Institut für Hochenergiephysik
Fachbereich Algorithmen und Softwareentwicklung

Dissertation

Material estimation and low momentum tracking in Belle II

Moritz Nadler

Abgabe 22. Mai 2013
Verbesserungen 1. Juli 2013

Betreut durch Univ.-Doz. Rudolf Frühwirth

Kurzfassung

Die Messung der Trajektorien geladener Teilchen ist eine wichtige Aufgabe in den meisten Teilchenphysik-Experimenten. Spezielle Komponenten, die Spurdetektoren oder Tracker genannt werden, liefern die nötige Information dazu. Um die optimalen Spur- und Interaktionsvertex-Parameter trotz verschiedenster Fehlerquellen zu schätzen, werden speziell entwickelte und angepasste Algorithmen verwendet.

Diese Doktorarbeit behandelt sowohl die Entwicklung neuer Methoden, um die Spurschätzung weiter zu verbessern, als auch die Evaluierung moderner Spur- und Vertex-Rekonstruktionsalgorithmen für ein neues großes Teilchenphysik-Experiment, das gerade in der Konstruktionphase ist.

Der erste Teil stellt zwei neue Methoden vor, mit denen man die Berechnung der Spurparameter verbessern kann, indem man das Material, das die Teilchen durchqueren, haben aus den Messungen des Spurdetektors selbst mitschätzt. Eine Methode basiert auf einem linearisierten Kleinste-Quadrate-Schätzer, die andere auf einer Kombination aus einem vorwärts- und einem rückwärtslaufenden Kalman-Filter. Zwei Beispiele für eine Anwendung eines solchen Algorithmus sind: a) Auch wenn eine genaue Beschreibung des Materials vorhanden ist, wird aus Geschwindigkeitsgründen oft eine vereinfachte Geometrie verwendet; diese kann durch eine Materialschätzung optimiert werden. b) Es befinden sich – aus welchen Gründen auch immer – im Weg der Teilchen eine oder mehrere Lagen Material, deren Eigenschaften nicht genau bekannt sind. Die auf dem Kalman-Filter basierende Methode wurde anhand von Daten aus einem Teststrahl überprüft und konnte die Qualität der Spurrekonstruktion tatsächlich verbessern.

Der zweite Teil stellt die Arbeit vor, die im Rahmen der Software- und Tracking-Gruppen des zukünftigen Experiments Belle II geleistet wurde. Der Autor beteiligte sich an der Entwicklung des neuen Softwareframeworks, das zur Datenauswertung in Belle II genutzt werden wird. Zwar standen große Teile der Spur- und Vertex-Fit Algorithmen schon in Form von detektorunabhängigen Bibliotheken zur Verfügung, diese mussten jedoch zunächst in die Software von Belle II integriert werden. Im Anschluss daran mussten zahlreiche Überprüfungen, Auswertungen und Verbesserungen vorgenommen werden. Es wird in dieser Arbeit gezeigt, dass einige wichtige Fälle schon jetzt besser rekonstruiert werden können, als es mit der Software des Vorgängerexperiments Belle möglich war; dies betrifft besonders die Rekonstruktion von niederenergetischen Spuren, deren exakte Behandlung ein Schwerpunkt dieser Arbeit war.

Abstract

The measurement of the trajectories of charged particles is an important task in most particle physics experiments. Special components, called tracking detectors or trackers, supply the necessary information. Customized algorithms are used to estimate the optimal track and interaction vertex parameters from this information in the presence of many different sources of error.

This thesis covers both the development of new methods to further improve the tracking in particle detectors, and the evaluation of modern tracking and vertexing algorithms in an important experiment now under construction.

The first part presents two new methods to improve the reconstruction of particle tracks by explicit estimation of the amount of material passed by the tracks. One is based on a linearised least-squares estimator, the other on the combination of a forward and a backward Kalman filter. There are two main scenarios where this could be useful in track reconstruction: a) A detailed description of the materials in the detector is available, but the track reconstruction uses a simplified model. The material estimation method could improve such a model. b) There is is a device with unknown material budget in the particle path. The method is based on a forward and a backward Kalman filter. It was validated with beam test data and was shown to lead to improved track fit results.

The second part presents the work done as member of the software and tracking group of the future Belle II experiment, currently built as an upgrade of the Belle detector at the KEK accelerator research facility in Japan. The author took part in the development of a new software framework that will be used to analyse the data of Belle II. While large parts of the track and vertex fit algorithms were already implemented in the form of detector independent software libraries, they had to be interfaced with the new framework, tested, evaluated, and improved. It is demonstrated that for some examples relevant for the physics program of Belle II the performance already surpasses the one of Belle, particularly in the domain of low-energy track reconstruction, a main focus of the present work.

Acknowledgements

I want to thank everybody who contributed to the creation of this thesis. I am particularly grateful to:

- Rudi Frühwirth who supervised this thesis and always gave questions and problems brought to him by his student the highest priority;
- Christoph Schwanda for his suggestion of interesting comparisons of the Belle and Belle II tracking capabilities;
- the ATLAS 3D collaboration and the ATLAS IBL collaboration for the permission to use their test beam data for a validation of the material estimation method, and especially Håvard Gjersdal, who prepared the test beam data;
- all my colleagues from the Belle II software and Belle II tracking groups, especially Oksana Lutz, Johannes Rauch, Christian Pulvermacher and Jakob Lettenbichler;
- the Austrian tax payer for his and her willingness to finance basic research.

Thank you very much!

Contents

1 Introduction and motivation

Particle physics is the search for the most fundamental constituents of nature and their interactions. High energy physics tries to achieve this goal by concentrating large amount of energy in very small spaces and measuring the result. To reach the highest energy densities large machines called colliders are used to accelerate particles and collide them at specific points. Around these points detectors are build to record the result of the collisions.

The current state of particle physics is called the standard model (of particle physics). It is a collection of the properties of the known fundamental particles and their interactions, mathematically formulated as quantum field theories. While the standard model had great success in the past and so far no effect or particle has been found that clearly violates it, it is consensus amongst physicist that extensions or new theories with more explanative power must exist.

A few examples of the predictions of the standard model that were found to be true were the existence of additional particles such as charm, bottom and top quarks, the W and Z bosons, and most recently the Higgs boson. Additionally a constituent of the standard model – quantum electrodynamics – is the most accurate theory of all physics, if defined as the number of digits agreeing between theory and experiment.

The challenges to the standard model are also well understood. They hint at the existence of more general theories and the standard model being just be an effective theory of those, valid in the energy and precision range that has been accessible by the experiments of the last decades. Examples of these issues are the high number of parameters, such as the elementary particle masses and the coupling constants of the forces, that have to be determined by the experiment, the exclusion of gravity, and the hierarchy problem. As an illustration of the approach to the discovery of new physics, the physics goals of the Belle II experiment are described in section 4.3.

Today, new discoveries in particles physics are normally made in specialised research facilities, where experiments are conducted at very large machines, financed and run by international collaborations. The most famous and largest example is CERN, where several thousand researchers from all over the world are involved. The Belle II collaboration, of which the author is a member, already counts several hundred participants. Needless to say, most individual contributions to these projects are highly specialised.

The experiments in CERN and the Belle II experiment at the KEK facility are detectors built around the collision points of high-energy or high-intensity particle beams. One of the key ingredients to the successful analysis of the physics processes occurring in these collisions is the measurement of the momentum and the origin of the charged particles that are observed in the detector. This part of the data analysis, called track and vertex reconstruction, is the topic of the present work.

The thesis consists of two parts. The first part presents new methods to improve track reconstruction by estimating the amount of material crossed by a charged particle, based on the interaction of the particle with the material. The second part describes the development, adaptation and validation of modern track and vertex reconstruction algorithms for the Belle II project. It is shown how these methods improve certain aspects of tracking and vertexing in comparison to its predecessor Belle.

This thesis presents new methods to improve track reconstruction with in detectors unknown material budgets as well as the adaptation and validation of modern tracking and vertexing algorithms for the Belle II project. It is shown how these methods improve certain aspects of tracking in comparison to Belle I.

2 Tracking in high energy physics

2.1 Purpose of tracking and introduction

The main objective of tracking is the estimation of track parameters. In case of charged particles in the presence of a constant magnetic field these parameters are the ones needed to describe a helix. The most important physical properties that can be extracted from the estimated track parameters are the particle momentum and its place of origin (vertex).

The input for the tracking normally comes from dedicated tracking detectors producing some kind of position measurement. Many modern general purpose detectors use solid state detectors or a combination of solid state and gaseous detectors.

The tracking detectors currently under construction for the Belle II are an example for such a combined system of solid state and gaseous detectors. They are briefly described in chapter 4.2. More details about the physical working principles as well as strengths and weaknesses of different detector types can be found in many textbooks. A good example is [1].

2.1.1 Typical process flow in tracking

The first step in tracking is pattern recognition, which means track finding in this context: sorting the detector hits into groups representing particle tracks, plus a group for noise or background hits. Figure 2.1 gives an idea of the difficulty of this task especially in the presence of high track multiplicity and a large number of background hits.

The next step is the track fitting. That is the actual estimation of the track parameters. The number and kind of the track parameters depend on the chosen track model. The track model choice is constrained by the physical properties of the environment – such as the existence of a magnetic field – and the estimation method.

A third step is the vertex reconstruction, there bunches of tracks originated from one vertex are identified and the position of their common vertex is estimated. The identification of vertices and the assignment of tracks – called vertex finding in analogy to the track finding – called can be done purely with geometrical properties of the reconstructed tracks or when available with additional information from the particle identification detectors. This enables the grouping of tracks based on the knowledge of particle decays.

Of course tracking is not necessarily strictly divided into these steps. Algorithms might solve the track finding and fitting problem in a single step; for instance, the combinatorial Kalman filter works in this way. Also the separation of the stages might be blurred when

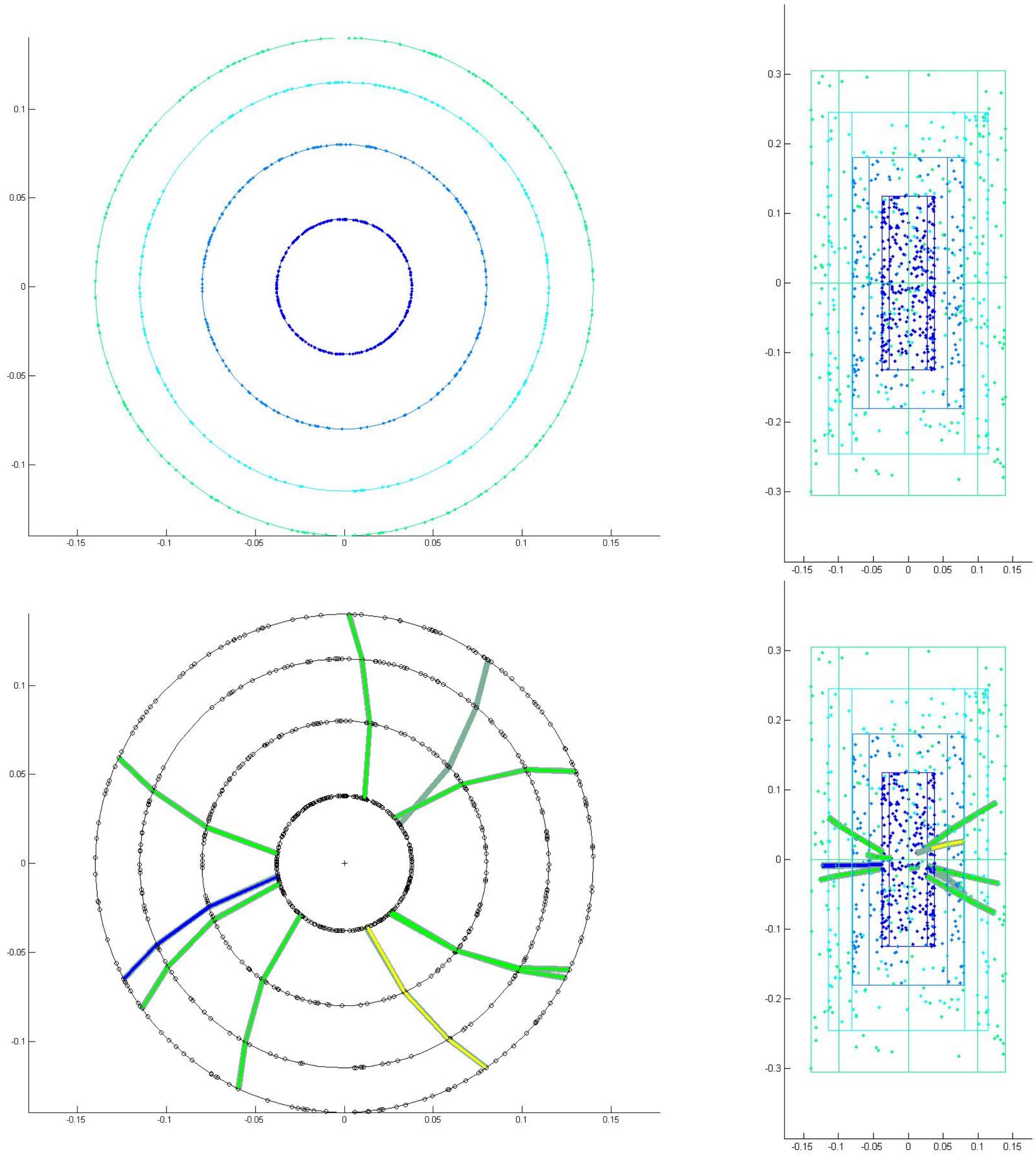


Figure 2.1: Track finding problem in the Belle II SVD (a 4 layer Si-detector) with artificially increased background density to test the finder under a worst case scenario. Figure and track finder by Jakob Lettenbichler [2]

for example the track fitting algorithm incorporates means for signal/noise classification as the Deterministic Annealing Filter does. Additionally some steps might be repeated if better information is available. For example refitting the tracks after the vertex fit to include the common vertex position as additional information in the track fit.

2.2 Estimation or fitting methods

Because a measurement always suffers from some kind of random error, an estimation method is needed to extract the optimal information (optimal in a statistical sense). The estimators generally used in track fitting are based on the linear model. The linear model connects the measurements $\boldsymbol{m}$, the model description $\mathbf{M}$, the true values of the quantities to be estimated $\boldsymbol{x}_\mathrm{t}$ and the random error $\boldsymbol{\epsilon}$ in the following way:

$$\boldsymbol{m} = \mathbf{M}\boldsymbol{x}_\mathrm{t} + \boldsymbol{\epsilon} \tag{2.1}$$

A number of established and well analysed estimation methods based on this model exist. Before they can be applied one has to make sure the two main assumptions of the linear model are fulfilled "well enough": The linear dependency of the measurement $\boldsymbol{m}$ on the true state $\boldsymbol{x}_\mathrm{t}$, and the additivity of the errors $\boldsymbol{\epsilon}$. The least-squares method is a very common example of a linear estimator.

If a problem is – not too strongly – non linear ($\boldsymbol{m} = \boldsymbol{M}(\boldsymbol{x}_\mathrm{t}) + \boldsymbol{\epsilon}$), one can approximate the non linear model $\boldsymbol{M}(\boldsymbol{x}_\mathrm{t})$ by a linear one $\mathbf{M}\boldsymbol{x}_\mathrm{t}$, using a first-order Taylor expansion. As the expansion point determines the quality of the linear approximation, it has to be chosen very carefully.

2.2.1 Global least-squares method

If a measurement process can be mathematically modelled as in equation (2.1), the estimated $\boldsymbol{x}$ given by the least-squares method is equal to:

$$\boldsymbol{x} = (\mathbf{M}^\mathrm{T}\mathbf{V}^{-1}\mathbf{M})^{-1}\mathbf{M}^\mathrm{T}\mathbf{V}^{-1}\boldsymbol{m}, \tag{2.2}$$

where $\mathbf{V} = \mathrm{Cov}(\boldsymbol{\epsilon})$ is the known covariance matrix of the error vector $\boldsymbol{\epsilon}$. If no prior information about the error exists, $\mathbf{V} = \mathbf{1}$ and equation 2.2 simplifies accordingly.

The uncertainties of the estimated results $\boldsymbol{x}$ given as a covariance matrix $\mathbf{C}$ are $\mathbf{C} = (\mathbf{M}^\mathrm{T}\mathbf{V}^{-1}\mathbf{M})^{-1}$

When calculating the solution most of the CPU time is used for the matrix inversions. Two non trivial inversions have to be done per track; one if $\mathbf{V}$ is diagonal. $\mathbf{V}$ has size $\dim(\boldsymbol{m}) \times \dim(\boldsymbol{m})$ and $\mathbf{M}^\mathrm{T}\mathbf{V}^{-1}\mathbf{M}$ has size $\dim(\boldsymbol{x}) \times \dim(\boldsymbol{x})$. As the complexity of matrix inversion – if a direct inversion solver is used – is cubic in measurements per track, so will be the complexity of the estimator.

The least-squares estimator has well-known mathematical properties: If $\boldsymbol{\epsilon}$ is normally distributed and the model is exactly linear and not just a linear approximation, then the estimation is efficient, unbiased and consistent. If $\boldsymbol{\epsilon}$ is still normally distributed, but the model is a linear approximation of a non-linear problem, then the estimation is asymptotically efficient, asymptotically unbiased and consistent. Finally, if $\boldsymbol{\epsilon}$ is not normally distributed and the model is linear, then the estimator has the smallest variance possible among all linear estimators, and it is unbiased and consistent.

2.2.2 Iterative least-squares method: The Kalman filter

Today most track and vertex fitting algorithms used in particle physics are based on the Kalman filter. The Kalman filter is also a least-squares estimator, but instead of calculating the estimation in one go it works with an alternating prediction/update scheme. The Kalman filter solves two problematic aspects of the global least-squares method. First, it reduces the size of matrices that have to be inverted and makes there size independent of the number of measurements per track; second, it makes the treatment of material effects such as multiple scattering, energy loss and bremsstrahlung much more straightforward.

The Kalman filter as used in particle tracking works in the following way. It starts with some prior information of the track state $\boldsymbol{x}$ and its covariance matrix $\mathbf{C}$, which normally comes as a rough estimate from the pattern recognition. These initial values are also called the seed. The filter consists of alternating prediction/update steps. The prediction is obtained by propagating $\boldsymbol{x}$ onto the next measurement plane, using the equation of motion of a charged particle in a magnetic field, and linear error propagation of $\mathbf{C}$. In the propagation step the changes to $\boldsymbol{x}$ and $\mathbf{C}$ caused by the interaction with material are calculated according to some appropriate model. The update step performs the correction of the prediction with the new measurement information, which is tantamount to a weighted mean of the predicted track parameters and the measured position.

When using the global linear model, every system property has to be modeled by the single model matrix $\mathbf{M}$. Because the alternation of prediction and update follows the actual track, additional system properties can be incorporate at places in the formalism corresponding to their occurrence in the detector. This is the reason for the more straightforward inclusion of material effects.

Mathematical formulation

The equations describing the Kalman filter use the following convention: If a variable such as $\boldsymbol{x}_{k|k}$ has two indices, the first indicates at which measurement layer k the prediction is defined, and the second gives this information for the latest update step. If only one index is given it means this quantity is the same for the update and prediction step.

