N_{TOS}} \\ &= \frac{N_{TISTOS}(N_{TIS} + N_{TOS} - N_{TISTOS})}{N_{TIS}N_{TOS}} = \frac{N_{TISTOS}N_{trig}}{N_{TIS}N_{TOS}} \end{aligned} \tag{12.7}$$

This form of the trigger efficiency can be used on both simulated events and real data because only triggered events and the number of TIS, TOS and TISTOS events among them are used, all of which are easily available in data and simulated events. It can

[1] Since $\sigma_{N_{TIS}}$, $\sigma_{N_{TOS}}$ and $\sigma_{N_{TISTOS}}$ are obtained from a fit to the invariant mass distribution of the two pion combination, there is a small possibility of getting an undefined value for the resulting uncertainties on TIS and TOS trigger efficiencies. In these rare cases, $\sigma_{N_{TIS}} = \sqrt{N_{TIS}}$, $\sigma_{N_{TOS}} = \sqrt{N_{TOS}}$ and $\sigma_{N_{TISTOS}} = \sqrt{N_{TISTOS}}$ are substituted to regularise the computation.

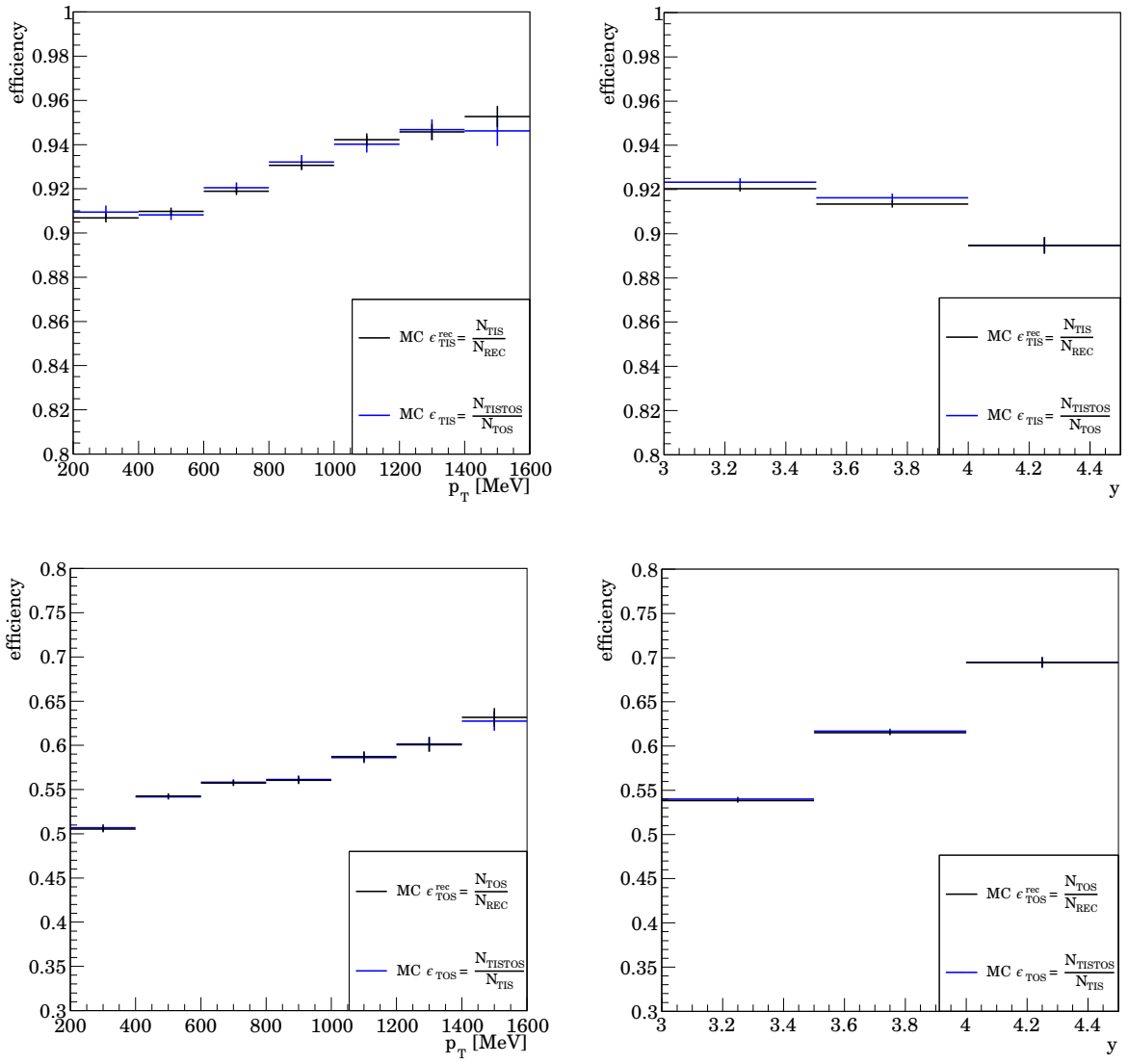


Figure 12.1.: *ε_{TIS} and ε_{TOS} measured in simulated events determined using two different approaches. The efficiencies obtained agree well with each other.*

therefore be used to study the degree of agreement of the trigger efficiency in data and simulated events.

However, on simulated events, the traditional method of using $\varepsilon_{trig} = \frac{N_{trig}}{N_{rec}}$, $\varepsilon_{TIS} = \frac{N_{TIS}}{N_{rec}}$ and $\varepsilon_{TOS} = \frac{N_{TOS}}{N_{rec}}$ is more sensitive because the underlying sample of events is larger than the sample that would be used to determine efficiencies according to equations 12.4 and 12.7. Since both methods agree well, the traditional method will be used on simulated events while equations 12.4 and 12.7 will be used on data.

12.1.2. Results

Figure 12.2 shows the results for TIS and TOS trigger efficiencies in 2009 data and simulated events. The agreement is acceptable, with simulated events having a lower trigger efficiency than data.

The main reason for this discrepancy lies in the different track multiplicities of 2009 data and simulated events (cf. Figure 12.3). With simulated events having the lower track multiplicities, it is not surprising that the trigger efficiency on simulated events should also be lower because there are fewer tracks in the event which can actually cause a positive trigger decision. For this reason, the sample of simulated events is reweighted to reproduce the downstream track multiplicity on selected events in 2009 data for the determination of the trigger efficiency. The corresponding per-event weights as function of downstream track multiplicity can also be found in Figure 12.3.

12.1.3. Results with track multiplicity reweighting

Figure 12.4 shows the results for TIS and TOS trigger efficiencies in 2009 data and simulated events which were reweighted to exhibit the same downstream track multiplicity found in 2009 data. Clearly, the data/Monte Carlo agreement is improved somewhat.

For TIS events, 1% systematic uncertainty is assigned for data/Monte Carlo agreement. For TOS events, 5% are used. Since the fraction of TOS events which are not also TIS is only 2.5% (cf. Table 12.1), the total systematics assigned to the trigger efficiency ε_{trig} is 2.5%.

12.2. Determination of trigger efficiency

The total trigger efficiency is determined on Monte Carlo by using in bins of p_T and y:

$$\begin{aligned} \varepsilon_{trig} &= \frac{N_{trig}}{N_{rec}} \\ \sigma_{\varepsilon_{trig}} &= \frac{1}{N_{rec}} \sqrt{(1-2\varepsilon_{trig})\sigma^2_{N_{trig}} + \varepsilon^2_{trig}\sigma^2_{N_{rec}}} \end{aligned} \tag{12.8}$$

The results can be found in Table 12.2. As a cross-check, the total trigger efficiency was determined using equation 12.7) on selected events, both for simulated events and for data, as a function of p_T. As can be seen in Figure 12.5, the agreement is good within the 2.5% of systematic uncertainty.

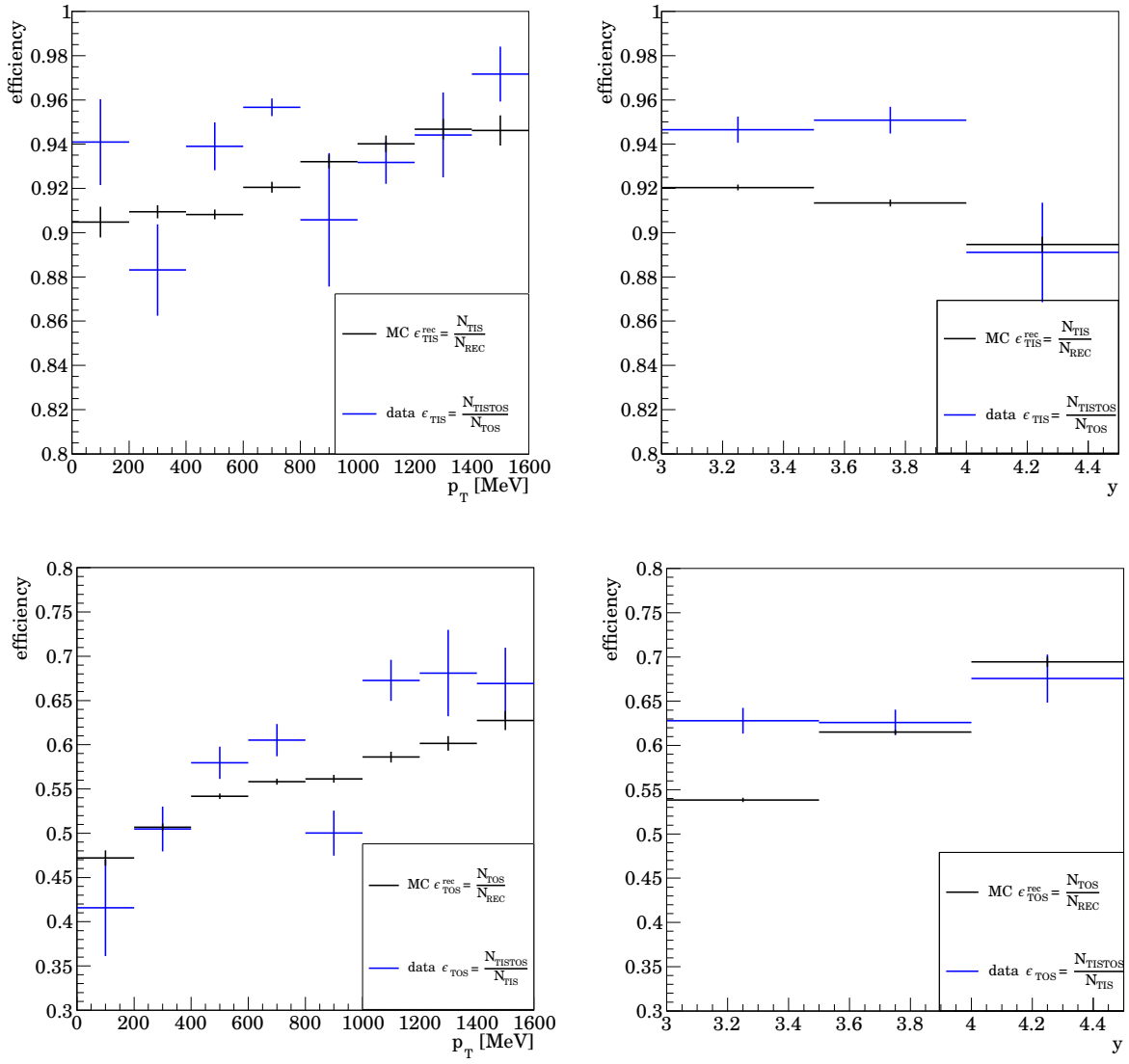


Figure 12.2.: *ε_{TIS} and ε_{TOS} measured in 2009 data and in simulated events as function of p_T and y.*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0– 200	–	96.9±0.3	100± –
200– 400	96.5±0.1	97.0±0.2	97.6±0.5
400– 600	97.2±0.1	97.5±0.1	97.2±0.4
600– 800	97.5±0.1	97.6±0.2	98.0±0.1
800– 1000	97.7±0.2	97.7±0.2	98.4±0.4
1000– 1200	99.3±0.1	98.8±0.2	98.3±0.6
1200– 1400	99.0±0.3	97.6±2.4	98.8±1.3
1400– 1600	99.1±0.3	98.7±0.3	99.0±1.0

Table 12.2: *Trigger efficiencies (ε_{trig}, in percent) derived from Monte Carlo data.*

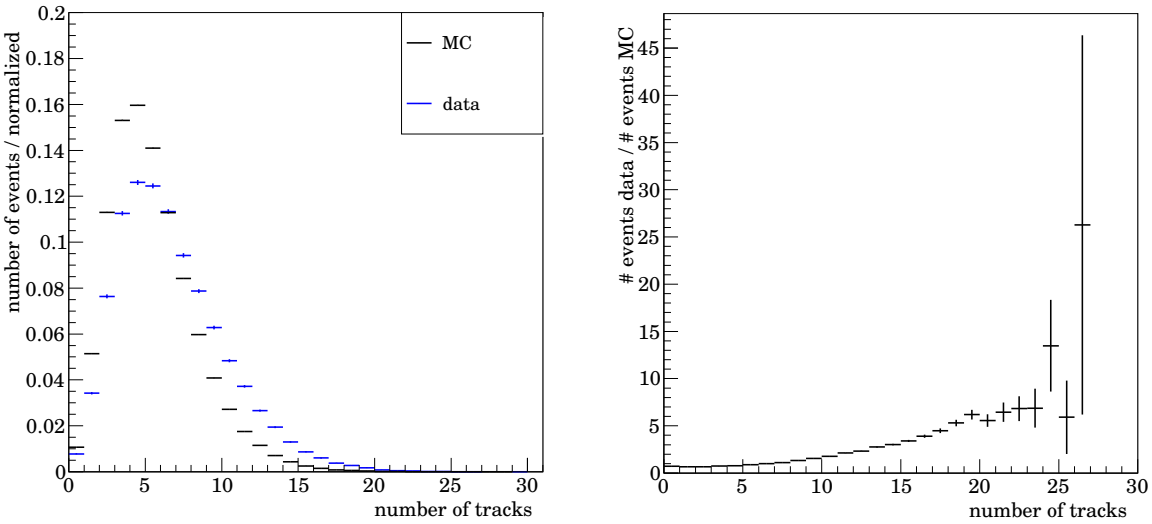


Figure 12.3.: *Downstream track multiplicity (left) in 2009 data and simulated events and the ratio used for reweighting (right).*

12.3. Trigger efficiencies in diffractive events

The standard LHCb Monte Carlo sample contains a fraction of 11% of diffractive events. With the reconstruction efficiency being equal for diffractive and non-diffractive events withing statistical uncertainties, the fraction of diffractive events among selected events is also 11%.

The trigger efficiency is different for different event types. The non-diffractive and diffractive components of the standard LHCb Monte Carlo (PYTHIA 6.4) were separated, and trigger efficiencies were determined on the two resulting subsamples. A sample of diffractive events generated with PYTHIA 8.1 was also studied, see Table 12.3 for the results for all three samples (the standard multiplicity reweighting has been applied).

To demonstrate that the difference is not due to different track multiplicities in diffractive and non-diffractive events which would need reweighting, Table 12.4 shows the trigger efficiencies for different downstream multiplicities (with the two K^0_S daughter tracks removed. The corresponding contribution to the systematic uncertainty will be discussed later.

12.4. Differences in trigger emulation in data and simulation

There is a minor inconsistency in how the LHCb software framework checks for L0 calorimeter triggers in data and simulation. The condition for an event to trigger on data is that it must have a 2×2 cluster in the hadronic calorimeter with more than 240 MeV of transverse energy and at least three hits in the tiles of the scintillating pad detector (SPD). In simulated events, the SPD is never checked by the software. This means that events which have only the charged K^0_S daughters but no other charged particles in the

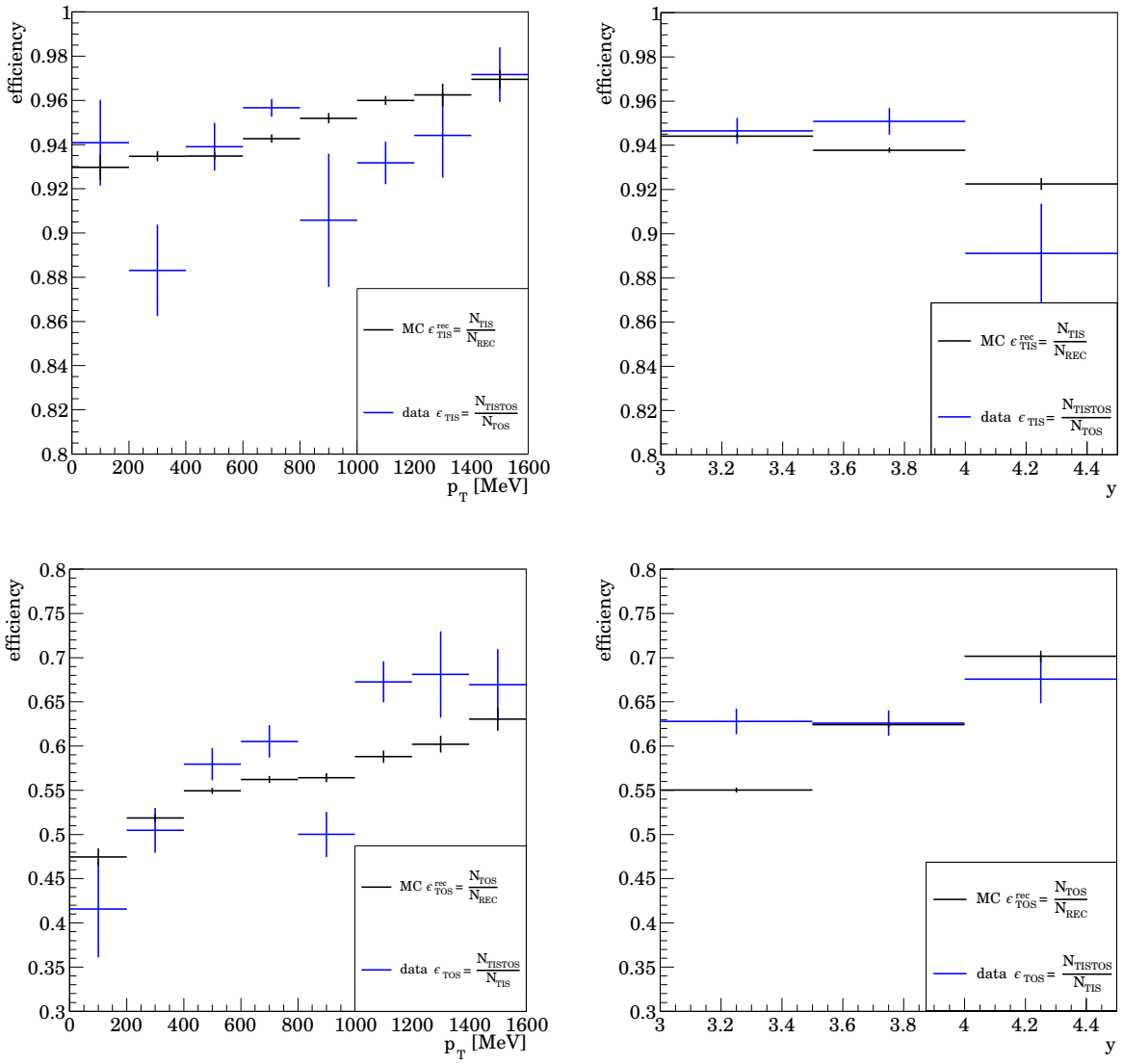


Figure 12.4.: *ε_{TIS} and ε_{TOS} measured in 2009 data and in Monte Carlo as function of p_T and y after correction for the downstream track multiplicity.*

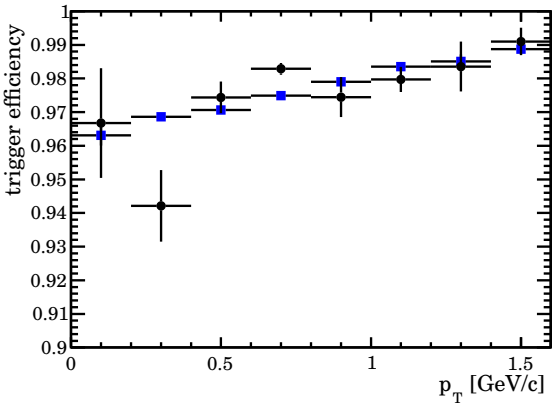


Figure 12.5: *Trigger efficiency for events containing a signal K^0_S decay in the downstream track selection, as a function of the K^0_S p_T , estimated both in data (black filled circles) and MC (blue open squares), using equation 12.7.*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0– 200	–	97.1±0.4	94.4±1.8
200– 400	97.7±0.1	98.1±0.1	97.3±0.6
400– 600	98.2±0.1	98.3±0.1	98.2±0.2
600– 800	98.2±0.1	98.2±0.1	98.4±0.3
800– 1000	98.1±0.2	98.1±0.2	98.4±0.5
1000– 1200	99.3±0.2	99.0±0.2	98.4±0.6
1200– 1400	98.9±0.5	98.9±0.3	98.8±1.3
1400– 1600	99.1±0.3	98.7±0.3	99.0±1.0

(a) *Trigger efficiencies (ε_{trig}, in percent) for non diffractive events.*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0– 200	–	88.6±0.4	90.6± 4.5
200– 400	86.6±1.0	87.7±0.8	85.3± 6.5
400– 600	86.7±0.9	89.9±0.8	92.1± 0.8
600– 800	87.8±1.1	92.7±1.1	92.7± 1.4
800– 1000	90.0±1.6	92.6±1.5	91.4± 2.5
1000– 1200	89.8±3.4	98.0±1.9	89.9±12.9
1200– 1400	100.0	97.9±4.4	100.0
1400– 1600	100.0	100.0	100.0

(b) *Trigger efficiencies (ε_{trig}, in percent) for diffractive events generated with* PYTHIA *6.4*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0– 200	–	95.6±1.0	92.3±8.5
200– 400	92.7±1.5	94.6±1.0	99.2±0.7
400– 600	93.5±1.0	94.8±1.0	94.9±0.9
600– 800	93.9±1.0	93.9±1.2	97.4±1.2
800– 1000	97.3±1.5	93.1±3.1	100.0
1000– 1200	100.0	83.9±6.4	100.0
1200– 1400	93.2±4.2	100.0	100.0
1400– 1600	100.0	100.0	100.0

(c) *Trigger efficiencies (ε_{trig}, in percent) for diffractive events generated with* PYTHIA *8.1.*

Table 12.3.: *Trigger efficiencies (ε_{trig}, in percent) for diffractive and non-diffractive events.*

Table 12.4: *Trigger efficiencies (ε_{trig}, in percent) as function of downstream track multiplicity for different data types.*

# of downstream tracks	non-diffractive	diffractive PYTHIA 6.4	diffractive PYTHIA 8.1
2–3	93.2± 0.2	83.1± 0.4	88.9± 0.7
4–5	97.4± 0.1	93.5± 0.3	95.8± 0.5
6–7	99.0± 0.1	97.8± 0.4	98.6± 0.4
8–9	99.6± 0.1	96.2± 1.2	99.1± 0.5
10–11	99.8± 0.1	100.0	100.0
12–13	99.9± 0.1	100.0	100.0
≥ 14	100.0	100.0	100.0

$p_T[MeV]/y$	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0− 200	-	3.3±0.1	2.8±0.1
200− 400	1.2±0.1	7.0±0.1	7.1±0.1
400− 600	3.4±0.1	11.3±0.1	11.5±0.1
600− 800	7.1±0.1	14.4±0.2	14.3±0.2
800−1000	10.7±0.2	16.4±0.2	15.0±0.3
1000−1200	13.6±0.3	17.7±0.4	14.4±0.4
1200−1400	15.2±0.4	18.1±0.5	12.8±0.6
1400−1600	16.8±0.6	18.1±0.7	11.9±0.7

Table 12.5: *Reconstruction efficiencies (in percent) when requiring in addition to the K^0_S daughters at least 3 prompt particles with $p > 1\,\mathrm{GeV}$ in the detector acceptance on generator level.*

acceptance of the detector will not trigger on data but will trigger on simulated events. The effect on the trigger efficiency is not larger than 0.5%. However, it might be possible that reconstruction efficiency is affected because certain simulated events are triggered in the trigger simulation which would be discarded by the trigger in real data.

