Track and vertex reconstruction in the ATLAS inner detector

Limper, M.

Chapter 6

Vertex fitting with kinematic constraints

This chapter introduces the ATLAS track and vertex reconstruction using the measurements provided by the inner detector. Track reconstruction aims to accurately reconstruct the trajectories of charged particles. Reconstruction of the common intersection point of a set of particle trajectories helps to identify the decay point, or vertex, of unstable particles.

In order to improve the resolution of the reconstructed decay vertices, information about the physical laws governing a particle decay can be exploited by adding kinematic constraints to the vertex fit. A method for kinematically constraining a vertex fit was developed and will be described in detail in this chapter.

One example of a kinematic constraint in a vertex fit is requiring the invariant mass of the decay-products to be equal to the mass of the decaying particle. This constraint can be used to improve the resolution of the momentum of reconstructed muons from the decay of J/ψ-particle as will be presented in this chapter.

6.1 Athena framework and applications

A common software framework is needed for the large set of software applications used for the reconstruction and simulation of events in the ATLAS detector. The ATLAS software is implemented in the Athena [51] framework. This framework takes care of the order of running the requested algorithms, and it offers various common services used by the different algorithms, such as access to information about the detector geometry, magnetic field and calibration constants.

The software is designed to reconstruct data taken by the ATLAS detector and also to produce simulated events of proton-proton collisions in ATLAS.

Samples of simulated events are of vital importance for testing the reconstruction algorithms and preparing data analysis before the start of the LHC. The steps to simulate and reconstruct events are schematically:

- **Event generation:** The first step in the simulation chain is the simulation of the proton-proton collision itself. The physics processes in high-energy collisions are described in event generators, such as Pythia [5], that simulate the particles and their momentum four-vectors as produced in a collision .

- **Detector simulation:** The second step is to simulate the interactions of the generated particles with the material in the detector. This task is performed by the GEANT4 [52] program which uses a detailed description of the ATLAS geometry and the material distribution in the detector. The GEANT4 algorithms transport the particles through the magnetic field and simulates material interactions such as multiple scattering, energy loss, and photon conversions. The decay of unstable particles is also simulated at this stage.

- **Detector response:** At the next stage, the response of the detector including electronics is simulated. This step is also called digitization. If a particle hits a detection volume (like a silicon sensor), a 'hit' is registered in the simulated detector response. The digitization consists of simulating the response of the detector to the energy deposits in these hits. Both the response of the detector and of the electronics are simulated in a fair amount of detail. The output format of the detector response simulation is equivalent to the real data after the bytestream conversion and mapping[1] took place.

- **Reconstruction:** Starting from the detector response data, which can be real or simulated, various algorithms are used to reconstruct the event. This includes algorithms that perform pattern recognition, track fitting, vertex determination, energy measurement and particle identification. On simulated data, the reconstructed objects can be matched and compared to the simulated input to validate the performance of the ATLAS reconstruction software.

All the calibration data required by the event reconstruction is stored in the ATLAS conditions database where it can be accessed by the different Athena software applications. Calibration constants are used to convert various raw detector data in position or energy measurements.

[1] Assigning the output of the read-out channels to the different detector elements.

6.2 Track reconstruction

A large number of charged particles will be produced in the proton-proton collision of the LHC, resulting in a large number of hits in the ATLAS inner detector. The track reconstruction software has to distinguish the hits from the different charged particles and determine a trajectory that best matches the measurements.

6.2.1 Track finding strategy

Track reconstruction in the inner detector uses the measurements of the pixel, SCT and TRT detectors. Several strategies for track finding in the ATLAS inner detector are available [53][54][55]. The main track reconstruction strategy is the "inside-out" strategy, which starts by finding a track candidate in the pixel and SCT detectors and then extends the trajectories of successfully fitted tracks to the TRT to reconstruct a full inner detector track.

Figure 6.1 shows the number of measurements made along a trajectory in the inner detector for tracks originating from the interaction point as a function of $|\eta|$ as found with the inside-out track strategy. The tracks have on average 3 pixel hits and 8 hits from the silicon strip detectors close to the interaction point. The tracks also have an average of 30 TRT-hits, at larger distances from the interaction point, that provide an accurate measure of the curvature of the track.

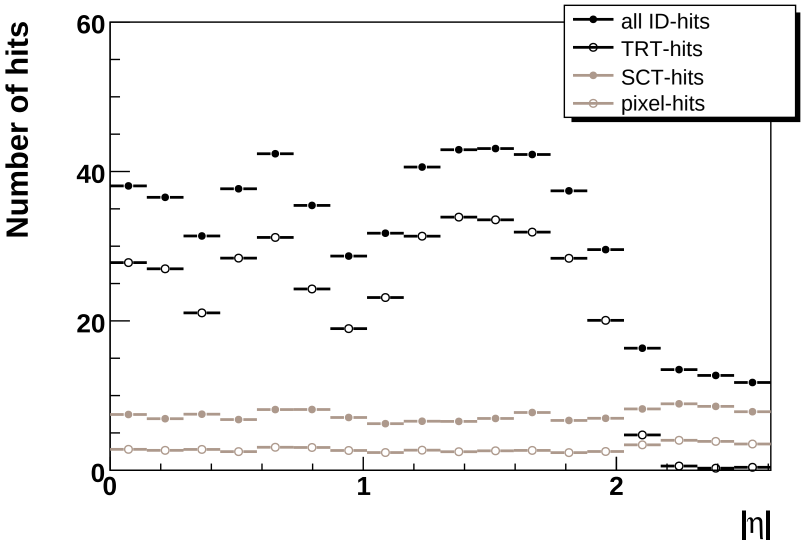

Figure 6.1: Number of hits by detector type used in reconstructed tracks in the inner detector as a function of $|\eta|$.

The main steps of the inside-out track strategy are as follows:

- **Pattern recognition**: The measurements in the pixel and SCT detectors form *silicon space-points*. Each silicon space-point represents a 3D precision point which is formed either from a cluster of silicon pixels or from the intersection of the front and backside silicon strips in an SCT module. The pattern recognition starts by finding track seeds that are formed of all three silicon space-points that are compatible with a minimum p_T cut of 500 MeV.
  The track seeds provide already enough directional information to associate additional silicon hits to the track segment by using a fast Kalman filter [56] to follow the trajectory. At this stage information from the separate silicon pixel and strip hits is used, rather than the silicon space-points. If the track seeds can be extended to contain a minimum of 7 silicon hits, a track candidate is formed.
  Depending on the underlying physics event, a large amount of track seeds can be found in each event. Figure 6.2 shows an example of the track seeds found in a simulated $t\bar{t}$ event in ATLAS. Many of the proposed track seeds represent fake track segments and can not be extended to form track candidates.

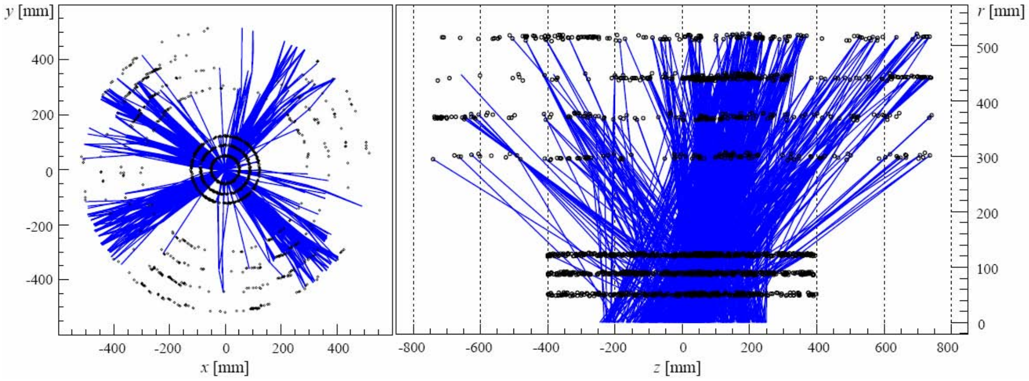

Figure 6.2: Track seeds consisting of three silicon space-points in the ATLAS inner detector barrel as found in a simulated $t\bar{t}$ event [53].