As a linear estimator the Kalman filter is based on a linear mapping of the true state $\boldsymbol{x}_{\mathrm{t}}$ onto the measurement; but whereas the global linear model maps onto all measurements, the Kalman filter maps the true state of the system $\boldsymbol{x}_{\mathrm{t}}$ at the current measurement layer k onto the measurements taken at this layer $\boldsymbol{m}_k$:

$$\boldsymbol{m}_k = \mathbf{H}_k \boldsymbol{x}_{\mathrm{t},k} + \boldsymbol{\epsilon}_k \tag{2.3}$$

Again the covariance matrix $\mathbf{V}_k = \mathrm{Cov}(\boldsymbol{\epsilon}_k)$ of the measurement error is assumed to be known. If the mapping $\mathbf{H}_k$ is non-linear, it is most of the times sufficient to linearize it, for instance by a first-order Taylor expansion. In order to connect the state in one measurement layer with the next one, one needs a so-called system equation:

$$\boldsymbol{x}_{\mathrm{t},k} = \boldsymbol{F}_k(\boldsymbol{x}_{\mathrm{t},k-1}) + \mathbf{P}_k \boldsymbol{\delta}_k \tag{2.4}$$

As before, equation (2.4) may be derived by using the first-order Taylor expansion of a non-linear system equation. In particle tracking, the first summand on the right hand side represents the deterministic change of the helix parameters $\boldsymbol{x}_\mathrm{t}$ when the particle is moving from one measurement layer to the next one; the second term represents the random changes of $\boldsymbol{x}_\mathrm{t}$ caused by material effects such as multiple scattering. Therefore $\boldsymbol{F}_k$ and its Jacobian matrix $\mathbf{F}_k$ have to be derived from the track model, i.e. the equations of motion of a particle in a magnetic field, and $\mathbf{P}_k$ has to be computed from the multiple scattering model. Also $\mathbf{Q}_k = \operatorname{Cov} \boldsymbol{\delta}_k$ can be calculated by using an adequate multiple scattering model.

Once the system equation is determined, the propagation step transforming the updated state from the last measurement layer to the prediction in the next layer is relatively straightforward:

$$\boldsymbol{x}_{k|k-1} = \boldsymbol{F}_k(\boldsymbol{x}_{k-1|k-1}) \tag{2.5}$$

$$\mathbf{C}_{k|k-1} = \mathbf{F}_k \mathbf{C}_{k-1|k-1} \mathbf{F}_k^\mathrm{T} + \mathbf{P}_k \mathbf{Q}_k \mathbf{P}_k^\mathrm{T} \tag{2.6}$$

Mathematically speaking, equation (2.6) is a linear error propagation.

The next two equations describe the update step. Both the updated covariance matrix $\mathbf{C}_{k|k}$ and the updated state $\boldsymbol{x}_{k|k}$ are the weighted means of the prediction at layer k and the measurement at layer k. The update step can be derived by the least-squares principle.

$$\mathbf{C}_{k|k} = (\mathbf{C}_{k|k-1}^{-1} + \mathbf{H}_k^\mathrm{T} \mathbf{V}_k^{-1} \mathbf{H}_k)^{-1} \tag{2.7}$$

$$\boldsymbol{x}_{k|k} = \mathbf{C}_{k|k}(\mathbf{C}_{k|k-1}^{-1} \boldsymbol{x}_{k|k-1} + \mathbf{H}_k^\mathrm{T} \mathbf{V}_k^{-1} \boldsymbol{m}_k) \tag{2.8}$$

With some matrix algebra an alternative formulation of the update step can be derived:

$$\mathbf{K}_k = \mathbf{C}_{k|k-1} \mathbf{H}_k^\mathrm{T} (\mathbf{V}_k + \mathbf{H}_k \mathbf{C}_{k|k-1} \mathbf{H}_k^\mathrm{T})^{-1} \tag{2.9}$$

$$\boldsymbol{x}_{k|k} = \boldsymbol{x}_{k|k-1} + \mathbf{K}_k(\boldsymbol{m}_k - \mathbf{H}_k \boldsymbol{x}_{k|k-1}) \tag{2.10}$$

$$\mathbf{C}_{k|k} = (\mathbf{1} - \mathbf{K}_k \mathbf{H}_k) \mathbf{C}_{k|k-1} \tag{2.11}$$

This alternative formulation has computational advantages in tracking, because only one instead of two matrix inversion have to be calculated, and the matrix to be inverted has the size $\dim(\boldsymbol{m}) \times \dim(\boldsymbol{m})$ with $\dim(\boldsymbol{m})$ normally being 1 or 2; in contrast, the former formulation inverts $\dim(\boldsymbol{x}) \times \dim(\boldsymbol{x})$ matrices, with $\dim(\boldsymbol{x})$ normally being 5.

In order to start the filter, the seed values $\boldsymbol{x}$ and $\mathbf{C}$ are extrapolated to the first measurement layer $k = 1$ according to equations (2.5) and (2.6). In the the Kalman filter nomenclature the indices of the seed values would therefore be $\boldsymbol{x}_{0|0}$ and $\mathbf{C}_{0|0}$. If $\boldsymbol{x}_{0|0}$ is calculated by the pattern reconstruction from the same information (hits) that is given to the Kalman filter, $\mathbf{C}_{0|0}$ should be large, so that the hit information is not used twice and the number of degrees of freedom (ndf) of the fit result is correct.

If there are n measurement layers, the final estimated state after the last update step is $\boldsymbol{x}_{n|n}$, along with its covariance $\mathbf{C}_{n|n}$. Only $\boldsymbol{x}_{n|n}$ contains the full information from all measurements. To make the full information available at every layer k – a procedure called “smoothing” – one can additionally run the Kalman filter backwards, starting with layer n and ending at layer 1, leading to $\boldsymbol{x}_{\mathrm{b},1|1}$ containing the full information. Finally, the forward and backward states are combined by a weighted mean:

$$\boldsymbol{x}_{\mathrm{sm},k} = \mathbf{C}^{-1}_{\mathrm{f},k|k-1}\boldsymbol{x}_{\mathrm{f},k|k-1} + \mathbf{C}^{-1}_{\mathrm{b},k|k}\boldsymbol{x}_{\mathrm{b},k|k} \tag{2.12}$$

$$\mathbf{C}_{\mathrm{sm},k} = (\mathbf{C}^{-1}_{\mathrm{f},k|k-1} + \mathbf{C}^{-1}_{\mathrm{b},k|k})^{-1} \tag{2.13}$$

In equations (2.12) and (2.13) the indices f and b respectively label the states coming from the forward and backward filter, and sm labels the smoothed states.

The smoothing procedure is useful for tuning the hit resolution and for identifying outlier hits (more details in section 2.2.4). Also, it is useful to have optimal estimates at both ends of the track if extrapolation into other detector modules is required.

Kalman filter with reference track

If the first few hits just by chance produce a strong bias in the track direction, it is possible that the function $\boldsymbol{F}_k$ calculated to propagate $\boldsymbol{x}_{k-1|k-1}$ will miss the layer k completely. This would lead to a failed track fit even if all consecutive measurements lie perfectly on the track. Even very good seed values cannot prevent this, as their large initial covariance matrix allow the first hits to completely change the track state. Also more robust iterative estimator prevent will not help, as no matter how strongly this estimator would down-weight the influence of the first badly aligned hits, it first has to propagate past them to reach the hits well on track.

A possible solution to this problem is to propagate and update the filter along a reference track. The seed state is propagated through all measurement layers without any update step, saving the reference states $x_{\mathrm{r},k}$ and Jacobian (propagation) matrices $\mathbf{F}_k$ created in this process. Next the the Kalman filter is run using modified equations (2.14) and (2.15) for propagation and update.

$$\Delta\boldsymbol{x}_{k|k-1} = \mathbf{F}_k\Delta\boldsymbol{x}_{k-1|k-1} \tag{2.14}$$

$$\Delta\boldsymbol{x}_{k|k} = \Delta\boldsymbol{x}_{k|k-1} + \mathbf{K}_k(\Delta\boldsymbol{m}_k - \mathbf{H}_k\Delta\boldsymbol{x}_{k|k-1}) \tag{2.15}$$

Δ denotes the difference of a quantity with respect to the reference states $x_{\mathrm{r},k}$, as defined in equations 2.16.

$$\Delta\boldsymbol{x}_{k|k} = \boldsymbol{x}_{k|k} - \boldsymbol{x}_{\mathrm{r},k} \quad \Delta\boldsymbol{x}_{k|k-1} = \boldsymbol{x}_{k|k-1} - \boldsymbol{x}_{\mathrm{r},k-1} \quad \Delta\boldsymbol{m}_k = \boldsymbol{m}_k - \mathbf{H}_k\boldsymbol{x}_{\mathrm{r},k} \tag{2.16}$$

The expansion point of he Taylor approximation $\mathbf{F}_k$ is now $x_{\mathrm{r},k-1}$ and not the updated state $\boldsymbol{x}_{k-1|k-1}$ any more. Using the pre-calculated matrices $\mathbf{F}_k$ prevents the generation of propagation matrices that lead the state away from a measurement layer because of a few bad hits.

As the reference track is calculated from the seed, it can only improve the number of successful fits when the seed state is a good representation of the track. In particular, calculating the seed simply from the first three hits would defeat the purpose of the reference track.

2.2.3 Testing fitted track parameters

Testing the plausibility of the calculated results is important in every field, but even more so when dealing with random input into an algorithm. While statistical estimators can in principle yield any result, one can use statistical tests to calculate how likely it is that a given outcome was produced by chance. The statistical tests themselves give a random variable distributed according to a specific function. If the test distribution is different from the theoretical one, the assumptions about the model or the implementation are probably wrong. If an estimate is made from simulated data one should of course make use of the this "truth" information by using additional tests. The following tests were used to validate the track fit results in this thesis.

Total χ^2 and p-values

The name of this test stems from the fact that the test statistic is a χ^2 distributed sum of χ^2 distributed values. The shape of a χ^2-distribution depends on the parameter "number of degrees of freedom" ndf $= m - n$, where $m = \sum_k \dim(\boldsymbol{m}_k)$ is the number of measurements and $n = \dim(\boldsymbol{x})$ the number of estimated parameters. If the seed value $\boldsymbol{x}$ does not come from the measurements used in the fit, but from an additional independent information source, the ndf has to be increased accordingly. If the seed value is calculated from the same measurements that are used in the fit, the seed covariance matrix $\mathbf{C}$ has to be very large to prevent the seed value from contributing to the total χ^2.

To calculate the total χ^2, one has to determine the residual $\boldsymbol{r}_k$ of every measurement $\boldsymbol{m}_k$ according to

$$\boldsymbol{r}_k = \boldsymbol{m}_k - \mathbf{H}_k \boldsymbol{x}_k, \tag{2.17}$$

where $\mathbf{H}$ is the projection of the track parameters onto the measurements, and the covariance matrix of $\boldsymbol{r}_k$ according to

$$\mathbf{R}_k = \mathbf{V}_k - \mathbf{H}_k \mathbf{C}_k \mathbf{H}_k^{\mathrm{T}}, \tag{2.18}$$

where $\mathbf{V}$ is the covariance matrix of the measurement and $\mathbf{C}$ is the updated covariance matrix of the Kalman filter. With all these quantities calculated the per-layer χ_k^2 is

$$\chi_k^2 = \boldsymbol{r}_k^{\mathrm{T}} \mathbf{R}_k^{-1} \boldsymbol{r}_k \tag{2.19}$$

and the total χ^2 is simply

$$\chi_{\mathrm{tot}}^2 = \sum_k \chi_k^2. \tag{2.20}$$

Note that with this equations the total χ^2 can be either calculated from the updated forward or updated backward state of the Kalman filter. If both filters start with same prior information and use the same track model the total χ^2 of the forward filter should be equal to the backward filter total χ^2 (besides small numerical fluctuations).

Equations (2.17) and (2.18) use the updated $\boldsymbol{x}$ and $\mathbf{C}$. If the predicted $\boldsymbol{x}$ and $\mathbf{C}$ are used equation (2.18) has to be changed to $\mathbf{R}_k = \mathbf{V}_k + \mathbf{H}_k \mathbf{C}_k \mathbf{H}_k^{\mathrm{T}}$. This must yield and identical χ^2_{tot} if the Kalman update step works correctly. Using the prediction has numerical advantages because $\mathbf{R}_k$ cannot get negative diagonal elements due to numerical fluctuations when $\mathbf{V}_k$ and $\mathbf{H}_k\mathbf{C}_k\mathbf{H}_k^{\mathrm{T}}$ are approximately equal.

Because different tracks might have different number of measurements, their total χ^2s come from different χ^2-distributions. Therefore calculating their mean and standard deviation does not make sense. With the information of their individual degrees of freedom ndf and the χ^2 cumulative distribution function the total χ^2s can be transformed to p-values, which should follow the same uniform distribution for all tracks. Because p-values are probabilities, they lie between 0 and 1.

In statistics the p-value is the probability to obtain the same result of a statistical test or a more extreme one as the one that was actually observed, under the assumption that the null hypothesis is true. The χ^2_{tot} is a result of a statistical test. A small p-value therefore indicates a bad fit, as the assumptions underlying the fit procedure would only lead to this fit result with the probability given by the small p-value. Of course one cannot tell from a single p-value what went wrong. It could be chance, bad input data (input data not following the assumed distribution, for instance a measurement outlier), errors in the track model or errors in the implementation. If on the other hand the p-value distribution is uniform, all the mentioned error sources have an impact on the result that is smaller than the effect of the unavoidable measurement errors. All model assumptions and the quality of the implementation can therefore be regarded as “good enough”.

Of course having a flat p-value distribution does not guarantee that everything is correct. As an example one can imagine a Monte-Carlo simulation in which the measurement error was set too large. If the simulation uses the same error consistently in the generation of the measurement and in the fit algorithm the p-values can be perfect while the average deviations of the estimate from the true value will be much larger than they should be.

Pulls and χ^2 from residuals calculated with the measurements

In particle physics a “pull” is what in statistics is called a standard score. It is a residual divided by its standard deviation: $z = r/\sigma_r$. The z-values should be (approximately) normally distributed with mean 0 and $\sigma_z = 1$.

To test the quality of the fit in each measurement layer individually, the pulls from the components of the residual vector from equation (2.17) can be calculated:

$$z_{i,k} = \frac{r_{i,k}}{R_{i,i,k}} \tag{2.21}$$

where $i \in [1, \dim(\boldsymbol{m}_k)]$. The corresponding χ^2 per layer is the one defined in equation (2.19).

One can calculate the pulls and χ^2s from forward updated, forward predicted, backward updated, backward predicted, and smoothed states and covariance matrices. If the prediction $\boldsymbol{x}_{k|k-1}$ is used, equation (2.18) becomes:

$$\mathbf{R}_k = \mathbf{V}_k + \mathbf{H}_k\mathbf{C}_k\mathbf{H}_k^{\mathrm{T}}. \tag{2.22}$$

Note that the minus sign has changed to a plus sign in equation (2.22), and $\mathbf{C}_k$ is now $\mathbf{C}_{k|k-1}$ instead of $\mathbf{C}_{k|k}$.

For every track every χ^2_k calculated with the updated state has to be identical to the χ^2_k calculated with the predicted state. If this is not the case, it indicates an error in the “update step” of the Kalman filter.

Every χ^2_k should be (approximately) χ^2 distributed with $\dim(\boldsymbol{m}_k)$ degrees of freedom. This is not true for the first few k but only after all estimated parameters $\boldsymbol{x}$ are determined. If on the other hand the fit starts with prior information (a fully determined $\boldsymbol{x}$ with corresponding $\mathbf{C}$ from a different information source) all χ^2_k should be distributed with $\dim(\boldsymbol{m}_k)$ degrees of freedom.

If the pulls are correct (standard normal distributed) but the χ^2_k is not, there may be a problem in the calculation of the off-diagonal elements (covariances) in $\mathbf{C}_k$.

Pulls and χ^2 from residuals calculated with the truth information

The per-layer pulls and and χ^2 calculated with the truth information are similar the ones from the last subsection, but instead of using the residual of the measurement $\boldsymbol{m}_k$ and the projection of the estimated state onto the measurement $\mathbf{H}_k\boldsymbol{x}_k$, one calculates the difference of the estimated state and the true state $\boldsymbol{x}_{k,\mathrm{t}}$:

$$\boldsymbol{r}_k = \boldsymbol{x}_k - \boldsymbol{x}_{k,\mathrm{t}}. \tag{2.23}$$

The covariance of $\boldsymbol{r}_k$ is

$$\mathbf{R}_k = \mathbf{C}_k, \tag{2.24}$$

as $\boldsymbol{x}_{k,\mathrm{t}}$ has no error and $\mathbf{C}_k$ is the covariance matrix of the state $\boldsymbol{x}_k$. The equations to calculate $z_{i,k}$ and χ^2_k stay the same, but the number of pull distributions per layer is now equal to the number of estimated parameters. Therefore $i \in [1, \dim(\boldsymbol{x})]$ instead of $i \in [1, \dim(\boldsymbol{m}_k)]$, and correspondingly χ^2_k now has $\dim(\boldsymbol{x})$ degrees of freedom and not $\dim(\boldsymbol{m}_k)$. Again $\boldsymbol{x}_k$ and $\mathbf{C}_k$ can come from the forward updated, forward predicted, backward updated, backward predicted, or smoothed state. In contrast to the test using residuals calculated from the measurements, these tests are not necessarily producing identical χ^2 values with the updated and the predicted state.

In addition to the measurement layers this test can be done at any point along the track where an extrapolation (prediction state) and a true value exists. A useful point to test the track fit is the point of closest approach to the origin or vertex of the track. To this end the state at the innermost measurement is extrapolated to the point of closest approach to the true vertex position, and the χ^2 and pulls with respect to the true state at this vertex are calculated. In a collider experiment the next step in the track reconstruction chain – the vertex fit – will start with fitted track parameters extrapolated close to the likely vertex.

2.2.4 Robust estimation

In statistics an estimation method is robust when it is insensitive to a certain amount of outliers. A rough definition of an outlier is: A datum in the sample that does not come from the distribution assumed when developing the estimation to extract a quantity from the data. The global and iterative least-squares estimators introduced in this section are not robust, meaning that a single arbitrary large outlier can lead to an estimate arbitrarily far away from the true value. The robustness is typically measured by the maximal proportion of outliers an estimator can handle without assuming arbitrary values. This proportion is called the breakdown point of an estimator. The breakdown point of the least-squares estimators discussed above is 0.

The mean and the median can be used to illustrate the concept of the breakdown point. To determine the centre of a data sample frequently the mean is used. A robust alternative to the mean is the median. It has a breakdown point of 0.5, which is the highest possible value, but it also comes with disadvantages: it is computationally more costly, and in many cases has a larger spread than the mean if it used to estimate the central value of a symmetric distribution.

Higher computational complexity is a typical property of robust estimators. They also tend to have worse statical properties like being only asymptotically unbiased when a non-robust estimator calculating the same quantity would simply be unbiased, for example.

The Kalman filter as described in section 2.2.2 is rarely used in production software. In fact, some measures have to be taken to make the Kalman filter more robust. The standard way of doing that is to introduce cuts on certain test statistics. In the case of track fitting this can be done on both track and hit level. For example: Every track not meeting a p-value threshold will be excluded from the set of successful reconstructed tracks, and every hit causing a χ^2 increment above a certain value will be excluded from the track fit. An improvement to this scheme is to base the hit rejection on the χ^2_k calculated with the smoothed state and to redo the fit if at least one hit was discarded.

There are obvious disadvantages of this approach. If, for example, a track starts with outliers, the first “real” hit might be falsely identified as an outlier and discarded. A more sophisticated way to make track fitting more robust is presented in section 2.2.5.

2.2.5 The Deterministic Annealing Filter (DAF)

The deterministic annealing filter (DAF) [3] is a robust track fitter designed to cope with bad seed parameters and outlier or background hits. The DAF utilises a Kalman filter and adds weights and annealing. In addition to estimating the track parameters the DAF calculates the probability of a hit belonging to a track. This probability then modifies the influence of the hit on the estimation. This procedure is called “soft assignment”.

The DAF is an iterative procedure. It starts by running a forward and backward Kalman filter plus smoother. The equations of the Kalman filter are changed to incorporate the weights. If no additional prior information is available the initial weights will treat all measurements as equal. The smoothed states and their covariance matrices are used to calculate the weights for the next iteration of the DAF. This procedure is repeated until a given number of maximum iterations is reached or some convergence criteria are met, for instance when the weights have settled to their final values.

It is possible to pass several hits per measurement layer to the DAF and to let them compete for inclusion in the track. Therefore the DAF to some extent blurs the strict separation between track finding and track fitting tasks.