To check that the difference this causes in the sample of events which is selected for the determination of reconstruction efficiency is negligible, the reconstruction efficiency is determined again with the default definitions from last chapter, but events are required to have at least three generated prompt tracks with $p > 1\,\mathrm{GeV}$ in the detector acceptance ($2 \leq \eta \leq 5$). This ensures that the SPD will trigger. The results of this study can be found in Table 12.5. When compared with the figures in Table 11.1, there is practically no difference. The effect of this difference in trigger emulation on the measured trigger efficiencies will be treated as systematics.

13. Systematics

In this chapter, the different sources of systematic uncertainties are presented.

13.1. Tracking efficiency

In Sections 10.2.2 and 10.3, two methods were developed to evaluate the difference in tracking efficiency in data and Monte Carlo, one based on Monte Carlo studies, the other data-driven.

13.1.1. Monte Carlo driven method

As described in Section 10.2.2, two samples of simulated events were prepared, under- or over-correcting for the effective hit finding efficiency. Assuming the different effective hit finding efficiencies are the driving force behind any differences in tracking efficiency between data and simulated events, it is possible to obtain an estimate for the systematic uncertainty due to the effective hit finding efficiency correction by checking the total reconstruction efficiencies between the over- and the under-corrected sample. The reconstruction efficiencies for these two samples can be found in Table 13.1; for each bin, Table 13.2 shows whichever sample has the larger difference with respect to the reconstruction efficiency with the standard hit efficiency correction factors.

13.1.2. Data-driven method

In Section 10.3, a data-driven method to evaluate systematic uncertainties due to differences in tracking efficiency between data and Monte Carlo was introduced. As a result of

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	–	3.4±0.1	2.9±0.1
200 – 400	1.3±0.1	7.2±0.1	7.7±0.1
400 – 600	3.6±0.1	11.8±0.1	12.4±0.1
600 – 800	7.4±0.1	15.2±0.2	15.3±0.2
800 – 1000	11.1±0.2	17.3±0.2	16.0±0.3
1000 – 1200	14.1±0.3	18.7±0.4	15.5±0.4
1200 – 1400	15.9±0.4	19.2±0.5	14.0±0.6
1400 – 1600	17.6±0.6	18.9±0.7	12.4±0.7

(a) *"+RMS"*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	–	3.0±0.1	2.6±0.1
200 – 400	1.2±0.1	6.6±0.1	6.4±0.1
400 – 600	3.3±0.1	10.5±0.1	10.4±0.1
600 – 800	6.7±0.1	13.4±0.2	12.8±0.2
800 – 1000	10.0±0.2	15.2±0.2	13.3±0.3
1000 – 1200	12.7±0.3	16.4±0.3	12.7±0.4
1200 – 1400	14.4±0.4	17.3±0.5	11.8±0.5
1400 – 1600	15.8±0.6	16.2±0.7	10.3±0.7

(b) *"-RMS"*

Table 13.1.: *Reconstruction efficiencies (in percent) derived from Monte Carlo, over-correcting ("+RMS") or under-correcting ("-RMS") for the effective hit finding efficiency.*

Table 13.2: *Absolute systematic uncertainties to the reconstruction efficiency (in percent) assigned to the remaining data and Monte Carlo discrepancy with the Monte Carlo driven method.*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	–	0.3	0.2
200 – 400	0.1	0.4	0.7
400 – 600	0.2	0.8	1.1
600 – 800	0.4	1.0	1.5
800 – 1000	0.7	1.2	1.7
1000 – 1200	0.9	1.3	1.7
1200 – 1400	0.9	1.1	1.2
1400 – 1600	1.0	1.9	1.6

Table 13.3: *Relative systematic uncertainty (percent) on the (sideband subtracted) yield on data due to discrepancies between data and Monte Carlo on reconstruction efficiencies. The numbers in brackets are the limit we place due to the statistical uncertainties on the measurement of reconstruction efficiency in data.*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	–	8	6
200 – 400	8	7	8
400 – 600	5 (6)	4 (6)	5 (6)
600 – 800	3 (6)	2 (6)	3 (6)
800 – 1000	1 (6)	1 (6)	2 (6)
1000 – 1200	1 (6)	1 (6)	2 (6)
1200 – 1400	<1 (6)	<1 (6)	<1 (6)
1400 – 1600	1 (6)	0 (6)	0 (6)

this method, a relative reconstruction efficiency of data with respect to Monte Carlo was established as function of the track p_T: $\varepsilon_{rel} = \frac{\varepsilon_{data}}{\varepsilon_{MC}} = 91\%$ for $p_T < 200\,\text{MeV}$, $\varepsilon_{rel} = 99\%$ for $200\,\text{MeV} \leq p_T \leq 400\,\text{MeV}$ and $\varepsilon_{rel} = 101\%$ for $p_T > 400\,\text{MeV}$. This can be used to reweight K^0_S candidates in data based on the p_T of the daughter tracks; specifically, each daughter track is assigned a weight of $1/\varepsilon_{rel}(p_T)$, and the weights of the two daughters are multiplied to obtain the weight for the K^0_S. Yields are extracted using sideband subtraction (the reweighting can distort the shape of the peak in low statistics bins, influencing the fit result), and these yields are compared to the non-reweighted yields obtained using sideband subtraction in Table 9.5. The relative difference is given in Table 13.3. It will be used as an estimate for the systematic uncertainty of the total reconstruction efficiency due to data/Monte Carlo track reconstruction efficiency differences.

The statistical precision of the tracking efficiency determination is not better than 2-3% per track, so data/Monte Carlo agreement cannot be verified to better than 6% per K^0_S. For that reason a minimum systematics of 6% is quoted in each bin.

Since this data-driven estimate is using Velo-Calo tracks, the tracks used were restricted to the reduced acceptance of the half-open Velo. It is not clear that the simulation of the half-open Velo perfectly reflects the situation in the detector during the 2009 run. Hence, it is necessary to check the data/Monte Carlo agreement in variables which are sensitive to the different acceptance of tracks with Velo information and downstream tracks. If the agreement is good, the results of the data-driven method can be transferred from tracks in the (reduced) Velo acceptance to the downstream track acceptance.

To this end, the distributions of the reconstructed angle ϕ (angle of rotation around the z axis) and position z of the K^0_S decay vertex are shown in Figure 13.1. The shapes for data and simulated events agree well and are not expected to depend on any specific

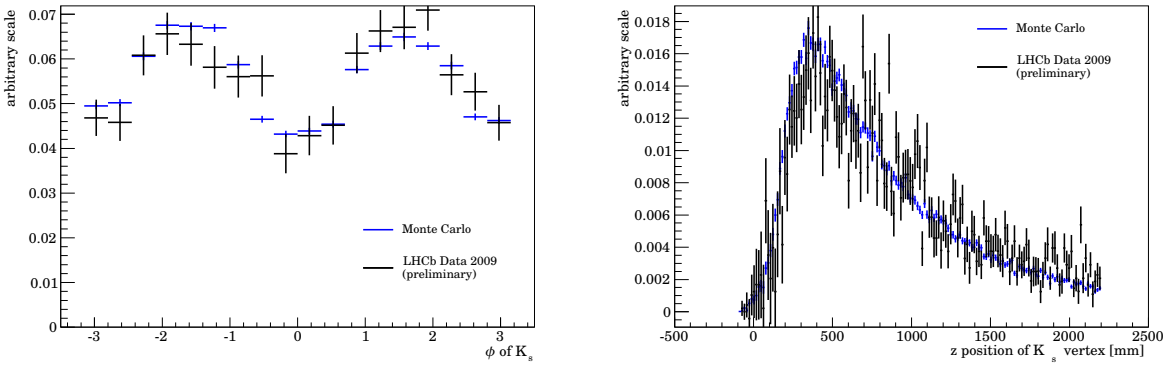


Figure 13.1.: *Data and Monte Carlo comparison for reconstructed ϕ of K^0_S signal candidates and for the z position of decay vertex of K^0_S signal candidates.*

model. Therefore, the reconstruction efficiency in data and simulation behave very similar as function of these two quantities.

In general, the results of the data-driven and the Monte Carlo-driven method are in good agreement (roughly within 10% relative to the measured efficiency). As expected, the data-driven (and p_T dependent) correction affects low-momentum bins more than high momentum bins. For this analysis, the data-driven method will be used as the estimate of tracking efficiency related systematics. The Monte Carlo based method remains a valuable cross check.

13.2. Variation of reconstruction efficiency inside a bin

The efficiency corrections which will be applied have been determined in bins of p_T and y. For given bins of p_T and y, the correction factors obtained on simulated events can only be transferred to data if the distributions of candidates inside each bin in data and simulation approximately agree or if the efficiency inside a bin is essentially constant in the bin.

To check how well this assumption holds, the reconstruction efficiency was also determined in smaller bins by subdividing each bin in p_T and y into four subbins. For each subbin, the efficiency-corrected yield on data is extracted by sideband subtraction using the inverse efficiency in that subbin as weight (cf. Table 13.4). The reason for using sideband subtraction is that the statistics in some subbins is so low that the fit will no longer converge reliably. For each of the original bins, the efficiency-corrected yield extracted using the standard efficiency correction weights obtained with the wider bins can be compared to yield in the corresponding four finer subbins. The result can be found in Table 13.5. The difference between the standard and the fine binning is typically small, except for low p_T and y. The affected bins are those bins in which the reconstruction efficiency in the corresponding four subbins shows a large variation, so the picture is consistent with expectations. The relative differences in Table 13.5 will be used as a bin-wise estimate of the systematic error.

p_T [MeV] / y	2.5 - 2.75	2.75 - 3.0	3.0 - 3.25	3.25 - 3.5	3.5 - 3.75	3.75 - 4.0
0 – 100	—	—	1.20±0.13	1.53±0.14	1.37±0.13	0.73±0.10
100 – 200	—	—	3.41±0.13	4.77±0.14	4.21±0.14	2.57±0.11
200 – 300	0.07±0.02	1.67±0.08	4.70±0.12	6.90±0.15	6.89±0.16	4.77±0.14
300 – 400	0.35±0.04	2.86±0.09	6.80±0.15	9.60±0.18	9.38±0.18	7.29±0.17
400 – 500	0.89±0.06	4.68±0.12	9.05±0.17	11.99±0.21	11.45±0.21	9.83±0.21
500 – 600	2.12±0.09	6.60±0.16	11.36±0.21	13.44±0.24	13.32±0.26	11.80±0.26
600 – 700	3.60±0.13	9.09±0.22	13.19±0.27	14.49±0.29	14.40±0.31	13.34±0.33
700 – 800	5.54±0.19	11.16±0.27	14.89±0.33	15.64±0.35	15.89±0.40	13.26±0.39
800 – 900	7.81±0.27	12.45±0.34	15.87±0.41	16.80±0.45	17.07±0.50	13.51±0.49
900 – 1000	8.78±0.33	14.71±0.45	16.57±0.51	16.79±0.55	16.43±0.58	13.68±0.58
1000 – 1100	10.53±0.43	15.99±0.55	16.49±0.61	17.12±0.67	15.93±0.74	13.81±0.74
1100 – 1200	12.38±0.55	16.58±0.68	18.47±0.77	19.70±0.88	14.62±0.87	11.53±0.82
1200 – 1300	13.16±0.67	17.06±0.60	17.51±0.91	18.65±1.04	12.73±0.95	12.10±0.98
1300 – 1400	13.79±0.80	18.58±1.01	16.66±1.03	20.89±1.29	13.56±1.15	11.41±1.16
1400 – 1500	14.26±0.98	18.42±1.21	18.99±1.29	16.93±1.35	11.52±1.22	10.36±1.32
1500 – 1600	15.85±1.16	19.52±1.39	18.41±1.47	16.11±1.48	12.22±1.46	11.17±1.58

Table 13.4.: *Reconstruction efficiencies (in percent) derived from Monte Carlo in finer bins.*

Table 13.5: *Difference in efficiency corrected K^0_S signal yield in data when applying a weight to each event according to its efficiency in fine bins with respect to the standard procedure. Signal yields have been extracted by sideband subtraction*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	-	-279	-257
200 – 400	-2373	109	-89
400 – 600	-410	45	35
600 – 800	-113	14	-13
800 – 1000	0	6	37
1000 – 1200	18	11	-9
1200 – 1400	-5	3	-15
1400 – 1600	-4	-5	-12

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	–	< 0.1	< 0.1
200 – 400	< 0.1	< 0.1	0.1
400 – 600	< 0.1	< 0.1	< 0.1
600 – 800	< 0.1	0.1	< 0.1
800 – 1000	< 0.1	0.1	0.3
1000 – 1200	< 0.1	0.1	0.1
1200 – 1400	0.1	< 0.1	0.1
1400 – 1600	0.1	0.1	0.1

(a) *Difference in reconstruction efficiency on Monte Carlo*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	–	6.3	1.0
200 – 400	1.4	1.9	10.0
400 – 600	1.2	6.8	11.4
600 – 800	2.2	6.1	2.4
800 – 1000	4.4	3.7	10.4
1000 – 1200	0.3	5.0	3.5
1200 – 1400	0.4	3.6	2.2
1400 – 1600	1.1	1.0	0.4

(b) *Difference in K^0_S yield in data*

Table 13.6.: *Difference in reconstruction efficiency (in percent) for K^0_S on Monte Carlo (a) and in (absolute) K^0_S yield (b) on data assuming an exponential background model with respect to the default fit.*

13.3. Stability of selection cuts

As explained in Section 11.7, the difference in reconstruction efficiency due to selection differences between data and Monte Carlo $\Delta\varepsilon = (\varepsilon_{data} - \varepsilon_{MC})/\varepsilon_{MC} = (7 \pm 4)\%$. The 7% difference will be corrected for, and 4% is assigned as systematic uncertainty of the correction.

13.4. Stability of fit to K^0_S invariant mass

13.4.1. Uncertainties related to default fit model

The uncertainties associated with the default fit model have been studied in Section 11.3 (cf. Table 11.3). A per-bin systematic uncertainty will be assigned accordingly.

13.4.2. Binning effects

Table 11.2 summarises the systematic uncertainties assigned due to binning effects. They are typically small but will of course enter the final calculation of systematic uncertainties as a bin-wise contribution.

13.4.3. Variation of the background model

To check the influence of the background model, an alternative fit is performed using an exponential background model. Changes with respect to the default fit enter in two places in the analysis: First, the signal yields on data might change, second, the yield changes on simulated events lead to changes in reconstruction efficiencies. Both of these effects must be assessed when estimating systematic uncertainties. The results can be found in Table 13.6. The effect is not typically small.

13.4.4. Fixing mass peak shape from Monte Carlo

To further check the fit stability, we take the shape of the mass peak from simulated events in each bin and fix widths, means and the fractions of the narrower Gaussians

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	-	-4.2	-4.9
200 – 400	-9.9	-7.7	-2.4
400 – 600	-2.5	8.2	7.8
600 – 800	-1.7	-0.5	-3.2
800 – 1000	-2.7	-4.0	5.8
1000 – 1200	-1.8	-7.6	-5.3
1200 – 1400	-1.5	0.8	0.3
1400 – 1600	-0.5	-3.6	-5.5

(a) *Difference ΔN in K^0_S yield*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	-	-0.4	-0.6
200 – 400	-1.0	0.4	-0.1
400 – 600	-0.2	<0.1	0.4
600 – 800	-0.1	<0.1	-0.2
800 – 1000	-0.2	-0.3	0.4
1000 – 1200	-0.1	-0.5	-0.5
1200 – 1400	-0.1	0.1	<0.1
1400 – 1600	-0.4	-0.4	-1.0

(b) *Relative difference $\Delta N/\sigma_N$ in K^0_S yield*

Table 13.7.: *Difference Δ_N and relative difference $\Delta N/\sigma_N$ in K^0_S signal yield in data fixing the fit shape from Monte Carlo with respect to the default fit.*

(in case of double Gaussian fit) of the mass peak in the corresponding bin in data to the values observed in simulation. To account for the different mass resolutions seen in data and simulation, the widths in simulation are scaled by a factor of 1.28 which corresponds to the ratio of average widths observed in data and simulation over the entire sample ($\sigma^{data}_{av}/\sigma^{MC}_{av}$, cf. Table 9.3).

Figures 13.2, 13.3 and 13.4 show the mass fits in bins of p_T and y to the 2009 pilot run data with the shapes fixed from Monte Carlo as explained above; the difference with respect to the standard fit is shown in Table 13.7. The observed difference is well within the statistical uncertainties of the yields, and there is no apparent pattern to the deviations. Therefore, not systematic uncertainty was assigned for the result of this cross-check.

13.5. Trigger-related sources of systematics

As shown in Chapter 12, the difference in trigger efficiency between the two definitions (one data-driven, to be used on data, the other based on Monte Carlo truth for use on simulated events) agree to better than ±1% when both definitions are applied to simulated events. The agreement in tracking efficiency as measured on data and simulated events (reweighted to reproduce the downstream track multiplicity seen on data) is also good to within ±1%. To account for the small inconsistency in how the trigger emulation is applied to data and simulated events (see 12.4), an overall systematic uncertainty of 2.5% is applied to trigger efficiencies.

In addition to this, a contribution coming from the uncertainty in the amount and modelling of diffraction in simulated events has to be taken into account. The fraction of diffractive events in the sample of simulated events is varied by ±50%. Two sets of trigger efficiencies are calculated for each fraction of diffractive events, one set for a sample combined from diffractive and non-diffractive events from PYTHIA 6.4, the other set for a sample with diffractive events from PYTHIA 8.1 and non-diffractive from PYTHIA 6.4. To obtain the trigger efficiencies on these samples, the information in Table 12.3 is used. The largest deviation of any of these samples to the standard sample is given in Table 13.8 and is taken as an estimate of the systematic uncertainty in trigger efficiency due to diffraction modelling. The deviations are small (below 1%) and will

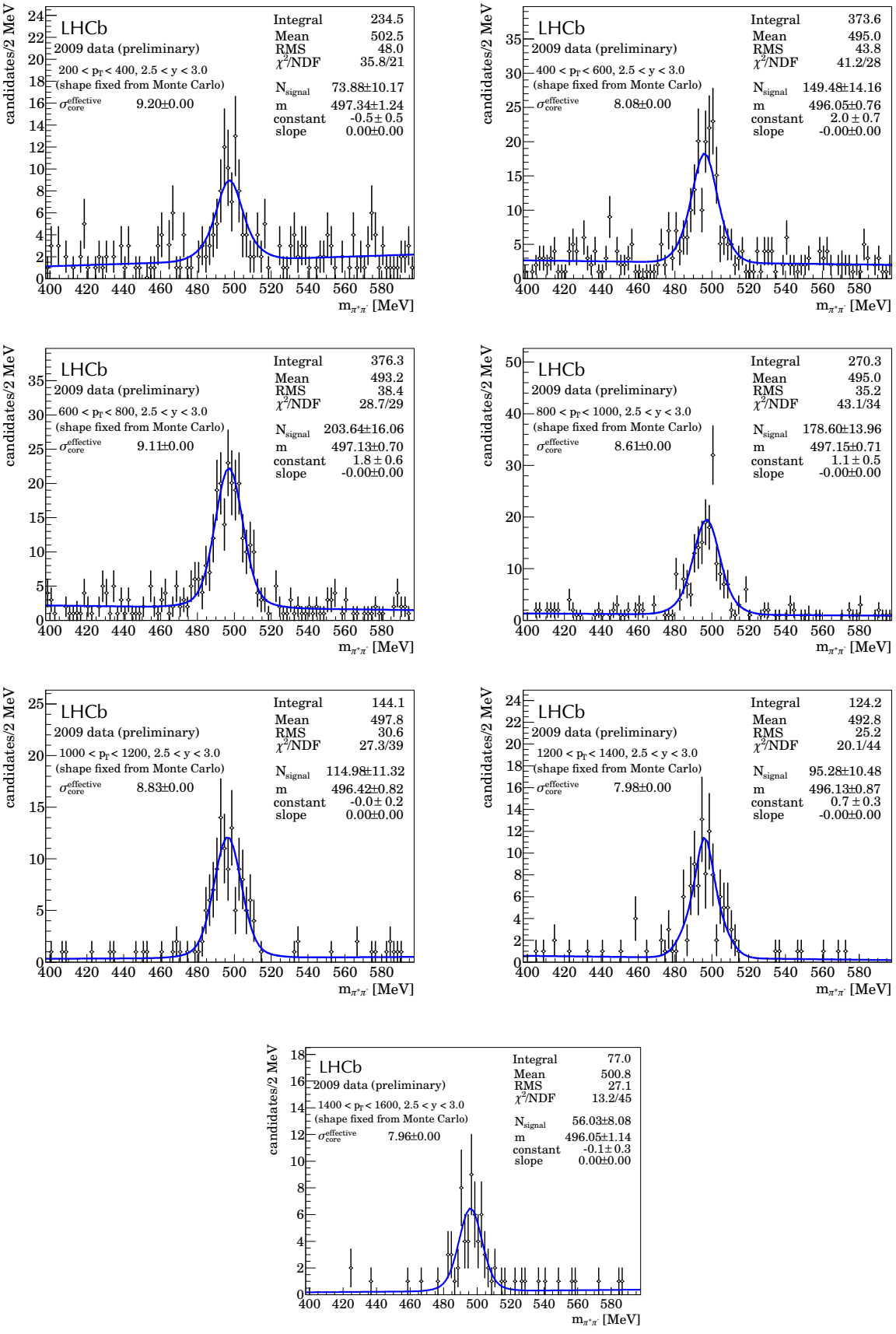


Figure 13.2.: *Fits to the K_s candidates mass distribution on data with fit shape fixed from Monte Carlo for different p_T bins with $2.5 < y < 3.0$.*

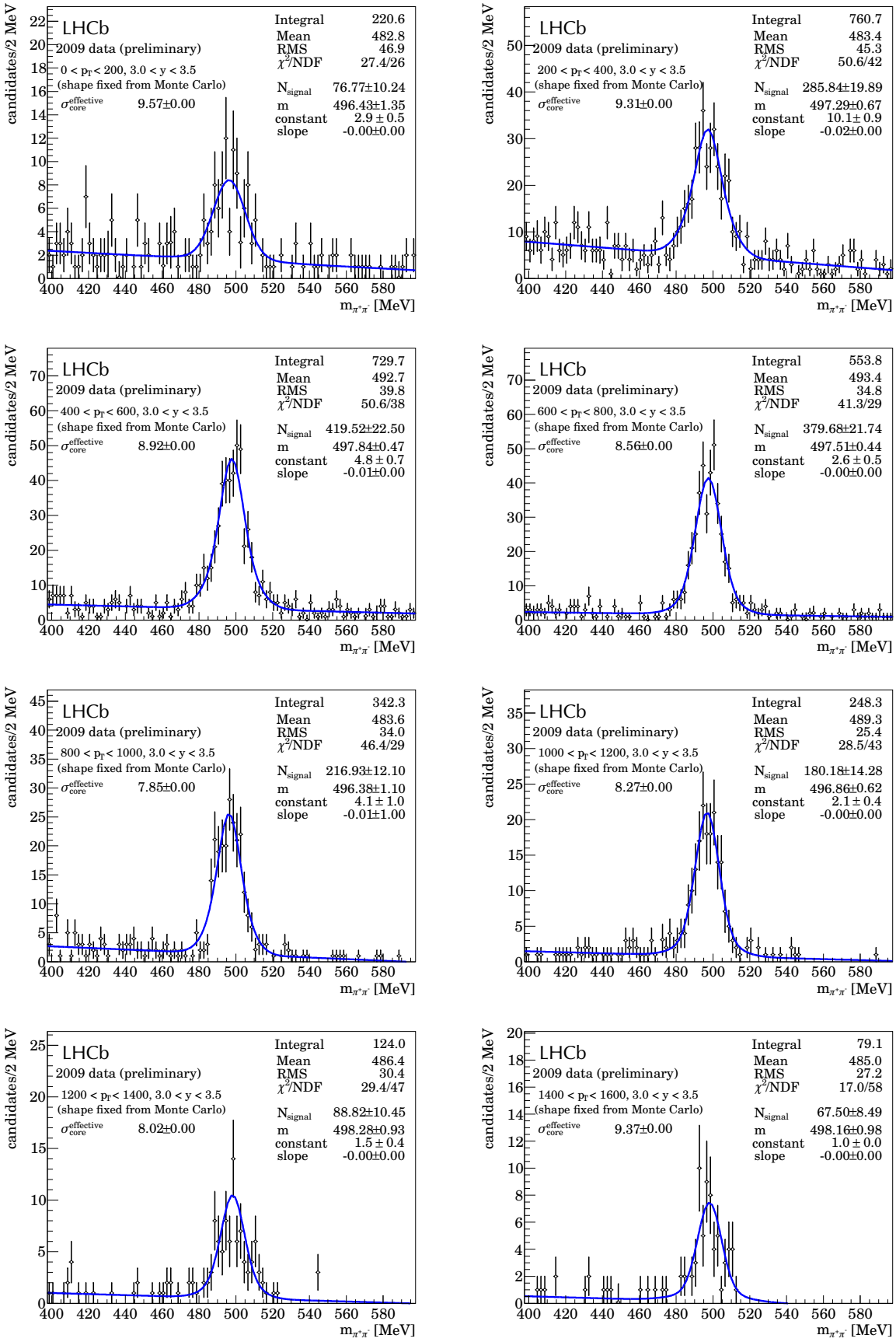


Figure 13.3.: *Fits to the K_s candidates mass distribution on data with fit shape fixed from Monte Carlo for different p_T bins with $3.0 < y < 3.5$.*