- **Ambiguity solving**: Many of the initial track candidates share hits, are incomplete or are fake tracks. The ambiguity solving process assigns a 'score' to each track candidate to indicate the likelihood that the track candidate originated from a real particle trajectory. In general, each hit associated with the track leads to higher score, in order to favor fully reconstructed tracks over smaller track segments. On the other hand, 'holes', defined as the passage of the track through a detector element without producing a hit, will lower the track score. Hits that are shared between tracks are assigned to the track with the highest score and removed from the other track. Track candidates that after the ambiguity solving process have less than the minimum number of 7 hits are neglected for further processing.

- **TRT track extension**: The fitted silicon track parameters at the outermost silicon measurement, closest to the TRT, are used to define a road through the TRT, and any drift radius measurements that are within 10 mm of this road are assigned to the silicon track.
  The track is re-fitted with the addition of the TRT measurements and compared with the original silicon-only track using the track scoring mechanism. In the case that the track score of the silicon track was higher than the extended version, the silicon track is kept.

This track reconstruction sequence is complemented by an 'outside-in' strategy. The outside-in strategy starts from unassigned TRT segments and looks for matching hits in the pixel and SCT detectors. This type of track reconstruction is mainly aimed at late decays of neutral particles and photon conversions to e^+e^- pairs, and is also able to recover the remaining trajectory after a catastrophic energy loss.

Figure 6.3 shows the track reconstruction efficiency as a function of $|\eta|$ for isolated muons, pions and electrons with p_T=5 GeV, as was obtained using simulated events [16]. In addition to multiple scattering, pions are affected by hadronic interactions in the inner detector material, while electrons are subject to even larger reconstruction inefficiencies due to bremsstrahlung. As a result, the efficiency curves as a function of $|\eta|$ for pions and electrons reflect the increase of material in the inner detector in the forward direction.

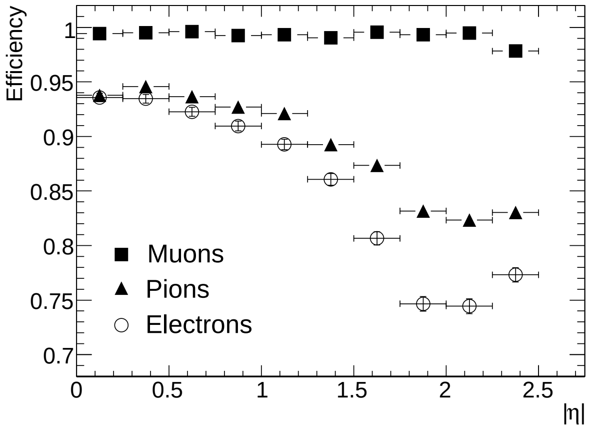

Figure 6.3: Track reconstruction efficiencies in the inner detector as a function of $|\eta|$ for muons, pions and electrons with p_T= 5 GeV [16].

6.2.2 Track parametrization

A track in ATLAS is parametrized at the point of closest approach with the global Z-axis using five *perigee parameters* (as illustrated[2] in figure 6.4):

- $\mathbf{\frac{q}{p}}$: the charge of the particle divided by the momentum
- $\boldsymbol{\phi_0}$: the angle with the x-axis in the X-Y plane at the perigee point
- $\boldsymbol{\theta_0}$: the angle with the z-axis in the R-Z plane
- $\mathbf{d_0}$: the signed distance to the z-axis. The sign of d_0 is positive, when $\phi - \phi_0 = \frac{\pi}{2} \mod (2\pi)$, where ϕ denotes the angle to the perigee position in the x-y plane, as shown in figure 6.4.
- $\mathbf{z_0}$: the z-coordinate of the track at the point of closest approach to the global Z-axis.

These track parameters can then be propagated to different positions within the ATLAS detector by using the field map of the ATLAS magnetic field.

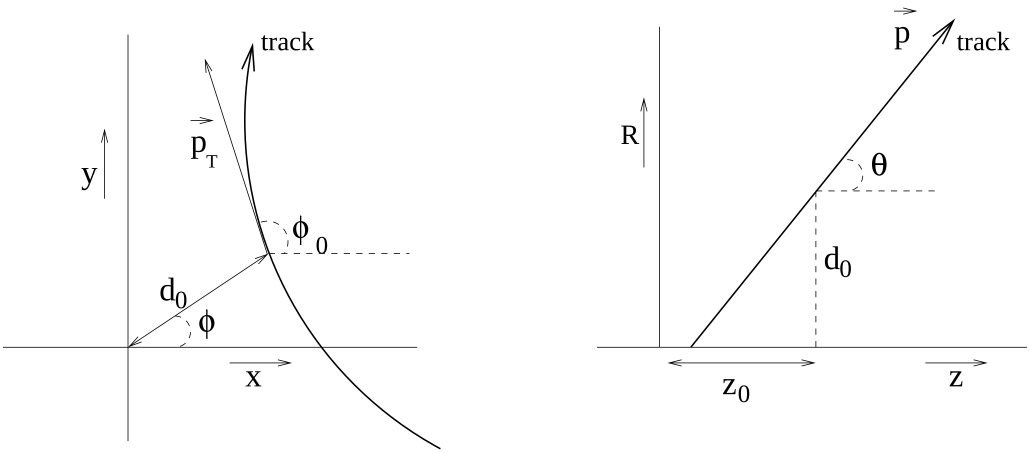

Figure 6.4: Illustration of the perigee parameters of a track in the transverse plane (left) and RZ-plane (right), as defined in the global ATLAS tracking frame.

Apart from the perigee parameters, a reconstructed track object can also contain a set of track parameters at various other positions along the trajectory. These extra track parameters can be used to describe interactions of the particle trajectory with material in the inner detector, which can change the momentum and direction of the track at the scattering centers along the track.

[2] The illustration in figure 6.4 approximates the helix in the RZ-plane as a straight line close to the interaction point.

6.2.3 Track fitting

The pattern recognition provides a list of measurements along a particle trajectory, as well as a first-estimate of the track parameters of the track. The task of the track fitting method is to compute the best possible estimate of the track parameters. The two track fitting techniques which are widely using in high energy physics, the *global least-squares fit* and the *Kalman filter* are both implemented in the ATLAS software framework. In order for the track reconstruction to run with a reasonable CPU time, both track fitters use "material maps" instead of the full ATLAS geometry, to correct the track prediction for multiple scattering and energy loss effects. The material maps in the ATLAS tracking geometry [57] represent the material in the form of layers, each with assigned thickness and radiation length. The track fitting methods also rely on the same *propagator* tool which calculates track parameters and covariance matrices at a destination surface while taking into account the bending of the charged particle in the magnetic field. In ATLAS, the presence of an inhomogeneous magnetic field means that a numerical approximation has to be used. The default propagator in ATLAS uses the Runga-Kutta method [58], which follows the path of particle by taking one step at a time, while optimizing the step-size to reach the required propagation precision. In ATLAS the propagation errors are shown to be within a few microns over distances of up to 10 meters, which is negligible compared to the internal detector resolution.