The annealing scheme is implemented by multiplying the measurement covariances $\mathbf{V}_k$ by a "temperature" factor T_I during the weight calculation – and only there: during the Kalman fit the $\mathbf{V}_k$ are modified only by the weight of the corresponding hit. T_I depends on I the iteration index of the DAF (one full Kalman filter run plus weight calculation). A large T leads to weights close to 0.5, while a small T leads to weights close to 0 or 1. The smallest possible value of T is 0, which forces the weights to be exactly either 0 or 1 making the "soft assignment" a "hard assignment". The sequence of the T_I make up the annealing scheme. One usually starts with a high value of T, which is then lowered gradually with increasing I. This prevents a premature down-weighting of hits because of bad seed values for the Kalman fit.

Mathematical formulation

The DAF has two parts which are repeated during the iteration over I: first, running a modified Kalman filter; second, calculating the weights w_i for the $I+1$ step. The propagation procedure of the Kalman filter does not need any changes. However, the update procedure needs to take into account the weight that is attached to every hit, and that there might be several competing hits, indexed by i, per measurement layer k:

$$\boldsymbol{x}_{k|k} = \boldsymbol{x}_{k|k-1} + \mathbf{K}_k \sum_i w_{k,i} (\boldsymbol{m}_{k,i} - \mathbf{H}_k \boldsymbol{x}_{k|k-1}) \tag{2.25}$$

$$\mathbf{K}_k = \mathbf{C}_{k|k-1} W_k \mathbf{H}_k^{\mathrm{T}} (\mathbf{V}_k + \mathbf{H}_k \mathbf{C}_{k|k-1} \mathbf{H}_k^{\mathrm{T}})^{-1} \tag{2.26}$$

with $W_k = \sum_i w_{k,i}$. Using the DAF gain matrix from equation (2.26), the update of the covariance matrix does not change compared to a standard Kalman filter and still reads as:

$$\mathbf{C}_{k|k} = (\mathbf{1} - \mathbf{K}_k \mathbf{H}_k) \mathbf{C}_{k|k-1} \tag{2.27}$$

After the calculation of the smoothed state and covariance matrix with equations (2.12) and (2.13), the weights for the next iteration have to be calculated. As the weights should express the probability of a hit to belong to the track, it makes sense to base them on the weighted residual (= χ^2 value) $\boldsymbol{r}^{\mathrm{T}} \mathbf{R}^{-1} \boldsymbol{r}$. On order to ensure that large residuals will quickly lead to small weights, their computation is based on a Gaussian probability density function. First the auxiliary quantities ϕ_i are computed:

$$\phi_i = \frac{1}{(2\pi)^{m/2} \sqrt{\det(T_I \mathbf{V}_i)}} \exp\left(-\frac{1}{2} (\boldsymbol{m}_i - \mathbf{H}_i \boldsymbol{x}_{\mathrm{sm}})^{\mathrm{T}} \frac{1}{T_I} \mathbf{V}_i^{-1} (\boldsymbol{m}_i - \mathbf{H}_i \boldsymbol{x}_{\mathrm{sm}}) \right), \tag{2.28}$$

where $m = \dim(\boldsymbol{m}_k)$ is the dimension of the measurement in the current layer. Only $\mathbf{V}$ instead of the full $\mathbf{R}$ is used in equation (2.28). This is a simplification that saves calculation time without deteriorating the fit result, as described in [3].

To arrive at probabilities the ϕ_i are normalised:

$$w_i = \frac{\phi_i}{\Phi + \Phi_{\mathrm{cut}}} \quad \text{with } \Phi = \sum_i \phi_i \text{ and } \Phi_{\mathrm{cut}} = \sum_i \phi_{\mathrm{cut},i} \tag{2.29}$$

The definition of $\phi_{\text{cut},i}$ is

$$\phi_{\text{cut},i} = \frac{1}{(2\pi)^{m/2}\sqrt{\det(T_I \mathbf{V}_i)}} \exp\left(-\frac{c_m(\alpha)}{2T_I}\right), \tag{2.30}$$

where $c_m(\alpha)$ is the $(1-\alpha)$-quantile of the χ^2 function with $\text{ndf} = m$ degrees of freedom. For example $c_2(0.001) = 13.81551$. The cut-off value $\phi_{\text{cut},i}$ forces the weight to be close to 0 if the probability of a hit falls below a chosen value. The presence of the $\phi_{\text{cut},i}$ in equation (2.29) is the formal reason that measurements without competition have probabilities slightly smaller than one. Larger $c_m(\alpha)$ values will lead to systematically lower weights – or stronger suppression of hits – if all other parameters stay the same.

Note that the index k was omitted in equations (2.28), (2.29) and (2.30) to increase readability, but of course the w_i need to be calculated in every measurement layer separately.

If a measurement for the DAF is defined in the following way

$$\overline{\boldsymbol{m}}_k = \overline{\mathbf{V}}_k \sum_i w_{k,i} \mathbf{V}_{k,i}^{-1} \boldsymbol{m}_{k,i} \tag{2.31}$$

with the covariance matrix

$$\overline{\mathbf{V}}_k = \left(\sum_i w_{k,i} \mathbf{V}_{k,i}^{-1}\right)^{-1}, \tag{2.32}$$

one can simplify the DAF Kalman update step defined in equation (2.25) and (2.26) and simply use the original Kalman update formulas (2.10) and (2.9). Basically the competing measurements are now treated by the update step as a single measurement formed by a weighted mean. This increases code reusability when an implementation of the Kalman filter is already available. Note that equations (2.31) and (2.32) allow every competing measurement $\boldsymbol{m}_i$ to have a different $\mathbf{V}_i$, while equations (2.25) and (2.26) assume they have the same $\mathbf{V}$.

An example of a convergence criterion to decide when to stop the iteration over I is

$$\max_{k,i} |w_{k,i,I} - w_{k,i,I-1}| < a \tag{2.33}$$

A suitable value of a can be determined by simulation studies.

2.2.6 Testing the Deterministic Annealing Filter (DAF)

The DAF tunes the state estimation of the current iteration with the weights calculated from the measurement/smoothed state residuals of the previous iteration. Therefore one cannot expect the same mathematical properties from the total χ^2s and p-values after tracks were fitted with the DAF. In particular the p-values distribution of a perfectly fitted track sample will not be flat any more. The traditional track fit quality indicator “p-value” therefore loses some of its appeal and other DAF specific quality indicators like the weight distribution might be better.

If one does want to calculate the total χ^2 and ndf values from DAF results one has to take the weights w associated with every measurement into account:

$$\chi^2_{\mathrm{tot}} = \sum_k \sum_i w_{k,i} (\boldsymbol{m}_{k,i} - \mathbf{H}_k \boldsymbol{x}_{k|k-1})^{\mathrm{T}} (\mathbf{V}_{k,i} + \mathbf{H}_k \mathbf{C}_{k|k-1} \mathbf{H}_k^{\mathrm{T}})^{-1} (\boldsymbol{m}_{k,i} - \mathbf{H}_k \boldsymbol{x}_{k|k-1}) \quad (2.34)$$

$$\mathrm{ndf} = -\dim(\boldsymbol{x}) + \sum_k \sum_i w_{k,i} \dim(\boldsymbol{m}_{k,i}) \quad (2.35)$$

All the symbols in equations (2.34) and (2.35) have the same meaning as in subsection 2.2.5: index k denotes a measurement layer and index i the competing hits in one layer k. Note that ndf is not a non-negative integer any more, but a non-negative real number. Again with the help of the χ^2 cumulative distribution function one can transform the χ^2_{tot} and ndf into a p-value. One has to take care to use an implementation of χ^2 cumulative distribution function that takes non-integer ndf as arguments, which is not a problem as the χ^2-distribution is special case of the Γ distribution.

As mentioned before, the final p-value distribution is not uniform any more. In fact, there tends to be an excess of p-values around the mean value of 0.5 and fewer p-values at the edges (0 and 1). See figure 5.21 as an example. Very small p-values will still indicate a bad fit result, as will a small value of ndf. Figure 5.21 actually shows the p-values of a vertex fit but the algorithm used is essentially the DAF scheme adapted to vertex fitting.

2.3 Track model and material effects

2.3.1 Equations of motion

A charged particle moving in a magnetic $\boldsymbol{B}(\boldsymbol{r})$ feels the Lorentz force. Its motion is therefore described by the differential equation

$$\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = \kappa q \boldsymbol{v}(t) \times \boldsymbol{B}(\boldsymbol{r}(t)), \quad (2.36)$$

where $q = Q/e$ is the charge in units of e, $\boldsymbol{r}$ is the position vector, $\boldsymbol{p} = \gamma m \boldsymbol{v}$ is the relativistic momentum, and $\kappa = 0.29979$ is a proportionality factor necessary if $\boldsymbol{B}$ is in the SI unit (T) while $\boldsymbol{p}$ is in units of GeV. When $\boldsymbol{B}$ is known, the six initial values $\boldsymbol{r}_0$ and $\boldsymbol{v}_0$ determine the trajectory of the particle. If the initial parameter are defined on a surface, for instance on a plane with know position and orientation, only five degrees of freedom are sufficient to fully define the particle trajectory. These parameter do not need to be $\boldsymbol{r}$ and $\boldsymbol{v}$, because equation (2.36) can be reformulated to depend on other geometrical quantities.

In the case of a homogeneous magnetic field $\boldsymbol{B}_0$ and the absence of material, the analytical solution of equation (2.36) is a helix. The presence of thin scatterers leads to a continuous sequence of helices. In the case of an arbitrary magnetic field and material distribution there is no analytic solution and a numerical solver has to be applied. For this purpose the Runge-Kutta-Nyström algorithm, a 4th order Runge-Kutta solver for second order differential equations, is often applied in track reconstruction software.

When the general form of equation (2.36) is solved, one can derive a propagator $\boldsymbol{F}_k(\boldsymbol{x}_{k-1})$ as introduced in equation (2.4). While the propagator of the state vector needed for the Kalman filter procedure as shown in equation (2.7) can be the non-linear function $\boldsymbol{F}_k(\boldsymbol{x}_{k-1})$, the propagation of the covariance matrix shown in equation (2.8) $\boldsymbol{F}_k(\boldsymbol{x}_{k-1})$ requires a linear approximation by a first-order Taylor expansion:

$$\mathbf{F}_k = \left.\frac{\partial \boldsymbol{x}_k}{\partial \boldsymbol{x}_{k-1}}\right|_{\boldsymbol{x}_{k-1}=\boldsymbol{x}_{k-1,0}} \tag{2.37}$$

$\mathbf{F}_k$ is the first order derivative or Jacobian matrix of $\boldsymbol{F}_k(\boldsymbol{x}_{k-1})$ at the expansion point $\boldsymbol{x}_{k-1,0}$.

The choice of the track parameters $\boldsymbol{x}$ depends on many factors, such as detector geometry, quality of the linear expansion, computational efficiency, or interfaces to numerical libraries. Another general requirement is the need for approximately normally distributed estimates. This means that the curvature is a better choice than the radius or the inverse momentum is better that the momentum itself. Also different parameterisations for different parts of the track reconstruction are possible. For the update step of the Kalman filter the number of parameters must be the minimal possible number or else the covariance matrices would not be invertible. As mentioned this number is five for a helix restricted to pass through known measurement planes.

2.3.2 Material effects

When particles pass matter their kinetic energy is reduced and the direction of their trajectory is changed. It is important for a correct track fit to incorporate these effects into the estimation method. From the point of view of track fitting it makes sense to distinguish between deterministic effects which change the state vector $\boldsymbol{x}$, and random effects, which change the covariance matrix $\mathbf{C}$. In the real interaction between a charged particle and the electrons of atoms this would not be a clear distinction as the electromagnetic interaction would be responsible for both the scattering and the excitement or ionisation of the hull electron.

The formulas used to incorporate material effects into the track fit do not describe the quantum interactions of single particles, but the average result of many such interactions for a macroscopic piece of material. In addition, they are semi empirical: while the main relations between variables stated in them are derived from fundamental equations their constants are chosen to fit experimental results.

Electro-magnetic multiple scattering

Electro-magnetic multiple scattering leads to a random change in the direction of the incident particle, for example a non zero scattering angle α. While the change in direction cannot be predicted, the “decrease in knowledge” about the trajectory should be compensated by increasing $\mathbf{C}$. To derive a formula for this increase one starts with the differential cross section $\mathrm{d}\sigma/\mathrm{d}\alpha$ for a single scattering event caused by an atom nucleus, as this is the most dominant contribution for particles heavier than electrons. The electron cloud of the atom only causes a modification of the differential cross section due to screening effects as does the finite size of the nucleus. From this differential cross section the standard deviation of α can be calculated. With the known atomic density of the material and the total cross section the expected number of scattering processes N per path length d in the material can be calculated. As the single scattering angles are independent of each other, the standard deviation of their distribution can be multiplied with $\sqrt{N}$ to obtain the multiple scattering standard deviation.

For sufficiently large N (or d) the projected multiple scattering angle distribution is Gaussian, because of the central limit theorem. The formulas describing this reasoning can be found in [1]. To correct for the non-Gaussian distribution in thin scatterers, Highland [4] introduced a logarithmic correction term to the multiple scattering standard deviation. The resulting formula reads:

$$\sigma_\alpha = \sqrt{\langle\alpha\rangle} = \frac{13.6\,\mathrm{MeV}}{\beta c p} q \sqrt{\frac{d}{X_0}} \left(1 + 0.038 \ln\left(\frac{d}{X_0}\right)\right) , \tag{2.38}$$

where p is the momentum and β is the speed in units of the speed of light c. X_0 is a material constant called the radiation length of the material.

Equation (2.38) is called the Highland formula, in this form taken from [5]. Its parameters are tuned to give a best Gaussian fit of the core of the non-Gaussian multiple scattering angle distribution, where the “core” is defined to consist of 98 % of the probability distribution.

The Highland variance has to be projected onto the the coordinate system used by the tracking algorithm. If the spherical coordinates ϕ and θ are used to describe the direction of the state, the MSC covariance matrix to be added to the state covariance matrix is

$$\mathbf{Q}_{\mathrm{MSC}} = \mathrm{Cov}(\Delta\phi, \Delta\theta) = \sigma_\alpha^2 \begin{pmatrix} (\sin\theta)^{-2} & 0 \\ 0 & 1 \end{pmatrix} . \tag{2.39}$$

If the direction cosines $a_x = \mathrm{d}x/\mathrm{d}s$ and $a_y = \mathrm{d}y/\mathrm{d}s$ are used, the MSC covariance matrix is

$$\mathbf{Q}_{\mathrm{MSC}} = \mathrm{Cov}(\Delta a_x, \Delta a_y) = \sigma_\alpha^2 \begin{pmatrix} (1-a_x)^2 & -a_x a_y \\ -a_x a_y & (1-a_y)^2 \end{pmatrix} . \tag{2.40}$$

Note that both formulas for $\mathbf{Q}_{\mathrm{MSC}}$ neglect the increase of the uncertainties of the spacial coordinates (e.g. x and y) but they can be found in [1] for example.

Electro-magnetic energy loss

The standard way of calculating the loss of momentum Δp due to excitation and ionisation of material atoms by an incident particle uses the Bethe–Bloch formula:

$$\Delta p = \int_0^d = \frac{\mathrm{d}p}{\mathrm{d}s}\mathrm{d}s = \int_0^d \frac{1}{\beta}\frac{\mathrm{d}E}{\mathrm{d}s}\mathrm{d}s = \int_0^d \frac{K}{\beta^3}\left(\ln\left(\frac{2m_\mathrm{e}c^2\beta^2\gamma^2}{I}\right) - \beta^2\right)\mathrm{d}s, \tag{2.41}$$

where K is a material constant, m_e is the electron mass, γ is the Lorentz factor and I the average ionisation potential (also a material dependent constant). For thin material layers the integral $\int_0^d \ldots \mathrm{d}s$ can be replaced by its linear approximation $\ldots \Delta s$ with Δs being simply the absolute of the difference of the vector marking entry and exit.

Δp is the average momentum loss. The actual momentum loss is Landau distributed around Δp. To account for the increased uncertainty of p an energy loss variance can be calculated and added to the state covariance matrix $\mathbf{C}$.

Bremsstrahlung

Bremsstrahlung is electromagnetic radiation caused by changes in the velocity of charged particles. In matter this velocity change is caused by electromagnetic fields surrounding the material atoms and results in the emittance of a bremsstrahlung photon by the incident particle. As bremsstrahlung is caused by changes in velocity it is clear that light particles will lose more energy due to bremsstrahlung than heavier ones when passing the same material.

Because energy loss by bremsstrahlung is roughly proportional to the energy of the total energy of the incident particle, but energy loss by ionisation is only proportional to $\ln E$ for higher energies, one can define a critical energy E_c where the effects are equal. A very rough estimation of E_c for e± in solid or liquid material is $E_\mathrm{c} = \frac{610}{Z+1.24}\,\mathrm{MeV}$ [5].

Among the set of typical particles and typical energy ranges in high energy physics experiments only e± will suffer from strong bremsstrahlung, while it is negligible for the other particles passing tracking detectors. This fact is also illustrated in figure 2.2 showing bremsstrahlung (called "radiative" in figure 2.2) dominating the energy loss for muons only above a momentum of 500 GeV.

The Bethe–Heitler model of bremsstrahlung uses the following distribution for the fraction of remaining energy ($a = 1 - \Delta E/E$) after the material is crossed:

$$f(a) = \frac{(-\ln a)^{b-1}}{\Gamma(b)}, \tag{2.42}$$

where $b = d/(X_0 \ln(2))$ and Γ is the Gamma function. As in equation 2.38 d/X_0 denotes material thickness in units of radiation length.

Based on (2.42) and the high energy approximation $p \approx E$ one can derive

$$\mathrm{mean}(\Delta p) = p(2^{-b} - 1) \quad \text{and} \tag{2.43}$$

$$\mathrm{var}(\Delta p) = p^2(3^{-b} - 4^{-b}). \tag{2.44}$$

Simply subtracting mean(Δp) from the track state $\boldsymbol{x}$ and adding var(Δp) to the track covariance $\mathbf{C}$ during the propagation step in a Kalman filter corresponds to a Gaussian approximation to the highly non-Gaussian distribution (2.42). If this approximation is not good enough, more sophisticated methods are available [6].

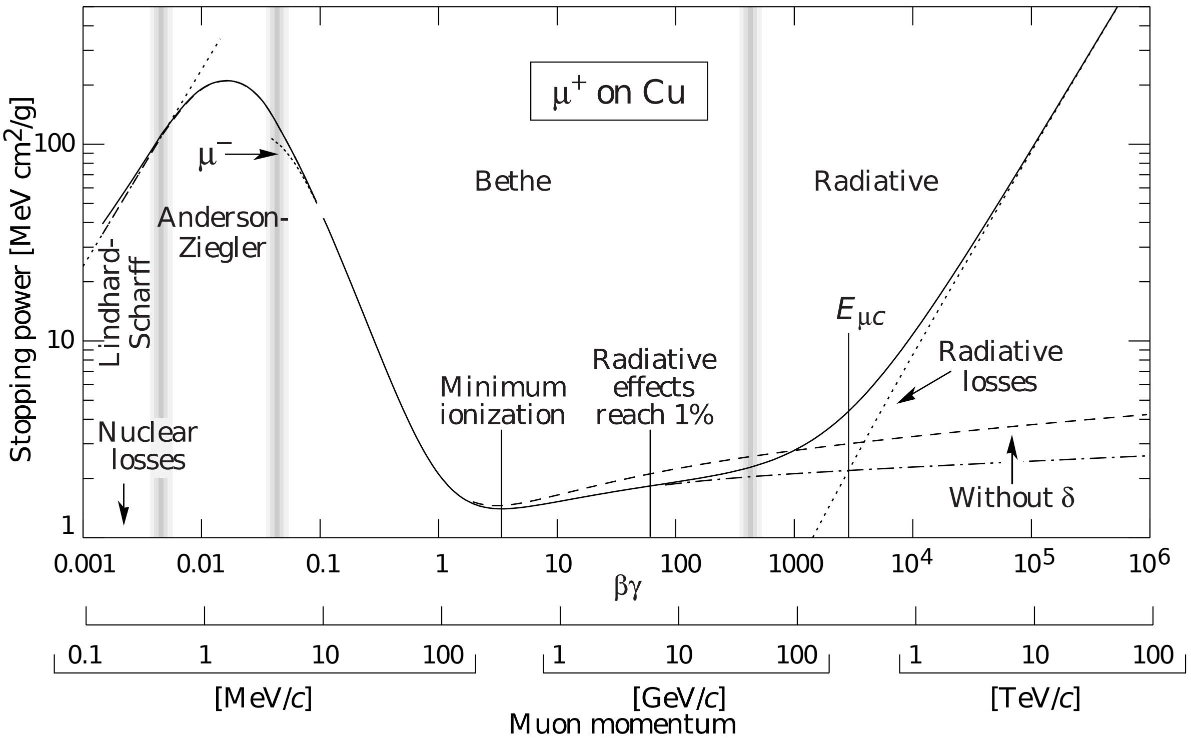


Figure 2.2: Stopping power of copper acting on incident muons depending on the muon momentum. The material effects which dominate at a specific momentum scale are named in the figure. Figure taken from [5].