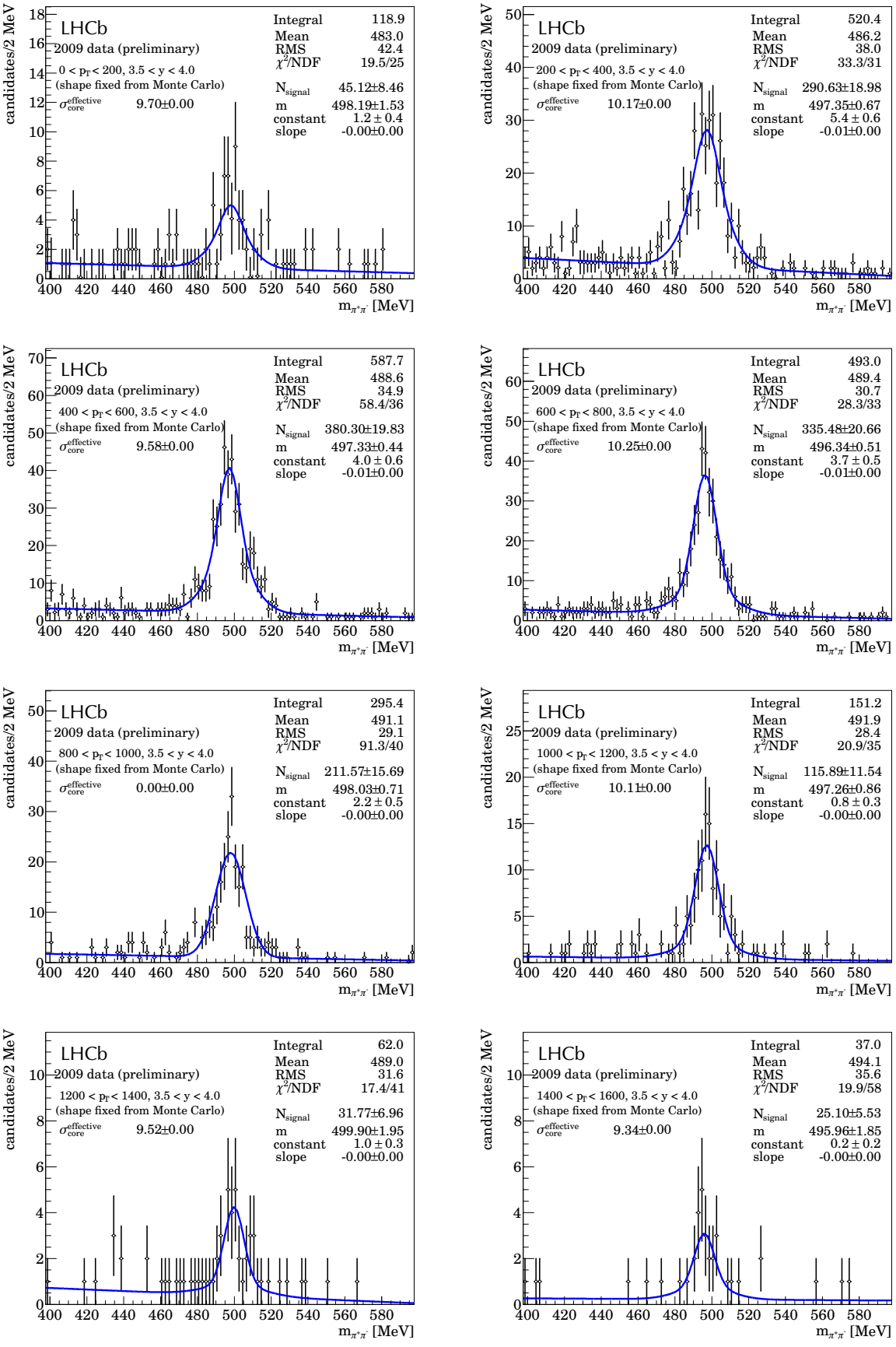


Figure 13.4.: *Fits to the K_s candidates mass distribution on data with fit shape fixed from Monte Carlo for different p_T y bins with* $3.5 < y < 4.0$.

Table 13.8: *Systematic uncertainties on trigger efficiencies (in percent) related to the unknown fraction and type of diffractive events in data.*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	–	0.5	0.2
200 – 400	0.6	0.5	0.6
400 – 600	0.6	0.4	0.3
600 – 800	0.5	0.3	0.3
800 – 1000	0.4	0.3	0.4
1000 – 1200	0.5	0.1	0.5
1200 – 1400	0.1	0.1	0.1
1400 – 1600	0.1	0.1	0.1

therefore not dominate the systematic uncertainties.

13.6. Small Effects

This section collects a few contributions to the total systematic uncertainty which turn out to be almost negligible and were therefore grouped together.

13.6.1. Correction for non-prompt K^0_S

For simulated events, the fraction of reconstructed non-prompt K^0_S is 0.6%. The default reconstruction efficiency definition implicitly corrects the K^0_S yields measured on data for the non-prompt contamination. Since the simulation might not properly reproduce the fraction of non-prompt K^0_S found in data, an additional 0.5% systematic uncertainty is assigned to the reconstruction efficiency; this corresponds to a (relative) uncertainty of 83% in the fraction of non-prompt K^0_S. Since the simulation is not expected to differ that much from data, the estimate seems to be conservative.

13.6.2. Material interaction and hard scattering

Significant material interaction affects about 10% of all reconstructible K^0_S daughters, or equivalently about 19% of K^0_S have at least one pion which is affected. Pions unaffected by significant material interactions are reconstructed with 82.5% efficiency; this drops to 79.1% efficiency for pions which interact significantly. On K^0_S level, the corresponding figures are a reconstruction efficiency of 79.1% if both daughter pions do not suffer from significant material interaction, which drops to 65.3% for K^0_S if at least one daughter has a significant interaction.

Since there are no samples of simulated events corresponding to the 2009 pilot run conditions which over- or under-estimate the amount of material in the detector, the numbers given above are used to estimate the effect of a ±10% variation in material budget. Such a variation would result in 10% fewer or more pions which undergo a significant material interaction. This would result in 20.8% (17.2%) of K^0_S with at least one interacting daughters (compared to 19.0% with the nominal amount of material). This would cause a fluctuation in combined K^0_S reconstruction efficiency (i.e. K^0_S reconstruction efficiency irrespective of whether the daughters undergo a significant material scattering) of less than ±0.5%. Since a ±10% variation in detector material is expected to be a

pessimistic estimate, the effect of material interaction and hard scattering is deemed negligible.

13.6.3. Beam gas subtraction

The weighting factor for beam gas subtraction was found to be $\beta = (0.9075 \pm 0.0148)$. It was varied upwards and downwards by its statistical uncertainty. The resulting differences in K^0_S yield with respect to the unvaried β were smaller than the significant digits quoted for the K^0_S yields. Therefore, this source of systematic uncertainties was neglected.

13.6.4. Branching ratio uncertainties

The branching ratio of $K^0_S \longrightarrow \pi^+\pi^-$ does not directly enter the analysis, however, it was used as input to the simulation which was used to derive the reconstruction efficiencies. This branching ratio was measured to be $(69.20 \pm 0.05)\%$ ([7]). In principle, its statistical uncertainty should enter the systematic uncertainties of the reconstruction efficiency. In practice, the contribution is tiny compared to the other sources of systematic uncertainties, so influence of the uncertainty in branching ratio may be neglected.

14. Determination of prompt K^0_S production cross section

In this chapter, the final results of the analysis are presented and summarised. Since the prompt K^0_S production cross section is connected to the efficiency corrected yields by multiplication with the measured luminosity and thus only presents an overall scaling, the first step is to present efficiency corrected K^0_S yields. In a second step, these yields are then converted to the cross section measurement.

14.1. Determination of efficiency corrected K^0_S yields

In this section, the determination of the efficiency corrected yields is presented. To this end, the results, statistical and systematic uncertainties are combined. Contributions to the systematic uncertainty of a quantity are added in quadrature. When doing so, a distinction is made between contributions which are uncorrelated between different bins in p_T and y and contributions which may show (some) correlation. In the latter category, a further distinction is made between correlated uncertainty contributions which are evaluated on a bin-by-bin basis and those which are a global correlated contribution which is applied to all bins.

14.1.1. Uncorrected K^0_S yield

The uncorrected K^0_S yield is taken from Table 9.4. The variation of the fit model (cf. Tables 11.3 and 13.6b) provides an uncorrelated contribution to the systematic uncertainty. A second contribution comes from the uncertainty of the beam gas correction factor β which causes a global contribution of 0.06% of the yield. The final result for the uncorrected K^0_S yield is given in Table 14.1.

14.1.2. Reconstruction efficiency

The K^0_S reconstruction efficiency and its statistical uncertainty is taken from Table 11.1, with an upwards correction of $\Delta\varepsilon = 7\%$ relative to the value given in Table 11.1 due to selection differences between data and simulated events, according to Section 11.7. This upwards correction gives rise to a 4% relative contribution to the systematic uncertainties which is correlated between all bins and applied globally.

Conceptually, the contribution from finer efficiency binning in simulated events in Table 13.5 is an uncorrelated bin-per-bin contribution to systematic uncertainties. Because the uncertainties given in Table 13.5 is given in terms of efficiency corrected yields and these uncertainties are large in some bins, this uncertainty contribution is not applied here. Applying it here would involve converting it to an efficiency uncertainty.

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	—	73.6±10.1± 6.7±–±0.0	40.2± 7.8± 1.6±–±0.0
200 – 400	64.0± 9.6±1.5±–±0.0	278.1±20.5± 6.4±–±0.2	288.2±20.5±13.8±–±0.2
400 – 600	147.0±15.2±4.2±–±0.1	427.7±24.1±10.1±–±0.3	388.1±21.3±15.2±–±0.2
600 – 800	202.0±16.2±2.4±–±0.1	379.2±22.3±10.0±–±0.2	332.3±20.0± 7.9±–±0.2
800 – 1000	175.9±14.7±4.5±–±0.1	213.0±16.1± 7.2±–±0.1	217.4±16.9±10.4±–±0.1
1000 – 1200	113.2±11.4±0.9±–±0.1	172.6±14.0± 5.1±–±0.1	110.6±11.7± 5.1±–±0.1
1200 – 1400	94.7±10.9±2.0±–±0.1	89.6±10.3± 3.6±–±0.1	32.0± 7.5± 2.3±–±0.0
1400 – 1600	55.6± 8.0±2.0±–±0.0	63.9± 8.4± 2.8±–±0.0	19.6± 4.9± 0.6±–±0.0

Table 14.1.: *Final result on the measured yields including statistical, uncorrelated, correlated bin-per-bin evaluated systematical uncertainties and correlated overall uncertainties. The "–" in the third component of each measurement indicates that there is no correlated bin-per-bin contribution to the systematic uncertainties. The "0" in the fourth component means that there is a contribution which was evaluated, and it came out as zero. The same notation will also be used in the following tables.*

The reconstruction efficiency enters in the denominator of the final result (the K^0_S cross section) and the "(systematic) fluctuations" around the mean efficiency are large for some bins. Therefore linear error propagation cannot be used for these bins. To avoid this problem, the contribution from the finer binning is postponed and later applied as a systematic to the efficiency corrected yields.

The contribution from different reconstruction efficiencies in data and simulation as evaluated with the data driven method (Table 13.3) are at least partially correlated among bins (the data-driven method works in bins of daughter track p_T, not K^0_S p_T). Contributions to the systematic uncertainty from fit model variation (cf. Table 13.6a) and binning effects (Table 11.2) are both correlated among the individual bins and are thus treated accordingly.

Contributions from non-prompt K^0_S (0.5% relative) and material interactions (0.5% absolute) contribute to the correlated global systematic uncertainties (see Sections 13.6.1 and 13.6.2).

The statistical uncertainties are almost 100% correlated to the statistical uncertainties of the trigger efficiency because the reconstruction efficiency numerator is exactly the denominator of the trigger efficiency. Therefore, the statistical uncertainties for the reconstruction efficiency will be listed here, but not used in the calculation of efficiency corrected yields to avoid including the contribution twice. Table 14.2 gives the final result on the K^0_S reconstruction efficiency.

14.1.3. Trigger efficiency

The efficiency to trigger an event containing a reconstructed K^0_S is taken from Table 12.2. For each bin, there is a (correlated) contribution to the systematic uncertainty from the unknown fraction of diffractive events which is taken from Table 13.8. Moreover, the remaining discrepancy between data and simulated events was estimated to be a 2.5% global (correlated) contribution to the systematic uncertainties (Section 13.5). The resulting figures for the trigger efficiency can be found in Table 14.3.

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	—	3.5±0.1±–±0.3±0.1	3.0±0.1±–±0.2±0.1
200 – 400	1.3±0.1±–±0.1±0.1	7.5±0.1±–±0.5±0.3	7.6±0.1±–±0.6±0.3
400 – 600	3.6±0.1±–±0.2±0.1	12.1±0.1±–±0.7±0.5	12.3±0.1±–±0.7±0.5
600 – 800	7.6±0.1±–±0.5±0.3	15.4±0.2±–±0.9±0.6	15.3±0.2±–±0.9±0.6
800 – 1000	11.4±0.2±–±0.7±0.5	17.5±0.2±–±1.1±0.7	16.1±0.3±–±1.0±0.6
1000 – 1200	14.6±0.3±–±0.9±0.6	18.9±0.4±–±1.1±0.8	15.4±0.4±–±0.9±0.6
1200 – 1400	16.4±0.4±–±1.0±0.7	19.4±0.5±–±1.2±0.8	13.8±0.6±–±0.9±0.6
1400 – 1600	18.0±0.6±–±1.1±0.7	19.4±0.7±–±1.2±0.8	12.7±0.7±–±0.8±0.5

Table 14.2.: *Final result on the reconstruction efficiency given in percent including statistical, uncorrelated, correlated bin-per-bin evaluated systematical uncertainties and correlated overall uncertainties.*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	—	96.9±0.3±–±0.5±2.5	100.0±0.0±–±0.2±2.5
200 – 400	96.5±0.1±–±0.6±2.5	97.0±0.2±–±0.5±2.5	97.6±0.5±–±0.6±2.5
400 – 600	97.2±0.1±–±0.6±2.5	97.5±0.1±–±0.4±2.5	97.2±0.4±–±0.3±2.5
600 – 800	97.5±0.1±–±0.5±2.5	97.6±0.2±–±0.3±2.5	98.0±0.1±–±0.3±2.5
800 – 1000	97.7±0.2±–±0.4±2.5	97.7±0.2±–±0.3±2.5	98.4±0.4±–±0.4±2.5
1000 – 1200	99.3±0.1±–±0.5±2.5	98.8±0.2±–±0.1±2.5	98.3±0.6±–±0.5±2.5
1200 – 1400	99.0±0.3±–±0.1±2.5	97.6±2.4±–±0.1±2.5	98.8±1.3±–±0.1±2.5
1400 – 1600	99.1±0.3±–±0.1±2.5	98.7±0.3±–±0.1±2.5	99.0±1.0±–±0.1±2.5

Table 14.3.: *Final results on the trigger efficiencies given in percent including statistical, uncorrelated, correlated bin-per-bin evaluated systematical uncertainties and correlated overall uncertainties.*

14.1.4. Resulting efficiency corrected yield

Tables 14.1, 14.2 and 14.3 are combined to form the efficiency corrected prompt K_S^0 yield for the 2009 pilot run. The contribution from in-bin reconstruction efficiency variations from Table 13.5 which was postponed above is applied as well. The result can be found in Table 14.4a.

14.2. Prompt K_S^0 production cross section

The efficiency corrected K_S^0 yields in Table 14.4a can be converted to the prompt K_S^0 production cross section by dividing by the integrated luminosity of $L_{int} = 6.8\,\mu\text{b}^{-1}$. The 14.7% relative uncertainty in the luminosity is treated as a correlated global contribution to the systematic uncertainty. Table 14.4b shows the results.

14.3. Final results and summary

The prompt K_S^0 production cross section measured with a selection based on downstream tracks was presented above. Summing up the K_S^0 production cross section measured

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	—	2151.1±295.3±340.1±182.9±104.2	1341.8±260.3±262.7± 80.6± 64.6
200 – 400	5165.2±774.8±2376.2±414.5±256.0	3827.8±282.3±139.8±268.7±184.2	3886.9±277.2±206.5±316.0±186.7
400 – 600	4157.1±429.9± 426.7±250.7±201.2	3628.0±204.5± 96.5±218.2±174.0	3244.9±178.6±131.5±196.7±155.7
600 – 800	2727.1±218.7± 117.7±164.2±131.1	2521.6±148.4± 68.1±152.4±120.8	2216.1±133.4± 54.3±133.1±106.0
800 – 1000	1572.5±131.5± 40.6± 94.6± 75.4	1242.4± 93.9± 42.5± 75.0± 59.5	1376.5±107.2± 75.7± 86.7± 65.8
1000 – 1200	783.4± 78.9± 19.1± 47.2± 37.3	922.4± 74.8± 29.6± 55.8± 44.0	730.2± 77.4± 34.6± 44.2± 34.9
1200 – 1400	584.3± 67.3± 13.1± 35.2± 27.9	474.0± 55.7± 19.4± 28.4± 22.7	234.6± 55.1± 22.3± 14.6± 11.2
1400 – 1600	312.1± 44.9± 11.9± 18.9± 14.9	334.3± 44.0± 15.6± 20.4± 16.0	155.5± 38.9± 13.0± 9.7± 7.4

(a) *Final efficiency corrected K^0_S yields, including statistical, uncorrelated and correlated systematical uncertainties.*

p_T [MeV] / y	2.5 - 3.0	3.0 - 3.5	3.5 - 4.0
0 – 200	—	316.3±43.4±50.0±26.9±49.0	197.3±38.3±38.6±11.8±30.5
200 – 400	759.6±113.9±349.4±61.0±117.9	562.9±41.5±20.6±39.5±87.1	571.6±40.8±30.4±46.5±88.4
400 – 600	611.3± 63.2± 62.8±36.9± 94.6	533.5±30.1±14.2±32.1±82.5	477.2±26.3±19.3±28.9±73.8
600 – 800	401.0± 32.2± 17.3±24.2± 62.0	370.8±21.8±10.0±22.4±57.4	325.9±19.6± 8.0±19.6±50.4
800 – 1000	231.3± 19.3± 6.0±13.9± 35.8	182.7±13.8± 6.2±11.0±28.3	202.4±15.8±11.1±12.7±31.3
1000 – 1200	115.2± 11.6± 2.8± 6.9± 17.8	135.6±11.0± 4.4± 8.2±21.0	107.4±11.4± 5.1± 6.5±16.6
1200 – 1400	85.9± 9.9± 1.9± 5.2± 13.3	69.7± 8.2± 2.9± 4.2±10.8	34.5± 8.1± 3.3± 2.1± 5.3
1400 – 1600	45.9± 6.6± 1.7± 2.8± 7.1	49.2± 6.5± 2.3± 3.0± 7.6	22.9± 5.7± 1.9± 1.4± 3.5

(b) *Final prompt K^0_S production cross section from the analysis using downstream tracks in μb, including statistical, uncorrelated, per-bin correlated and overall correlated systematical uncertainties.*

Table 14.4.: *Final efficiency corrected yields (a) and K^0_S production cross section measurements (b) in bins of p_T and y.*

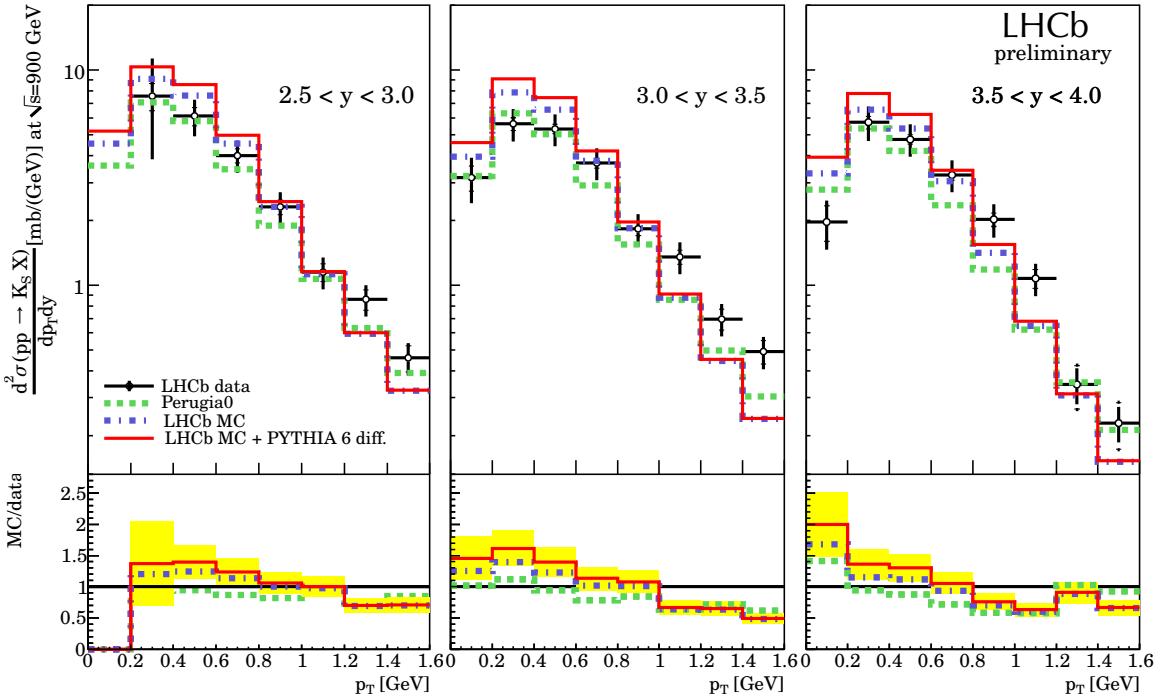


Figure 14.1.: *Double differential prompt K^0_S production cross section in pp collisions at 900 GeV centre-of-mass energy as function of rapidity and transverse momentum. The points represent LHCb data; total uncertainties are represented with vertical error bars while statistical uncertainties are indicated by tick marks on these error bars. Monte Carlo predictions for several settings of the* PYTHIA *generator (see text for details) are superimposed as histograms. The lower plot forms the ratio of Monte Carlo/data with the shaded band representing the total uncertainty of one of these ratios. Figure from [3].*

in all bins in the range of $0\,\text{MeV} \leq p_T \leq 1600\,\text{MeV}$ and $2.5 \leq y \leq 4$ (excluding the bin $(0\,\text{MeV} \leq p_T \leq 200\,\text{MeV}) \times (2.5 \leq y \leq 3.0)$ where no measurement was performed) amounts to $\sigma(pp \longrightarrow K^0_S X) = (6410.3 \pm 169.6 \pm 364.3 \pm 285.9)\,\mu\text{b}$.