Global least-squares fit

The global least-squares fit [59] obtains the best estimate for the track parameters by directly minimizing the following χ^2 function:

$$\chi^2 = \sum_{i=1}^{N_{meas}} \left(\frac{\Delta_i}{\sigma_i}\right)^2 + \sum_{i=1}^{N_{layer}} \left(\frac{\theta_{scat}^2}{\sigma_{scat}^2} + \frac{(\sin\theta_{loc})^2 \phi_{scat}^2}{\sigma_{scat}^2}\right) \tag{6.1}$$

where the first term is the sum of all the detector hits of the residuals, Δ_i, divided by the by the error on the measurement σ_i. The residuals are given by the distance between the measured and the predicted value of the hit position along the track.

The second term in equation 6.1 is a sum over the number of material layers along the track, N_{layer}. Here the possible change of track direction at a scattering center is implemented by adding two extra fit-parameters, ϕ_{scat} and θ_{scat}. The scattering angles are added to the χ^2 as measurements with zero as initial value and with an error σ_{scat} that is calculated from the traversed material thickness in terms of radiation lengths, see equation 3.1 in section 3.1. The fit also reduces the momentum of the track after traversing a material layer by an amount equal to the average energy loss corresponding to the traversed material.

The minimization of the χ^2 in the global least-squares fit involves the inversion of a large matrix [60] with a size of N × M, with N the number of fit-parameters and M the number of measurements plus the number of scattering angles used in the fit. The CPU time required for the inversion of the matrix scales with M^2 so for fits with a large number of measurement and scattering planes this inversion can cost a significant amount of CPU time.

Kalman filter

The Kalman filtering technique [56] is a widely used method for track fitting. This techniques determines the optimal track parameters through an iterative process of stepping from one measurement to the next. At each measurement surface the local track parameters are described by the track state vector and the covariance matrix of the state vector.

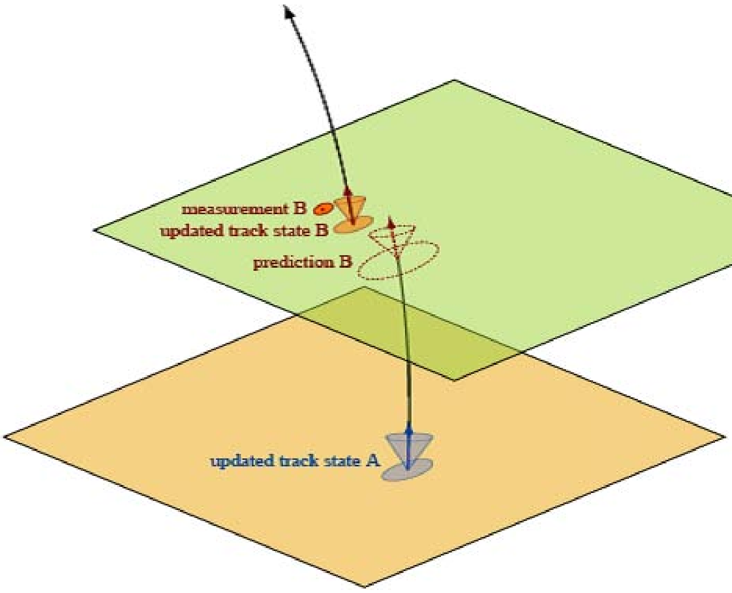

Figure 6.5: A single filtering step in the Kalman method. The different sizes of the ellipses and the cones indicate the change of the errors of the track parameters during a filtering step.

In each filtering step (illustrated by figure 6.5) the track parameters are extrapolated to the next measurement surface to calculate a prediction of the track state at the next surface. Extrapolation involves the propagation through the magnetic field of the track state vector and its covariance matrix, as well as updating the errors when the track passes through a material layer.

The predicted value together with the actual measurement are used to update the track state at the next measurement surface. The filtering process is bi-directional with the aim to get precise parameters both at the perigee position and at the outer end of the tracks and to have a balanced sensitivity to outliers[3]. One advantage of the Kalman filter is that all calculations are made locally (i.e. one measurement at a time) and the algorithm remains fast, in terms of CPU time, even in the presence of many measurements and multiple scattering planes, since no large matrix inversion is required.

Track fitting performance

The different track fit methods should give identical results, as both are attempting to find to optimum track trajectory by minimizing the hit residuals of the tracks. Figure 6.6 shows the resolutions obtained with the two fitting methods on the impact parameters and the transverse momentum, as function of the transverse momentum, obtained by fitting simulated single muon tracks in the inner detector. It can be seen that the results in figure 6.6 are indeed identical for the two fit methods. The figures show that the resolution on the impact parameter improves for increasing values of p_T, while $\Delta p_T/p_T$ improves at lower values of p_T.

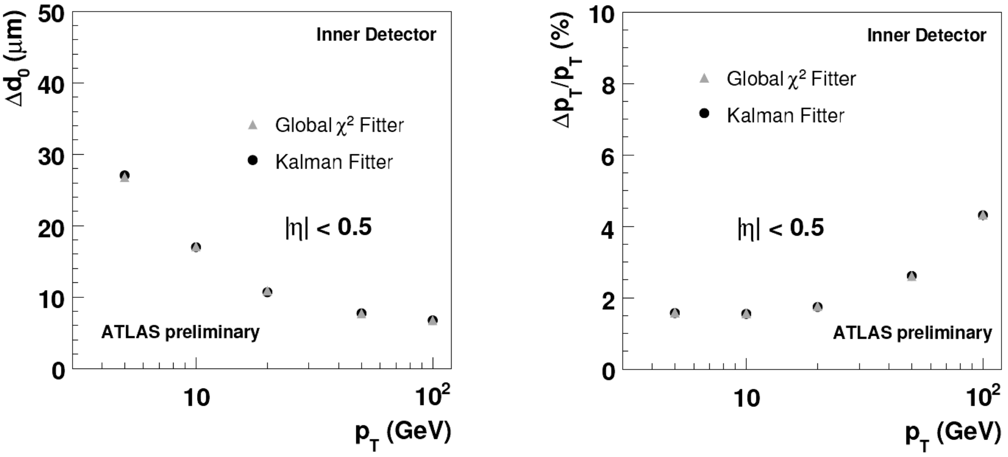

Figure 6.6: Obtained resolutions on the impact parameter (left) and transverse momentum (right) for Kalman filter and global least-squares fit methods for simulation single muon tracks in the inner detector [60].

The time required to fit a track depends on the number of hits and the number of material layers along the trajectory. On average, each inner detector track contains 35 hits and has 15 material layers along the trajectory. In the case of the global least-squares fit this leads to a total of 35 fit parameters (5 from the perigee, 2 × 15 from the scattering angles). Using fast matrix algorithms the global least squares method requires about twice as much CPU

[3] Hits that are incorrectly associated to the track and typically give a large χ^2 contribution.

time as the Kalman filter method. Though the Kalman filter technique is faster than the global least-squares fit, the global least-squares fit has the advantage of the explicit calculation of the scattering angles, which can be used to study the material distributions along the tracks.

At the event reconstruction stage, one of the two track fitters can be selected. At the time of writing the global χ^2 fitter was the default track fitter used for track reconstruction in the inner detector.

6.2.4 Muon identification and combined reconstruction

This thesis focuses on the reconstruction of $J/\psi \to \mu^+\mu^-$ events, which requires tracks that are identified as muons. To identify a track in the inner detector as originating from a muon, the track has to be matched with a track or track segment in the muon spectrometer. The ATLAS reconstruction software provides several strategies to identify and reconstruct a muon track.