Other effects

For charged particles electro-magnetic effects are by far the most dominant. While hadronic particles additionally interact via the strong force with material, these occur much less frequent than electromagnetic interactions. On the other hand, because of the short range nature of the strong force, they do they have very strong effects on the trajectory if they occur. For this reason hadronic interactions are not considered by most tracking algorithms. The effort to treat them correctly would be large yet they affect relatively few particles.

Momentum dependency of material effects

As one can see in equations (2.41) and (2.38) the strength of material effect grows with decreasing momentum of the incident particle. Figure 2.2 shows this rise in case of the Bethe–Bloch formula (2.41) applied to muons in copper.

2.4 Vertexing

After the reconstruction of tracks it must be determined which tracks originate at the same point. These points are called vertices, and have to be reconstructed, too. A vertex is called “primary” when it is the point of the collision of beam particles; it is called “secondary” when it is the point of a particle decay. Vertex reconstruction has many similarities to track reconstruction. One can also divide the process into a finding step, i.e. the partition of the set of track into subsets of tracks originating at the same point, and a fitting step in which the actual common vertex position for each of the subsets is estimated.

The vertex finding can be done with generic clustering algorithms, topological methods or iterated vertex fitters. In the last option there are no distinct finding and fitting steps as all tracks are fitted to a common vertex, then all tracks that are outliers with respect to this vertex are removed and put into a new set that is fitted again. This continues until not enough tracks for a new vertex fit are left.

The Kalman filter is – as for track fitting – today’s standard technique to estimate the optimal vertex parameters. Instead of position measurements, as in the case of track fitting, an update now adds the track parameters of one track to the estimated vertex position. Again the Kalman filter can be made robust to some degree when tracks leading to large χ^2 increments are removed from the sample. The p-value from the total χ^2 of the Kalman vertex fit is again a widely used indicator of fit quality. Every track adds two degrees of freedom. Information with an ndf of three is the minimum necessary to fit a 3D vertex position.

Vertexing algorithms normally offer the possibility to add prior information of the approximate vertex position. For example, when fitting a primary vertex one would add the 3D location and uncertainty of the beam crossing region.

The vertex fit can be extended by the inclusion of physical constraints. It is then called a kinematic fit. If for example a particle that has decayed is known from the identification of its daughters, their track parameters can be constrained by the invariant mass of their mother particle. This should improve the resolution of the decay vertex position of the mother particle. The standard way of adding mass, energy or momentum constraints is the Lagrange multiplier method.

In recent years new robust vertex reconstruction algorithms were developed. The Adaptive Vertex Fitter (AVF) [7] for example incorporates and annealing scheme and weights inspired by the DAF into an iterated Kalman filter. Similar to the DAF the weights are interpreted as the probability of a track really belonging to the fitted vertex. The AVF, together with the Adaptive Vertex Reconstruction (AVR), is part of the vertexing library RAVE [8].

The AVR finds vertices with the iterated fitter method mentioned above. It inspects the weights to determine which tracks do not belong to the first fitted vertex, assigns them to a new track set, and restarts the fitting procedure with the AVF using the new track set as input.

3 Material estimation

This chapter presents the result of the development of explicit material estimation methods from tracking measurements. Two different methods to estimate the amount of material and its distribution into layers using only hit measurements from tracking detectors were developed. Both methods were implemented and tested in simulation studies and one of them was used to real tracking data from a test beam. The two methods, simulation results and the algorithm's performance on real test beam data were published by the author and his thesis supervisor [9].

3.1 Motivation and Introduction

As it is clear from the formulas in chapter 2 one needs a description of the material obstructing the particle trajectories to get optimal tracking results. Missing or erroneous information about the material will deteriorate the estimation of the track parameters. As part of this thesis a method was developed that explicitly estimates the material in a tracking detector using only its position measurements. Such a material estimator can be useful in several scenarios.

One is the improvement of simplified tracking geometries. For performance reasons track reconstruction often does not uses the full detector geometry as used by the full simulation, but some simplified description. These simplified geometry models are normally made by transforming complex geometry details with different material budget into some kind of average material layers. The exact details of the simplification are often decided by guesses and assumptions of the responsible person. With a material estimation method one can check which one of different material descriptions is best suited to the requirements of optimal track reconstruction.

Another scenario is the existence of material with unknown budget in the tracking detector. This can be the case in when the performance of a new type of sensor is evaluated in a beam test, where there might be no or insufficient knowledge of the material budget of the support structures or cooling devices.

As an assumption about the material budget has to be used in the reconstruction of the tracks, the estimation method is necessarily iterative. If the unknown thickness of the material layers is considered as a latent variable, the iteration can be interpreted as an expectation-maximization (EM) algorithm [10]. In this interpretation the expectation step corresponds to the estimation of the thickness, and the maximization step to the estimation of the track parameters, using the most recent estimate of the thickness.

The methods proposed here use charged tracks that interact with the material of the detector only by multiple scattering and energy loss by ionization. The methods can and should be regarded as a complement to other approaches, which use pair production of electrons or strong interactions of hadrons for obtaining information about the distribution and amount of material present in a detector. The main restriction of the approach is the assumption that the material is concentrated in the measurement layers.

3.2 Estimation Methods

The basic idea of material estimation is to exploit the effects of the material on the particle trajectories. The following studies use a cylindrical detector model in a constant magnetic field. The track model is therefore a helix with five parameters to describe the trajectory between material layers. The parameters are $(\mathit{\Phi}, z, \theta, \beta, \kappa)$, where $\mathit{\Phi}$ is the azimuth angle of a point on the track, z is the Cartesian coordinate parallel to the beam, θ is the polar angle between the momentum vector and the z-axis, $\beta = \phi - \mathit{\Phi}$, where ϕ is the azimuthal angle of the momentum vector, and κ is the curvature of the projection of the helix on the x-y-plane. The estimation algorithm is of course independent of the particular track model and can easily be reformulated in a different coordinate system.

The mapping of the track parameters onto the measurements of one layer $\boldsymbol{m}_k$ is given by

$$\mathbf{H}_k = \begin{pmatrix} R_k & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \tag{3.1}$$

where R_k is the radius of the kth layer.

Two material effects are used in this investigation:
(i) multiple scattering, as described by the Highland formula [4, 5]:

$$\begin{aligned} \sigma_{\mathrm{ms},\theta} &= \frac{0.015E}{p^2}\sqrt{\frac{X}{\sin(\theta)\cos(\beta)}}\left(1 + 0.038\ln\left(\frac{X}{\sin(\theta)\cos(\beta)}\right)\right), \\ \sigma_{\mathrm{ms},\beta} &= \frac{\sigma_{\mathrm{ms},\theta}}{\sin(\theta)}, \end{aligned} \tag{3.2}$$

where X is the nominal thickness of the material traversed in units of radiation lengths, E the total particle energy and p the particle momentum; and
(ii) energy loss, as described by the Bethe–Bloch formula [5]:

$$\Delta E(E) = \frac{n_\mathrm{e} e^4}{4\pi m_\mathrm{e}\epsilon_0^2}\left(\frac{1}{1 - \frac{m^2}{E^2}}\right)\left(\ln\left(\frac{2m_\mathrm{e}(E^2 - m^2)}{Im^2}\right) + \frac{m^2}{E^2} - 1\right)\Delta x, \tag{3.3}$$

where n_e is electron volume density, e is elementary charge, ϵ_0 is the electric constant, m is the mass of the passing particle, m_e is the electron mass and I the mean excitation potential. Equations (3.2) and (3.3) use SI units, with the exception of GeV for E, GeV/c for p and GeV/c^2 for m. Our method only estimates the amount of material. The type of material must be known to the algorithm.

The Highland formula in equation (3.2) appears a bit different from equation (2.38) because in equation (3.2) the speed of light c was set to 1, a slightly different empirical factor was used (15 MeV instead of 13.6 MeV) and the projection onto the coordinate system (equation (2.39)) was already applied. Also the Bethe–Bloch formula (3.3) is the same as in equation (2.41) just reformulated in exactly the way it was implemented in the material estimation program described in section (3.3.1).

The estimation of the layer thicknesses $\boldsymbol{X}$ from reconstructed track data is a non-linear problem because the track reconstruction depends on the knowledge of $\boldsymbol{X}$. A common approach to this kind of problem is to apply an iteration scheme that resembles an EM algorithm [10]. It starts with a prior guess of $\boldsymbol{X}_0$, then uses a linear estimator to get $\boldsymbol{X}_1$. This is reiterated until a convergence criterion is met. The linear estimator is derived by Taylor-expanding the non-linear model to first order.

Two linear estimators were developed to work in conjunction with the iteration scheme. The first one is a global linear estimator that estimates directly all $\boldsymbol{X}$, all scattering angles $\boldsymbol{\alpha}$ and all their correlations. It can be considered as an extended track fit with breakpoints [11, 12]. The second one is an indirect estimation of the material effects in every layer from energy loss and multiple scattering, by combining a forward and backward Kalman filter [13].

3.2.1 Global linear estimator

The global linear estimator uses a linearised model that models the position measurements as a function of the initial track parameters $\boldsymbol{p}_1$, the thicknesses $\boldsymbol{X}$, and the actual scattering angles $\boldsymbol{\alpha}$. It incorporates prior information about the scattering angles by using their mean value (zero) as a virtual measurement and their variance in the covariance matrix of the model. The model therefore reads:

$$\begin{pmatrix} \boldsymbol{m} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{B} & \mathbf{A} & \mathbf{D} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix} \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{X} \\ \boldsymbol{\alpha} \end{pmatrix} + \boldsymbol{\epsilon} \quad \text{with } \operatorname{Cov}(\boldsymbol{\epsilon}) = \mathbf{E} = \begin{pmatrix} \mathbf{V} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} \end{pmatrix}. \tag{3.4}$$

The left hand side of equation (3.4) are the position measurements $\boldsymbol{m}$, consisting of the $(R\Phi, z)$ measurements of the L detector layers, and the expectation values of the scattering angles $\boldsymbol{\alpha}$, which are all equal to zero. The covariance matrix $\mathbf{E}$ of the error vector $\boldsymbol{\epsilon}$ is diagonal because both $\mathbf{V}$, which holds the variances of the $(R\Phi, z)$ measurements, and $\mathbf{Q}$, which holds the variances of the multiple scattering angles, are diagonal.

The parameters to be estimated are (i) the five initial track parameters $\boldsymbol{p}_1$, (ii) the layer thicknesses $\boldsymbol{X}$, and (iii) the scattering angles $\boldsymbol{\alpha}$ in all layers. By convention a measurement is taken at the beginning of the material of a layer therefore there is no measurement after the material of the last layer and the its material cannot be estimated. $\mathbf{B}$, $\mathbf{A}$ and $\mathbf{D}$ are the Jacobian matrices of $\boldsymbol{m}$ with respect to the estimated parameters:

$$\mathbf{B} = \frac{\partial \boldsymbol{m}}{\partial \boldsymbol{p}_1}, \quad \mathbf{A} = \frac{\partial \boldsymbol{m}}{\partial \boldsymbol{X}}, \quad \mathbf{D} = \frac{\partial \boldsymbol{m}}{\partial \boldsymbol{\alpha}}. \tag{3.5}$$

To get expressions for $\mathbf{B}$, $\mathbf{A}$, and $\mathbf{D}$ that can be implemented one has to write them as products of partial derivatives. $\mathbf{B}$ is assembled from L submatrices $\mathbf{b}_k \in \mathbb{R}^{2\times5}$, $k =$

$1, \dots, L$:

$$\mathbf{b}_k = \frac{\partial \boldsymbol{m}_k}{\partial \boldsymbol{p}_1} = \frac{\partial \boldsymbol{m}_k}{\partial \boldsymbol{p}_k} \prod_{i=k-1}^{1} \frac{\partial \boldsymbol{p}_{i+1}}{\partial \boldsymbol{p}_{\mathrm{ME},i}} \frac{\partial \boldsymbol{p}_{\mathrm{ME},i}}{\partial \boldsymbol{p}_i}, \tag{3.6}$$

where $\boldsymbol{p}_i$ is the track state in layer i before material effects and $\boldsymbol{p}_{\mathrm{ME},i}$ is the track state after material effects. $\partial \boldsymbol{m}_k / \partial \boldsymbol{p}_k$ is equal to $\mathbf{H}_k$, the mapping of the track state on the measurements; $\partial \boldsymbol{p}_{i+1} / \partial \boldsymbol{p}_{\mathrm{ME},i}$ is equal to $\mathbf{F}_i$, the propagation matrix extrapolating the track parameter from layer i to layer $i+1$; and

$$\frac{\partial \boldsymbol{p}_{\mathrm{ME},i}}{\partial \boldsymbol{p}_i} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \frac{\partial \kappa_{\mathrm{ME},i}}{\partial \kappa_i} \end{pmatrix}, \tag{3.7}$$

because κ is the only track parameter that is changed by the material effects. Note that the linear model is computed from a reference track that does not incorporate multiple scattering.

A is assembled from $L(L-1)$ submatrices $\boldsymbol{a}_{i,j} \in \mathbb{R}^{2\times 1}$, $i = 1, \dots, L$, $j = 1, \dots, L-1$:

$$\boldsymbol{a}_{i,j} = \frac{\partial \boldsymbol{m}_i}{\partial X_j} = \begin{cases} \frac{\partial \boldsymbol{m}_i}{\partial \boldsymbol{p}_i} \left(\prod_{k=i-1}^{j+1} \frac{\partial \boldsymbol{p}_{k+1}}{\partial \boldsymbol{p}_{\mathrm{ME},k}} \frac{\partial \boldsymbol{p}_{\mathrm{ME},k}}{\partial \boldsymbol{p}_k} \right) \frac{\partial \boldsymbol{p}_{j+1}}{\partial \boldsymbol{p}_{\mathrm{ME},j}} \frac{\partial \boldsymbol{p}_{\mathrm{ME},j}}{\partial X_j} & \text{if } i > j+1, \\ \frac{\partial \boldsymbol{m}_i}{\partial \boldsymbol{p}_i} \frac{\partial \boldsymbol{p}_i}{\partial \boldsymbol{p}_{\mathrm{ME},j}} \frac{\partial \boldsymbol{p}_{\mathrm{ME},j}}{\partial X_j} & \text{if } i = j+1. \\ 0 & \text{otherwise.} \end{cases} \tag{3.8}$$

$\partial \boldsymbol{m}_i / \partial \boldsymbol{p}_i$ and $\partial \boldsymbol{p}_{k+1} / \partial \boldsymbol{p}_{\mathrm{ME},k}$ are the same as in **B**, and

$$\frac{\partial \boldsymbol{p}_{\mathrm{ME},j}}{\partial X_j} = \begin{pmatrix} 0 & 0 & 0 & 0 & \frac{\partial \kappa_{\mathrm{ME},j}}{\partial X_j} \end{pmatrix}^{\mathrm{T}}. \tag{3.9}$$

D is assembled from $L(L-1)$ submatrices $\mathbf{d}_{i,j} \in \mathbb{R}^{2\times 2}$, $i = 1, \dots, L$, $j = 1, \dots, L-1$:

$$\mathbf{d}_{i,j} = \frac{\partial \boldsymbol{m}_i}{\partial \boldsymbol{\alpha}_j} = \begin{cases} \frac{\partial \boldsymbol{m}_i}{\partial \boldsymbol{p}_i} \left(\prod_{k=i-1}^{j+1} \frac{\partial \boldsymbol{p}_{k+1}}{\partial \boldsymbol{p}_{\mathrm{ME},k}} \frac{\partial \boldsymbol{p}_{\mathrm{ME},k}}{\partial \boldsymbol{p}_k} \right) \frac{\partial \boldsymbol{p}_{j+1}}{\partial \boldsymbol{p}_{\mathrm{ME},j}} \frac{\partial \boldsymbol{p}_{\mathrm{ME},j}}{\partial \boldsymbol{\alpha}_j} & \forall i > j+1 \\ \frac{\partial \boldsymbol{m}_i}{\partial \boldsymbol{p}_i} \frac{\partial \boldsymbol{p}_i}{\partial \boldsymbol{p}_{\mathrm{ME},j}} \frac{\partial \boldsymbol{p}_{\mathrm{ME},j}}{\partial \boldsymbol{\alpha}_j} & \forall i = j+1 \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

$\partial \boldsymbol{m}_i / \partial \boldsymbol{p}_i$ and $\partial \boldsymbol{p}_{k+1} / \partial \boldsymbol{p}_{\mathrm{ME},k}$ are the same as in **B** and **A**, and

$$\frac{\partial \boldsymbol{p}_{\mathrm{ME},j}}{\partial \boldsymbol{\alpha}_j} = \begin{pmatrix} 0 & 0 & 1 & 0 & -\kappa_j \frac{\cos(\theta_j)}{\sin(\theta_j)} \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}^{\mathrm{T}}. \tag{3.11}$$

To fully assemble $\mathbf{M}$ one therefore needs to calculate two Jacobian matrices numerically per track and layer (equations (3.7) and (3.9)), because all the matrices $\mathbf{F}_k$ are already calculated during the generation of the reference track.

Finally, the least-squares estimator is given by

$$\begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{X} \\ \boldsymbol{\alpha} \end{pmatrix} = \left(\mathbf{M}^{\mathrm{T}}\mathbf{E}^{-1}\mathbf{M}\right)^{-1}\mathbf{M}^{\mathrm{T}}\mathbf{E}^{-1}\begin{pmatrix} \boldsymbol{m} \\ \mathbf{0} \end{pmatrix} \quad \text{with } \mathbf{M} = \begin{pmatrix} \mathbf{B} & \mathbf{A} & \mathbf{D} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}. \tag{3.12}$$

3.2.2 Forward-backward Kalman filter

To estimate the layer thickness with a Kalman filter, first one has to estimate the track state in every layer before the material effects with the forward filter, and after the material effects with the backward filter. Then the differences $\Delta\beta$, $\Delta\theta$ and ΔE have to be calculated. It is important that in the layer in which the current difference is computed the update step is done either by the forward or by the backward filter, so that every measurement is used exactly once when forming the difference. In the current implementation of the estimator the forward filter includes the measurement of the current layer.