These results also are shown in Figure 14.1, together with predictions for the prompt K^0_S production cross section from three settings of the PYTHIA generator:

- "LHCb MC": This is the standard LHCb Monte Carlo generated using PYTHIA 6.4 (see Chapter 11 for details); diffractive events are excluded.
- "LHCb MC + Pythia 6 diff.": Same as "LHCb MC", but with (soft) diffractive event generated by PYTHIA 6.4.
- "Perugia 0": This is the "Perugia 0" tune for PYTHIA from [74] excluding diffractive events.

The results agree reasonably well with Monte Carlo predictions, although the predictions tend to have a higher (lower) cross section near the low (high) end of the p_T range

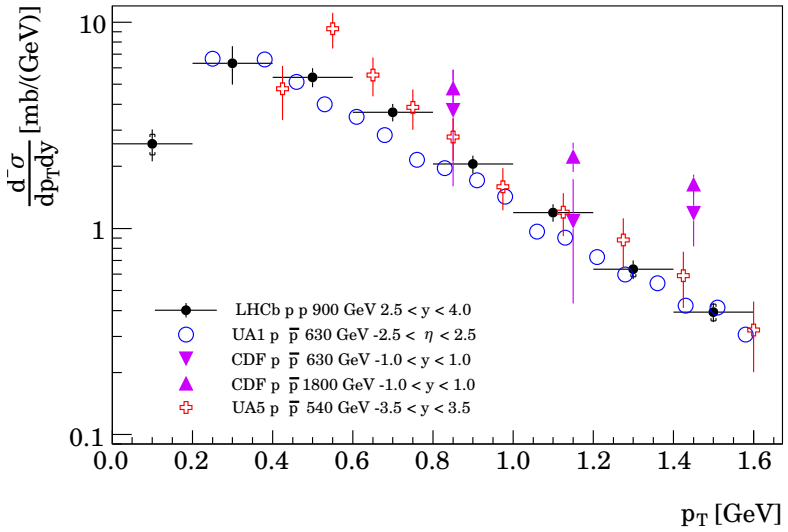


Figure 14.2.: *Absolute measurement of the prompt K^0_S production cross section as function of p_T, performed by UA5 ([63]), UA1 ([66]) and CDF ([65]) and LHCb (this thesis). The measurements were performed at different centre-of-mass energies in pp or $p\bar{p}$ collisions and in different rapidity (y) or pseudorapidity (η) ranges.*

studied. While the Monte Carlo generators used for the predictions have (obviously) been tuned to reproduce selected physics distributions available at the time this measurement was made, previous measurements of prompt K^0_S production in pp or $p\bar{p}$ collisions at similar centre-of-mass energies have had their focus on measurements in the central region, so the relatively good agreement between the measurement and Monte Carlo predictions is not necessarily to be expected. It is therefore encouraging to see this relatively good agreement, and the results presented here can be used as input in future attempts to improve Monte Carlo generator tunes.

Previous measurements of the prompt K^0_S production cross section have been performed by UA5 ([63]), UA1 ([66]) and CDF ([65]) in pp and $p\bar{p}$ collisions at different centre-of-mass energies and in different kinematic regimes. Results there have been quoted in the form of invariant differential cross sections $E\,d^3\sigma/d^3p$ as function of p_T. To be able to compare this to this result given as $d^2\sigma/(dp_T dy)$, the other results are multiplied by $2\pi p_T$ while limiting the p_T range of previous measurements to the range measured here.

The result can be seen in Figure 14.2. The general trend is reproduced well by the measurement described in this thesis; however, there is some spread between the different experiments because of the different centre-of-mass energies and rapidity/pseudorapidity ranges for which the measurements were performed. This plot also shows that LHCb is able to contribute measurements at high rapidities and down to low p_T, thus complementing existing measurements.

The fact that this measurement (which is almost what was published as the first

measurement by LHCb in [3]) was performed demonstrates that the LHCb detector performs well and that it is well understood — a prerequisite for the many future physics analyses to come.

15. Summary and conclusion

In the first part of this thesis, an algorithm to find and reconstruct charged particles in the main tracking system of LHCb was implemented. The efficiency to find and reconstruct a track is over 95%, a value which is retained even when operating LHCb under high pile-up conditions of up to 3 simultaneous pp interactions. The fraction of wrongly reconstructed tracks in such a sample is below 4%. With respect to an earlier algorithm, the developed tracking algorithm takes about half the time to execute and is more efficient, especially at high pseudo-rapidities. Due to its good properties, this algorithm has become the default algorithm for the standalone main tracker reconstruction in LHCb.

Its small execution time and flexibility also make it suitable for use at the input of the first stage of the LHCb software trigger: The decision of the hardware trigger based on information from the muon or calorimeter system can be accepted or rejected by attempting to reconstruct a confirming track in the main tracking system. With the algorithm presented in this thesis, this type of track reconstruction takes less than 1 ms per hardware trigger candidate.

Moreover, the algorithm has played a crucial role during the commissioning phase of the experiment because it is the only algorithm in LHCb which can reliably reconstruct cosmic muons in the main tracking system which are needed for alignment and detector calibration purposes.

The second part of this thesis presents a measurement of the production cross section of prompt K_S^0 in pp collisions at a centre-of-mass energy of 900 GeV in the data from the 2009 pilot run on a data sample of $\mathscr{L} = (6.8 \pm 1.0)\,\mu\text{b}^{-1}$. The K_S^0 are reconstructed via their decay to two charged pions, so an excellent understanding of the track reconstruction is required to perform the analysis. The total cross section in the range $0\,\text{MeV} \leq p_T \leq 1600\,\text{MeV}$ and $2.5 \leq y \leq 4$ (excluding the bin $(0\,\text{MeV} \leq p_T \leq 200\,\text{MeV}) \times (2.5 \leq y \leq 3.0)$ where no measurement was performed) amounts to $(6410.3 \pm 169.6 \pm 463.1)\,\mu\text{b}$; Figure 15.1 shows the result as function of K_S^0 transverse momentum p_T and rapidity y.

The results are in line with previous measurements (cf. Figure 14.2) and in rough agreement with predictions from Monte Carlo event generators. However, the agreement with Monte Carlo event generators is far from perfect. As these generators have not been tuned in the forward region covered by the LHCb detector, discrepancies between Monte Carlo predictions and measurements are to be expected. The results presented here can be used to improve the simulation. The results of this analysis contributed significantly to the first LHCb publication ([3]).

The main challenge in the presented analysis was the relatively poor agreement of data and simulation for this very early data sample. To compensate, techniques were developed which improve the data/Monte Carlo agreement sufficiently to permit the use of simulated events to measure selection, reconstruction and trigger efficiencies. Special care was taken to cross-check the remaining data/Monte Carlo discrepancy in several

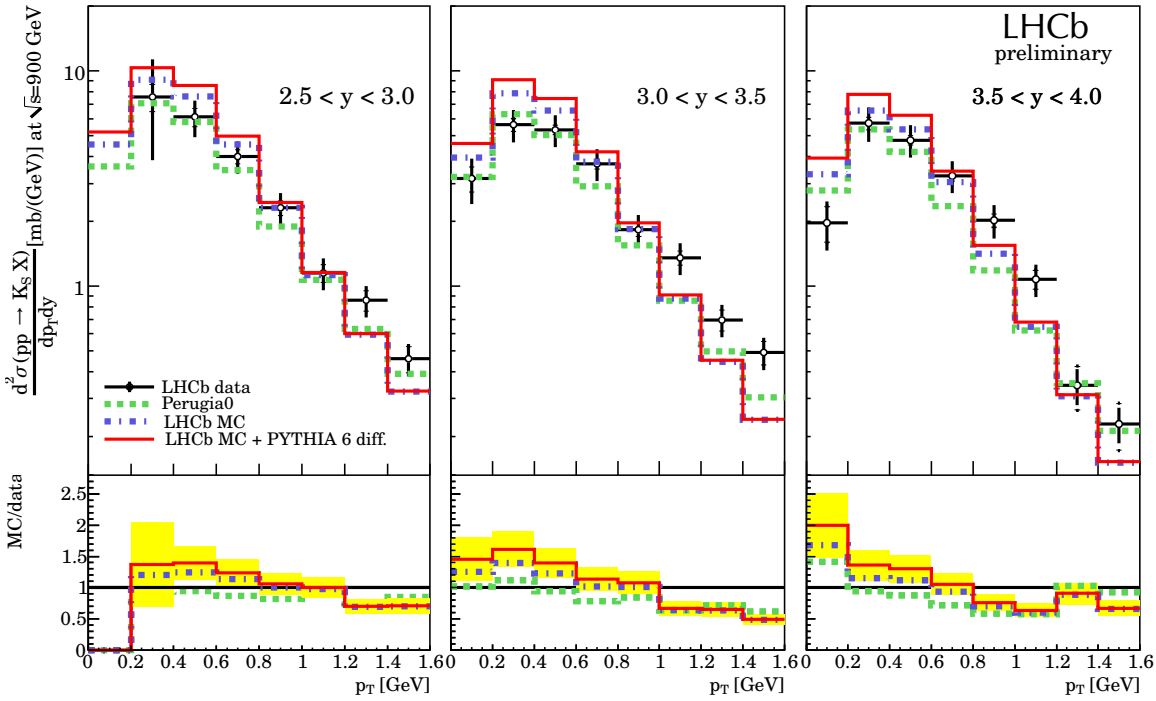


Figure 15.1.: *Double differential prompt K^0_S production cross section in pp collisions at 900 GeV centre-of-mass energy as function of rapidity and transverse momentum. The points represent LHCb data; total uncertainties are represented with vertical error bars while statistical uncertainties are indicated by tick marks on these error bars. Monte Carlo predictions for several settings of the* PYTHIA *generator (see text for details) are superimposed as histograms. The lower plot forms the ratio of Monte Carlo/data with the shaded band representing the total uncertainty of one of these ratios. Figure from [3].*

ways and to assign appropriate systematic uncertainties.

The understanding of the LHCb detector, the simulation and the analysis procedures employed is at a level which is advanced enough for more complicated analyses of the LHCb physics programme to proceed.

A. Triple Gaussian fits

In this thesis, triple Gaussian fits are used in several places to extract the widths and fractions of the *core* of a peaking distribution. Here, the *core* refers to the peaking part of the distribution; there is usually a much wider distribution under the core which will be called background in this discussion. The aim of the fit is the extraction of some reproducible measure of the width of the core and also the fraction of all histogram entries which are contained in the core part of the distribution. These parameters are used as a means to compare in a consistent way distributions of a quantity obtained using different methods. Note that there is no need for the fits to have a very good χ^2/NDF to enable comparisons as long as the fitted function reproduces the general shape of the distribution.

As an example of the idea, see Figure 6.6 where the distribution of Outer Tracker residuals is compared for different methods to resolve the ambiguities in the Outer Tracker (see section 6.3.3 for details).

The following model is used for the fit:

$$\begin{aligned} N(x) &= N \cdot \Big(f_{core} \big(f_{narrow} G(x; \mu_{core}, \sigma_{core}^{narrow}) + \\ &\qquad (1 - f_{narrow}) G(x; \mu_{core}, \sigma_{core}^{wide}) \big) + \\ &\qquad (1 - f_{core}) G(x; \mu_{bg}, \sigma_{bg}) \Big) \end{aligned}$$

In the formula above, $G(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}$ is a normal distribution with mean μ and width σ and N is the total number of entries. The core of the distribution is characterised by the fraction of the total number of entries f_{core}, its mean μ_{core} and relative fraction of the narrower contribution to the core and the widths of the narrower and the wider Gaussian contributions, f_{narrow}, σ_{core}^{narrow} and σ_{core}^{wide}. The background and/or the tails of the distribution accounts for the remaining fraction of histogram entries, and it is further characterised by its mean σ_{bg} and its width σ_{bg}.

The following steps are used to obtain a fit:

- Fit a single Gaussian to the distribution in a $\pm 4.5\sigma$ range around the mean of the previous single Gaussian fit, using mean and RMS of the distribution as staring point. This fit is repeated ten times, updating the fit range with the values of the previous iteration. From the last iteration, an estimate of the core width σ_{core}^{est} is obtained.

- Initialise the triple Gaussian fit model by fixing N to the total number of entries in the histogram, the fractions f_{core} and f_{core}^{narrow} to 0.75, μ_{core} and μ_{bg} to zero, and the widths σ_{core}^{narrow}, σ_{core}^{wide} and σ_{bg} to $0.75\sigma_{core}^{est}$, $1.5\sigma_{core}^{est}$ and 2.5 times the RMS of the distribution.

- Fit the distribution by keeping all parameters but one fixed, checking that fractions, means and widths stay in the allowed range and resetting them to sane values if they leave their ranges (fractions smaller than 0 or larger than 1 are set to 0.01 and 0.99, respectively), and the means and widths are set to their initial values defined above if their absolute value is larger than five times the RMS of the distribution for the means or ten times the RMS of the distribution for the widths. This is repeated for each parameter except N until none of the single-parameter fits moves any parameter outside of its allowed range.

- Fit the distribution by keeping all but two parameters fixed. The parameters are kept in the allowed range by the same method as above, and the method is repeated until all possible two parameter combinations not involving N have been exhausted and not caused any of the fit parameters to move outside their allowed ranges.

- Fit with all parameters but N floating, and repeat the fit until none of the parameters move outside their allowed ranges.

This method is useful because it quickly obtains reasonable starting values for the fit with all parameters floating. A simple χ^2 fit is used, since the number of entries in the histograms where this method is used is large enough.

To obtain an effective width of the core, a weighted mean of the core widths is formed:

$$\sigma_{core}^{effective} = \sqrt{f_{core}^{narrow}(\sigma_{core}^{narrow})^2 + (1 - f_{core}^{narrow})(\sigma_{core}^{wide})^2}$$

This definition has the nice property that the result $\sigma_{core}^{effective}$ is equal to the RMS of the core part of the fit model.

B. PatSeedingTool options and default values

This appendix lists the default job option values of PatSeeding, grouped roughly according to their use which is explained in detail in Chapters 6 and 7.

B.1. Hit selection

option name	default value
reusePatSeeding	true
UseForward	false
InputTracksName	LHCb::TrackLocation::Forward
DriftRadiusRange	-0.6 mm to 2.8 mm

B.2. Track model

option name	default value
dRatio	-3.2265e-3
InitialArrow	4.25307e-9
MomentumScale	35.31328
zReference	StateParameters::ZMidT

B.3. Track search in $x-z$ projection

option name	default value
MinMomentum	500 MeV
CurveTol	5 mm
zMagnet	5383.17 mm
xMagTol	2000 mm
TolCollectOT	3 mm
TolCollectIT	0.3 mm
MinXPlanes	5
CloneMaxXDistIT	3 mm
CloneMaxXDistOT	7 mm
CommonXFraction	0.7
QualityWeights	1.0, -0.2

B.4. Track search in $y-z$ projection

option name	default value
zForYMatch	9000 mm
MaxRangeOT	150 mm
MaxRangeIT	15 mm
yCorrection	4.73385e15
MaxYAtOrigin	400 mm
MaxYAtOriginLowQual	1500 mm
YTolSensArea	40 mm

B.5. Track search for tracks migrating between regions

option name	default value
TolExtrapolate	4 mm
TolCollectITOT	0.6 mm

B.6. Track fit and final track selection

option name	default value
MaxChi2HitOT	30
MaxChi2HitIT	10
MaxTrackChi2	15
MaxFinalChi2	12.25
MaxFinalTrackChi2	9
MaxTrackChi2LowMult	6
MinTotalPlanes	9
MaxMisses	1
OTNHitsLowThresh	17
MinPlanesPerStation	1
MaxHoles	2
ITStubLooseChi2	1500
ITStubTightChi2	80

B.7. Maximum fraction of used hits

option name	default value
MaxUsedFractPerRegion	0.30
MaxUsedFractITOT	0.15
MaxUsedFractLowQual	0.05

B.8. State estimates

option name	default value
StateErrorX2	4
StateErrorY2	400
StateErrorTX2	6e-5
StateErrorTY2	1e-4
FastMomentumToolName	"FastMomentumEstimate"
ZOutput	StateParameters::ZBegT, StateParameters::ZMidT, StateParameters::ZEndT

B.9. Running in tandem with PatForward

option name	default value
UseForwardTracks	false
ForwardCloneMergeSeg	true
ForwardCloneMaxShared	0.3
ForwardCloneMaxXDist	10 mm
ForwardCloneMaxYDist	50 mm
ForwardCloneMaxTXDist	0.005

B.10. Special applications

option name	default value
MeasureTime	false
FieldOff	false
EnforceIsolation	false
ITIsolation	15 mm
OTIsolation	20 mm
Cosmics	false

C. Further tracking studies

This appendix presents further tracking studies which are less important for the current running conditions of LHCb. They are nevertheless interesting because they document to some extent the evolution of the tracking algorithms for the main tracking system in the last four years.

Section C.1 discusses the benefits from using hit cleaning and/or Global Event Cuts in the pattern recognition. Hit cleaning was introduced to help cope with high occupancy events in the simulation a few years ago.

Global Event Cuts discard very busy events based on hit multiplicity (in this case, T station multiplicity). This is especially helpful in the software trigger, where very busy events can be discarded very quickly without introducing dead time.

Section C.2 studies the performance of PatSeeding under the influence of very large detector misalignments. The studies dates back to the time before data was available and shows how an ultra-clean track sample for alignment can be obtained for an initial alignment.

This chapter also is a convenient place to keep the plots that were omitted earlier for brevity; they can be found near the end of the chapter.

C.1. Influence of hit cleaning and Global Event Cuts

It is interesting to study the performance of both PatSeeding and TsaSeeding with consistent settings for hit cleaning, and it is also interesting to see how PatSeeding behaves if no Global Event Cuts (GEC) are applied. Since both hit cleaning and Global Event Cuts show their biggest effect in high multiplicity events, only the $\nu = 3$ scenario will be investigated. The overall changes in terms of reconstruction efficiency and average time needed to reconstruct an event are rather small, therefore, I will only give the summarising tables and a few representative plots.

C.1.1. Influence of hit cleaning

Tables C.1, C.2, C.3 and C.4 show reconstruction efficiencies and ghost rates for both PatSeeding and TsaSeeding, with and without hit cleaning at, $\nu = 3$. It can be seen that the loss in efficiency caused by hit cleaning is rather small (in the 0.1% to 0.2% range), and PatSeeding continues to be more efficient than TsaSeeding if hit cleaning is disabled. In fact, the use of hit cleaning for PatSeeding has been activated fairly recently to cope better with the high ν running conditions which LHCb experienced in 2010.

Figures C.1 and C.2 show reconstruction efficiencies and ghost rates as functions of p, p_T, η and as function of the number of visible interactions for both algorithms, with and without hit cleaning. The plots which show efficiency and ghost rates as function of the number of visible interactions demonstrate that the effect of hit cleaning is a small one

	long reconstructible		T reconstructible	
	eff. (clone) [%]	hit pur./eff. [%]	eff. (clone) [%]	hit pur./eff. [%]
all tracks	94.7 (0.2)	97.1 / 93.5	80.5 (0.2)	97.0 / 92.6
— $p > 5$GeV	95.4 (0.3)	97.2 / 94.4	95.0 (0.4)	97.4 / 94.0
B daughters	95.4 (0.1)	97.4 / 94.3	85.9 (0.1)	97.3 / 93.6
— $p > 5$ GeV	96.2 (0.2)	97.4 / 95.3	95.9 (0.2)	97.4 / 95.0
good B daughters	96.1 (0.1)	97.2 / 94.8	94.7 (0.2)	97.3 / 94.6
— $p > 5$ GeV	96.2 (0.1)	97.2 / 95.4	96.2 (0.1)	97.3 / 95.2
K^0_S/Λ daughters	94.5 (0.1)	97.0 / 92.9	78.6 (0.2)	97.0 / 92.1
— $p > 5$GeV	95.0 (0.3)	96.9 / 93.9	94.8 (0.4)	97.4 / 93.6
	tr. avg.	ev. avg.	tr. avg.	ev. avg.
ghost fraction	6.3	4.2	6.3	4.2

Table C.1.: *Reconstruction efficiencies and ghost rate for PatSeeding without hit cleaning (at $\nu = 3$).*

as long as the number of visible pp interaction does not rise too much. Only plots for all long reconstructible tracks have been included for brevity, since the corresponding plots for b or K^0_S/Λ daughters do not show anything new.