The most commonly used packages for combined tracking are *STACO* and *MUID*. Both algorithms identify muons by pairing tracks reconstructed in the muon spectrometer with tracks reconstructed in the inner detector. MUID produces the combined track from a re-fit of the inner detector track with the addition of the measurements from muon spectrometer track, while STACO determines the combined track parameters from the weighted average of the two sets of track parameters.

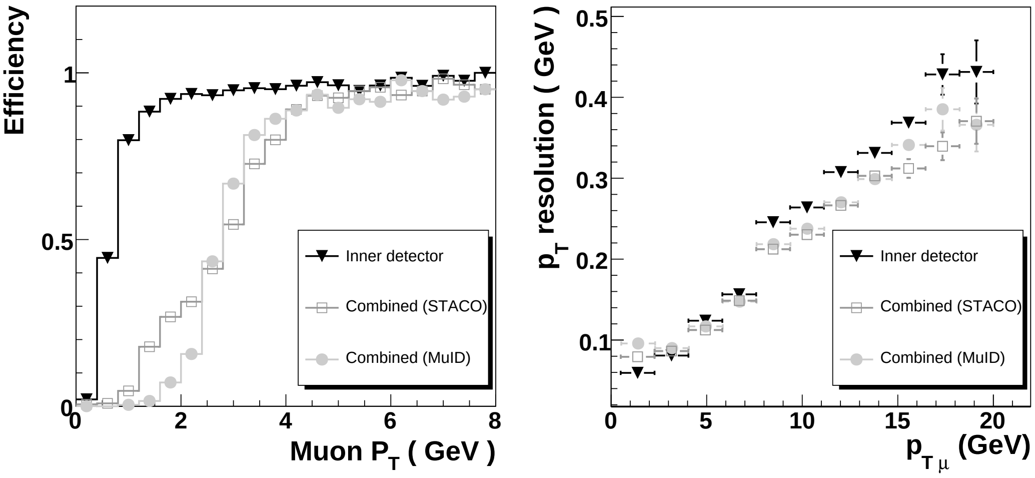

Figure 6.7: Efficiency (left) and p_T resolution (right) of muons in simulated $J/\psi \to \mu^+\mu^-$ events reconstructed by the combined muon algorithms.

Figure 6.7 shows the efficiency and p_T resolution of tracks reconstructed by the inner detector track reconstruction and the two combined reconstruction strategies. The plots were obtained using a sample of simulated $J/\psi \to \mu^+\mu^-$

events, in which one of the two muons was required to have a p_T of at least 9 GeV. The plots shows that muon identification only becomes efficient (>90%) for tracks with a transverse momentum greater than 5 GeV, as muon tracks with low transverse momenta are stopped in the calorimeter. For high momentum muons, using the information from the combined measurement in the inner detector and muon spectrometer is expected to improve the p_T resolution of the reconstructed muon. However, this effect is not very significant for $J/\psi \to \mu^+\mu^-$ events where the muons have relatively low transverse momenta.

6.3 Vertex reconstruction

Finding the common intersection points between sets of reconstructed tracks allows to identify the proton-proton interaction point as well the decay vertices of unstable particles produced in the collision. Identifying and reconstructing different vertices within one event relies on precise track reconstruction and helps to study various physics processes.

Most of the reconstructed tracks from a proton-proton collision in ATLAS will originate from the collision point, indicating the *primary vertex* of that collision. The beam spot defines in which region the proton-proton collisions take place and can be described by a Gaussian with the standard deviation of approximately 5 cm in the direction of the beam and approximately 15 μm in the perpendicular plane. Within this region, especially for high luminosity runs of the LHC, more than one proton-proton interaction per bunch-crossing can take place.

Some particles, such as b-hadrons, can decay at measurable distance from the primary vertex, and tracks originating from such decays can be used to identify *secondary vertices.* A primary vertex often is formed from the intersection of 20 or more reconstructed tracks while secondary vertices are often fitted with just 2 or 3 trajectories. Tracks from secondary vertices can often be identified by a significantly high value of the impact parameter (d_0) relative to the primary interaction.

The work described in this chapter used simulated $b\bar{b} \to X + (J/\psi \to \mu^+\mu^-)$ events to validate the performance of a new method for vertex fitting with kinematic constraints. In this type of event, the secondary vertex of the b-decay can be identified from the intersection point of two reconstructed muon tracks with an invariant mass around the J/ψ value.

6.3.1 Vertex fitting

The vertex fit has to determine a common intersection point between the trajectories, given the track parameters and their covariance matrices as determined from the track fit. Several strategies for vertex fitting are currently implemented in the ATLAS reconstruction software [61].

One example of a vertex fit method implemented in ATLAS is the *Billoir vertex fitting method* that follows the scheme presented in [62]. This method finds the vertex position $\vec{x}$ and momentum vector of the tracks $\vec{p_i}$, that minimizes the following χ^2:

$$\chi^2 = \sum_{i=1}^{N} (\vec{\eta}(\vec{x},\vec{p_i}) - \vec{\eta}_{0i})^T V_{\eta 0i}^{-1} (\vec{\eta}(\vec{x},\vec{p_i}) - \vec{\eta}_{0i}), \tag{6.2}$$

where $\vec{\eta}_{0i}$ is the vector of track parameters from the original track fit, $\vec{\eta}(\vec{x},\vec{p_i})$ is the prediction of the track parameters at the perigee and $V_{\eta 0i}$ is the covariance matrix of the track parameters as obtained from the original track fit. In the Billoir method the vertex position is estimated by approximating the equations of motion of a charged particle with a first order Taylor expansion to obtain an analytical solution for the χ^2-minimization. Apart from reconstructing the vertex position and its covariance matrix, the Billoir method also refits the parameters of the incident tracks with the knowledge of the vertex.

The Billoir method for vertex fitting is widely used and the approach is similar to the method for vertex fitting with kinematic constraints that is presented in this thesis. For this reason the Billoir method for vertex fitting will be used as reference in comparing results of the kinematic vertex fit in section 6.5.

6.4 Vertex fitting with kinematic constraints

The use of kinematic constraints in the vertex fit can help to improve the resolution of the fitted parameters and to test the hypothesis of a certain interaction type.

Constrained vertex fitting is implemented in several packages in the ATLAS software framework. As part of the work done here, a tool for constrained vertex fitting [63] has been developed, based on the method of applying kinematic constraints using Lagrange multipliers in the Cartesian tracking frame [64]. At the time of writing, the only implemented kinematic constraints for this tool are the vertex constraint and the mass constraint but the framework of the new method facilitates the implementation of other types of constraints in the future.

6.4.1 Track parameters in the Cartesian frame

The Cartesian coordinates are the momenta and the energy of a track and the position on the track where the momenta are defined:

$$\vec{\alpha} = (p_x, p_y, p_z, E, x, y, z) \tag{6.3}$$

The advantage of using Cartesian coordinates is that they provide a track representation which uses physically meaningful quantities for reconstruction of particle decays and allow the track to be parametrized at the position representing its decay vertex, rather than a global reference point. The use of the Cartesian representation therefore simplifies the formulation of most kinematic constraints.