The differences in energy, θ and β can be used to get three largely independent estimates of X. In order to compute ΔE, one has to transform the two κ values to the momentum p and then to the energy E. Because a linear approximation of the highly non-linear transformation $p = h(\kappa) = \left|\frac{B_z}{\kappa \sin(\theta)}\right|$ introduces a bias in p, p was Taylor-expanded up to second order in κ:

$$p(\kappa) \approx h(\kappa_0) + h'(\kappa_0)(\kappa - \kappa_0) + \frac{1}{2}h''(\kappa_0)(\kappa - \kappa_0)^2. \tag{3.13}$$

If κ_0 is chosen as the local estimate of κ, which is the mean of its posterior distribution, taking the expectation in equation (3.13) gives the second-order transformation

$$p = \frac{B_z}{|\kappa|\sin(\theta)} + \frac{B_z\sigma_\kappa^2}{\sin(\theta)|\kappa^3|}. \tag{3.14}$$

The second-order variance can be computed by using the relation $\sigma_p^2 = \mathrm{E}(p^2) - \mathrm{E}(p)^2$. If equation (3.13) is substituted in this relation, in the Gaussian assumption the variance turns out to be

$$\sigma_p^2 = \left(\frac{B_z}{\sin(\theta)}\right)^2 \frac{\sigma_\kappa^2}{\kappa^4} + 2\left(\frac{B_z}{\sin(\theta)}\right)^2 \frac{\sigma_\kappa^4}{\kappa^6} + \left(\frac{B_z\cos(\theta)}{\kappa\sin^2(\theta)}\right)^2 \sigma_\theta^2, \tag{3.15}$$

where the last term is the contribution of θ to the variance of p. The final transformation from p to E uses standard linear error propagation:

$$E = \sqrt{p^2 + m^2}, \quad \sigma_E^2 = p\left(p^2 + m^2\right)^{-\frac{1}{2}}\sigma_p^2. \tag{3.16}$$

The variance of the difference ΔE is equal to $\sigma^2_{\Delta E} = \sigma^2_{E,\mathrm{f}} + \sigma^2_{E,\mathrm{b}}$, i.e., the sum of the variances of the forward and the backward filter estimates. At this point the Bethe–Bloch formula (equation (3.3)) is used to compute Δx, the layer thickness in meters. E, the particle energy before the material effects, is also needed in equation (3.3); therefore the correlation between ΔE and E has to be taken into account when calculating the variance on Δx:

$$\sigma^2_{\Delta x} = \begin{pmatrix} \frac{\mathrm{d}\Delta x}{\mathrm{d}\Delta E} & \frac{\mathrm{d}\Delta x}{\mathrm{d}E} \end{pmatrix} \mathrm{Cov}((\Delta E, E)) \begin{pmatrix} \frac{\mathrm{d}\Delta x}{\mathrm{d}\Delta E} \\ \frac{\mathrm{d}\Delta x}{\mathrm{d}E} \end{pmatrix}, \tag{3.17}$$

with

$$\mathrm{Cov}((\Delta E, E)) = \begin{pmatrix} \sigma^2_{\Delta E} & \sigma^2_{E,\mathrm{f}} \\ \sigma^2_{E,\mathrm{f}} & \sigma^2_{E,\mathrm{f}} \end{pmatrix}. \tag{3.18}$$

After a final transformation from Δx to X and $\sigma_{\Delta x}$ to σ_X, the estimated thickness from N tracks is computed by the weighted mean

$$\bar{X} = \frac{\sum_i^N \frac{X_i}{\sigma^2_{X,i}}}{\sum_i^N \frac{1}{\sigma^2_{X,i}}}.$$

For the multiple-scattering based estimation of X a maximum-likelihood estimator was developed. In each layer, the observed angle $\Delta\theta$ between the track segments estimated by the forward and the backward filter, respectively, is assumed to be normally distributed with mean 0 and a variance σ^2. This can be written as

$$f(\Delta\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\Delta\theta^2}{2\sigma^2}\right), \tag{3.19}$$

where σ^2 is the sum of σ^2_{ms}, the multiple scattering variance given by the Highland formula (equation (3.2)) in the layer where X is to be estimated, and σ^2_{ext}, the variance of all other error sources contributing to the error of $\Delta\theta$. So σ^2_{ext} is just the sum of the variances of θ from the forward and the backward filter therfore it includes all errors form all material effects and measurements in all layers besides the contribution of the MSC error of the current material layer. The log-likelihood function therefore reads:

$$\ln L(X) = \ln \prod_i f_i = K + \sum_i \left[-\ln(\sigma^2_{\mathrm{ms},i} + \sigma^2_{\mathrm{ext},i}) - \frac{\Delta\theta_i^2}{\sigma^2_{\mathrm{ms},i} + \sigma^2_{\mathrm{ext},i}} \right], \tag{3.20}$$

where the product runs over all reconstructed tracks. $\sigma^2_{\mathrm{ms},i}$ is replaced by the Highland formula, and the log-likelihood function is maximised with respect to X. K includes all terms in $\ln L(X)$ that do not depend on X and therefore are irrelevant for the maximization. Solving the equation $\ln L - \ln L_{\max} + \frac{1}{2} = 0$ for X gives the standard deviation of the maximum-likelihood estimator.

The log-likelihood estimator of X that uses the $\Delta\beta$ is done by repeating this process and replacing $\Delta\theta$ with $\Delta\beta$ and σ^2_{ms} with $\sigma^2_{\mathrm{ms},\beta}$.

3.3 Simulation Experiments

3.3.1 The program

The MATLAB [14] program has three main parts: (i) the "simulation" part to create the true particle trajectories or to generate a reference track from given starting values; (ii) the "estMat" part which iteratively estimates the thicknesses $\boldsymbol{X}$ using a forward-backward Kalman filter; and (iii) the "globalEst" part that implements the global linear regression on the thicknesses $\boldsymbol{X}$ and the scattering angles $\boldsymbol{\alpha}$.

The detector simulation starts with the five true track parameters in the innermost layer and propagates them through the detector. It is assumed that the material to be estimated is concentrated in the measurement layers. In each layer, the energy loss is calculated with the Bethe–Bloch formula (equation 3.3), and random scattering angles are drawn from a normal distribution with a standard deviation according to the Highland formula (equation 3.2). Then the track parameters are propagated to the next layer. As a simplification, the multiple scattering uses the momentum after the full energy loss. The position measurements are smeared by normally distributed errors.

Both the Kalman filter and the global regression model need a reference track. It is constructed by adding random errors to the true state vector in the first layer, and propagating the track through all layers, using the helix model with energy loss but without multiple scattering. Part of the reference track generation is the calculation of the propagation matrices $\mathbf{F}_k$ that are the Jacobian matrices of the non-linear track propagator.

The iteration scheme in "estMat" is dampened in order to avoid large oscillations of the estimated thicknesses. In each iteration, the thickness used as an input for the next iteration is 0.34 times the current estimate plus 0.66 times the previous estimate. Other numerical values for the dampening were tried but this ones lead to a fast convergence without risking a breakdown of the iteration scheme.

3.3.2 Setup

As a test case for the material estimation methods presented above, a simplified model of the CMS silicon tracker was implemented. Table 3.1 shows the parameters of this model. Apart from the X values they were taken from [15]. The thicknesses were chosen to add up to the total amount of material stated in [15], i.e., 0.4 radiation lengths of silicon. One layer was assigned a different value in order to check if the estimation methods could recognize the difference.

The simulated events consist of single randomly chosen π^+ or π^- particles. The initial true track parameters are selected from uniform random distribution, the ranges of which are given in Table 3.2. The transverse momentum p_T was chosen as 1 GeV for all tracks, and the homogeneous magnetic field B_z was set to 4 T.

The results shown in this section use some ideal conditions: perfect alignment, exact knowledge of detector resolution and perfect particle identification. Additional tests to check the influence of wrong particle identification or underestimation of the detector

Layer	R/m	$\sigma_{R\Phi}/(\mu\mathrm{m\,rad})$	σ_z/mm	X/X_0
1	0.044	20	0.02	0.03
2	0.073	20	0.02	0.03
3	0.102	20	0.02	0.03
4	0.225	23	0.23	0.03
5	0.339	23	0.23	0.03
6	0.4185	35	34	0.04
7	0.498	35	34	0.03
8	0.608	53	0.53	0.03
9	0.692	53	0.53	0.03
10	0.78	53	52	0.03
11	0.868	53	52	0.03
12	0.965	35	52	0.03
13	1.08	35	52	0.03

Table 3.1: Parameters used in the simplified CMS silicon tracker model. From left to right: layer number, radius R, resolution in $R\Phi$, resolution in z and the thickness of the layer as a fraction of the radiation length.

	Φ/rad	z/m	θ/rad
min	0	-0.1	$\pi/4$
max	2π	0.1	$\pi/2$

Table 3.2: Range of the uniform distributions that are used to generate the initial track parameters.

resolution were also conducted.

3.3.3 Results from Global linear estimator

Table 3.3 shows the results of the estimation of $\boldsymbol{X}$ with the global estimator. No iteration scheme was used, and the true thicknesses were used as the initial values. Table 3.3 shows that the global estimator, while working in principle, requires a very large number of tracks – in the order of 10^7 – to estimate $\boldsymbol{X}$ with sufficient accuracy.

The final estimate of $\boldsymbol{X}$ is a weighted mean over the estimates from all tracks. Table 3.3 shows the weighted means, their standard error and their standard scores with respect to the true values. The errors scale as expected with the inverse square-root of the number of tracks. With 10^5 tracks the errors are still so large that negative estimates of the thickness occur. With 10^7 tracks the errors are a few percent of the thickness, therefore the estimates can be considered as satisfactory.

An inspection of the joint covariance matrix of all X estimates in table 3.4 reveals very strong correlations between the thicknesses, up to 0.97. Adjacent layers always have negative correlation, so that there is strict alternation between positive and negative

tracks	10^5			10^6			10^7		
layer	X	σ_X	z_X	X	σ_X	z_X	X	σ_X	z_X
3	424	90	1.37	328	29	0.971	286	9	−1.510
4	154	76	−1.93	295	24	−0.196	307	8	0.950
5	576	117	2.36	328	37	0.759	306	12	0.475
6	−81	157	−3.07	344	50	−1.130	397	17	−0.198
7	812	151	3.38	371	48	1.480	301	15	0.088
8	−175	158	−3.02	231	50	−1.380	306	16	0.368
9	819	181	2.87	370	57	1.220	298	18	−0.088
10	−128	182	−2.36	257	57	−0.732	304	18	0.260
11	642	173	1.98	346	55	0.850	309	17	0.531
12	48	145	−1.73	263	46	−0.807	296	15	−0.246

Table 3.3: Results of the estimation of X, its standard error (both in units of $10^{-4}X_0$), and standard scores z_X calculated with the global linear estimator for different track numbers. See table 3.1 for the true values of X.

1	−0.762	0.616	−0.528	0.455	−0.403	0.369	−0.338	0.315	−0.297
−0.762	1	−0.908	0.781	−0.674	0.597	−0.546	0.500	−0.466	0.439
0.616	−0.908	1	−0.935	0.813	−0.723	0.661	−0.605	0.564	−0.532
−0.528	0.781	−0.935	1	−0.950	0.851	−0.779	0.713	−0.664	0.626
0.455	−0.674	0.813	−0.950	1	−0.944	0.868	−0.795	0.741	−0.699
−0.403	0.597	−0.723	0.851	−0.944	1	−0.966	0.893	−0.833	0.786
0.369	−0.546	0.661	−0.779	0.868	−0.966	1	−0.966	0.906	−0.855
−0.338	0.500	−0.605	0.713	−0.795	0.893	−0.966	1	−0.973	0.922
0.315	−0.466	0.564	−0.664	0.741	−0.833	0.906	−0.973	1	−0.977
−0.297	0.439	−0.532	0.626	−0.699	0.786	−0.855	0.922	−0.977	1

Table 3.4: Correlation matrix of the 10 estimated thicknesses (layer 3 to 12) calculated by the global linear estimator using 12500000 tracks

correlations. This explains the alternating signs of the standard scores in table 3.3.

3.3.4 Results from forward-backward Kalman filter

Table 3.5 shows the result of the material estimation with a single step of the forward-backward Kalman with the true thicknesses as the initial values. The true values are reproduced within the statistical errors. Note that the estimator requires two well-defined track segments from both filters, so that there are no reliable estimates in the innermost two and outermost two layers. The fact that there is no estimate of X in layer 11 is only due to the poor resolution in the outer layers. A test run with smaller resolution produced an estimate in layer 11 with the same quality as the other estimates.

tracks	10^4				10^5			
layer	X	σ_X	z_X	r_X	X	σ_X	z_X	r_X
2	287	6	−2.01	−0.043	291	2	−4.40	−0.030
3	300	7	−0.01	−0.000	306	2	2.40	0.018
4	278	8	−2.81	−0.073	302	3	0.65	0.005
5	305	10	0.55	0.018	305	3	1.73	0.018
6	423	9	2.51	0.059	402	3	0.75	0.005
7	309	9	0.92	0.029	303	3	0.97	0.010
8	298	9	−0.19	−0.005	302	3	0.67	0.006
9	313	10	1.40	0.043	307	3	2.31	0.023
10	310	12	0.82	0.032	300	4	0.04	0.000

Table 3.5: Results of the estimation of X its σ_X (both in $10^{-4}X_0$), the standard score z_X, and the relative error $r_X = (X - X_{\mathrm{tr}})/X_{\mathrm{tr}}$ calculated with the forward-backward Kalman estimator for different track numbers. The track parameters used for the material estimation were reconstructed with the true $\boldsymbol{X}$ values. They can be found in Table 3.1

The comparison of Table 3.5 with Table 3.3 shows that the estimator based on the Kalman filter needs much less tracks to achieve the same accuracy. This can be explained by noting that the scattering angles are estimated from two independent track segments, without injecting any prior information about their variance. Additionally only the direct estimation of $\boldsymbol{X}$ was used in the global regression model. Actually, 10^4 tracks of low momentum are sufficient to estimate the thickness to a relative precision of a few percent.

If the starting values of the thickness are different from the true ones, the iteration scheme with dampening has to be applied. A test was performed with starting thicknesses of 2% in all layers. The improvement of the track reconstruction in the first four iteration is illustrated by the distributions of the p-values of the total χ^2 of the forward filter (Figure 3.1). In the course of four iterations, the distribution quickly converges to a nearly perfect uniform one. The estimated thicknesses are shown in Table 3.6. The four iterations took approximately 10 minutes on an Intel Core i3 CPU with 3.33 GHz.

The iteration is stopped as soon as the mean of the total χ^2 and the smoothed χ^2 values of the individual layers stabilise. Further iterations actually tend to degrade the smoothed χ^2-distributions and estimated thicknesses while the total χ^2-distribution and the total material budgets remain stable. The reason for this behaviour is the presence of strong correlations between adjacent layers which are not taken into account in the Kalman filter approach.

The quick stabilization total material budget of all layers is visible in table 3.6 while table 3.7 shows the same for the total χ^2. Table 3.6 also shows how the slow but clear deviation of the estimated individual layer thickness form their true values when continuing the iteration after the total thickness has reached a stable value.

Table 3.8 shows the smoothed χ^2 for the individual layers after 4 iterations and

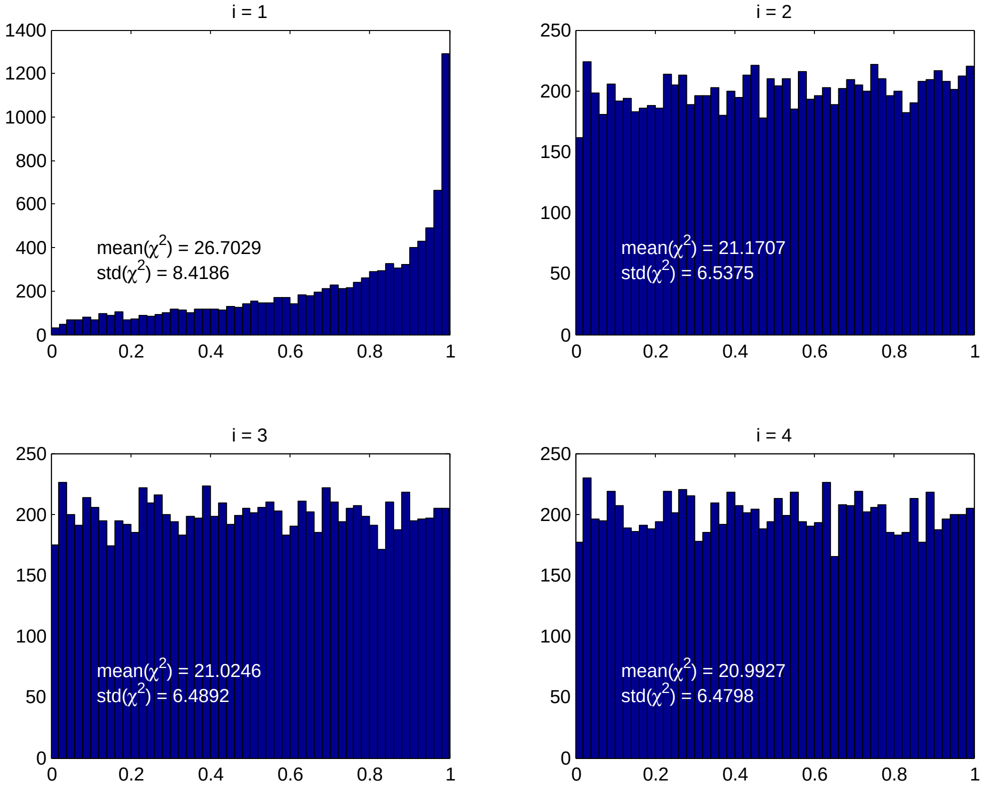


Figure 3.1: Distribution of the p-values of the total χ^2 of the forward filter, for the first four iterations. The p-values should be uniformly distributed.

i \layer	2	3	4	5	6	7	8	9	10	total
0	200	200	200	200	200	200	200	200	200	2900
1	274	303	333	355	345	322	299	283	241	3965
2	287	287	303	365	364	330	318	313	256	4030
3	301	289	276	362	371	324	317	327	268	4043
4	311	303	259	357	377	319	310	332	277	4056
10	269	376	242	327	389	331	299	319	292	4056

Table 3.6: Estimated X values in $10^{-4}X_0$ for the first four and the 10th iteration. Because X cannot be estimated in layers 1, 11, 12 and 13, they are set to the true value. The row $i = 0$ contains the starting values (a-priori guesses) of X. See Table 3.1 for the true values. The sum of all X including the non-estimated ones is also shown.

i	1	2	3	4	10
mean($\chi^2_{\mathrm{tot,f}}$)	26.7029	21.1707	21.0246	20.9927	20.9479
std($\chi^2_{\mathrm{tot,f}}$)	8.4186	6.5375	6.4892	6.4798	6.479
mean($\chi^2_{\mathrm{tot,b}}$)	26.6040	21.1462	21.0059	20.9743	20.9306
std($\chi^2_{\mathrm{tot,b}}$)	8.3311	6.5131	6.4677	6.4588	6.4628

Table 3.7: Total χ^2 values for the first 4 and the 10th iterations from forward and backward filter. They remain nearly constant after the third iteration

layer	1	2	3	4	5	6	7
mean(χ^2_{sm})	2.0062	1.9948	2.0175	2.0676	1.9137	1.9821	1.9544
std(χ^2_{sm})	1.9954	1.9367	1.9766	2.0376	1.8823	2.0067	1.9683
layer	8	9	10	11	12	13	
mean(χ^2_{sm})	1.9428	1.9728	2.0665	2.0266	2.0038	1.9789	
std(χ^2_{sm})	1.9658	1.9815	2.0734	1.9856	1.9956	1.963	

Table 3.8: Mean and standard deviation of the χ^2-distribution created from the smoothed track parameters in every layer after 4 iterations.

layer	1	2	3	4	5	6	7
mean(χ^2_{sm})	2.1167	2.0556	1.87	2.1824	1.986	1.9563	1.9402
std(χ^2_{sm})	2.101	1.9989	1.8273	2.1582	1.9505	1.9802	1.9539
layer	8	9	10	11	12	13	
mean(χ^2_{sm})	1.9878	1.9849	2.0151	2.014	2.0041	1.9837	
std(χ^2_{sm})	2.0119	1.9933	2.0189	1.9728	1.9957	1.9694	

Table 3.9: Mean and standard deviation of the χ^2-distribution created from the smoothed track parameters in every layer after 10 iterations

table 3.9 after 10 iterations. One can see the stronger deviation from the ideal χ^2-distribution with ndf = 2 after 10 iterations in comparison to four iterations. Therefore by using this method one can detect if too many iterations were used without knowing the true thickness values, just by inspecting the smoothed χ^2.

In later extensions of this material estimations method [16] the total amount of material proved to be a good convergence criteria.