Table C.5 gives details of the timing behaviour of the the four alternatives at $\nu = 3$. PatSeeding is still faster than TsaSeeding if neither algorithm uses hit cleaning. However, TsaSeeding appears to be more well-behaved in the worst case. The timing behaviour of both algorithms without hit cleaning is shown in more detail in Figure C.3, the following parametrisations were fitted:

- PatSeeding without hit cleaning

$$\begin{aligned} t(n_{\text{hits}}) &= (0.6605 \pm 0.0003) \cdot 10^{-9} \cdot n^3_{\text{hits}} + (-4.8004 \pm 0.0048) \cdot 10^{-6} \cdot n^2_{\text{hits}} + \\ &\quad (15.8601 \pm 0.0213) \cdot 10^{-3} \cdot n_{\text{hits}} + (-11.3903 \pm 0.0274) \end{aligned}$$

$$\begin{aligned} t(n_{\text{vis.}}) &= (0.1030 \pm 0.0008) \cdot n^3_{\text{vis.}} + (2.1000 \pm 0.0089) \cdot n^2_{\text{vis.}} + \\ &\quad (-1.4530 \pm 0.0304) \cdot n_{\text{vis.}} + (5.5784 \pm 0.0307) \end{aligned}$$

- TsaSeeding without hit cleaning.

$$\begin{aligned} t(n_{\text{hits}}) &= (0.0435 \pm 0.0003) \cdot 10^{-9} \cdot n^3_{\text{hits}} + (3.0514 \pm 0.0042) \cdot 10^{-6} \cdot n^2_{\text{hits}} + \\ &\quad (-6.5389 \pm 0.0193) \cdot 10^{-3} \cdot n_{\text{hits}} + (10.6224 \pm 0.0260) \end{aligned}$$

$$\begin{aligned} t(n_{\text{vis.}}) &= (-0.0191 \pm 0.0007) \cdot n^3_{\text{vis.}} + (3.1768 \pm 0.0087) \cdot n^2_{\text{vis.}} + \\ &\quad (1.1621 \pm 0.0302) \cdot n_{\text{vis.}} + (8.1843 \pm 0.0307) \end{aligned}$$

	long reconstructible		T reconstructible	
	eff. (clone) [%]	hit pur./eff. [%]	eff. (clone) [%]	hit pur./eff. [%]
all tracks	89.7 (0.2)	97.4 / 87.3	74.4 (0.1)	97.3 / 87.1
— $p > 5$GeV	94.3 (0.3)	97.5 / 89.0	91.8 (0.2)	97.6 / 89.6
B daughters	90.2 (0.1)	97.7 / 87.9	79.7 (0.1)	97.6 / 87.6
— $p > 5$ GeV	94.7 (0.2)	97.7 / 89.7	93.5 (0.2)	97.8 / 89.8
good B daughters	93.1 (0.1)	97.6 / 88.9	91.7 (0.1)	97.7 / 88.9
— $p > 5$ GeV	94.8 (0.1)	97.6 / 90.0	94.8 (0.1)	97.7 / 90.1
K^0_S/Λ daughters	87.7 (0.1)	97.4 / 85.9	71.8 (0.1)	97.3 / 86.4
— $p > 5$GeV	93.7 (0.3)	97.3 / 88.1	90.9 (0.3)	97.6 / 89.1
	tr. avg.	ev. avg.	tr. avg.	ev. avg.
ghost fraction	7.1	4.8	7.1	4.8

Table C.2.: *Reconstruction efficiencies and ghost rate for TsaSeeding without hit cleaning (at $\nu = 3$).*

	long reconstructible		T reconstructible	
	eff. (clone) [%]	hit pur./eff. [%]	eff. (clone) [%]	hit pur./eff. [%]
all tracks	94.3 (0.2)	97.2 / 93.1	80.2 (0.2)	97.1 / 92.3
— $p > 5$GeV	95.1 (0.3)	97.3 / 94.0	94.7 (0.4)	97.5 / 93.7
B daughters	95.1 (0.1)	97.5 / 94.0	85.6 (0.1)	97.4 / 93.3
— $p > 5$ GeV	95.9 (0.2)	97.4 / 95.0	95.5 (0.2)	97.5 / 94.7
good B daughters	95.7 (0.1)	97.3 / 94.5	94.4 (0.2)	97.4 / 94.3
— $p > 5$ GeV	95.8 (0.1)	97.2 / 95.0	95.8 (0.1)	97.4 / 94.9
K^0_S/Λ daughters	94.1 (0.1)	97.1 / 92.4	78.3 (0.2)	97.1 / 91.8
— $p > 5$GeV	94.5 (0.2)	97.1 / 93.5	94.5 (0.4)	97.5 / 93.3
	tr. avg.	ev. avg.	tr. avg.	ev. avg.
ghost fraction	5.3	3.7	5.3	3.7

Table C.3.: *Reconstruction efficiencies and ghost rate for PatSeeding with hit cleaning (at $\nu = 3$).*

	long reconstructible		T reconstructible	
	eff. (clone) [%]	hit pur./eff. [%]	eff. (clone) [%]	hit pur./eff. [%]
all tracks	89.6 (0.2)	97.4 / 87.3	74.3 (0.1)	97.3 / 87.1
— $p > 5$GeV	94.2 (0.3)	97.5 / 89.1	91.7 (0.2)	97.6 / 89.6
B daughters	90.1 (0.1)	97.7 / 87.9	79.7 (0.1)	97.6 / 87.6
— $p > 5$ GeV	94.7 (0.2)	97.7 / 89.7	93.4 (0.2)	97.8 / 89.8
good B daughters	93.1 (0.1)	97.6 / 88.9	91.6 (0.1)	97.7 / 88.9
— $p > 5$ GeV	94.7 (0.1)	97.6 / 90.0	94.7 (0.1)	97.7 / 90.1
K^0_S/Λ daughters	87.5 (0.1)	97.4 / 86.0	71.7 (0.1)	97.3 / 86.4
— $p > 5$GeV	93.5 (0.3)	97.3 / 88.1	90.8 (0.3)	97.6 / 89.1
	tr. avg.	ev. avg.	tr. avg.	ev. avg.
ghost fraction	7.0	4.8	7.0	4.8

Table C.4.: *Reconstruction efficiencies and ghost rate for TsaSeeding (with hit cleaning enabled by default, at $\nu = 3$).*

		no hit cleaning		hit cleaning used	
ν		PatSeeding	TsaSeeding	PatSeeding	TsaSeeding
3	average time [ms]	26.08	42.43	22.50	42.07
	maximum time [ms]	981.6	638.2	473.9	513.8

Table C.5.: *Average and maximum time needed to reconstruct an event for both PatSeeding and TsaSeeding, with and with out hit cleaning in inclusive $b\bar{b}$ events.*

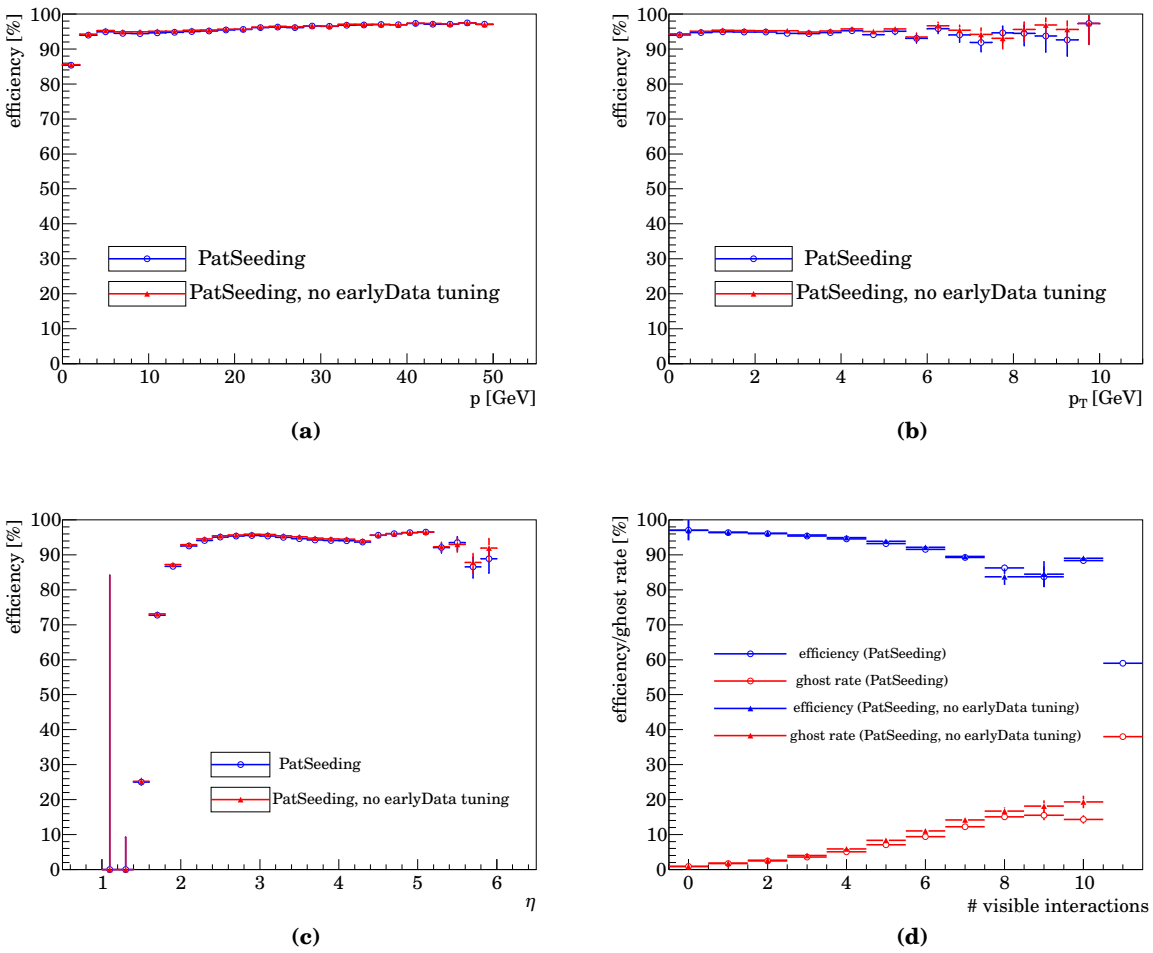


Figure C.1.: *Efficiency versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for tracks found by PatSeeding with (circles) and without (triangles) hit cleaning.*

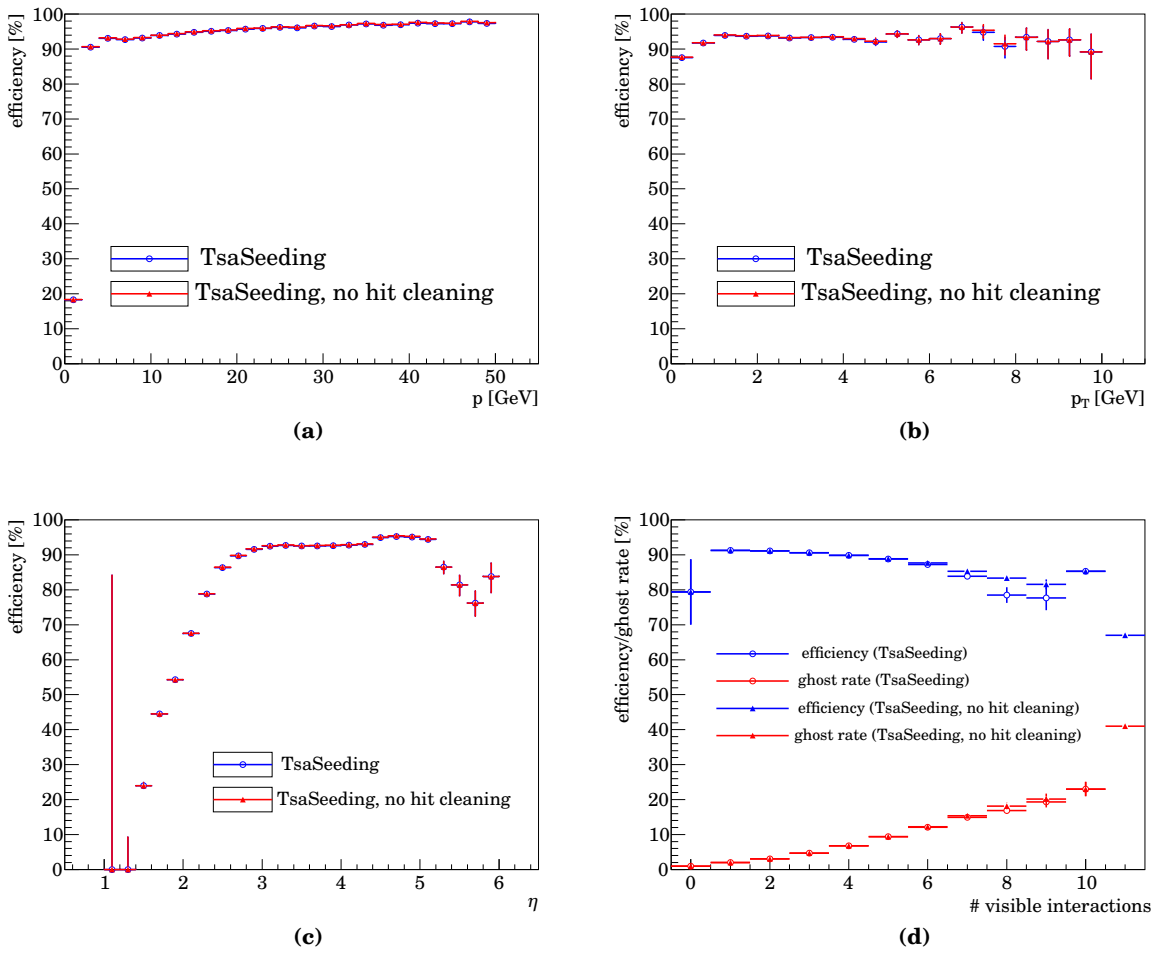
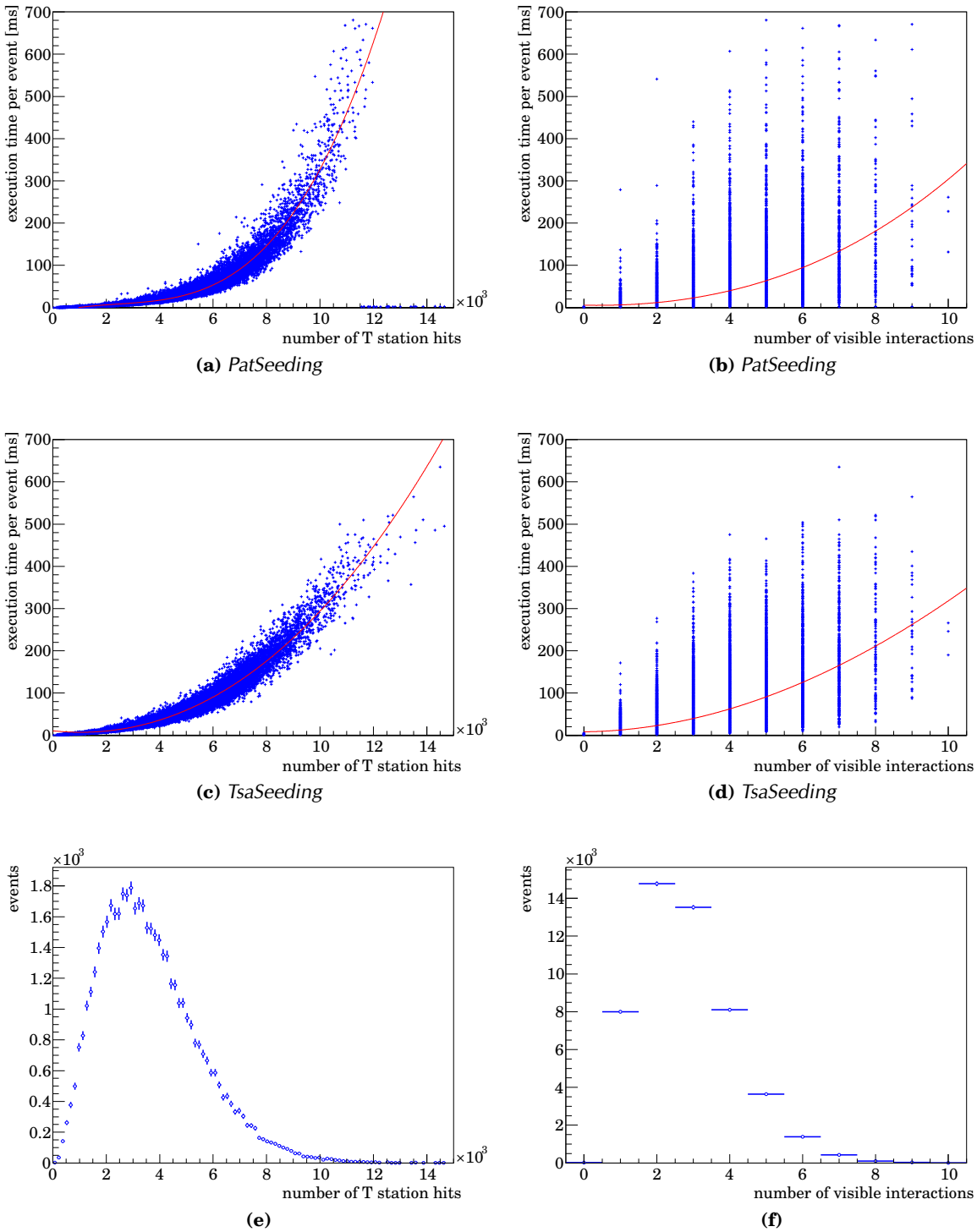


Figure C.2.: *Efficiency versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for tracks found by TsaSeeding with hit cleaning disabled (triangles) and in its standard configuration (circles).*



(a) *PatSeeding* **(b)** *PatSeeding*

(c) *TsaSeeding* **(d)** *TsaSeeding*

(e) **(f)**

Figure C.3.: *Execution time at $v = 3$ of PatSeeding and TsaSeeding with hit cleaning disabled as function of the number of main tracker hits n_{hits} (a, c) and the number of visible interactions $n_{\text{vis.}}$ (b, d). A third order polynomial has been used to fit the graphs (a-d). (e) and (f) contain the distribution of the number of main tracker hits and the number of visible interactions in the events which entered the plots. The parametrisations which have been fitted can be found in Equations C.1, C.1, C.1 and C.1. The corresponding plots with hit cleaning enabled can be found in Figure 5.11.*

	long reconstructible		T reconstructible	
	eff. (clone) [%]	hit pur./eff. [%]	eff. (clone) [%]	hit pur./eff. [%]
all tracks	94.9 (0.2)	97.1 / 93.5	80.7 (0.2)	97.0 / 92.6
— $p > 5$GeV	95.6 (0.3)	97.2 / 94.4	95.2 (0.4)	97.4 / 94.0
B daughters	95.5 (0.1)	97.4 / 94.3	85.9 (0.1)	97.3 / 93.6
— $p > 5$ GeV	96.3 (0.2)	97.4 / 95.3	95.9 (0.2)	97.4 / 95.0
good B daughters	96.1 (0.1)	97.2 / 94.8	94.8 (0.2)	97.3 / 94.6
— $p > 5$ GeV	96.2 (0.1)	97.2 / 95.4	96.2 (0.1)	97.3 / 95.2
K_S^0/Λ daughters	94.7 (0.1)	97.0 / 92.9	78.7 (0.2)	97.0 / 92.1
— $p > 5$GeV	95.2 (0.3)	96.9 / 93.9	94.9 (0.4)	97.4 / 93.6
	tr. avg.	ev. avg.	tr. avg.	ev. avg.
ghost fraction	6.4	4.2	6.4	4.2

Table C.6.: *Reconstruction efficiencies and ghost rate for PatSeeding without hit cleaning and without Global Event Cuts (at $v = 3$).*

v		with GEC	without GEC
3	average time [ms]	26.08	27.25
	maximum time [ms]	981.6	1681.6

Table C.7.: *Average and maximum time needed to reconstruct an event for PatSeeding with and without Global Event Cuts (GEC) (both without hit cleaning).*

C.1.2. Influence of Global Event Cuts

In this subsection, the behaviour of PatSeeding is investigated in the case that "unfiltered" detector data is fed into the algorithm, so both Global Event Cuts and hit cleaning are disabled. This study gives some insight into how big a loss in reconstruction efficiency is due to Global Event Cuts, and how big a gain in terms of execution time is being realised.

Table C.6 shows reconstruction efficiency and ghost rate for PatSeeding at $v = 3$. The efficiency loss due to hit cleaning can be up to 0.2% (when compared to PatSeeding with Global Event Cuts (GEC), but without hit cleaning, see Table C.1); Figure shows track reconstruction efficiency and ghost rate as a function of p, p_T, η and the number of visible interactions for long reconstructible tracks. The corresponding plots for b and K_S^0/Λ daughter tracks do not show anything new compared to the ones for long reconstructible tracks, they have been omitted for that reason tracks do not show anything new compared to the ones for long reconstructible tracks, they have been omitted for that reason. Table C.7 gives the average and maximum execution time per event; the average does not change much if GEC are disabled, while the maximum execution time almost doubles at $v = 3$ when GEC are disabled.

The following parametrisations were fitted to timing plots for PatSeeding without hit cleaning and GEC:

- whatever is left:

$$t(n_{\text{hits}}) = (0.8386 \pm 0.0003) \cdot 10^{-9} \cdot n_{\text{hits}}^3 + (-7.0966 \pm 0.0042) \cdot 10^{-6} \cdot n_{\text{hits}}^2 + (24.4497 \pm 0.0193) \cdot 10^{-3} \cdot n_{\text{hits}} + (-19.8962 \pm 0.0260) \quad \text{(C.1)}$$

$$t(n_{\text{vis.}}) = (0.4946 \pm 0.0007) \cdot n_{\text{vis.}}^3 + (-1.2820 \pm 0.0087) \cdot n_{\text{vis.}}^2 + (7.3425 \pm 0.0302) \cdot n_{\text{vis.}} + (-0.6370 \pm 0.0307) \quad \text{(C.2)}$$

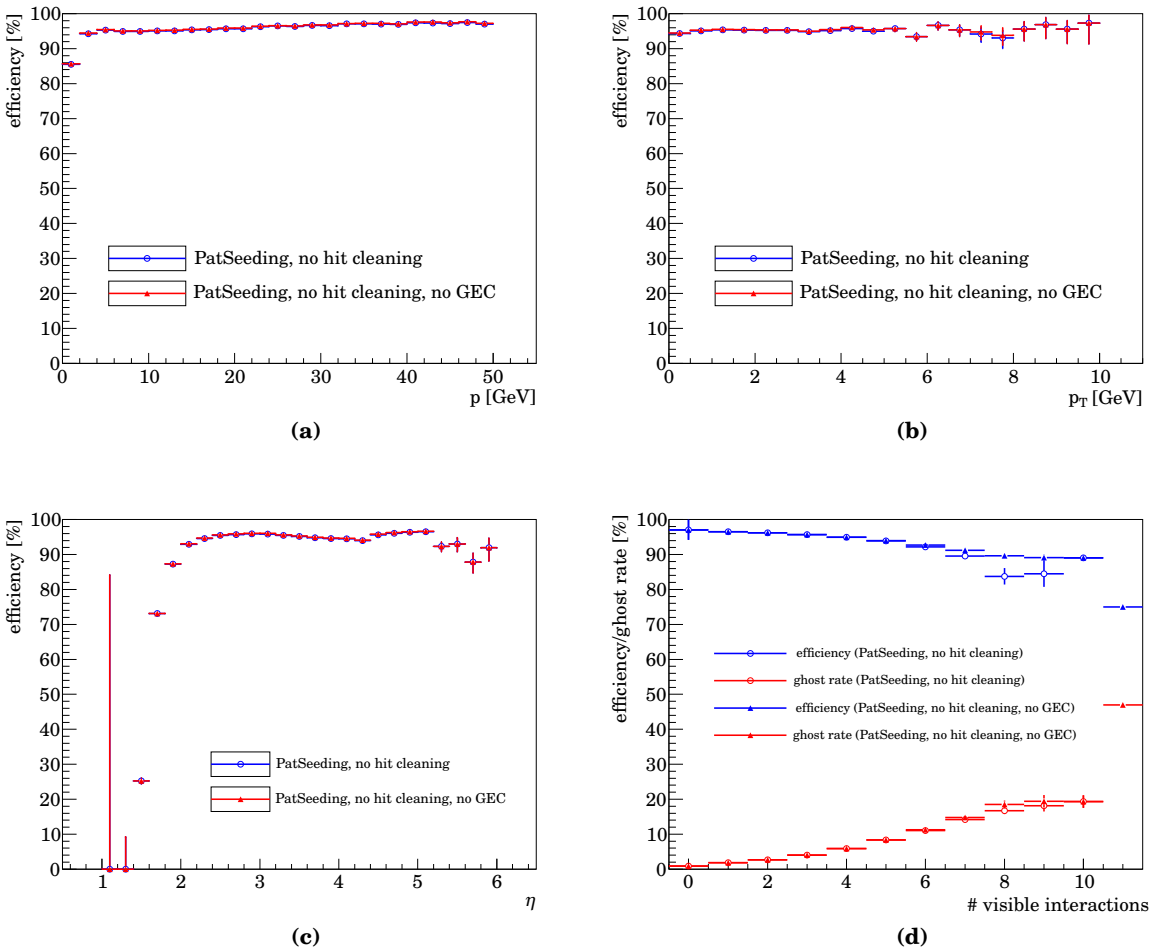


Figure C.4.: *Efficiency versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for tracks found by PatSeeding without Global Event cuts and without hit cleaning (triangles); the corresponding plot for PatSeeding with GEC but without hit cleaning has been added for comparison (circles).*

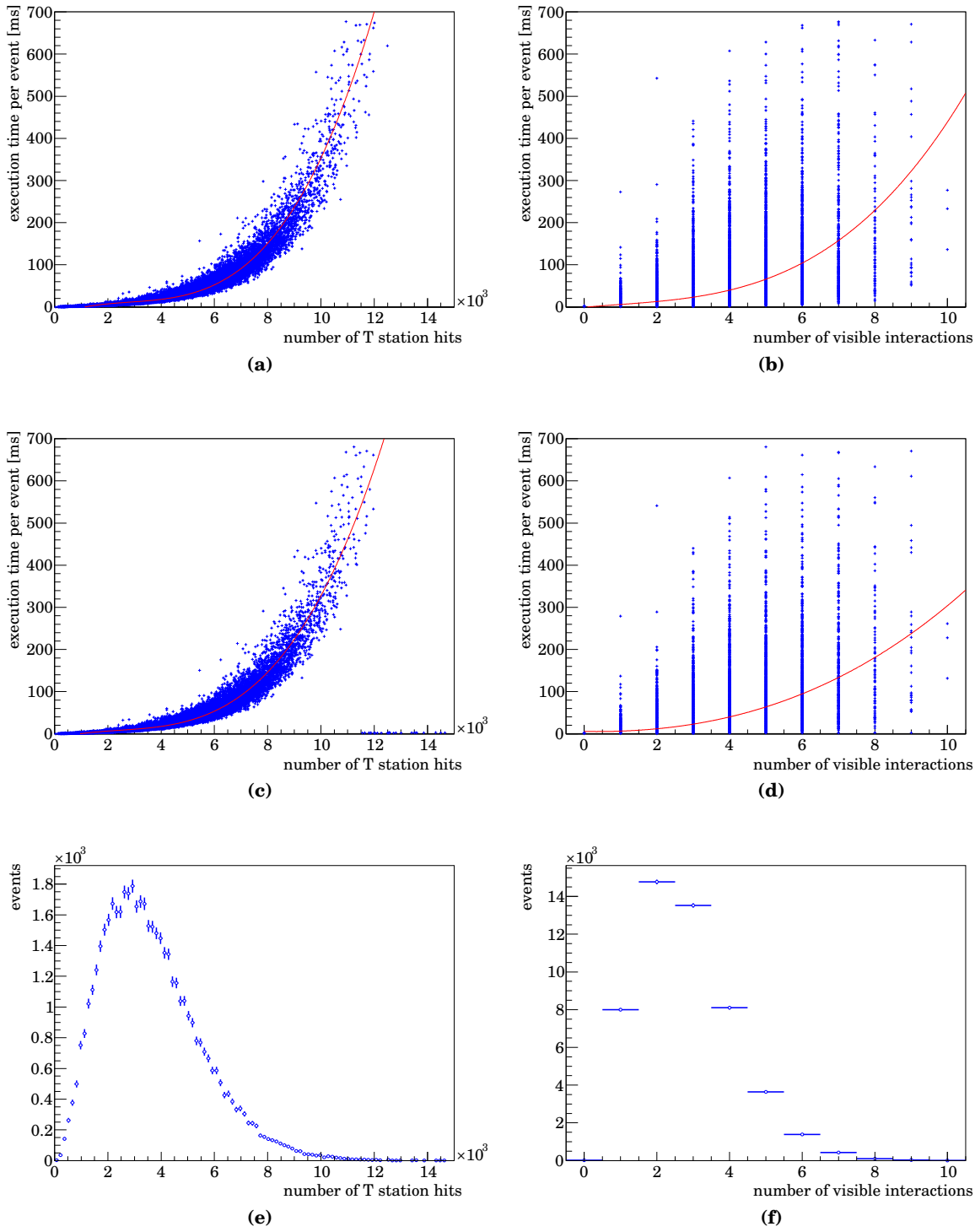


Figure C.5.: *Execution time at $\nu = 3$ of PatSeeding without Global Event Cuts as function of the number of main tracker hits n_{hits} (a, c) and the number of visible interactions $n_{\text{vis.}}$ (b, d). A third order polynomial has been used to fit the graphs; (a, b) are for PatSeeding without hit cleaning and GEC, (c, d) are for PatSeeding with GEC but without hit cleaning. (e) and (f) contain the distribution of the number of main tracker hits and the number of visible interactions in the events which entered the plots. The parametrisations which have been fitted can be found in Equations C.1 and C.2.*

C.2. Tracking with a misaligned detector

In this section, the effect of a misaligned detector on the pattern recognition is studied; the studies in this section are somewhat older than the rest of the document and date back to the time before any data was taken. Since the LHCb geometry does not favour alignment with cosmic muons, there used to be some concern in the collaboration that the initial misalignment of the detector would be so large that LHCb would have a hard time to obtain a track sample to do an initial alignment of the detector. While this did not turn out to be a problem, studies of this kind were important to show that tuning the pattern recognition to obtain very clean samples (even if reconstruction efficiency is low) is indeed possible. The issue is therefore discussed from the point of view from back then.