The Cartesian coordinates are related to the standard perigee track coordinates: $\vec{\eta} = (d_0, z_0, \phi_0, \theta, \frac{q}{p})$ through the following:

$$\begin{aligned}
p_x &= |\frac{q}{p}|^{-1}\cos\phi\sin\theta \\
p_y &= |\frac{q}{p}|^{-1}\sin\phi\sin\theta \\
p_z &= |\frac{q}{p}|^{-1}\cos\theta \\
E &= \sqrt{|\frac{q}{p}|^{-2} + m^2} \\
x &= x_{\text{ref}} - d_0\sin\phi \\
y &= y_{\text{ref}} + d_0\cos\phi \\
z &= z_{\text{ref}} + z_0 \qquad (6.4)
\end{aligned}$$

It can be noted that in the Cartesian frame the track is represented by a set of 7 parameters instead of 5. The extra parameters make the Cartesian frame an over-constrained parametrization, which is reflected by the correlations in the corresponding 7 × 7 covariance matrix, V_α.

The covariance matrix V_α, that represents the errors and correlations of the track parameters in the Cartesian frame, is obtained by transforming the original 5 × 5 matrix, V_η, with the Jacobian matrix $\mathbf{J}$:

$$V_\alpha = \mathbf{J}V_\eta\mathbf{J}^T \qquad (6.5)$$

The Jacobian matrix $\mathbf{J}$ for this coordinate change can be computed by taking the differentials of the equations in 6.4 as is explained in appendix B.

6.4.2 Lagrange multiplier technique for adding constraints

Each kinematic constraint is expressed as a vector of equations $\mathbf{H}(\vec{\alpha}_i, \vec{x})$, where $\vec{\alpha}_i$ is a vector of Cartesian parameters of all tracks participating in the fit and $\vec{x}$ is the vertex position. The constraint equations are formulated such that when $\mathbf{H}(\vec{\alpha}_i, \vec{x}) = 0$, the constraint is satisfied.

The constraint equations are linearized by using the first order Taylor expansion around a convenient point $(\vec{\alpha}_A;\vec{x}_A)$:

$$\begin{aligned} 0 &= \left[\frac{\partial\mathbf{H}(\vec{\alpha}_A,\vec{x}_A)}{\partial\vec{\alpha}}\right](\vec{\alpha}-\vec{\alpha}_A)+\left[\frac{\partial\mathbf{H}(\vec{\alpha}_A,\vec{x}_A)}{\partial\vec{x}}\right](\vec{x}-\vec{x}_A)+\mathbf{H}(\vec{\alpha}_A,\vec{x}_A) \\ &\equiv \mathbf{D}(\vec{\alpha}-\vec{\alpha}_A)+\mathbf{E}(\vec{x}-\vec{x}_A)+\mathbf{d}, \end{aligned} \tag{6.6}$$

where $\mathbf{d}$ is the vector of values of the constraint equations at the expansion point $(\vec{\alpha}_A;\vec{x}_A)$, $\mathbf{D}$ is the matrix of partial derivatives of the constraint equations at the expansion point with respect to the track parameters and $\mathbf{E}$ is the matrix of the partial derivatives of the constraint equations at the expansion point with respect to the vertex coordinates.

The constrained vertex fit has to find the solution for the values of $\vec{\alpha}$ and $\vec{x}$ in equation 6.6, and at the same time minimize the χ^2 contribution of the original track fit. The method of Lagrange multipliers solves these problems by introducing an extra set of variables, $\vec{\lambda}$, so that the χ^2 function to be minimized is given by:

$$\chi^2=(\vec{\alpha}-\vec{\alpha}_0)^T V_{\alpha_0}^{-1}(\vec{\alpha}-\vec{\alpha}_0)+2\lambda^T(\mathbf{D}(\vec{\alpha}-\vec{\alpha}_A)+\mathbf{E}(\vec{x}-\vec{x}_A)+\mathbf{d}), \tag{6.7}$$

where the first term is the χ^2 contribution from the original track fits and the second term is the contribution from the constraints, with $\vec{\lambda}$ the vector of Lagrange multipliers.

The quantities $\vec{\alpha}_0$ and V_{α_0} in equation 6.7 are the old track parameters and the covariance matrix from the original track fit. The original error matrix $V_{\alpha 0}$ contains in block-diagonal form the covariance matrices of the Cartesian track parameters:

$$V_{\alpha 0}=\begin{pmatrix} V_{\alpha 1} & \emptyset & \cdots & \emptyset \\ \emptyset & V_{\alpha 2} & \cdots & \emptyset \\ \vdots & \vdots & \ddots & \vdots \\ \emptyset & \emptyset & \cdots & V_{\alpha N} \end{pmatrix}, \tag{6.8}$$

where the zero entries on the off-diagonal elements of $V_{\alpha 0}$ represent the lack of correlations between the track parameters of different tracks before the constraints are applied.

The values of track parameters $\vec{\alpha}$ and vertex position $\vec{x}$ that satisfy the given set of constraints can be found by minimizing the χ^2 in equation 6.7 with respect to $\vec{\alpha}$, $\vec{x}$ and $\vec{\lambda}$, resulting in the following equations:

$$\begin{aligned} V_{\vec{\alpha}_0}^{-1}(\vec{\alpha}-\vec{\alpha}_0)+\mathbf{D}^T\lambda &= 0, \\ \mathbf{E}^T\lambda &= 0, \\ \mathbf{D}(\vec{\alpha}-\vec{\alpha}_A)+\mathbf{E}(\vec{x}-\vec{x}_A)+\mathbf{d} &= 0, \end{aligned} \tag{6.9}$$

where the second equation is the same as equation 6.6 which shows that the solution of the χ^2 minimization satisfies the (linearized) constraints.

The solution for the updated track parameters $\vec{\alpha}$ and the vertex prediction $\vec{x}$ is given by [64]:

$$\begin{aligned} \vec{\alpha} &= \vec{\alpha}_0 - V_{\alpha_0}\mathbf{D}^T\lambda & (6.10)\\ \vec{x} &= \vec{x}_A - V_E E^T\lambda_0 & (6.11) \end{aligned}$$

where the vectors λ_0 and λ and the matrix V_E are given by:

$$\begin{aligned} \lambda_0 &= V_D(\mathbf{D}(\vec{\alpha}_0 - \vec{\alpha}_A) + d), \\ V_E &= (\mathbf{E}V_D\mathbf{E}^T)^{-1}, \\ \lambda &= \lambda_0 + V_D\mathbf{E}\vec{x}, & (6.12)\\ \text{where } V_D &= (\mathbf{D}V_{\alpha_0}\mathbf{D}^T)^{-1}. \end{aligned}$$

This solution was obtained using linearized constraint-equations at a chosen expansion point, and so the updated track parameters and vertex position represent only a first guess of the actual solution. The solution only becomes exact once the chosen expansion point is equal to the vertex position and updated track parameters that minimize the χ^2. For this reason, the vertex fit uses an iterative procedure that updates the expansion points $\vec{\alpha}_A; \vec{x}_A$ to be equal to the new values obtained by equation 6.11 and repeats the calculation of the **D**, **E** and **d** to find an improved solution. The iterative procedure continues until the vertex prediction for one iteration becomes equal to that of the previous iteration.

After the final iteration the covariance matrices of the vertex and updated track parameters are determined. The covariance matrix of the vertex position, V_x, is equal to V_E. The covariance matrix of the updated track parameters is given by:

$$\begin{aligned} V_\alpha &= V_{\alpha 0} - V_{\alpha 0}\mathbf{D}^T V_{\tilde{D}}\mathbf{D}V_{\alpha 0}, & (6.13)\\ \text{where } V_{\tilde{D}} &= V_D - V_D\mathbf{E}V_E\mathbf{E}^T V_D. \end{aligned}$$

The new covariance matrix V_α now contains the covariance matrices of the updated track parameters as well as the correlation between the track parameters of different tracks created by imposing the constraints.