Table 3.10 shows the weights used to combine the information from the three different X estimates. Two of them are based on the multiple scattering in θ and β, respectively, and the third one on the change of κ due to energy loss. The table shows the result of the 4th iteration. One can see that most of the information comes from the θ scattering in the inner layers and from β in the outer layers. This is a direct result of the good z resolution of the inner pixel detectors and the bad z resolution of the outer strip detectors.

layer	2	3	4	5	6	7	8	9	10
θ	0.988	0.890	0.629	0.068	0.063	0.070	0.000	0.000	0.000
β	0.012	0.109	0.361	0.893	0.897	0.887	0.963	0.967	0.967
κ	0.000	0.001	0.011	0.039	0.040	0.043	0.037	0.033	0.033

Table 3.10: Weights (normalised inverse variances) of the X estimates in all layers after the 4th iteration. The weights are used to combine information from multiple scattering in θ and β and from energy loss via the change in κ.

The relative contribution of the estimation from energy loss to the weighted mean is very small and never exceeds 5 %. The contribution is even less when using particles with higher momenta. The results shown here would therefore hardly change if the estimation from energy loss would be discarded.

In general the material estimation works better with low momentum tracks because the material effects are stronger. Higher momentum can to some extent be compensated by using more tracks for the estimation. Table 3.11 shows a comparison of (A) the estimated thicknesses of the standard setup, (B) the estimates from 10000 tracks with a $p_\mathrm{T} = 10\,\mathrm{GeV}$, and (C) estimates from 40000 tracks with $p_\mathrm{T} = 10\,\mathrm{GeV}$. The resulting total χ^2 is still perfect with 10000 tracks with $p_\mathrm{T} = 10\,\mathrm{GeV}$, having a mean of 20.9076 and a standard deviation of 6.4097 after four iterations.

It is also interesting to test the effect of wrong assumptions on the detector resolution and on the particle mass. If the detector resolution used in the track reconstruction is 10 % better than the one used to generate the measurements, the average χ^2 is 22.8238, and the standard deviation is 7.0804, somewhat too large. The estimated thicknesses are shown in setup D. If the generated particles are muons and the Kalman filter uses the pion mass instead (setup E), the effect is very small: the average $\langle\chi^2\rangle$ is 20.9645, and the standard deviation is 6.4715.

For the results in this section, σ_θ^2 was neglected in equation 3.15. This term increases σ_p^2 and through error propagation σ_X^2, the variance of the X estimate from energy loss. This effect is visible mostly in the outer layers because of their lower z resolution. As a consequence, the estimate of X from energy loss has even smaller weight, and the results in this section are virtually the same if σ_θ^2 is included in equation 3.15.

3.4 Validation with Real Data

3.4.1 Setup

Thanks to the generous assistance of the ATLAS 3D collaboration and the ATLAS IBL collaboration, it was possible to validate the forward-backward Kalman filter estimator on real data, taken in a test beam with the EUDET telescope. The setup was similar to the one described in [17]. It consisted of three layers of EUDET pixel detectors with known thickness, 3 devices under test plus infrastructure with unknown material

setup \layer		2	3	4	5	6	7	8	9	10
A	X	333	330	225	348	389	310	297	341	293
	std(X)	6.73	7.25	8.23	9.64	9.14	9.59	9.03	9.46	11.8
B	X	502	320	172	345	229	387	470	72.7	720
	std(X)	51.5	52.7	39.5	28.2	28	46.5	49.2	855	191
C	X	274	344	165	371	322	388	213	440	257
	std(X)	25.6	26.1	19.7	13.8	13.3	22.7	25.9	40.5	101
D	X	332	362	214	369	387	341	300	340	306
	std(X)	6.74	7.23	8.3	9.63	9.17	9.55	9.03	9.46	11.7
E	X	331	329	222	347	387	309	296	341	288
	std(X)	6.69	7.21	8.19	9.59	9.09	9.55	8.99	9.41	11.7

Table 3.11: Estimated X values and their error in $10^{-4}X_0$ after four iterations, for different setups (see Table 3.1 for the true values). A is the standard setup. B uses $p_\mathrm{T} = 10\,\mathrm{GeV}$ instead of $1\,\mathrm{GeV}$. C uses 40000 tracks with $p_\mathrm{T} = 10\,\mathrm{GeV}$. In setup D the filter uses a resolution that is 10 % better than the one used in generating the measurements. In setup D the generated particles are muons while the filter assumes the pion mass for reconstruction. In comparison to Table 3.6 this table shows the direct estimation results before they get dampened with the previous estimation.

budget, and three more EUDET pixel layers of the same type as the first three layers. The devices under test were samples of 3D silicon pixel sensors. There was no magnetic field. The beam particles were pions with an energy of 120 GeV. Table 3.12 shows the parameters used to estimate the material budget of the three central layers. It should be stressed that the actual sensors are only a small fraction of the material budget, which is dominated by the infrastructure.

The main difference to the “CMS” setup is the missing magnetic field. As a consequence, the tracks do not have a curvature κ, and only the effect of multiple scattering can be used to estimate the thickness. It should be noted, however, that the contribution of energy loss is almost negligible in any case. The reference track was computed using the first position measurement, the beam direction, and the known beam momentum.

To limit the amount of outliers in the measurements used for the material estimation a preliminary track reconstruction with several cuts was applied. The cuts were tuned to give a well shaped χ^2-distribution after four iterations of the material estimation. This resulted in a sample of 99966 tracks.

As an additional cross-check, a sample of 99966 simulated tracks was generated in the test-beam setup, using the estimated amount of material in the three central layers.

3.4.2 Results

Figure 3.2 shows the development of the p-values of the total χ^2 of the tracks. It is obvious that the initial guess of the material budget is too small. The estimates converge

Layer	z/cm	σ_x/µm	σ_y/µm	$X/(10^{-3}X_0)$
1	0	4.3	4.3	7.33
2	15	4.3	4.3	7.33
3	30	4.3	4.3	7.33
4	39.27	114	14.4	?
5	44.17	111	11	?
6	50.17	111	12.8	?
7	68	4.3	4.3	7.33
8	83	4.3	4.3	7.33
9	98	4.3	4.3	7.33

Table 3.12: Parameters known about the test-beam setup. From left to right: layer number, distance in beam direction z, resolution σ in x, resolution σ in y and the thickness of the layer X

	real data				simulation			
i \layer	4	5	6	total	4	5	6	total
0	641	641	641	1923	641	641	641	1923
1	952	946	923	2821	947	945	923	2815
2	967	954	924	2845	961	957	933	2851
3	974	955	918	2847	963	958	932	2853
4	980	955	912	2847	964	958	930	2852

Table 3.13: Test beam data. Estimated X values (in $10^{-4}X_0$) for the first 4 iterations. $i = 0$ are the are the initial guesses of X. The sum of all X includes only the 3 estimated layers. Left: real data; right: simulated data.

to around 0.95% of a radiation length per layer, including the entire infrastructure (see Table 3.13). The corresponding plots with simulated tracks in figure 3.3 look very similar.

By combining the forward and the backward filter estimates, individual (smoothed) chi-squares can be obtained for each layer. Their distribution is summarised in Table 3.14, both for real and simulated data. While the distributions from simulated data are perfect, the real data show some discrepancies. These could be due either to deviations of the actual resolution from the nominal one, or to the fact that the material is not really concentrated in the detector layers.

Unfortunately, it was not possible to validate the material estimates by comparing them to the true values because the latter were never determined by the experimenters. In this test run preliminary and bulky cooling and support structures accounted for most of the material budget, but they have been modified in the meantime. The starting values shown in Table 3.13 were rough guesses by a member of the collaboration [18]. Actually it turned out that the result of the material estimation did indeed improve the quality

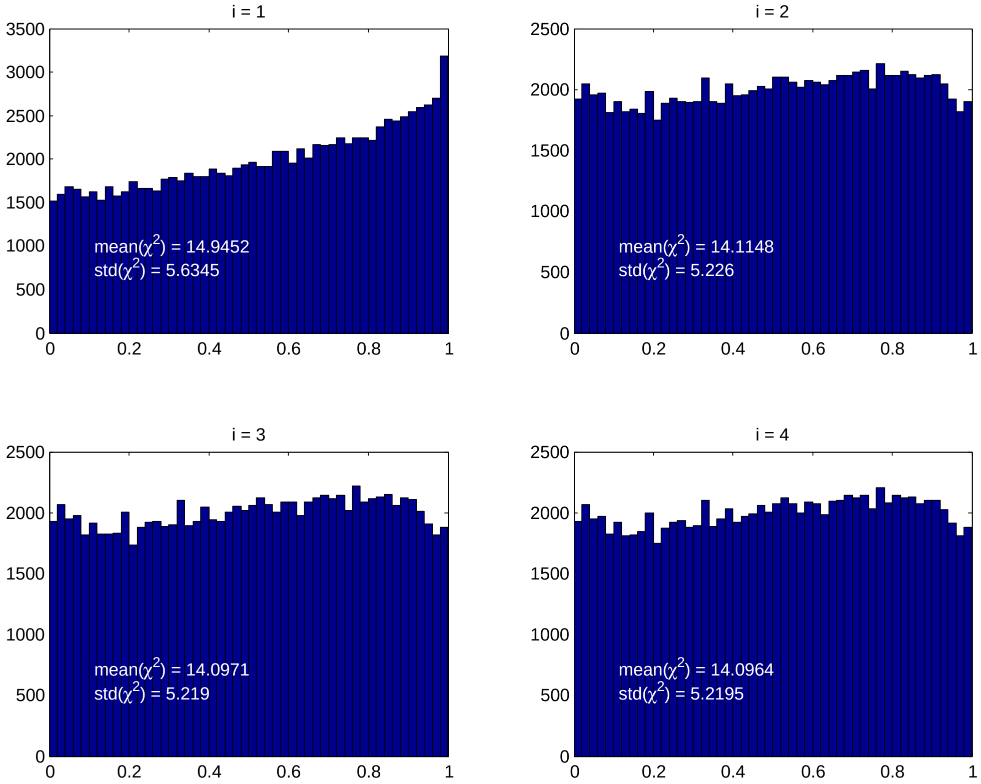


Figure 3.2: Test beam data. Distribution of the p-values of the total χ^2 of the forward filter, for the first four iterations. The p-values should be uniformly distributed.

layer	1	2	3	4	5	6	7	8	9
mean	1.888	1.611	2.086	2.026	2.290	2.020	2.307	1.702	1.900
std	1.755	1.575	2.035	1.635	2.509	1.734	2.280	1.676	1.790
mean	2.079	2.007	2.004	2.002	2.012	1.993	1.991	1.996	1.997
std	2.077	2.005	2.004	2.003	2.004	2.001	1.988	1.999	1.989

Table 3.14: Mean and standard deviation (std) of the smoothed χ^2-distributions in every layer, after four iterations. The two top rows are from real data, the two bottom rows are from simulated data.

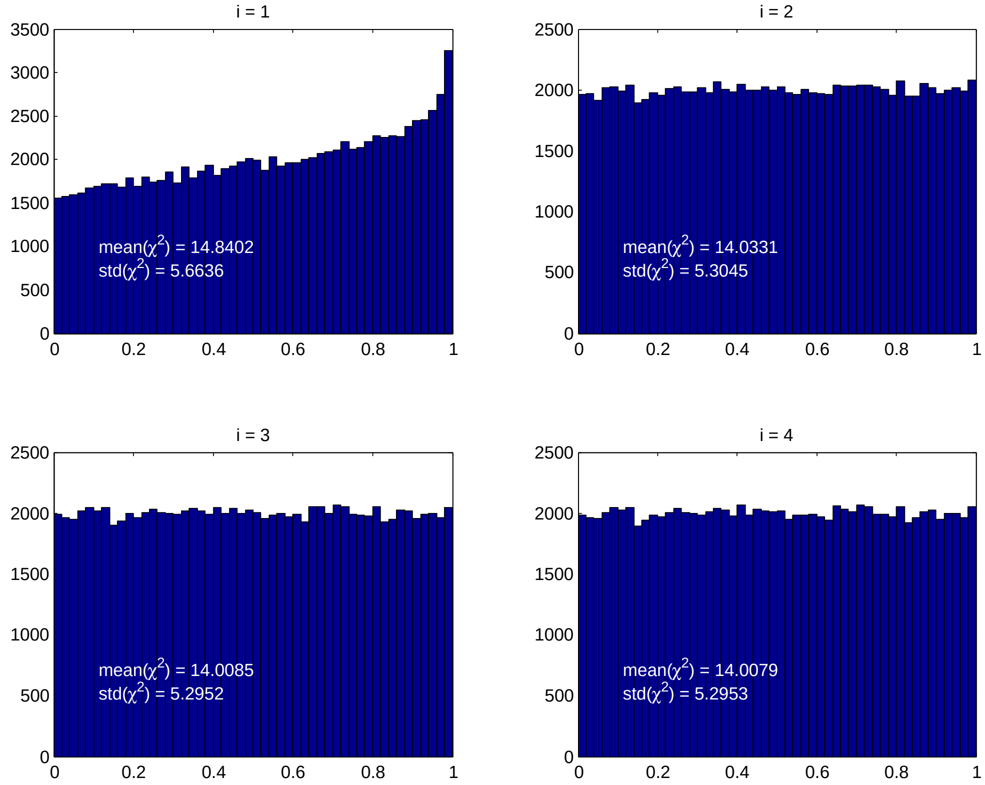


Figure 3.3: Simulated test beam data. Distribution of the p-values of the total χ^2 of the forward filter, for the first four iterations. The p-values should be uniformly distributed.

indicators of the track reconstruction in their analysis framework [18].

3.5 Conclusion and Outlook

Two methods for estimating the amount of material in a detector using charged tracks have been presented. Simulation studies show that only the forward-backward Kalman filter approach is feasible in terms of number of tracks required to obtain satisfactory estimate. It was possible to validate the latter method with test-beam data, in spite of a relatively high beam momentum of 120 GeV. Four iterations over the track sample are sufficient to obtain a reliable estimate of the total material budget and a nearly perfect χ^2-distribution.

It should be noted that the individual thicknesses do not always converge to their true values, although the total material budget does. In fact, there tends to be an alternating over- and underestimation of the thickness. The most likely reason for this is neglecting the correlations between the state vectors in different layers. These correlations can in principle be computed by error propagation in both the forward and the backward

Kalman filter. Once they are known, they can be used to set up a global covariance matrix for the estimated multiple scattering angles $\Delta\theta$ and $\Delta\beta$ in all layers.

A restriction in the approach presented above is the assumption that the material is concentrated in the measurement layers. This is clearly not the case in many tracking detectors. A future development of the method will have to relax this assumption and deal with material between the measurement layers, possibly spread over a larger distance.

An unexpected outcome of the simulation studies is the fact that compared to multiple scattering energy loss by ionization contributes very little to the estimate of the material thickness. It could be worth studying whether energy loss of electrons by bremsstrahlung can make a significant contribution to the material estimate.

In a more recent paper, H. Gjersdal et. al. combined the forward backward Kalman estimation method with a measurement resolution estimator to give a simultaneous estimation of both material budget and sensor resolution. A description of this work can be found in [16].

4 Tracking in Belle II

4.1 Motivation and introduction

The Belle II experiment is currently under construction at the “The High Energy Accelerator Research Organization” or “Kō Enerugī Kasokuki Kenkyū Kikō” in Japanese (common abbreviation: **KEK**) in Tsukuba, Japan. It will be a large electron-positron collider experiment using the SuperKEKB accelerator also currently under construction. SuperKEKB/Belle II is the successor and upgrade of the successful KEKB/Belle experiment and will consist of a mixture of reused and upgraded parts and newly developed and built parts.

The tunnel of 3.0 km circumference used for SuperKEKB originally housed the TRISTAN accelerator, a 60 GeV centre of mass collider constructed in 1986.

While SuperKEKB will have the same centre of mass energy as its predecessor, it aims at a 40-fold increase in luminosity. This, in addition to the higher precision of the Belle II detector, will make possible a new range of measurements in heavy flavour physics. It is planned to uncover details of new physics processes that are not accessible by current hadron collides.

The tracking detectors for Belle II are new and not reused ones from Belle. Also there will be a new two-layer pixel detector (PXD) which did not exist in Belle In addition to the silicon strip detector (SVD) and the central drift chamber (CDC).

The software for the analysis of the Belle II detector output is also a new project started from scratch, rather than an adaptation of the old “Belle Analysis Software Framework” (basf). As a member of the Belle II tracking group the author of this theses helped implementing the tracking modules and developed several tools for tracking quality assurance and tracking code debugging for the “Belle Analysis Software Framework 2” (basf2). This also included adding features and debugging work for GENFIT, the fitting library used in basf2.

In particular, the author was assigned the task of studying the reconstruction of low-momentum tracks. The goal of the Belle II tracking group is to correctly reconstruct particles with transverse momenta as low as approximately 50 MeV. This will increase the accuracy of the analysis of decays like $\mathrm{D}^{*+} \longrightarrow \mathrm{D}^0 + \pi^+$, where the pion has an average momentum of 160 MeV and therefore the sensitivity of the experiment to new phenomena appearing in events of this kind will rise. The finding and fitting performance was studied in detail and presented in chapter 6.

4.2 SuperKEKB and Belle II

As mentioned above, the upgraded KEKB collider is called SuperKEKB. Figure 4.1 is an illustration of SuperKEKB. Some of the most important changes are marked in the figure. The centre of mass energy of the electron-positron collision will stay at 10.579 GeV, the mass of the 4S state of the Υ meson; however, runs with slightly different energies ($\Upsilon(5S)$) are also planned. The principal improvement over KEKB will be the 40-fold increase in luminosity to a value of $8 \times 10^{35}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$. This ambitious goal will be achieved by increasing the beam current by a factor of two and by decreasing the volume of the beams at the interaction point. The latter is called nano beam scheme by the accelerator group. The two beams have different energies, which gives the particles created in the collisions a boost in the travelling direction of the higher energy beam. This increases the vertex resolution.

Belle II is a multi-purpose particle detector or solid angle magnetic spectrometer. It will surround the only interaction point of the SuperKEKB accelerator. Its composition is similar to other large detectors used in collision experiments. The Belle II detector consist of the following sub-detectors:

- Pixel detector (PXD)
- Silicon vertex detector (SVD)
- Central drift chamber (CDC)
- Time of propagation counter (TOP)
- Aerogel ring imaging Cherenkov detector (ARICH)
- Electromagnetic calorimeter (ECL)
- K_L and μ detector (KLM)

Figure 4.2 is an illustration of the Belle II detector. The sub-detectors are marked with text boxes. Their relative size and position is visible.

The tracking detectors PXD, SVD and CDC are newly built for Belle II to cope with the 40-fold increase in luminosity and the decreased boost at the interaction point, while still providing a higher accuracy than the old Belle SVD and CDC sub-detectors. Being tracking detectors, their purpose is to provide the data to estimate the track parameters (momentum and track origin or vertex) of charged particles. Figure 4.3 shows a schematic illustration of size of the PXD and SVD layers in the R-z-plane.

The PXD consists of two layers of depleted field effect transistor (DEPFET) silicon pixel sensors. The innermost layer, with a radius of about 14 mm sits very close to the beam pipe (ca. 10 mm radius).