When reconstructing with a misaligned detector, one must take care of two things: First, one needs to widen search windows, χ^2 and outlier cuts, and second, it is usually necessary to suppress ghost tracks which would be killed by the tighter cuts applied in case of perfect alignment.

For the first category, a set of options will be given in C.2.2. The second category, killing ghosts, is tackled by requiring tracks to be isolated in the detector. This comes of course with a substantial loss in efficiency, but this is fine because for alignment, it is more important to have a clean track sample.

C.2.1. Isolated tracks

A track is called isolated in a layer if there are no hits within `ITIsolation` (the default is 15 mm, or 20 mm for `OTIsolation`, respectively) to the left and to the right of the hit in this layer. If the hit manager has marked a hit to potentially belong to a cluster, the other potential hit in the cluster is ignored for the check above.

A track is said to be isolated if:

- During the $x-z$ search per region, the three points forming the initial parabola must be isolated, and non-isolated hits are skipped when picking up further x hits around the parabola.
- While collecting stereo hits per region, a hit must either be isolated to be collected, or, if it is not, it must be the first hit which is not isolated in a strip of $\Delta y = 15\text{cm}$ (strips start at $y = 0$). This segmentation of stereo layers is done to avoid losing hits which are clearly separated in y, even if this is not apparent from their x coordinate. (Clearly, it would be better to demand the hit to be the only isolation-violating hit instead of the first, but the code would be more complex and slower.)
- When searching tracks spanning several regions, all hits must be isolated in their respective layers and regions.

To turn on the code enforcing these isolation criteria, the option `EnforceIsolation` must be set to true.

C.2.2. Performance

For this study, the same sample and software stack as in Section 7.4.1 was used. More specifically, four misalignment scenarios are investigated in which both Inner and Outer

tracker layers were shifted in x direction and rotated around all axes by a random amount, according to a Gaussian distribution. The width of these distributions (i.e. the parameter commonly called σ) is shown in Table C.8. To avoid very large misalignments, values differing from zero by more than 2.5σ were rejected, and another random number was drawn. The first two scenarios given in Table C.8 are chosen to represent deviations from detector survey results of the order of the accuracy claimed by these survey results. The second two scenarios are a worst-case study in case it turns out that the accuracy claimed by the survey are too optimistic.

All four scenarios were tested using 10,000 events, the two sets of options used can be found in Table C.9 (scenarios 2 and 4 with their large rotation angles require cuts to be wider). Please note that the set of options is not particularly well tuned, the objective is to get a clean track sample, not to have high reconstruction efficiency.

scenario	σ(shifts OT)	σ(shifts IT)	σ(rotations OT and IT)
1	1.0 mm	0.3 mm	1 mrad
2	1.0 mm	0.3 mm	10 mrad
3	4.0 mm	1.0 mm	1 mrad
4	4.0 mm	1.0 mm	10 mrad

Table C.8.: *Misalignment scenarios: Inner and Outer Tracker layers were shifted in x direction and rotated around all three axes according to a Gaussian distribution with mean σ.*

The aim of this study was to achieve very low a ghost fraction for alignment purposes, efficiency is considered less important in such a context. Table C.10 shows the results for the four scenarios. The track-averaged ghost fraction is reduced by a factor between roughly 6 and 9 depending on the misalignment scenario when compared to the figure given in Section 7.4 which was obtained on a simulated minimum bias sample with perfect alignment. Thus, a sizable reduction in ghosts was achieved, despite the misalignments applied.

option	value (scenarios 1, 3)	value (scenarios 2, 4)
CurveTol	12.0 mm	12.0 mm
TolCollectOT	5.5 mm	5.5 mm
TolCollectIT	1.5 mm	3.0 mm
MaxRangeOT	200. mm	200. mm
MaxChi2HitOT	400.	3600.
MaxChi2HitIT	400.	1800.
MaxFinalChi2	400.	3600.
MaxTrackChi2	400.	1800.
MaxTrackChi2LowMult	300.	1800.
MaxFinalTrackChi2	300.	1800.
MinTotalPlanes	11	11
CloneMaxXDist	20.0 mm	20.0 mm
QualityWeights	1.0, 0.0	1.0, 0.0
EnforceIsolation	true	true
MaxYAtOriginLowQual	600. mm	600. mm

Table C.9.: *Options used to reconstruct a misaligned detector*

		ghost fraction		
scenario	efficiency	track av.	event av.	clones
1	26.1%	1.8%	1.2%	3 in 67392
2	24.1%	2.8%	2.3%	3 in 49419
3	15.9%	2.0%	1.5%	0 in 38648
4	20.5%	2.3%	1.8%	2 in 44472

Table C.10.: *Performance for different misalignment scenarios (efficiency for long tracks with momenta above 5 GeV, number of clones among all reconstructed tracks).*

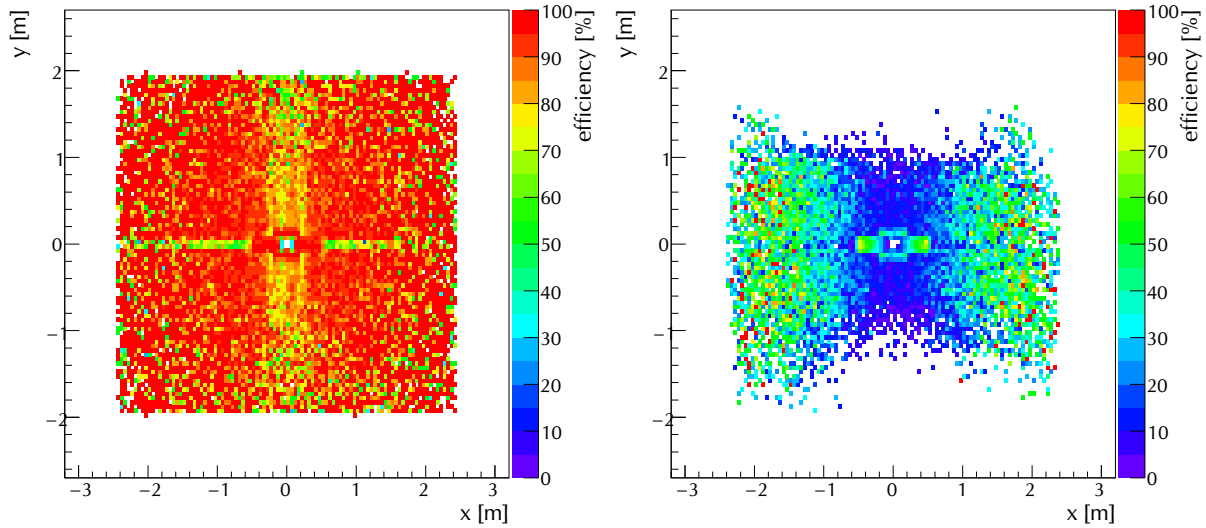


Figure C.6.: *Efficiency versus track position at $z = 7500$ mm. (a) is without misalignments, (b) is for scenario 4. The inefficiency at central x in the Outer Tracker is caused by high track multiplicity because the absence of the magnetic field means a larger number of low momentum particles can reach the main tracker. In (b), the inefficiency in the outer regions of the Outer Tracker is caused by rotations around the centre of the layers, leading to larger displacements of hits in these regions.*

There has been some concern in the collaboration that the isolation cuts used in this setup might cause the algorithm to fail completely in the most occupied regions of the detector, thus making alignment of these regions using this algorithm effectively impossible. Therefore, the efficiency of the algorithm was studied as function of the track position at $z = 7500$mm (cf Fig. C.6) for scenario 4 and for perfect alignment, also using the sample from 7.4.1.

In case of perfect alignment, the effect of the higher multiplicity in the main tracker (due to the absence of the magnetic field as filter for low momentum particles) can be seen in the central region of the Outer Tracker. In the misaligned case, the efficiency is clearly lower than under ideal conditions, but the algorithm manages to find tracks also in the dense regions of the detector. The plot for the misaligned case is not completely symmetric because Inner Tracker boxes can move independently, so that the actual misalignments applied are not symmetric themselves.

It must also be said that for the scenarios with large rotations (10 mrad), the algorithm has trouble reconstructing tracks in the Outer Tracker at large $|x|$ and particularly $|y|$. This is due to the large lever arm caused by the spatial extension of the Outer Tracker. However, rotations of 10 mrad, in particular around the z axis, seem to be quite big so that rotations are known to a better accuracy from survey measurements (10 mrad over the length of half an Outer Tracker module mean displacements of 2.5 cm).

C.3. Working in tandem with PatForward (missing plots)

This section contains the plots that were omitted from Section 7.3.

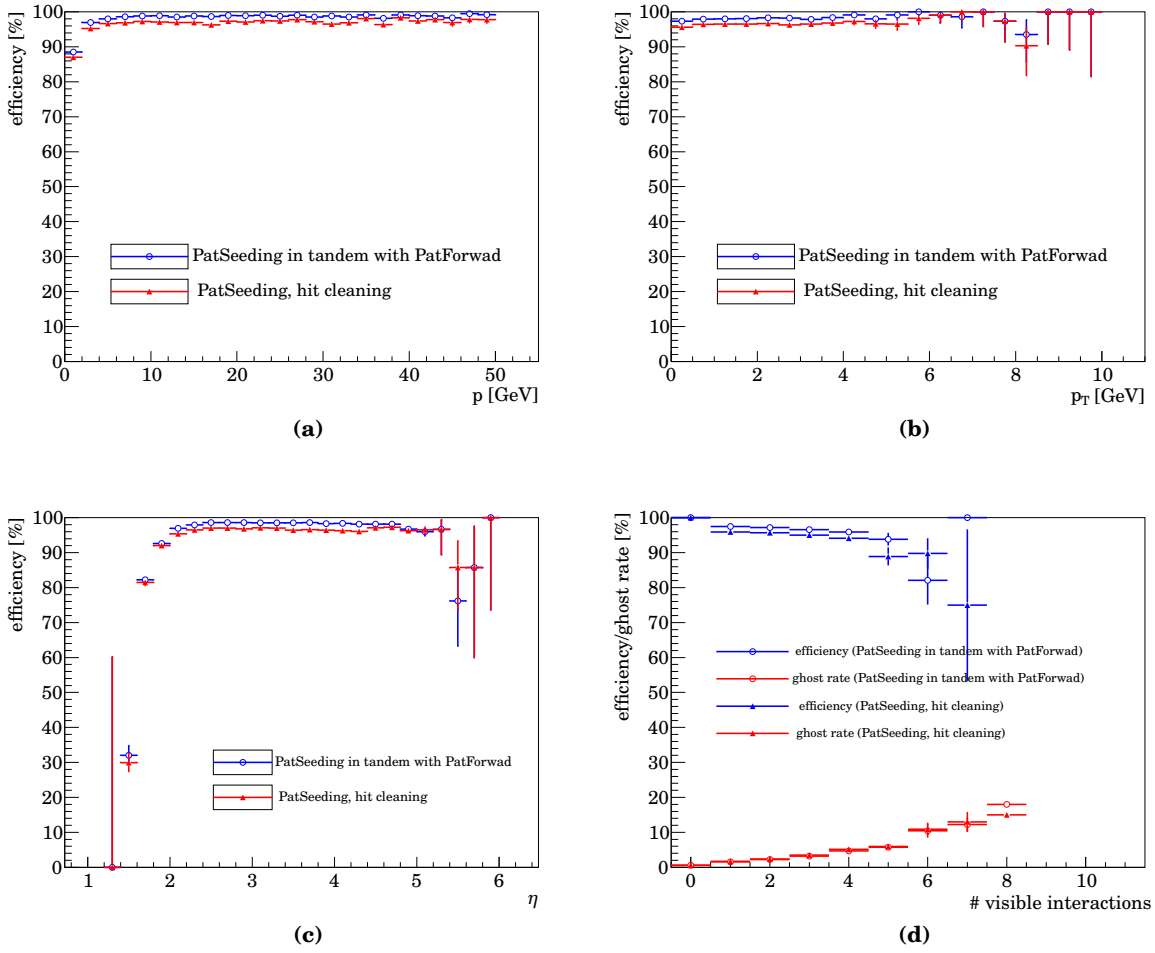


Figure C.7.: *Efficiency versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for b daughter tracks found by PatSeeding when running in tandem with PatForward (circles) and it its default configuration (triangles).*

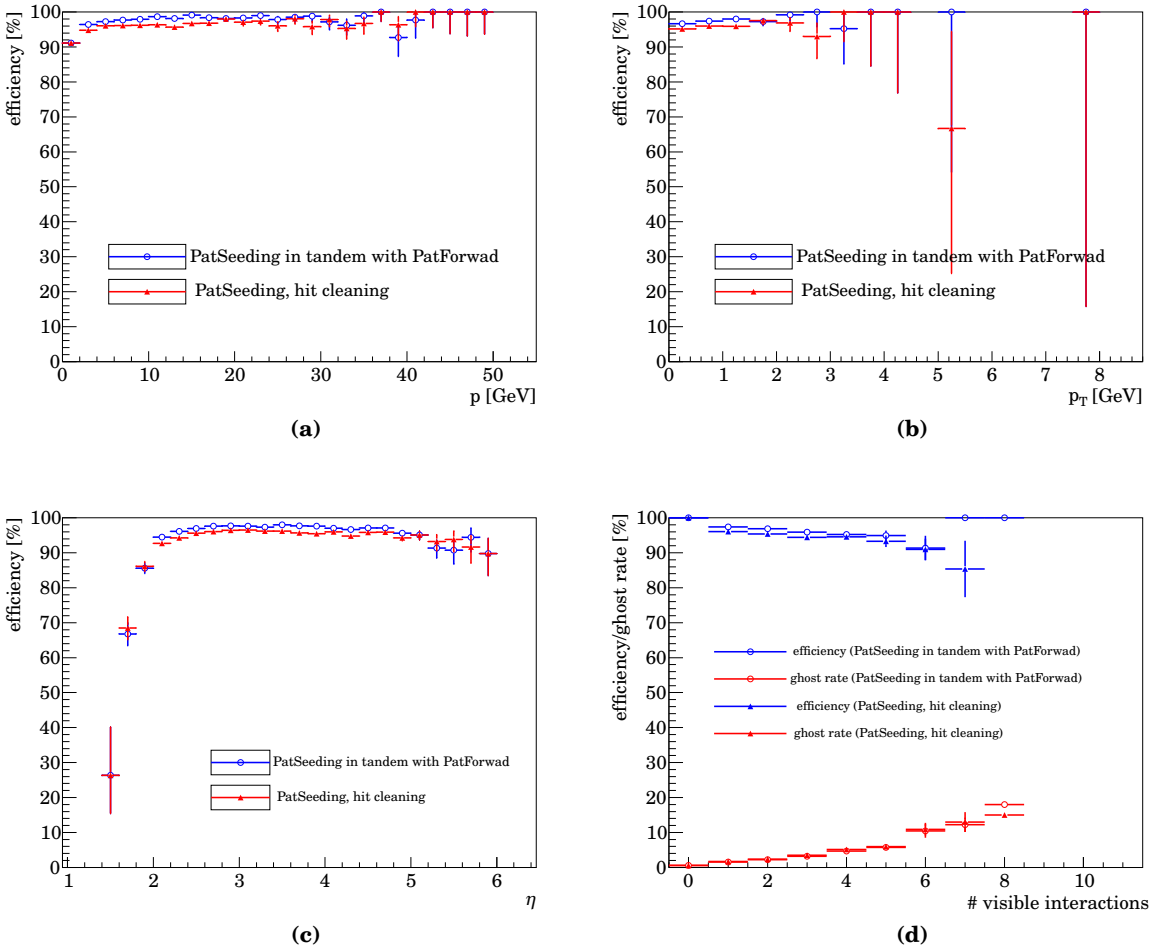


Figure C.8.: *Efficiency versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for K^0_S/Λ daughter tracks found by PatSeeding when running in tandem with PatForward (circles) and it its default configuration (triangles).*

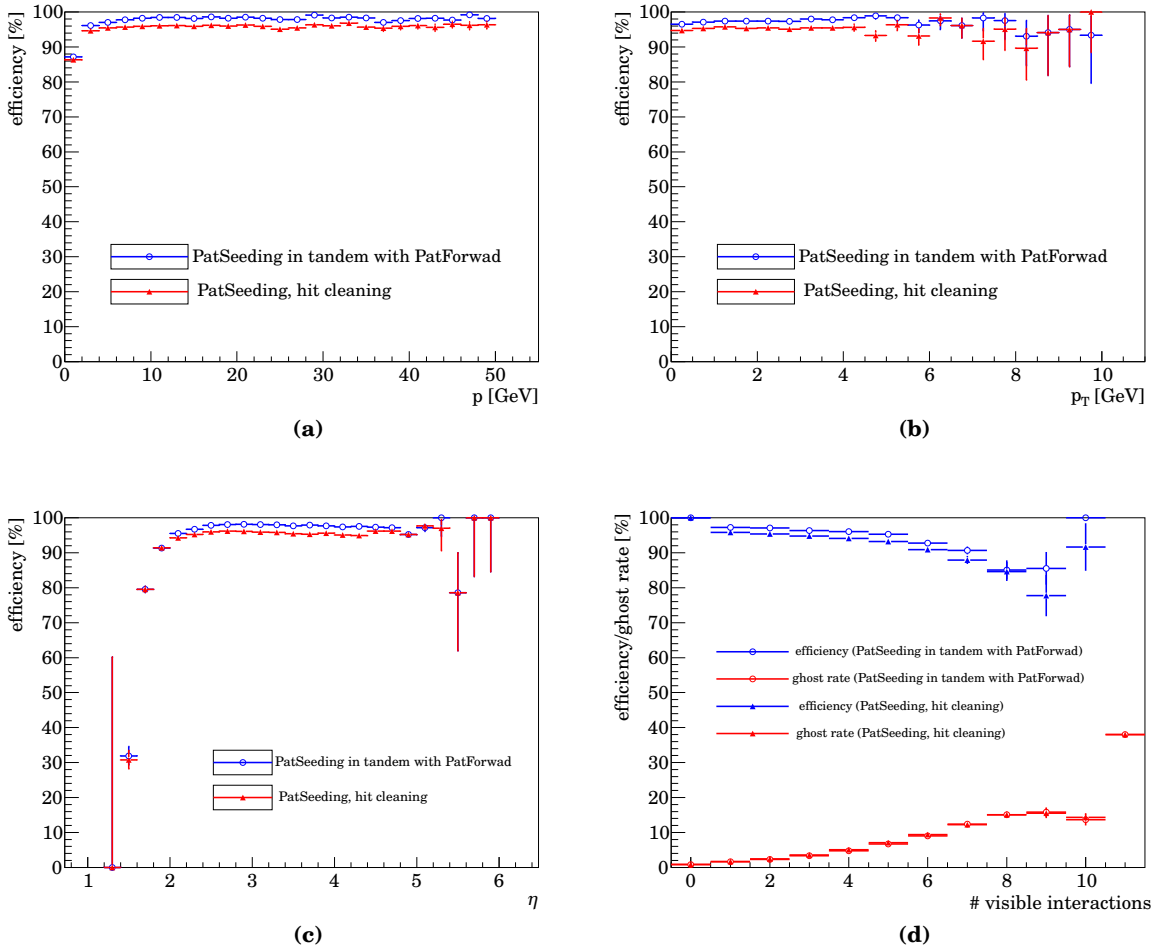


Figure C.9.: *Efficiency (at $\nu = 3$) versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for b daughter tracks found by PatSeeding when running in tandem with PatForward (circles) and in its default configuration (triangles).*

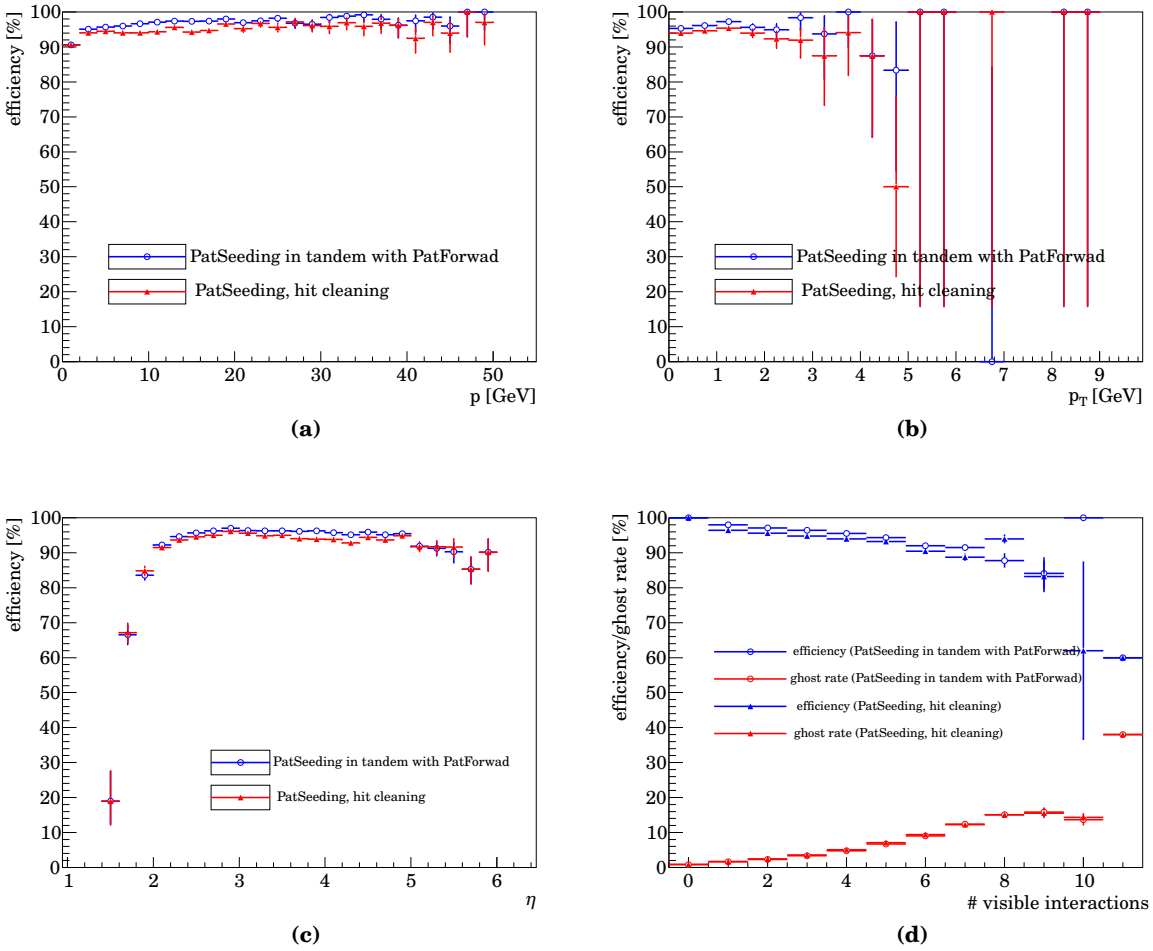


Figure C.10.: *Efficiency (at $v = 3$) versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for K^0_S/Λ daughter tracks found by PatSeeding when running in tandem with PatForward (circles) and in its default configuration (triangles).*

C.4. Tuning for early data taking conditions (missing plots)

This section contains plots omitted from Section 7.2.