6.4.3 Mass constraint

The equation to force a number of N trajectories to have a common invariant mass m_{constr} can be written as:

$$\mathbf{H}_0 = (\sum_{i=0}^{N} E_i)^2 - (\sum_{i=0}^{N} p_{xi})^2 - (\sum_{i=0}^{N} p_{yi})^2 - (\sum_{i=0}^{N} p_{zi})^2 - (m_{constr})^2 = 0, \quad (6.14)$$

where p_{xi}, p_{yi}, p_{zi} represent the momenta of the track at the decay vertex. These momenta are not the same as in the Cartesian coordinates of tracks, which were obtained from transforming the track parameters at the perigee position (see equation 6.4). In order to determine the value of p_{xi}, p_{yi}, p_{zi} at a the vertex position, the Cartesian coordinates have to be transported through the magnetic field. The magnetic field is approximated to be perfectly solenoidal so that the coordinates on any point along the track can be calculated as function of the path length s by:

$$\begin{aligned}
p_x &= p_{0x}\cos\rho s - p_{0y}\sin\rho s \\
p_y &= p_{0y}\cos\rho s + p_{0x}\sin\rho s \\
p_z &= p_{0z} \\
E &= E_0 \\
x &= x_0 + \frac{p_{0x}}{a}\sin\rho s - \frac{p_{0y}}{a}(1-\cos\rho s) \\
y &= y_0 + \frac{p_{0y}}{a}\sin\rho s + \frac{p_{0x}}{a}(1-\cos\rho s) \\
z &= z_0 + \frac{p_{0z}}{|\vec{p}|}s
\end{aligned} \tag{6.15}$$

where $(p_{x0}, p_{y0}, p_{z0}, E_0, x_0, y_0, z_0)$ are the original coordinates, $\rho = a/p$ and a is a bending factor which, for momentum units in MeV and coordinate units in mm, is given by $a = -0.2997 \cdot Bq$, where q is the charge of the particle and B the magnetic field strength in Tesla.
Eliminating s in equations 6.15, the momentum of the track can be defined in terms of its position:

$$\begin{aligned}
p_x &= p_{0x} - a(y - y_0) \\
p_y &= p_{0y} + a(x + x_0) \\
p_z &= p_{0z}
\end{aligned} \tag{6.16}$$

So the equation that forces a set of track parameters to have an invariant mass m_{constr} at a vertex $(x_{vtx}, y_{vtx}, z_{vtx})$ is written as:

$$\begin{aligned}
\mathbf{H}_0 &= \left(\sum_{i=0}^{N} E_i\right)^2 - \left(\sum_{i=0}^{N}(p_{xi} - a(y_{vtx} - y_i))\right)^2 - \left(\sum_{i=0}^{N}(p_{yi} + a(x_{vtx} - x_i))\right)^2 \\
&\quad - \left(\sum_{i=0}^{N} p_{zi}\right)^2 - (m_{constr})^2 = 0
\end{aligned} \tag{6.17}$$

Solving this constraint using the Lagrange multiplier technique requires the formulation of $\mathbf{D}$ and $\mathbf{d}$. In this case $\mathbf{d}$ is a one-dimensional vector containing the value of $\mathbf{H}_0(\vec{\alpha}_A)$ and $\mathbf{D}$ is a matrix of size $1 \times 7N$ that can be calculated from the derivatives of equation 6.17.

6.4.4 Vertex constraint

The vertex constraint consist of two constraint equations for each trajectory in the vertex fit. For a solenoidal magnetic field these equations are given by [64]:

$$\begin{aligned} H_{i+1} &= p_x \Delta y - p_y \Delta x - \frac{a}{2}(\Delta x^2 + \Delta y^2) = 0., \\ H_{i+2} &= \Delta z - \frac{p_z}{a} \sin^{-1} \left(a(p_x \Delta x + p_y \Delta y)/p_T^2 \right) = 0., \end{aligned} \tag{6.18}$$

where $\Delta x = x_{trk} - x_{vtx}$, and a is the bending factor as defined previously. The corresponding $\mathbf{E}$ and $\mathbf{D}$ matrices can be written for n tracks as:

$$\mathbf{E} = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_3 \end{pmatrix}, \ \mathbf{D} = \begin{pmatrix} D_1 & \emptyset & \dots & \emptyset \\ \emptyset & D_2 & \dots & \emptyset \\ \vdots & \ddots & \ddots & \vdots \\ \emptyset & \dots & \emptyset & D_n \end{pmatrix} \tag{6.19}$$

where $\mathbf{E}_i$ is a 2 × 3 matrix and $\mathbf{D}_i$ is a 2 × 7 matrix. The block diagonal form of $\mathbf{D}$ results from the vertex constraint equations of each track having no correlation with the track parameters of the other tracks since each track is separately constrained to pass through the vertex position. The matrix $\mathbf{E}_i$ can be determined from taking the derivatives of equations versus the vertex coordinates and the matrix $\mathbf{D}_i$ can be determined from taking the derivatives of equations versus the track parameters of the i-th track.

6.4.5 The *VertexKinematicFitter* software package

The method described in section 6.4 was implemented as an additional tool for vertex fitting in the ATLAS software framework. An important feature of the new *VertexKinematicFitter* software package is that it provides a general framework where any type of user-defined constraint can be added to a vertex fit, while other constrained vertex fitting methods available in ATLAS have always implemented specific constraints as part of a specific tool.

The inputs to the constrained vertex fitter are a set of *kinematic particle* objects, where each kinematic particle represents the track parameters and associated particles mass, together with a list of *kinematic constraints*. The output of the fit contains the vertex position and its covariance matrix as well as the updated track parameters defined at the vertex position.

The constraint equations and their derivatives are contained in the implementations of the different kinematic constraints, while the main method only holds the constraint equations for the vertex constraint. This makes the tool behave as a conventional χ^2 vertex fitter when no additional constraints are given.

6.5 Vertex fit with mass constraint results

In order to study the performance of the kinematic vertex fitting method, a sample of simulated $b\bar{b} \to X + (J/\psi \to \mu^+\mu^-)$ events is used. The simulated sample is based on events where one muon has $p_T > 6$ GeV and the other muon has $p_T > 4$ GeV, corresponding to one possible trigger scenario to be used in real ATLAS data-taking. For each event that had two inner detector tracks that were identified as muons by STACO (see 6.2.4) two types of vertex fits are made. One vertex fit uses the method described in section 6.4 to perform the fit while constraining the invariant mass to the exact value of the J/ψ. Another vertex fit uses the Billoir method for vertex fitting so that the performance of both fit methods can be compared.

Figure 6.8 shows the invariant mass of the reconstructed track parameters at the vertex position obtained with the Billoir method and the kinematic vertex fit with mass constraint. It can be seen that, as expected, the mass constrained vertex fit forces the invariant mass to be exactly equal to the value it was constrained to.