The SVD consists of four double-sided silicon strip sensor layers. The two sides have different pitch sizes, leading to a higher resolution in direction orthogonal to the beam direction (z-axis) than parallel to z. Polar angles within the acceptance range from 17°

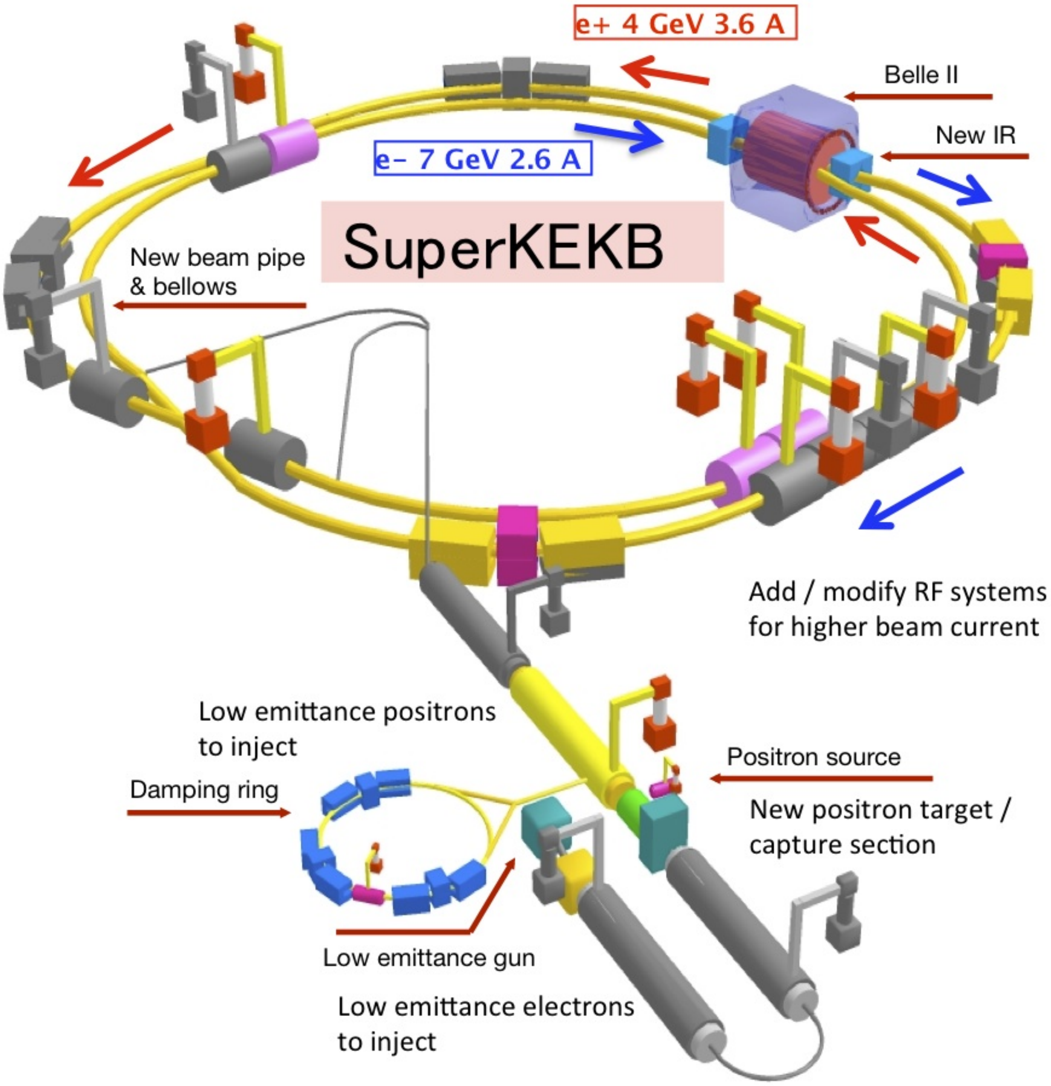
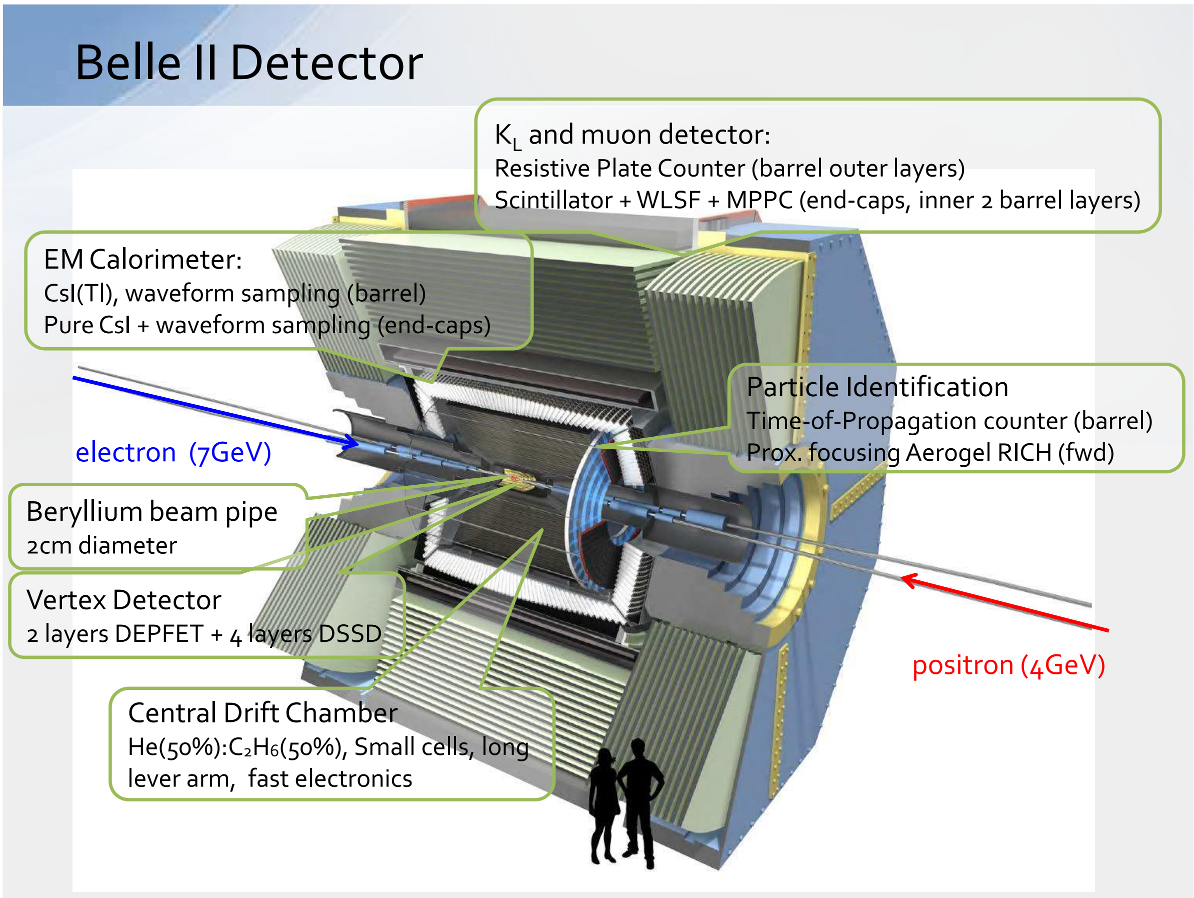


Figure 4.1: Illustration of the SuperKEKB accelerator. Areas of major change with respect to the KEKB accelerator are marked. Taken from `belle2.desy.de`. Original version from SuperKEKB website

to 150°; the innermost SVD layer (named layer 3) has a radius of 38 mm, while the outermost layer (named layer 6) has a radius of 135 mm.

The output of the PXD and the SVD from the point of view of tracking software are already clusterised hits. This means that if a particle lights several strips (or pixels), an average position is calculated using the signal heights as weights. This should result in a cluster hit resolution which is approximately twice as good as the digital resolution $pitch/\sqrt{12}$. The PXD provides two-dimensional hits, while the SVD provides two one-dimensional hits (one for each of the two sides of the double-sided strip sensor). If there are several particles crossing the same SVD sensor, ghost hits will appear as an



Belle II Detector

K_L and muon detector:
Resistive Plate Counter (barrel outer layers)
Scintillator + WLSF + MPPC (end-caps, inner 2 barrel layers)

EM Calorimeter:
CsI(Tl), waveform sampling (barrel)
Pure CsI + waveform sampling (end-caps)

Particle Identification
Time-of-Propagation counter (barrel)
Prox. focusing Aerogel RICH (fwd)

electron (7GeV)

Beryllium beam pipe
2cm diameter

Vertex Detector
2 layers DEPFET + 4 layers DSSD

positron (4GeV)

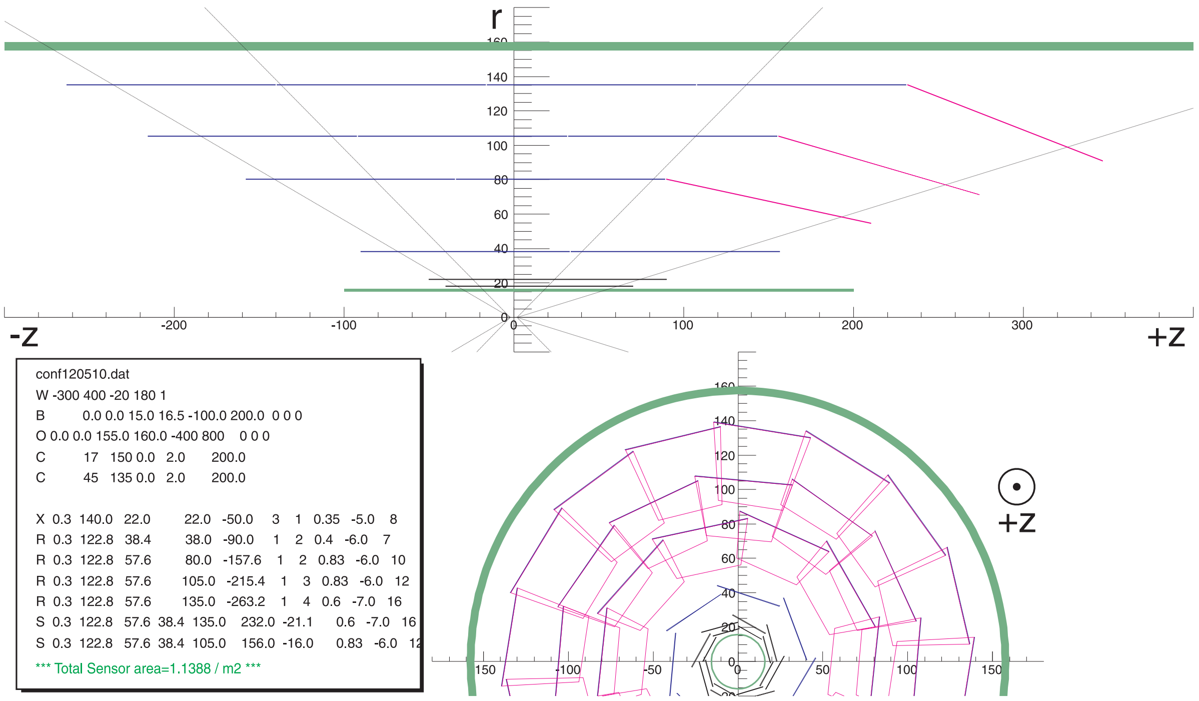
Central Drift Chamber
He(50%):C_2H_6(50%), Small cells, long lever arm, fast electronics

Figure 4.2: Illustration of the Belle II Detector with descriptions and human silhouettes for size comparison. Taken from Belle II web site.

additional challenge for the track finder and another source of outlier hits the track fitter has to deal with. Additionally it is possible in the SVD that only one side of the double-sided sensor generates a cluster, leading to a one-dimensional measurement of the track position in that case.

PXD and SVD together provide the main source of information for the z position of vertices, as the CDC has very poor z resolution (most of the wires are exactly parallel to z). On the other hand, the CDC contributes more to the momentum resolution of reconstructed tracks because of its much larger lever arm. Actually, the name “silicon vertex detector” is a bit misleading, as strictly speaking both PXD and SVD are silicon vertex detectors. The explanation is the existence of a 4-layer silicon strip detector in Belle, so the name was just kept for the new device.

The CDC has 56 layers of wires ranging from a radius of 160 mm to 1130 mm. The 56 wire layers are organised in 9 super-layers. These super-layers have an alternating configuration of axial layers (strictly parallel to z) and stereo layers, which are slightly slanted with respect to the z axis and therefore offer a small position information along z, while the axial layers only offer $R\Phi$ information. As a drift chamber the CDC measures



```
conf120510.dat
W -300 400 -20 180 1
B        0.0 0.0 15.0 16.5 -100.0 200.0  0 0 0
O 0.0 0.0 155.0 160.0 -400 800    0 0 0
C        17   150 0.0   2.0      200.0
C        45   135 0.0   2.0      200.0

X  0.3  140.0   22.0        22.0   -50.0    3    1   0.35   -5.0    8
R  0.3  122.8   38.4        38.0   -90.0    1    2   0.4   -6.0    7
R  0.3  122.8   57.6        80.0   -157.6   1    2   0.83   -6.0   10
R  0.3  122.8   57.6       105.0   -215.4   1    3   0.83   -6.0   12
R  0.3  122.8   57.6       135.0   -263.2   1    4   0.6   -7.0   16
S  0.3  122.8   57.6  38.4  135.0    232.0  -21.1      0.6   -7.0   16
S  0.3  122.8   57.6  38.4  105.0    156.0  -16.0      0.83   -6.0   12
*** Total Sensor area=1.1388 / m2 ***
```

Figure 4.3: Schematic illustration of PXD and SVD sensor layers. Scale in mm. Stated values might not be up to date with current design.

the drift time of the charge induced by passing particles. The drift time is translated to a drift length and therefore enters the track fit as a one-dimensional measurement. As the drift length has no sign, a left-right ambiguity arises, because it is impossible tell for a single hit on which side of the wire the particle has passed. This problem has to be addressed in the track reconstruction software.

The TOP provides particle identification in the barrel region by measuring the time of propagation of Cherenkov photons created in its quartz radiator. It is designed to give a good $K^{\pm}$ / $\pi^{\pm}$ separation.

The ARICH is used for particle identification in the forward region of Belle II. Like the TOP it is also a Cherenkov detector, but its radiator is made of an aerogel. It is also used to discriminate mainly between $K^{\pm}$ and $\pi^{\pm}$; under 1 GeV additionally between $e^{\pm}$.

The ECL measures the energy of incident particles. Mainly $e^{\pm}$ and γ cause an electromagnetic particle shower it its thallium-doped CsI crystals. The crystals and the mechanical structure are reused from Belle.

The KLM provides particle identification for particle with high penetration power – mainly $\mu^{\pm}$ and K_L^0. It consists of a barrel part (BKLM) oriented parallel to z, using the same resistive-plate chambers and iron absorber plates as Belle, and a new part for the end cap region (EKLM). The EKLM uses new scintillators instead of the resistive-plate chambers that were used in Belle.

Much more detailed information can be found in the technical design report [19], but if specific numbers are needed it is always best to contact the responsible team directly as and ask for the latest design parameters.

4.3 Belle II Physics Goals

The aim of Belle II is the discovery of physics beyond the current standard model of elementary particles at the rare/precision frontier. In Belle II it will be possible to measure deviations from known decays predicted by new theories that are beyond the measurement resolution of current experiments.

The main advantage of this second generation B factory will be the ability to perform measurements in all fields of heavy flavour physics simultaneously ($B^0/B^\pm$ meson decays, charm physics, τ lepton physics and pure electroweak measurements). This includes the the further confirmation or the finding of deviations of the Kobayashi-Maskawa mechanism for standard model CP-violation. Or in other words: Is there any new CP-violating phase, and can the hierarchy in the CKM matrix be explained with a new flavour symmetry? Cosmic models suggest a much larger CP-violation to explain the matter anti-matter asymmetry in the universe than the standard model theory of CP-violation can explain.

Other investigations will search for effects from new Higgs fields, new flavour violation such as lepton-flavour violation and the existence of right-handed currents.

Among the many measurements planned to answer these questions at Belle II are time dependant CP asymmetries in $B^0 \longrightarrow \phi K^0$ and $B^0 \longrightarrow \phi K^0_S \pi^0 \gamma$ branching fractions for $B \longrightarrow \tau \nu$, $B \longrightarrow D \tau \nu$ and $\tau \longrightarrow \mu \gamma$. These examples are very difficult to access at other currently running high-energy physics experiments.

More details for these examples and lots of other examples of planned physics analyses can be found in the Belle II technical design report from 2010 [19] and the Physics at Super B Factory article (also 2010) [20]. As the publication years indicate, these articles do not yet incorporate the latest exclusion limits from LHC for physics beyond the Standard Model, so the physics analysis program stated in these documents will certainly be revised until the start of the Belle II data taking.

4.4 The Belle II Analysis Software Framework

The Belle Analysis Framework 2 (basf2) is the computer program that handles all software-related tasks needed for the Belle II experiment. This includes basically every software task that is not run on specialised hardware but on general purpose computers. Examples are online tasks such as data acquisition and high level trigger, or offline tasks such as event reconstruction and analysis.

basf2 is a new software project written from scratch and not an adaptation of an existing high-energy physics experiment software. For its design existing software frameworks were of course analysed and therefore it shares many features with these.

As a framework basf2 calls the methods of modules which were implemented by developers or users. The modules that are executed by the framework are defined in a so called steering file. This is a pure Python source file, so all features of this programming language are available in the steering file. Additionally the modules can be grouped into paths which can be executed conditionally based on output values from modules.

In basf2 all functionality is organised into modules, even tasks like reading events from hard drive and storing them there or the generation of the metadata used by all other modules to keep track of the current experiment, run and event numbers. Modules can be written in C++ or Python. They communicate via a so-called datastore where the modules store and retrieve objects holding data.

As is the case with other high-energy physics frameworks, basf2 uses many third party libraries, notably GEANT4 [21] for detector simulation, EvtGen [22] for event generation and ROOT [23] for data I/O. GENFIT is a relatively young tracking library and is described in section 4.5.1. The third party libraries are located in special folder called "externals" and are part of a full basf2 installation so that the users do not have to install them themselves. This also ensures specific version numbers of these libraries.

In section 4.5.3 a few modules are described, and a typical simulation run by the author to test the track fitting is explained. This simulation is illustrated by a flow chart in figure 4.4. It shows the strict linear order of modules by black arrows pointing from one black ellipse representing a module to the next. Red boxes are StoreArray objects. The module that registers them the first time points to them with a red arrow. Dark blue arrows point from a StoreArray to the module that requires it as input. basf2 quits with a fatal error naming the missing StoreArray if the requirement is not met.

As mentioned above, the metadata that help modules to keep track of current experiment/run/event identifiers is itself an object in the DataStore, created by the module EvtMetaGen. The module Gearbox loads parameters from XML. Among them are the geometry parameters that are used by the Geometry module to build the GEANT4 master geometry. A peculiarity of the current framework is the fact that the Geometry module also registers the StoreArrays for the Hit classes. This is the reason for the red arrows from Geometry to the hit classes. The next module is the event generator – a simple particle gun in this example. FullSim is the interface module to GEANT4 filling the hit StoreArrays with hit objects. The last three modules are related to track reconstruction and are described in section 4.5.3.

More information on the structure and features of basf2 can be found in [19] and [24].

4.5 Goals and status of the tracking software

The minimum goal for the tracking system when Belle II was first conceived was to keep the momentum and vertex resolution of Belle despite the much higher luminosity and smaller beam energy difference and consequently smaller boost. Of course it is anticipated that the tracking system will provide better performance as in the Belle experiment because the design specifications of the sub-detectors should make that possible. Therefore the goal is a better momentum and impact parameter resolution and the possibility of reconstructing tracks with much lower momenta than was the case in Belle.