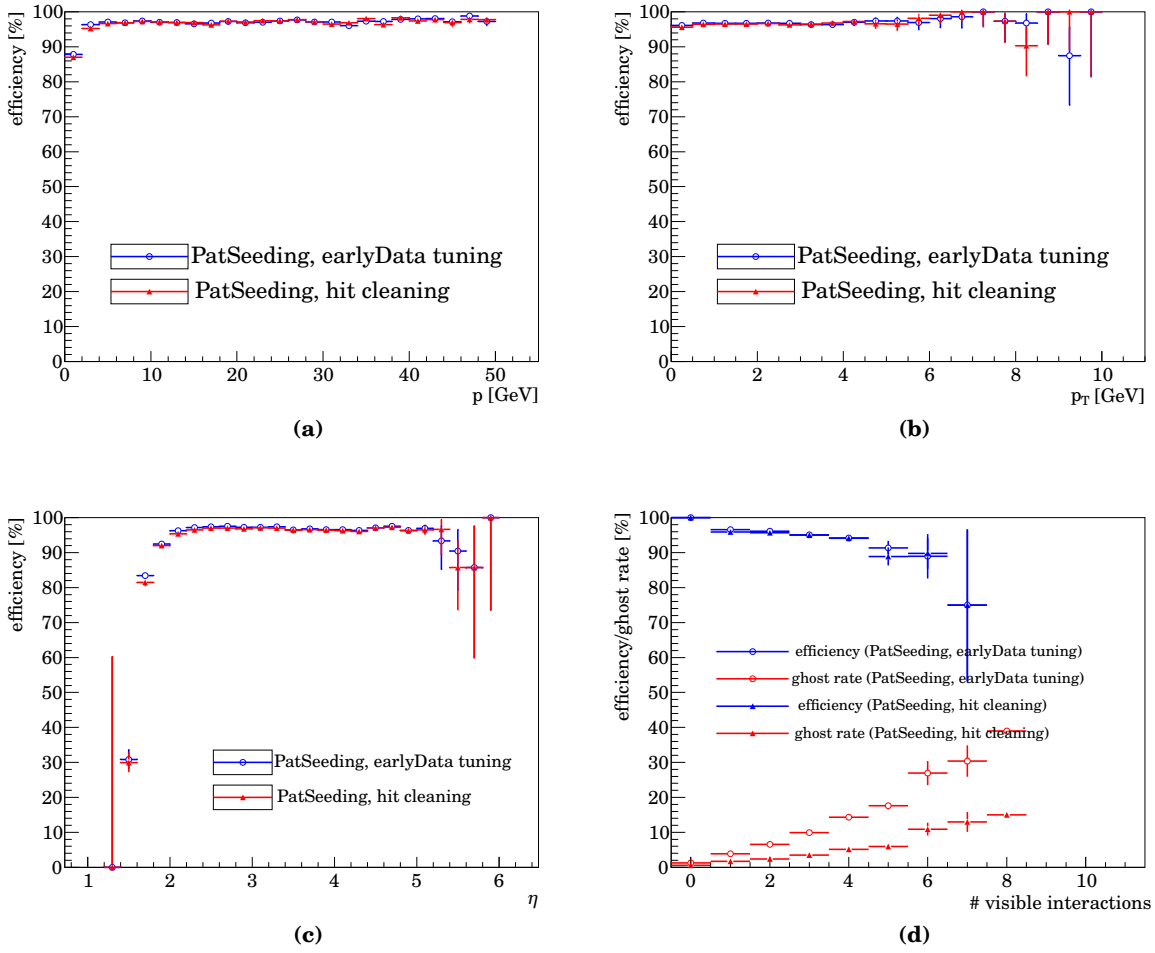


Figure C.11.: *Efficiency versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for b daughter tracks found by PatSeeding at $v = 1$ using a tuning for early data taking conditions (circles) and the standard tuning for comparison (triangles).*

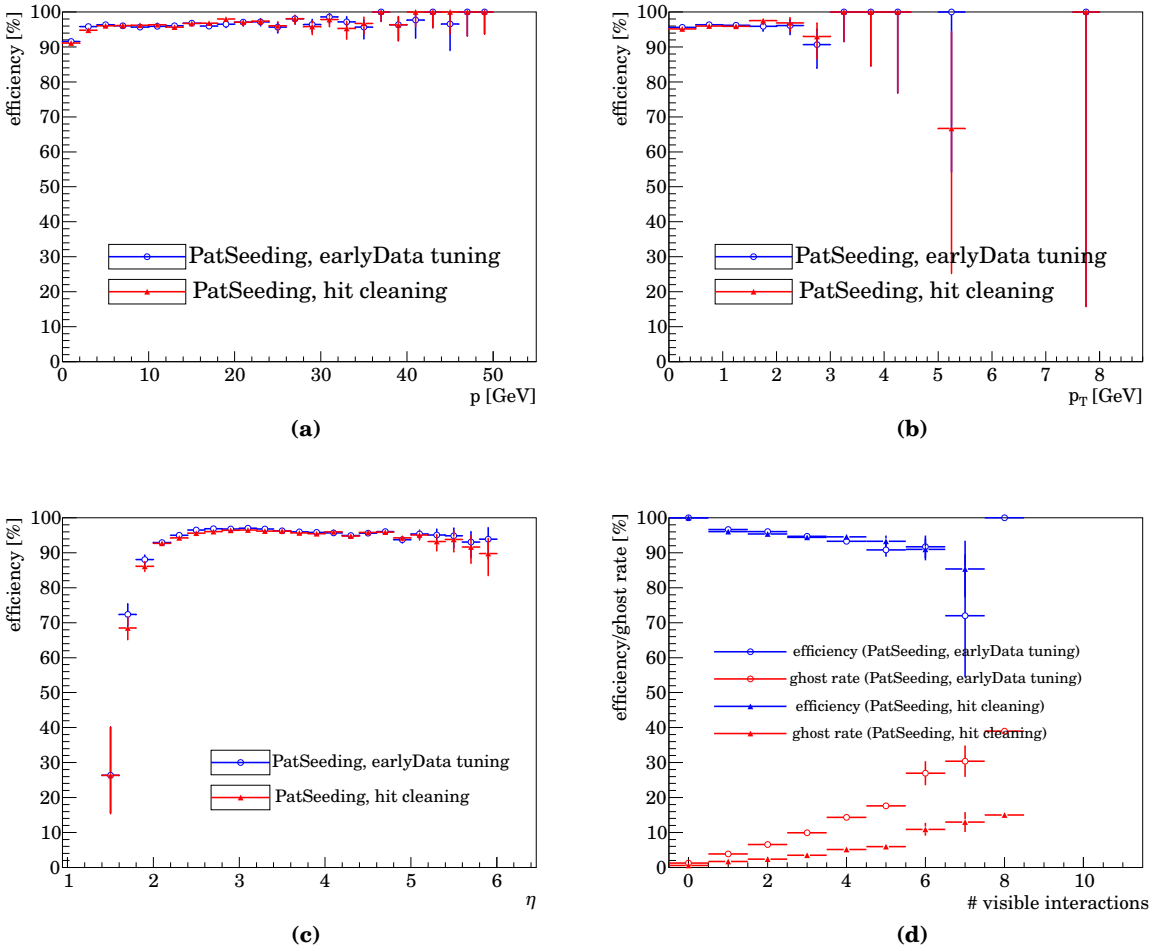


Figure C.12.: *Efficiency versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for K^0_S/Λ daughter tracks found by PatSeeding at $v = 1$ using a tuning for early data taking conditions (circles) and the standard tuning for comparison (triangles).*

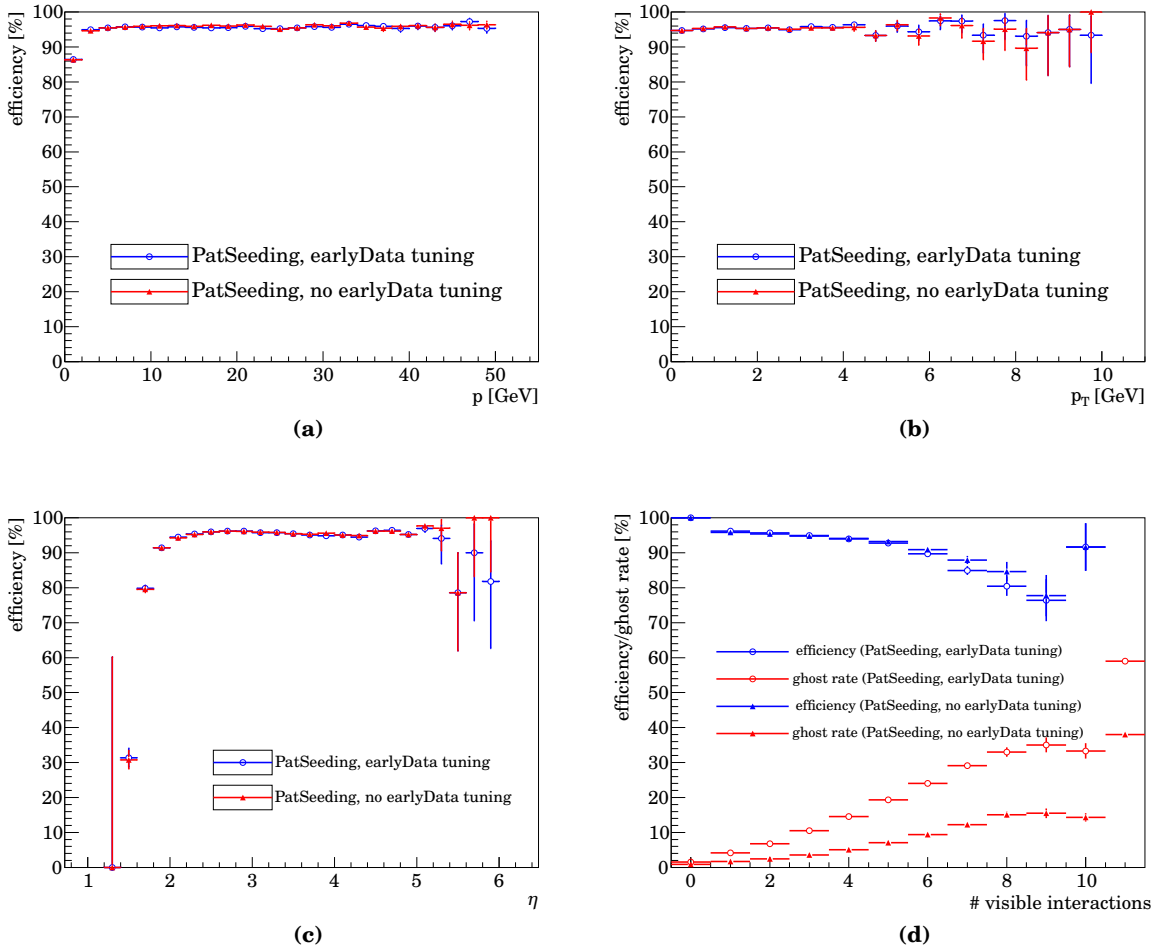


Figure C.13.: *Efficiency versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for b daughter tracks found by PatSeeding at $\nu = 3$ using a tuning for early data taking conditions (circles) and the standard tuning for comparisons (triangles).*

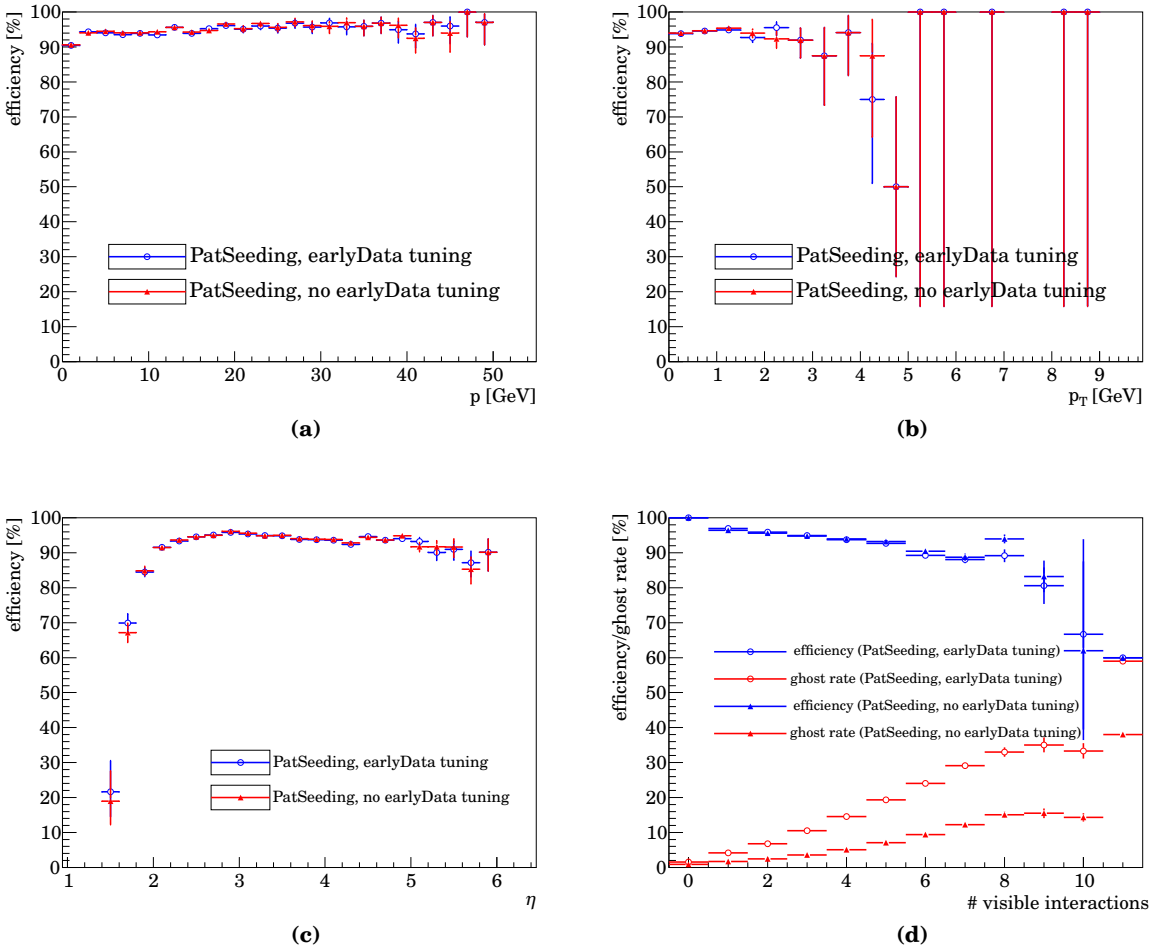


Figure C.14.: *Efficiency versus (a) p, (b) p_T, (c) η, and (d) efficiency and ghost fraction versus number of visible interactions for K^0_S/Λ daughter tracks found by PatSeeding at $v = 3$ using a tuning for early data taking conditions (circles) and the standard tuning for comparisons (triangles).*

D. Cholesky decomposition

Cholesky decomposition is a matrix decomposition method for symmetric positive definite matrices M. M is decomposed into a lower triangular matrix L times its transpose, $M = LL^T$. The components l_{ij} of L are computed from the components m_{ij} of the matrix M according to the following scheme:

$$l_{ij} = \begin{cases} 0 & i < j \\ \sqrt{m_{ii} - \sum_{k<i} l_{ik}^2} & i = j \\ (m_{ij} - \sum_{k<j} l_{ik} l_{jk})/l_{jj} & i > j \end{cases}$$

Cholesky decomposition has a number of advantages over the more conventional LU decomposition or (most) hand-written solvers because it takes advantage of the special form of the matrices in fit problems:

- reduced memory and time consumption: only lower half of the matrix and the decomposition needs to be stored (and computed), since the matrix is symmetric
- numerically stable: the decomposition does not add significantly to numerical problems (see below for details)
- bad tracks which lead to saddle points or maxima in χ^2 are easily caught and discarded because the Eigenvalues of M are no longer strictly positive which becomes apparent during decomposition

Hand optimised code for matrix dimensions 1 through 6 is available, and generic code exists for larger matrices. The code is part of the ROOT framework ([75]) since version 5.24 and can be found in `Math/CholeskyDecomp.h` for standalone applications with an emphasis on speed or as part of the more convenient `SMatrix` class family (more specifically, the `InvertChol` method for symmetric matrices, and the `SolveChol` function). In LHCb software, this code is used in most of the pattern recognition code and in the Kalman filter code for track fitting.

To see why Cholesky decomposition is numerically stable, we need to analyse what happens with the numerical roundoff when a linear system is solved. I will not derive this in any mathematical rigour, but only make plausible why this is the case. The rigorous proof would follow the outline I give below, though.

In its most general case, a decomposition of a matrix M has the from $M = TR$ where T is some transformation matrix and R is some matrix for which it is easy to compute the solution of a linear system (typically, R is an upper triangular matrix). To solve $Mx = r$, one needs to calculate $y = T^{-1}r$ and then solve $Rx = y$.

In a general matrix-vector multiplication like e.g. $y = T^{-1}r$ above, the total error due to numerical inaccuracies in y, e_y, will behave like $||e_y|| \leq c||e_r||$ where e_r is the

(numerical) error in r and c is a constant that depends on the numerical precision of the floating point implementation of the computer that is used for the computation ("machine precision"), the number of floating point operations needed to compute the matrix-vector multiplication and the eigenvalues of T^{-1}. To illustrate this last point, consider that doubling all eigenvalues of T^{-1} is equivalent to multiplying T^{-1} by a factor of 2 which will for sure also scale the numerical error due to roundoff by this factor. However, since we must preserve the product $TR = M$, doubling T^{-1} must be compensated by doubling R to preserve the overall result M. In the end, that means that one wants the absolute values of the eigenvalues of T^{-1} to be as close as possible to 1 (i.e. the overall scaling behaviour of M is kept in R), and that the size of c above is governed by the ratio of the largest to the smallest absolute value of the eigenvalues of T^{-1}.

The total error on x can then be written as $||e_x|| \leq c_T c_R ||e_r||$ where c_T is the constant we obtained above and c_R is the constant related to the matrix R. R will depend on the original matrix M and our choice of T. The trick to get a numerically stable decomposition scheme is to chose T^-1 in such a way that it does neither attenuate nor amplify the numerical roundoff along any of its eigenvectors (i.e. "it does not make numerical roundoff worse"), or, equivalently, that the absolute value of any eigenvalue of T^{-1} must be 1 (in which case the absolute values of all eigenvalues of T itself must also be 1). In that case, c_T becomes as small as one can possibly make it.

To make plausible why Cholesky decomposition is numerically well behaved, we first need to show that the Eigenvalues of such a decomposition have to be positive (it is also true that all symmetric real matrices M with positive Eigenvalues can be decomposed using Cholesky decomposition in the mathematical sense (i.e. numerical problems such as roundoff are neglected), but the proof is less interesting and will not be shown here). Consider a related decomposition, $M = L'DL'^T$, where the lower triangular matrix L' has only entries of 1 on its diagonal, and D has positive entries on its diagonal and is zero everywhere else (the connection between L, L' and D is $L = L'\sqrt{D}$ where $(\sqrt{D})_i = \sqrt{d_{ii}} = l_{ii}$ and d_{ii} are the elements on the diagonal of D). It is trivial to see that the Eigenvalues of D are positive (since $d_{ii} = l_{ii}^2$), and the k Eigenvalues of L' are all 1 (for a kxk matrix L' of the given form, the characteristic polynomial $\det(L' - \lambda\mathbb{1})$ goes to zero like $(1-\lambda)^k$ as λ goes to 1).

From that argument, it is clear that L' (which plays the role of our transformation matrix T above) has all eigenvalues equal to 1, i.e. it is optimal in the sense given above. For the real Cholesky decomposition, one has to deal with L instead of L'. Since L' and L have their eigenvectors in the same direction (one obtains L' from L by a scaling of eigenvectors), it does not really matter if one moves part of the scaling along the direction of an eigenvector from $R = DL'$ into L (to be precise, the "amount of scaling of eigenvectors" that is moved is exactly $\sqrt{D}$ as defined above). Therefore, Cholesky decomposition is numerically stable.

E. Options for `FlagHitsForPatternReco`

This chapter lists the options which were used to produce the Monte Carlo sample with the right fraction of hits dropped in each layer of TT, IT and OT. While this information is a little technical, it would be an omission to not actually include the correction factors determined.