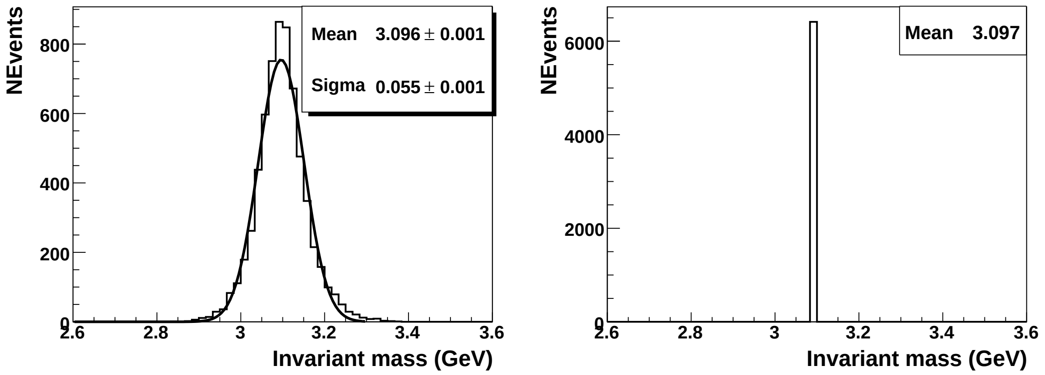

Figure 6.8: Invariant mass of the reconstructed muons at the vertex position as obtained with the Billoir method for vertex-fitting (left) and the kinematic fit with mass constraint (right) using a sample of simulated $b\bar{b} \to X + (J/\psi \to \mu^+\mu^-)$ events

6.5.1 Vertex fitting results

The method for vertex fitting with kinematic constraints implemented an iterative approach to find the best solution for the vertex prediction. Figure 6.9 shows the distance between the reconstructed and the true vertex as a function of the number of iterations of the fit. The initial value of the vertex prediction (corresponding to iteration=0) is always equal to $(0; 0; 0)$. It can be seen that for all the vertices in figure 6.9 the fit converged within 2 iterations.

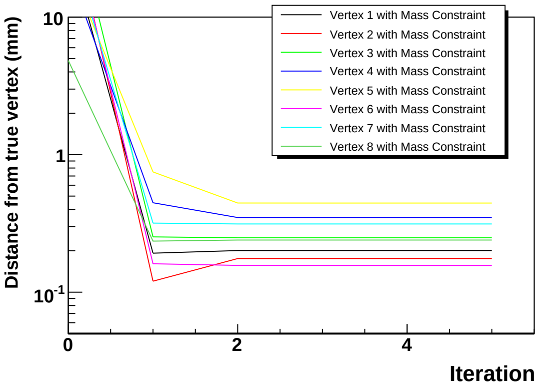

Figure 6.9: Distance between the reconstructed and the true vertex as a function of the number of iterations of the vertex fit with mass constraint, for a sample of 8 vertices from simulated $b\bar{b} \to X + (J/\psi \to \mu^+\mu^-)$ events.

The distance between the primary vertex and the secondary vertex reconstructed using the two muon tracks gives a measure of the decay-length of the b-hadron that produced the J/ψ. Figure 6.10 shows the signed transverse decay length, L_{XY}, as obtained using the secondary vertex reconstructed with the Billoir method and the mass constrained vertex fit. It can be seen that the two fitters obtain similar results for the transverse decay length.

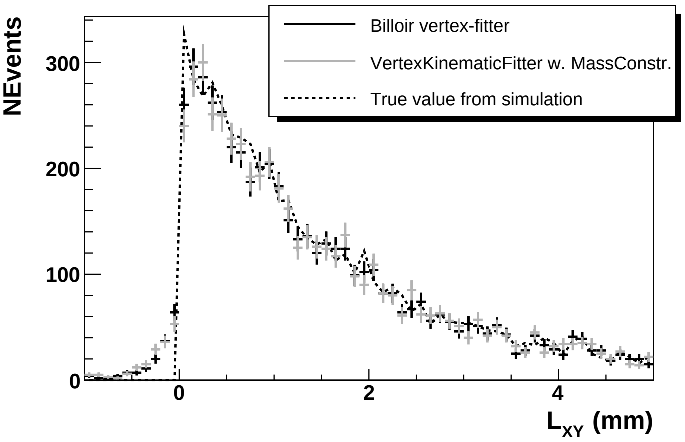

Figure 6.10: Signed transverse decay length as obtained from the secondary vertex of two muon tracks in simulated $b\bar{b} \to X + (J/\psi \to \mu^+\mu^-)$ events.

A measure of reliability of the vertex fit are the pull distributions of the vertex coordinates. The pull is defined as the difference between the reconstructed and the true vertex coordinate, divided by the estimated error on the coordinate. Figure 6.11 shows the pull distribution of the vertex coordinates as obtained with the mass constrained vertex fit. For comparison, figure 6.12

shows the same distributions obtained with the Billoir method. The Gaussians fitted to the pull distributions have, for both vertex fitting methods, a width that is significantly above one, indicating that the errors of the vertex fits were underestimated. It can also be seen that the vertex pull distribution of the mass constrained vertex fit are significantly wider as those for the Billoir method, implying that the errors were more severely underestimated by the mass constrained fit since the vertex resolution was the same for both fitters. This needs to be further investigated.

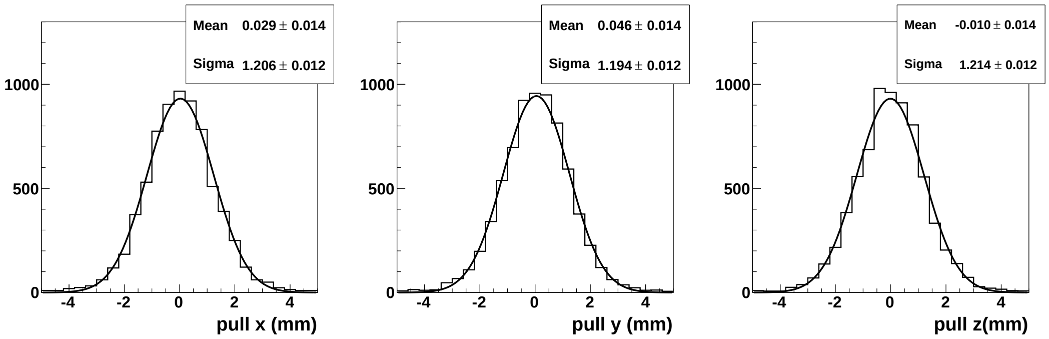

Figure 6.11: Pull distributions of the vertex coordinates for simulated $b\bar{b} \rightarrow X + (J/\psi \rightarrow \mu^+\mu^-)$ events, as obtained with the kinematic vertex-fit with mass constraint. All the distributions are fitted with a single Gaussian.

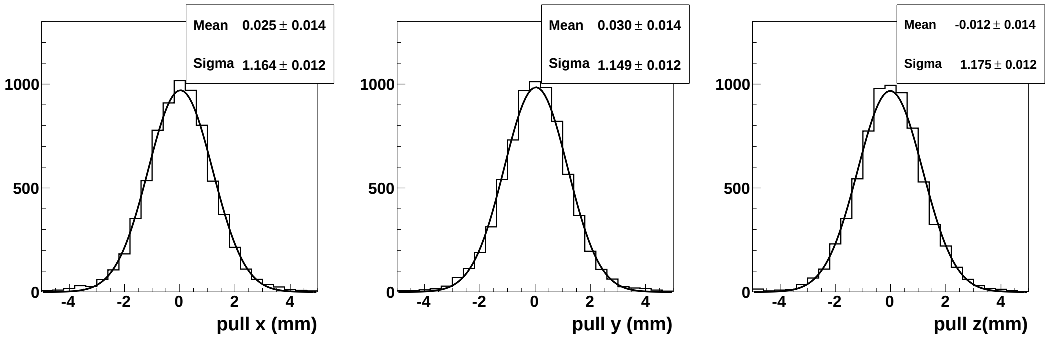

Figure 6.12: Pull distributions of the vertex coordinates for simulated $b\bar{b} \rightarrow X + (J/\psi \rightarrow \mu^+\mu^-)$ events, as obtained with the Billoir method for vertex fitting. All the distributions are fitted with a single Gaussian.

6.5.2 Track parameters at vertex

The updated track parameters at the vertex obtained by both the Billoir method and the mass constrained vertex fit are compared with the true track parameters from the simulated events to obtain the track parameter resolutions shown in figure 6.13. The results show that the addition of a mass-constraint in the vertex-fit helps to improve the measurement of q/p_T of the tracks. The other track-parameters are determined with a similar precision as with the Billoir method and are not significantly affected by the mass constraint.

It can be noted that the use of a mass constraint improves the resolution of q/p_T of the muon track by 20-25% over the whole momentum range. This improvement is a direct result of the mass constraint that constrains the possible momenta of the tracks to the true invariant mass of the J/ψ.

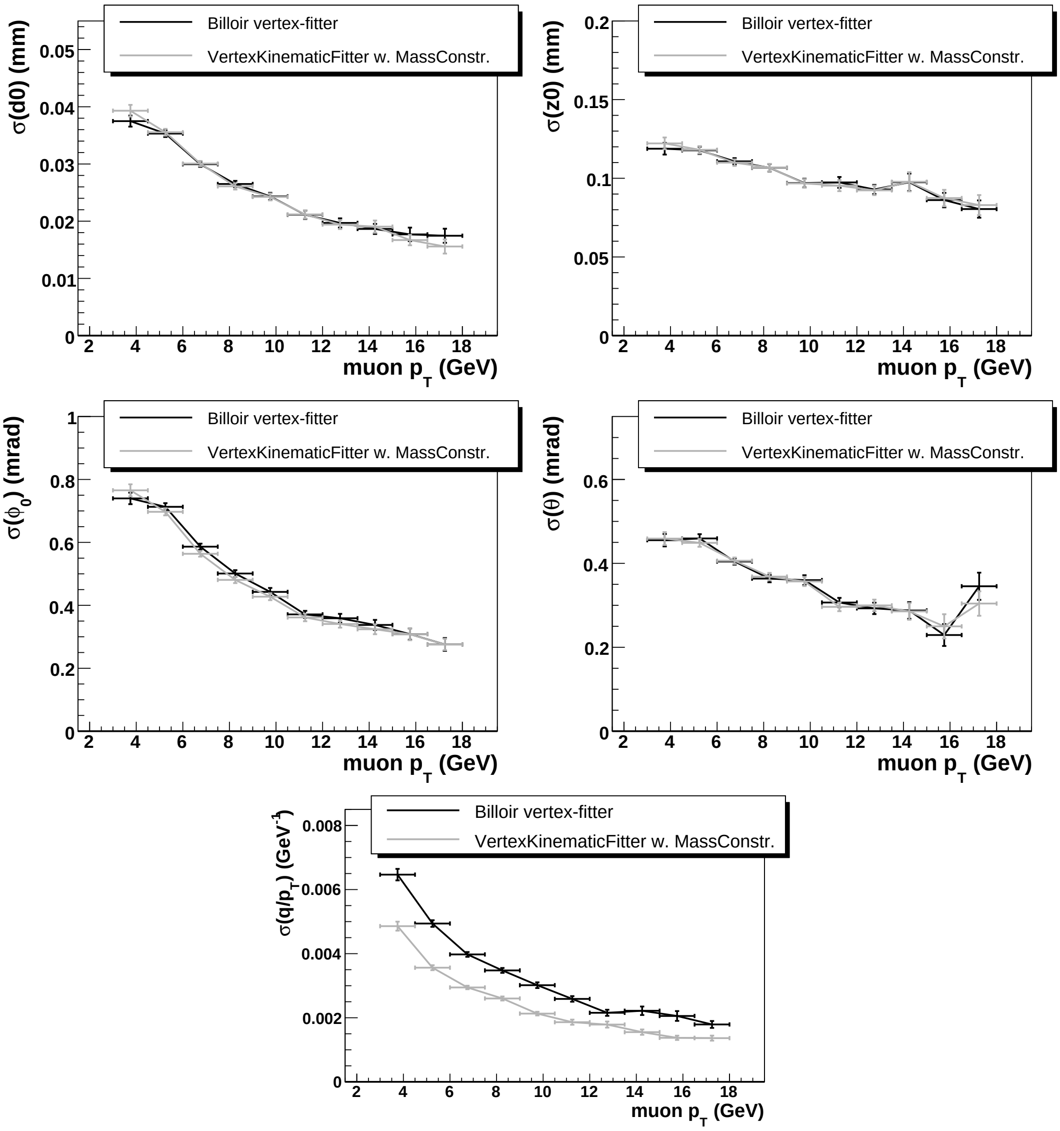

Figure 6.13: Resolutions of the reconstructed track parameters at the vertex as a function of the transverse track-momenta, as obtained with the two vertex fitting methods using simulated $b\bar{b} \rightarrow X + (J/\psi \rightarrow \mu^+\mu^-)$ events.

Figure 6.14 shows the distribution of the pull of q/p of the updated track parameters obtained with the different fit methods. The width of the pull of q/p using the mass-constraint is slightly increased from 1.30 to 1.34, which is most likely caused by the error on q/p being underestimated in the first place. The widths of the pull distribution for d_0, ϕ and θ are 1.2 for both fitters.

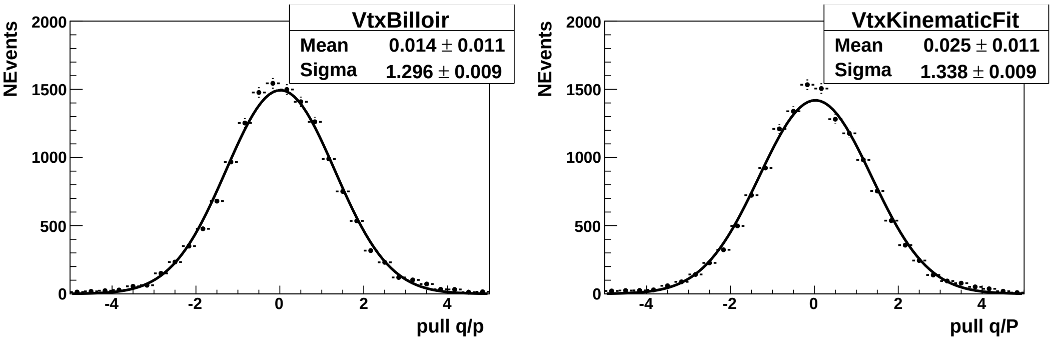

Figure 6.14: Pull distributions of q/p of the track parameters at the vertex obtained with the Billoir method (left) and the vertex fit with mass constraint (right) using simulated $b\bar{b} \rightarrow X + (J/\psi \rightarrow \mu^+\mu^-)$ events.

6.6 Conclusion

A tool for kinematically constraining a vertex fit in ATLAS was developed. This tool is based on minimization of the χ^2 with Lagrange multipliers in the Cartesian frame. This method allows any constraint that can be formulated as function of the Cartesian track parameters at the vertex position to be included in the vertex fit. The tool has been designed for use within the offline reconstruction and physics analysis algorithms and provides a general interface for vertex fitting with kinematic constraints. The constraints are implemented as independent modules which makes the addition of other constraints simple and flexible.

In a sample of simulated $b\bar{b} \rightarrow X + (J/\psi \rightarrow \mu^+\mu^-)$ events it has been shown that the use of a mass constraint improves the momentum resolution of the track parameters at the vertex while achieving similar results for the vertex resolution and resolution of the other track parameters as obtained with the Billoir method for vertex fitting.