The `FlagHitsForPatternReco` algorithm takes a list of strings which tells it which fraction of hits in each detector element to flag for hit dropping later in the pattern recognition. These strings are of the form `"DDAAAAE0.75"` where `DD` stands for the subdetector (here, one of TT, IT or OT), `AAAA` is a string which specifies the "address" of the subdetector element and an `E` followed by the efficiency to apply to that element (in the example string above, element `AAAA` of subdetector `DD` would have 25% of its hits dropped). The string `AAAA` is quite flexible and can contain the following constituents (which are all optional, and the list is not exhaustive, only what is needed to understand the strings below is given):

- an `S` followed by the station number in that subdetector
- an `L` followed by the layer number
- in OT, an `Q` followed by a number specifying which quarter is meant (each quarter corresponds to a quadrant in the $x-y$ plane)
- in IT and TT, this may be followed by `R` followed by a region number (regions in IT correspond to top, bottom, left and right boxes, in TT, they roughly correspond the the four quadrants in the $x-y$ plane)
- an `M` followed by a module number
- in the OT, an `OTIS` followed by an OTIS number which specifies which of the four OTIS TDC chips which feed the readout chip for a single OT module is meant

Counting starts at 0. The options used for the analysis are:

```
from Gaudi.Configuration import *
from GaudiConf.Configuration import *
from Configurables import GaudiSequencer

def patch_earlyDecoding():
        GaudiSequencer("RecoDecodingSeq").Members += [
                GaudiSequencer("RecoHitEffPatchingSeq") ]

appendPostConfigAction(patch_earlyDecoding)

from Configurables import ( GaudiSequencer,
   FlagHitsForPatternReco )
```

```
hiteffpatchSeq = GaudiSequencer("RecoHitEffPatchingSeq")
hiteffpatcher = FlagHitsForPatternReco("FlagHitsForPatternReco")
hiteffpatchSeq.Members = [ hiteffpatcher ]

hiteffpatcher.EfficiencyCorrections += [
                # hit efficiency correction (step 3), Repro-8
                    alignment
                # OT
                "OTS0L0E1.0",
                "OTS0L1E0.963745",
                "OTS0L2E0.980674",
                "OTS0L3E0.991592",
                "OTS1L0E0.991053",
                "OTS1L1E0.976663",
                "OTS1L2E0.972575",
                "OTS1L3E0.978313",
                "OTS2L0E0.995818",
                "OTS2L1E0.983799",
                "OTS2L2E0.982114",
                "OTS2L3E1.0",
                # TT
                "TTS0L0E0.99538",
                "TTS0L1E0.989285",
                "TTS1L0E0.993469",
                "TTS1L1E0.993944",
                # IT
                "ITS0L0R0E0.974497",
                "ITS0L0R1E0.974497",
                "ITS0L0R2E0.94269",
                "ITS0L0R3E0.94269",
                "ITS0L1R0E0.923044",
                "ITS0L1R1E0.923044",
                "ITS0L1R2E0.932099",
                "ITS0L1R3E0.932099",
                "ITS0L2R0E0.697963",
                "ITS0L2R1E0.697963",
                "ITS0L2R2E0.924244",
                "ITS0L2R3E0.924244",
                "ITS0L3R0E0.935116",
                "ITS0L3R1E0.935116",
                "ITS0L3R2E0.964464",
                "ITS0L3R3E0.964464",
                "ITS1L0R0E0.955833",
                "ITS1L0R1E0.955833",
                "ITS1L0R2E0.977474",
                "ITS1L0R3E0.977474",
                "ITS1L1R0E0.955529",
                "ITS1L1R1E0.955529",
                "ITS1L1R2E0.938101",
                "ITS1L1R3E0.938101",
                "ITS1L2R0E0.939121",
```

```
                "ITS1L2R1E0.939121",
                "ITS1L2R2E0.95211",
                "ITS1L2R3E0.95211",
                "ITS1L3R0E0.926615",
                "ITS1L3R1E0.926615",
                "ITS1L3R2E0.975615",
                "ITS1L3R3E0.975615",
                "ITS2L0R0E1.0",
                "ITS2L0R1E1.0",
                "ITS2L0R2E0.932171",
                "ITS2L0R3E0.932171",
                "ITS2L1R0E0.839948",
                "ITS2L1R1E0.839948",
                "ITS2L1R2E0.853523",
                "ITS2L1R3E0.853523",
                "ITS2L2R0E0.965543",
                "ITS2L2R1E0.965543",
                "ITS2L2R2E0.962833",
                "ITS2L2R3E0.962833",
                "ITS2L3R0E1.0",
                "ITS2L3R1E1.0",
                "ITS2L3R2E0.967506",
                "ITS2L3R3E0.967506"
]

from Configurables import ( GaudiSequencer,
    FlagHitsForPatternReco )

hiteffpatchSeq = GaudiSequencer("RecoHitEffPatchingSeq")
hiteffpatcher = FlagHitsForPatternReco("FlagHitsForPatternReco")
hiteffpatchSeq.Members = [ hiteffpatcher ]

hiteffpatcher.EfficiencyCorrections += [
        # Dirk Wiedner inspected run 63809 (23k events, 12. 12.
            2009 17:31-18:23)
        # and produced the following list of dead channels
        #
        # T1
        "OTS0L0Q1M0E0.0",       # no frontend box
        "OTS0L0Q3M0OTIS3E0.0",  # HV trip
        "OTS0L1Q1M1OTIS1E0.0",  # missing hits due to noise (ch
            0-15, 20-63)
        "OTS0L1Q1M7OTIS3E0.0",  # HV trip
        "OTS0L1Q2M7E0.0",       # disables FE (unresponsive to
            slow control)
        "OTS0L2Q0M6OTIS0E0.0",  # HV trip
        "OTS0L2Q0M6OTIS1E0.0",  # HV trip
        "OTS0L2Q0M7OTIS1E0.0",  # HV trip
        "OTS0L2Q0M7OTIS3E0.0",  # HV trip
        "OTS0L2Q2M4OTIS2E0.0",  # HV trip
        "OTS0L2Q2M6OTIS0E0.0",  # HV trip
        "OTS0L2Q2M6OTIS1E0.0",  # HV trip
```

```
        "OTS0L2Q2M7OTIS0E0.0",  # HV trip
        "OTS0L2Q2M7OTIS1E0.0",  # HV trip

        # T2
        "OTS1L0Q3M8OTIS3E0.0",  # ???
        "OTS1L1Q1M4OTIS0E0.0",  # HV trip
        "OTS1L1Q1M5OTIS1E0.0",  # HV trip
        "OTS1L1Q2M8OTIS3E0.0",  # ???
        "OTS1L2Q3M0OTIS1E0.0",  # ???
        "OTS1L3Q0M8E0.0",       # disabled (DLL lock lost, i.e.
           no time info)
        "OTS1L3Q2M8OTIS3E0.0",  # HV trip

        # T3
        "OTS2L0Q1M1OTIS0E0.0",  # ???
        "OTS2L1Q1M7OTIS2E0.0",  # HV trip
        "OTS2L2Q1M4OTIS1E0.0",  # ???
        "OTS2L2Q1M4OTIS2E0.0",  # ???
        "OTS2L3Q1M0OTIS1E0.0",  # HV broken
]
```

List of Tables

List of Figures

Bibliography

[1] Peter W. Higgs. “Broken Symmetries and the Masses of Gauge Bosons”. In: *Phys. Rev. Lett.* 13.16 (1964), pp. 508–509. DOI: 10.1103/PhysRevLett.13.508.

[2] LEP electroweak working group et al. “Precision Electroweak Measurements and Constraints on the Standard Model”. In: (2010). eprint: 1012.2367.

[3] R Aaij et al. “Prompt K^0_S production in pp collisions at $\sqrt{s} = 0.9$ TeV”. In: *Phys. Lett.* B693 (2010), pp. 69–80. DOI: 10.1016/j.physletb.2010.08.055. eprint: 1008.3105.

[4] Francis Halzen and Alan D. Martin. *Quarks and Leptons: An Introductory Course in Modern Particle Physics*. John Wiley & Sons Inc., 1984.

[5] Otto Nachtmann. *Ph"anomene und Konzepte der Elementarteilchenphysik*. Friedr. Vieweg & Sohn, 1986.

[6] Donald H. Perkins. *Introduction to High Energy Physics*. 4th. Cambridge University Press, 2000.

[7] K Nakamura and Particle Data Group. “Review of Particle Physics”. In: *Journal of Physics G: Nuclear and Particle Physics* 37.7A (2010), p. 075021. URL: http://stacks.iop.org/0954-3899/37/i=7A/a=075021.

[8] S. L. Glashow, J. Iliopoulos and L. Maiani. “Weak Interactions with Lepton-Hadron Symmetry”. In: *Phys. Rev. D* 2.7 (1970), pp. 1285–1292. DOI: 10.1103/PhysRevD.2.1285.

[9] Et al. “Roadmap for selected key measurements of LHCb”. In: (2009). eprint: 0912.4179.

[10] The LHCb Collaboration et al. “The LHCb Detector at the LHC”. In: *Journal of Instrumentation* 3 (Aug. 2008), pp. 8005–+. DOI: 10.1088/1748-0221/3/08/S08005.

[11] *CERN Workshop on Standard Model Physics (and more) at the LHC*. oai:cds.cern.ch:425440. CERN. Geneva: CERN, 2000. URL: http://cdsweb.cern.ch/record/425440.

[12] R. Aaij et al. “Measurement of $\sigma(pp \longrightarrow b\bar{b}X)$ at $\sqrt{s} = 7$TeV in the forward region”. In: *Physics Letters B* 694.3 (2010), pp. 209 –216. ISSN: 0370-2693. DOI: DOI:10.1016/j.physletb.2010.10.010. URL: http://www.sciencedirect.com/science/article/B6TVN-51726PD-3/2/0268b89e6e3e016e05f9bbf57733eec9.

[13] J Nardulli. “Reconstruction of two-body B decays in LHCb.” Presented on 04 Oct 2007. PhD thesis. Amsterdam: Vrije Univ. Amsterdam, 2007. URL: http://cdsweb.cern.ch/record/1057762.

[14] *LHCb : Technical Proposal*. Tech. Proposal. Geneva: CERN, 1998. URL: http://cdsweb.cern.ch/record/622031?ln=en.

[15] J. Van Tilburg and M. Merk. "Track simulation and reconstruction in LHCb." Presented on 01 Sep 2005. PhD thesis. Amsterdam: Vrije Univ. Amsterdam, 2005. URL: http://cdsweb.cern.ch/record/885750?ln=en.

[16] S Amato et al. *LHCb magnet: Technical Design Report*. Technical Design Report LHCb. Geneva: CERN, 2000. URL: http://cdsweb.cern.ch/record/424338?ln=en.

[17] P R Barbosa-Marinho et al. *LHCb VELO (VErtex LOcator): Technical Design Report*. Technical Design Report LHCb. Geneva: CERN, 2001. URL: http://cdsweb.cern.ch/record/504321?ln=en.

[18] R Antunes-Nobrega et al. *LHCb reoptimized detector design and performance: Technical Design Report*. Technical Design Report LHCb. Geneva: CERN, 2003. URL: http://cdsweb.cern.ch/record/630827?ln=en.

[19] P R Barbosa-Marinho et al. *LHCb inner tracker: Technical Design Report*. Technical Design Report LHCb. revised version number 1 submitted on 2002-11-13 14:14:34. Geneva: CERN, 2002. URL: http://cdsweb.cern.ch/record/582793?ln=en.

[20] P R Barbosa-Marinho et al. *LHCb outer tracker: Technical Design Report*. Technical Design Report LHCb. Geneva: CERN, 2001. URL: http://cdsweb.cern.ch/record/519146?ln=en.

[21] S Amato et al. *LHCb RICH: Technical Design Report*. Technical Design Report LHCb. Geneva: CERN, 2000. URL: http://cdsweb.cern.ch/record/494263?ln=en.

[22] S Amato et al. *LHCb calorimeters: Technical Design Report*. Technical Design Report LHCb. Geneva: CERN, 2000. URL: http://cdsweb.cern.ch/record/494264?ln=en.

[23] P R Barbosa-Marinho et al. *LHCb muon system: Technical Design Report*. Technical Design Report LHCb. Geneva: CERN, 2001. URL: http://cdsweb.cern.ch/record/504326?ln=en.

[24] R Antunes-Nobrega et al. *LHCb trigger system: Technical Design Report*. Technical Design Report LHCb. revised version number 1 submitted on 2003-09-24 12:12:22. Geneva: CERN, 2003. URL: http://cdsweb.cern.ch/record/630828?ln=en.

[25] "The Gaudi project". project web page. URL: http://proj-gaudi.web.cern.ch/proj-gaudi/.

[26] "The Gauss project". project web page. URL: http://lhcb-release-area.web.cern.ch/LHCb-release-area/DOC/gauss/.

[27] Torbjörn Sjöstrand, Stephen Mrenna and Peter Skands. "PYTHIA 6.4 physics and manual". In: *Journal of High Energy Physics* 2006.05 (2006), p. 026. URL: http://stacks.iop.org/1126-6708/2006/i=05/a=026.

[28] Anders Ryd et al. "EvtGen: A Monte Carlo Generator for B-Physics". In: (). EVTGEN-V00-11-07.

[29] S. Agostinelli et al. “GEANT4–a simulation toolkit”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506.3 (2003), pp. 250 –303. ISSN: 0168-9002. DOI: DOI:10.1016/S0168-9002(03)01368-8. URL: http://www.sciencedirect.com/science/article/B6TJM-48TJFY8-5/2/23ea98096ce11c1be446850c04cfa498.

[30] “The Boole project”. project web page. URL: http://lhcb-release-area.web.cern.ch/LHCb-release-area/DOC/boole/.

[31] “The Moore project”. project web page. URL: http://lhcb-release-area.web.cern.ch/LHCb-release-area/DOC/moore/.

[32] “The Brunel project”. project web page. URL: http://lhcb-release-area.web.cern.ch/LHCb-release-area/DOC/brunel/.

[33] “The DaVinci project”. project web page. URL: http://lhcb-release-area.web.cern.ch/LHCb-release-area/DOC/davinci/.

[34] “The Panoramix project”. project web page. URL: http://lhcb-release-area.web.cern.ch/LHCb-release-area/DOC/panoramix/.

[35] G Gracia et al. *Track reconstruction for LHCb*. Tech. rep. LHCb-98-045. Geneva: CERN, Feb. 1998. URL: http://cdsweb.cern.ch/record/684492?ln=en.

[36] R. Fruhwirth. “Application of Kalman filtering to track and vertex fitting”. In: *Nucl. Instrum. Meth.* A262 (1987), pp. 444–450.

[37] Pierre Billoir. “TRACK FITTING WITH MULTIPLE SCATTERING: A NEW METHOD”. In: *Nucl. Instr. Meth.* A225 (1984), p. 352.

[38] R. Kalman. “A new approach to linear filtering and prediction problems”. In: *Transactions of the ASME, Journal of Basic Engineering* D82 (1960), pp. 35–45.

[39] D. Hutchcroft. *VELO Pattern Recognition*. Tech. rep. LHCb-2007-013. CERN-LHCb-2007-013. Geneva: CERN, Mar. 2007. URL: http://cdsweb.cern.ch/record/1023540?ln=en.

[40] T Lastoviicka. *Generic VELO Pattern Recognition*. Tech. rep. LHCb-2007-002. CERN-LHCb-2007-002. Geneva: CERN, Feb. 2008. URL: http://cdsweb.cern.ch/record/1087659?ln=en.

[41] S Redford. *Profiling the VELO Pattern Recognition And Vertexing Algorithms*. Tech. rep. LHCb-2007-117. CERN-LHCb-2007-117. Geneva: CERN, Aug. 2007. URL: http://cdsweb.cern.ch/record/1054088?ln=en.

[42] O Callot. *FastVelo, a fast and efficient pattern recognition package for the Velo*. Tech. rep. LHCb-PUB-2011-001. CERN-LHCb-PUB-2011-001. LHCb. Geneva: CERN, Jan. 2010. URL: http://cdsweb.cern.ch/record/1322644?ln=en.

[43] O. Callot and S. Hansmann-Menzemer. *The Forward Tracking Algorithm and Performance Studies*. Tech. rep. LHCb-2007-015. CERN-LHCb-2007-015. Geneva: CERN, May 2007. URL: http://cdsweb.cern.ch/record/1033584?ln=en.

[44] M Benayoun and O Callot. *The forward tracking, an optical model method*. Tech. rep. LHCb-2002-008. revised version number 1 submitted on 2002-02-22 17:19:02. Geneva: CERN, Feb. 2002. URL: http://cdsweb.cern.ch/record/684710?ln=en.

[45] M. Needham and J. Van Tilburg. *Performance of the track matching*. Tech. rep. LHCb-2007-020. CERN-LHCb-2007-020. Geneva: CERN, Mar. 2007. URL: http://cdsweb.cern.ch/record/1020304?ln=en.

[46] M Needham. *Performance of the Track Matching*. Tech. rep. LHCb-2007-129. CERN-LHCb-2007-129. Geneva: CERN, Oct. 2007. URL: http://cdsweb.cern.ch/record/1060807?ln=en.

[47] O Callot. *Downstream Pattern Recognition*. Tech. rep. LHCb-2007-026. CERN-LHCb-2007-026. Geneva: CERN, Mar. 2007. URL: http://cdsweb.cern.ch/record/1025827?ln=en.

[48] S Stahl and S Hansmann-Menzemer. *Update on Downstream Tracking*. Tech. rep. LHCb-INT-2009-032. CERN-LHCb-INT-2009-032. Geneva: CERN, Dec. 2009. URL: http://cdsweb.cern.ch/record/1225910?ln=en.

[49] O Callot, M Kucharczyk and M Witek. *VELO-TT track reconstruction*. Tech. rep. LHCb-2007-010. CERN-LHCb-2007-010. Geneva: CERN, Apr. 2007. URL: http://cdsweb.cern.ch/record/1027834?ln=en.

[50] E Rodrigues. *Dealing with clones in the tracking*. Tech. rep. LHCb-2006-057. CERN-LHCb-2006-057. Geneva: CERN, Nov. 2006. URL: http://cdsweb.cern.ch/record/1000723?ln=en.

[51] A Perieanu. *A Fast Algorithm to Identify and Remove Clone Tracks*. Tech. rep. LHCb-2008-020. CERN-LHCb-2008-020. Geneva: CERN, May 2008. URL: http://cdsweb.cern.ch/record/1102942?ln=en.

[52] R. W. Forty and M. Needham. *Standalone Track Reconstruction in the T-stations*. Tech. rep. LHCb-2007-022. CERN-LHCb-2007-022. Geneva: CERN, Mar. 2007. URL: http://cdsweb.cern.ch/record/1024496?ln=en.

[53] R W Forty and M Needham. *Updated Performance of the T-Seeding*. Tech. rep. LHCb-2007-023. CERN-LHCb-2007-023. Geneva: CERN, Mar. 2007. URL: http://cdsweb.cern.ch/record/1023581?ln=en.

[54] M Needham. *The Tsa Reconstruction Framework*. Tech. rep. LHCb-2007-037. CERN-LHCb-2007-037. Geneva: CERN, Mar. 2007. URL: http://cdsweb.cern.ch/record/1025922?ln=en.

[55] C Adrover et al. "Search for the rare decays $B_{(s)}^0 \to \mu^+\mu^-$ with the LHCb experiment". In: LHCb-ANA-2011-007 ; CERN-LHCb-ANA-2011-007 (Mar. 2011). URL: http://cdsweb.cern.ch/record/1309568?ln=en.

[56] C Jones et al. "Data handling tools for tracking-interfaces & implementation". Talk at the LHCb tracking workshop, 30-August 2007. Aug. 2007. URL: http://indico.cern.ch/subContributionDisplay.py?subContId=0&contribId=14&confId=19148.

[57] I. Abt et al. *Cellular automaton and Kalman filter based track search in the HERA-B pattern tracker*. Tech. rep. 2002, pp. 546–558.

[58] J Albrecht et al. *Commissioning and Performance of the LHCb HLT1 muon trigger*. Tech. rep. LHCb-PUB-2011-006. CERN-LHCb-PUB-2011-006. LHCb-INT-2010-042. Geneva: CERN, Feb. 2011. URL: http://cdsweb.cern.ch/record/1331489.

[59] J Albrecht and M Schiller. *Fast T-Station Seeding for HLT1*. Tech. rep. LHCb-2009-012. CERN-LHCb-2009-012. Geneva: CERN, Jan. 2009. URL: http://cdsweb.cern.ch/record/1157939?ln=en.

[60] J Albrecht, M Needham and H Terrier. *L0 Confirmation with fast, Tsa based tracking in the T-stations*. Tech. rep. LHCb-2007-118. CERN-LHCb-2007-118. revised version submitted on 2007-10-29 16:30:33. Geneva: CERN, Oct. 2007. URL: http://cdsweb.cern.ch/record/1064479?ln=en.

[61] S Blusk. *LHCb Twiki: General Misalignment Samples*. URL: https://twiki.cern.ch/twiki/bin/view/LHCb/AlignmentSamples.

[62] D. Drijard et al. "Neutral strange particle production in proton-proton collisions at $\sqrt{s} = 63$ GeV". In: *Zeitschrift für Physik C Particles and Fields* 12 (3 1982). 10.1007/BF01558259, pp. 217–224. ISSN: 0170-9739. URL: http://dx.doi.org/10.1007/BF01558259.

[63] G. J. Alner et al. "Kaon production in reactions at a centre-of-mass energy of 540 GeV". In: *Nuclear Physics B* 258.C (1985), pp. 505–539. ISSN: 0550-3213. DOI: DOI:10.1016/0550-3213(85)90624-8. URL: http://dx.doi.org/10.1016/0550-3213(85)90624-8.

[64] R. E. Ansorge et al. "Kaon production in $p\bar{p}$ interactions at c.m. energies from 200 to 900 GeV". In: *Zeitschrift für Physik C Particles and Fields* 41 (2 1988). 10.1007/BF01566915, pp. 179–190. ISSN: 0170-9739. URL: http://dx.doi.org/10.1007/BF01566915.

[65] F. Abe et al. "K^0_S production in $p\bar{b}$ interactions at $\sqrt{s} = 630$ *and* 1800 GeV". In: *Phys. Rev. D* 40.112 (Dec. 1989), pp. 3791–3794. DOI: 10.1103/PhysRevD.40.3791.

[66] G. Bocquet et al. "Inclusive production of strange particles in collisions at with UA1". In: *Physics Letters B* 366.1-4 (1996), pp. 441–446. ISSN: 0370-2693. DOI: DOI:10.1016/0370-2693(95)01436-5. URL: http://dx.doi.org/10.1016/0370-2693(95)01436-5.

[67] D. Acosta et al. "K^0_S and Λ^0 production studies in $p\bar{p}$ collisions at $\sqrt{s} = 1800$ and 630 GeV". In: *Phys. Rev. D* 72.5 (2005), p. 052001. DOI: 10.1103/PhysRevD.72.052001.

[68] B. I. Abelev et al. "Strange particle production in $p + p$ collisions at $\sqrt{s} = 200$ GeV". In: *Phys. Rev. C* 75.6 (2007), p. 064901. DOI: 10.1103/PhysRevC.75.064901.

[69] M Knecht et al. "Measurement of V^0-particle production at LHCb with 2009 data". In: LHCb-ANA-2010-002 (Sept. 2010). URL: http://cdsweb.cern.ch/record/1295601?ln=en.

[70] S Hansmann-Menzemer et al. "Measurements of yield of K_s reconstructed with downstream tracks in 2009 data". In: (2010). URL: http://cdsweb.cern.ch/record/1295568?ln=en.

[71] F. James and M. Roos. “Minuit - a system for function minimization and analysis of the parameter errors and correlations”. In: *Computer Physics Communications* 10.6 (1975), pp. 343–367. ISSN: 0010-4655. DOI: DOI:10.1016/0010-4655(75)90039-9. URL: http://www.sciencedirect.com/science/article/B6TJ5-46FPXJ7-2C/2/58a665cfee07e122711a859fef69c955.

[72] N Zwahlen. *HLT Hadronic L0 Confirmation*. Tech. rep. LHCb-2006-040. CERN-LHCb-2006-040. Geneva: CERN, July 2006. URL: http://cdsweb.cern.ch/record/970989?ln=en.

[73] B Storaci et al. “Tracking Efficiency”. Talk at the LHCb tracking workshop, 4-June 2010. Aug. 2010. URL: http://indico.cern.ch/contributionDisplay.py?contribId=9&materialId=slides&confId=93325.

[74] Peter Zeiler Skands. *The Perugia Tunes*. Tech. rep. arXiv:1005.3457. MCNET-10-08. CERN-PH-TH-2010-113. Comments: 33 pages. May 2010. URL: http://cdsweb.cern.ch/record/1266776.

[75] R. Brun and F. Rademakers. “ROOT — An Object Oriented Data Analysis Framework”. In: *Nucl. Instr. and Meth.* A389 (1997). See also http://root.cern.ch/, pp. 81–86. URL: http://root.cern.ch.