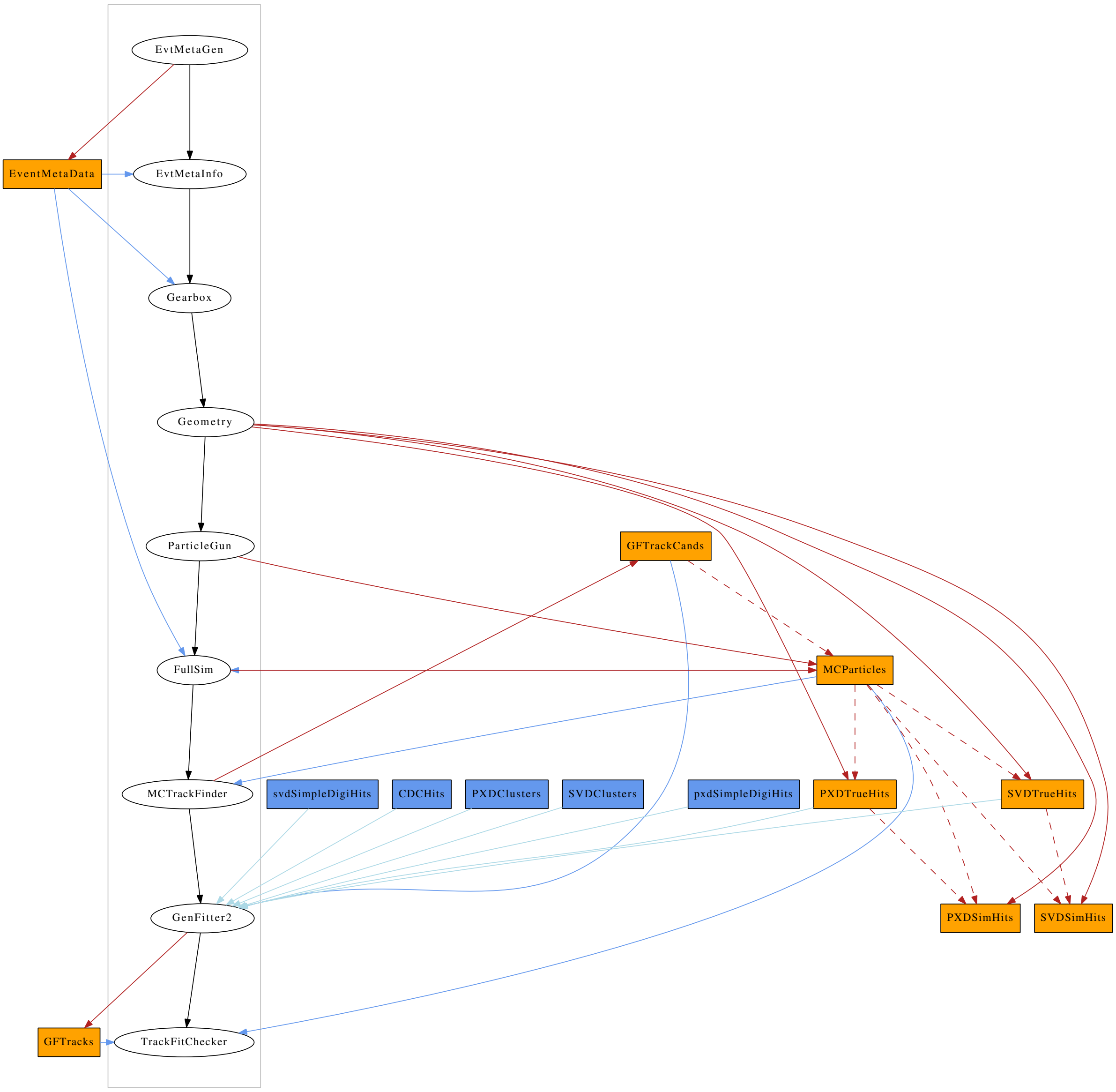


Figure 4.4: Module and StoreArray objects dependancy in an example steering file

The track finding/fitting strategy in Belle, described in a simplified way, was the following. A Hough finder was used to find tracks in the CDC. Its implementation is called TRASAN. The track candidates from TRASAN were fitted with a Kalman filter implementation named TRACK. Hits detected in the Belle SVD were added to the track if they were close enough to the extrapolated track from the CDC through the SVD. The vertex fitting was done with a Kalman filter based kinematic vertex fitter called KFitter. A dedicated vertex finder was not used, as it is of limited use in the clean environment of an e^+/e^- collider, where very reliable particle identification information is available at the vertex reconstruction stage.

Especially the requirement to reconstruct very low-momentum tracks rules out a pure reimplementation of the Belle tracking strategy. To find tracks with hits in the SVD only or a combination of hits in PXD and SVD, a dedicated low-momentum track finder is being developed[25]. Several algorithms for track finding in the CDC were implemented by the tracking group. There is a Hough finder [26], a Legendre finder and local track finder [27]. Additionally TRASAN was ported from BASF to basf2, so it is possible by every developer or user to see how the new algorithms/implementations perform in comparison to the old one. All new algorithms/implementations are currently tested, improved and tuned.

The planned full track finder for basf2 will make use of all algorithms taking into account their strength and weaknesses and combining the information of the PXD, SVD and CDC to produce the final track candidates, including tracks with hits only in the inner tracker or curling tracks making several turns in the inner tracker and the CDC. The details of the optimal strategy have yet to be worked out.

With the decision to use GENFIT as the fitting library in basf2, an implementation of a Kalman filter (see section 2.2.2) and a DAF (see section 2.2.5) was available, but needed to be interfaced to the basf2 data structure and thoroughly tested and improved, as GENFIT [28] is a relatively new software project.

As far as vertex reconstruction is concerned, two options are currently available: KFitter, the vertex fitter of basf, was ported to basf2, and the experiment independent vertex reconstruction library RAVE [8], for which the author implemented the interfaces with the Belle II data objects.

4.5.1 GENFIT

GENFIT is a experiment independent track fitting library developed at the Technical University of Munich [28]. GENFIT offers at the moment implementations of the Kalman filter and of the Deterministic Annealing Filter, and two different propagation codes. One is called RKTrackRep and based on a C port of GEANE (Geant3 track propagation plus track error propagation), but heavily modified and only under the control of the GENFIT developers. This propagation uses the Nyström algorithm to numerically solve the equation of motion of charged particles in an arbitrary magnetic field. It can handle arbitrary geometries if present in the ROOT TGeo format. The other option, namely using GEANE directly for propagation and material effects, is also possible, but this is not realised in basf2.

The implemented material effects are energy loss due to ionisation (using the Bethe–Bloch model (2.41)), bremsstrahlung (using the Gaussian approximation of equation (2.42)), and electromagnetic multiple scattering (the Highland formula (2.38) is one of two models). GENFIT uses two different coordinate system. The propagation and material effects as implemented in RKTrackRep use a global 7D Cartesian coordinate system:

$$\boldsymbol{x} = (x, y, z, a_x, a_y, a_z, q/p)^{\mathrm{T}}, \tag{4.1}$$

where $(a_x, a_y, a_z) = \boldsymbol{a} = \boldsymbol{p}/p$ is the unit vector in the track direction. The estimation algorithms work in a local 5D coordinate system defined in the measurement plane:

$$x = (q/p, u', v', u, v)^{\mathrm{T}}, \tag{4.2}$$

where u and v are the local coordinates in the plane. Before every update or propagation step the state is transformed accordingly .

GENFIT was originally developed for the PANDA experiment, but intended to be experiment independent from the start. At the moment GENFIT undergoes a major change as all requirements by alignment and outer detector groups of Belle II are incorporated into GENFIT, but still keeping its experiment independent nature.

To interface basf2 with GENFIT the GEANT4 geometry is converted to TGeo via a ROOT function, and for every tracking detector hit a corresponding GENFIT RecoHit is created. The RecoHit holds hit dimensionality, coordinates, covariance matrix and position, orientation and size of the plane where the hit is defined. The RecoHits need to be added to a GFTrack object, which is then given to a fitting algorithm object. As mentioned before, GFKalman and GFDaf fitting algorithm objects are the ones currently available.

At the time of writing GENFIT was undergoing a major revision with many changes in the public functions and therefore in the interface details. Describing the public functions therefore does not make much sense at the time being.

GENFIT 2 will feature easy access to all information needed for alignment, reference tracks and extrapolation functions that provide the additional information required by the outer detectors in Belle II.

4.5.2 RAVE

RAVE [8] is a detector independent software library for vertex reconstruction. It was originally developed inside the CMS community and is in continuous use by the CMS experiment. It comes with several vertex fitting algorithm, a combined vertex finder/fitter, and the possibility to make kinematic fits with mass constrains. A helix propagator assuming a constant magnetic field and vacuum is included. RAVE offers the option to recalculate the input track parameters to force them to the fitted vertex.

This covers typical use cases in collider experiments, where tracks are propagated to the vicinity of the likely vertex position before they are fed to the vertex fitter. The absence of material and a constant field are valid assumptions for the propagation of track parameter inside the beam pipe by the vertex fitter. If these assumptions are not met, RAVE offers the possibility to replace the built-in propagator with a more general one. A working example of this is GFRave, a part of GENFIT that replaces the vacuum propagator of RAVE with the RKTrackRep propagator of GENFIT, thus enabling the fitting algorithms in RAVE to deal with arbitrary magnetic fields and material distributions. Inside basf2 RAVE can be used directly or via GFRave, as they are both part of the basf2 externals.

Among the included vertex algorithms are the Kalman filter, the Adaptive Vertex Fitter (AVF) and the finder/fitter Adaptive Vertex Reconstructor (AVR). A very brief description on how they work can be found in section 2.4. More information can be found in [7] and [8].

4.5.3 Overview of Code written for basf2 and GENFIT

When the author joined the Belle II tracking group, no tracking was possible in basf2. The first task for the tracking group was to make a module interfacing basf2 and GENFIT and a module producing 100 % correct track candidates from the simulated truth. These modules are GenFitter and MCTrackFinder.

First the author contributed mainly the code adding the PXD and SVD hits and later became the primary maintainer of these modules adding features when need for them arose or request were made. It was also the author's responsibility to make sure basf2 would still compile after upstream changes in GENFIT. The PXD/SVDRecoHits were introduced by the sub-detector developers, but subsequently changed and extended by the author when necessary. The author wrote or made major contributions to the following basf2 modules or classes:

TrueHitTester

This module was developed to investigate/validate the simulated TrueHits directly before they are smeared to become digitised hits. It extracts the 5D track parameter in the format used by GENFIT, the scattering angles and the energy loss for every layer from the TrueHits. The result is written to a ROOT file for further inspection.

MCTrackFinder

The MCTrackFinder is an important development tool. It produces GFTrackCand objects that can be used to test GENFIT in basf2 under optimal conditions. Developers of track finding algorithms can compare their track candidates to the "perfect" ones produced by MCTrackFinder. In case of CDC hits MCTrackFinder tells GENFIT how to resolve the left/right ambiguity. MCTrackFinder offers many options for the tracking developer: Very fine grained selection of which kind of particles will result in a track candidate, optional smearing of initial track parameters for the fit (with or without passing the smearing cov to GENFIT), which kind of hits should go into the candidate etc..

GenFitter

As mentioned above, GenFitter is the current interface module between basf2 and GENFIT. The input are GFTrackCand objects produces by any of the track finders. The output are fitted tracks in two formats: Belle2::Tracks and GFTracks. The first one is a light object useful to store the fitted track parameters after the propagation into the beam pipe on disk; the second one is a heavy GENFIT object supporting track extrapolation to arbitrary points.

GenFitter2

A module just for tracking development. Features can be tested there before they are transferred to the GenFitter module. It offers additional options such as changing the multiple scattering model or deactivating specific material effects in GENFIT via a basf2 steering file.

TrackFitChecker

It takes GFTracks as its input and calculates a variety of statistical test quantities. All tests described in section 2.2.3 and 2.2.6 are implemented. Mean and standard deviation of all test quantities are calculated and written to the terminal and/or a test file, depending on user input. As an alternative to mean and std (standard deviation), median/MAD (median of the absolute deviation from the median) and truncated mean/std are also implemented. The MAD is scaled such that it is 1 for a standard normal distribution. The module detects automatically whether the DAF or the Kalman filter was used in the track fit, and extracts the weights if the former is used. Also, the presence of predicted states in the GFTrack objects is automatically detected; if so, the states are analysed along with the updated and smoothed stated.

Vertexer

This module takes all GFTracks from one StoreArray and feeds them to GFRave to perform vertex fits. The output are GENFIT objects of the class GFRaveVertex. The main purpose of the module was to test GFRave and RAVE and to show other developers how GFRave can be used inside basf2.

RaveInterface

RaveInterface is a collection of classes intended to become the principal interface between basf2 and RAVE. It frees the basf2 user from tasks such as passing the magnetic field to RAVE. Its public functions are designed to mimic the functions of KFitter to make the integration of RAVE into the work flow of the Belle II analysis package easy. The user creates a RaveVertexFitter object in his or her code, adds tracks to it using any format in use in basf2 (Belle2::Tracks, GFTracks, Belle2::Particles), selects a fitting algorithm, and retrieves the result form the RaveVertexFitter object.

VertexFitChecker

A module that performs statistical tests on fitted vertices. It takes GFRaveVertex objects as its input and extracts the p-values of the vertex fit. The residuals and pulls of the estimated vertex position are also calculated.

VXDSimpleBackground

VXDSimpleBackground produces artificial, precisely placeable background hits in the PXD and SVD detector. Its main purpose is the validation of the DAF implementation in GENFIT.

MCTrackCandCombiner

This module combines the track candidates from a CDC track finder with the track candidates from the VXD track finder using the simulated truth info. If at least 60 % of the hits in both track candidates are coming from the same particle they will be merged into one track candidate. The MCTrackCandCombiner serves as an intermediate step until a full track finder becomes available. The module is also useful to validate the output of the full track finder during development.

Contributions to GENFIT

The author's contribution to GENFIT concentrated on three areas: improvement of material effects treatment, improving the DAF implementation, and adapting GENFIT classes to the requirements of basf2 (mainly the GFTrackCand class). Of course, errors found in other parts of GENFIT where also fixed or reported.

The material effects improvements include the implementation of the Highland formula and the correct treatment of position variances and covariances in the multiple scattering noise matrix. Originally, the projection of the predicted covariance matrix onto the measurement plane was only done for the propagated covariance matrix and not for the material effects noise matrix that is added to the propagated covariance matrix before the update step. This was changed so that both matrices are projected before the are added to form the predicted covariance matrix.

The DAF implementation lacked the correct calculation of the total χ^2 and ndf according to equations (2.34) and (2.35), and the annealing scheme implementation was wrong. In addition to these features also a simple convergence criteria based on the DAF weights was implemented to have an adaptive number of iterations over the temperatures T_I. Now it is also possible to set any probability cut ($c_m(\alpha)$ in equation (2.30) instead of choosing from only three probability cut values.

The non zero elements of multiple scattering matrix in the 7D global Cartesian coordinate system ($\boldsymbol{x} = (x, y, z, a_x, a_y, a_z, q/p)^{\mathrm{T}}$) used in GENFIT are stated in equations 4.3 to 4.21.

$$\mathrm{Cov}(x,x)/\sigma_\alpha^2 = \frac{h^2}{3}(\cos^2(\theta) + \sin^2(\phi)\sin^2(\theta)) \tag{4.3}$$

$$\mathrm{Cov}(x,y)/\sigma_\alpha^2 = -\frac{h^2}{3}\sin(\phi)\cos(\phi)\sin^2(\theta) \tag{4.4}$$

$$\mathrm{Cov}(x,z)/\sigma_\alpha^2 = -\frac{h^2}{3}\cos(\phi)\cos(\theta)\sin(\theta) \tag{4.5}$$

$$\mathrm{Cov}(x,a_x)/\sigma_\alpha^2 = \frac{h}{2}(\cos^2(\theta) + \sin^2(\phi)\sin^2(\theta)) \tag{4.6}$$

$$\mathrm{Cov}(x, a_y)/\sigma_\alpha^2 = -\frac{h}{2}\sin(\phi)\cos(\phi)\sin^2(\theta) \tag{4.7}$$

$$\mathrm{Cov}(x, a_z)/\sigma_\alpha^2 = -\frac{h}{2}\cos(\phi)\cos(\theta)\sin(\theta) \tag{4.8}$$

$$\mathrm{Cov}(y, y)/\sigma_\alpha^2 = -\frac{h^2}{3}(\sin^2(\phi)\sin^2(\theta) - 1) \tag{4.9}$$

$$\mathrm{Cov}(y, z)/\sigma_\alpha^2 = -\frac{h^2}{3}\cos(\theta)\sin(\phi)\sin(\theta) \tag{4.10}$$

$$\mathrm{Cov}(y, a_y)/\sigma_\alpha^2 = -\frac{h}{2}(\sin^2(\phi)\sin^2(\theta) - 1) \tag{4.11}$$

$$\mathrm{Cov}(y, a_z)/\sigma_\alpha^2 = -\frac{h}{2}\cos(\theta)\sin(\phi)\sin(\theta) \tag{4.12}$$

$$\mathrm{Cov}(z, z)/\sigma_\alpha^2 = \frac{h^2}{3}\sin^2(\theta) \tag{4.13}$$

$$\mathrm{Cov}(z, a_z)/\sigma_\alpha^2 = \frac{h}{2}\sin^2(\theta) \tag{4.14}$$

$$\mathrm{Cov}(a_x, a_x)/\sigma_\alpha^2 = 1 - \sin^2(\theta)\cos^2(\phi) \tag{4.15}$$

$$\mathrm{Cov}(a_x, a_y)/\sigma_\alpha^2 = -\sin^2(\theta)\cos(\phi)\sin(\phi) \tag{4.16}$$

$$\mathrm{Cov}(a_x, a_z)/\sigma_\alpha^2 = -\cos(\theta)\sin(\theta)\cos(\phi) \tag{4.17}$$

$$\mathrm{Cov}(a_y, a_y)/\sigma_\alpha^2 = 1 - \sin^2(\theta)\sin^2(\phi) \tag{4.18}$$

$$\mathrm{Cov}(a_y, a_z)/\sigma_\alpha^2 = -\cos(\theta)\sin(\theta)\sin(\phi) \tag{4.19}$$

$$\mathrm{Cov}(a_z, a_z)/\sigma_\alpha^2 = \sin^2(\theta) \tag{4.20}$$

$$\tag{4.21}$$

Because the multiple scattering matrix is a covariance matrix the symmetric elements are not explicitly stated. Additionally the following symmetries apply: $\mathrm{Cov}(x, a_y) = \mathrm{Cov}(y, a_x)$, $\mathrm{Cov}(z, a_x) = \mathrm{Cov}(x, a_z)$ and $\mathrm{Cov}(y, a_z) = \mathrm{Cov}(z, a_y)$. The relations between the direction cosines a_x, a_y, a_z and the spherical angle coordinates ϕ and θ are $\sin\theta = \sqrt{1 - a_z^2}$, $\cos\theta = a_z$ and $\phi = \arctan(a_y/a_x)$ or more precisely $\phi = \mathrm{atan2}(a_y, a_x)$.

5 Validation of Belle II tracking software

This chapter describes the current state of the Belle II tracking software. It concentrates on the quality of fitted tracks that can currently be reached, starting with ideal conditions and changing them to more and more realistic ones.

When testing software under development it is important to state its version. The svn revision number of basf2 was 5459 when the first tests included in this thesis where conducted. During the testing basf2 was regularly updated and reached svn revision number 5538 when the testing ended. The externals version number was v00-03-04, but two components of the externals were altered: the GENFIT version was changed to the so- called "recohit" branch with revision number 990, and the multiple scattering model for muons in GEANT4 was changed. The reason for this is explained in section 5.1.

The change of the GENFIT version was undertaken to get new features which were not yet in the official basf2 externals: the improved material stepping and multiple scattering variance transformation, and the possibility to switch off resolution of the left/right ambiguity of wire hits by the DAF.

5.1 Input Validation

To validate the correctness of the track reconstruction implementations one first has to make sure that the input data are correct. This is especially true when the input data is generated by software that has never been used in production before, like basf2. The input for the track fit are the hits that belong to a track and the geometry of the detector/material. For the hits two things have to be checked, namely that their simulated true positions behave as expected, and that the simulation of the measurement errors works as expected.

The module TrueHitTester, developed by the author, reads in the exact hits in the SVD and PXD, called TrueHits. From them it calculates the change in the 5 helix parameters as well as the energy loss and scattering angles $\Delta\Phi$ and $\Delta\theta$ for every silicon layer. Figure 5.1 shows the comparison of two multiple scattering models implemented in GEANT4. It was produced with the output of the TrueHitTester module and is an example of which kind of tests can be done to validate the input. In this case, unexpected differences in the fit results of μ and π particles could be traced to the the fact that GEANT4 9.5 patch-01 (with default physics list "QGSP_BERT") uses a multiple scattering model for μ (the Wentzel model) which is different from the ones for all other types of particles (the Urban model). Figure 5.1 clearly shows that the Wentzel model

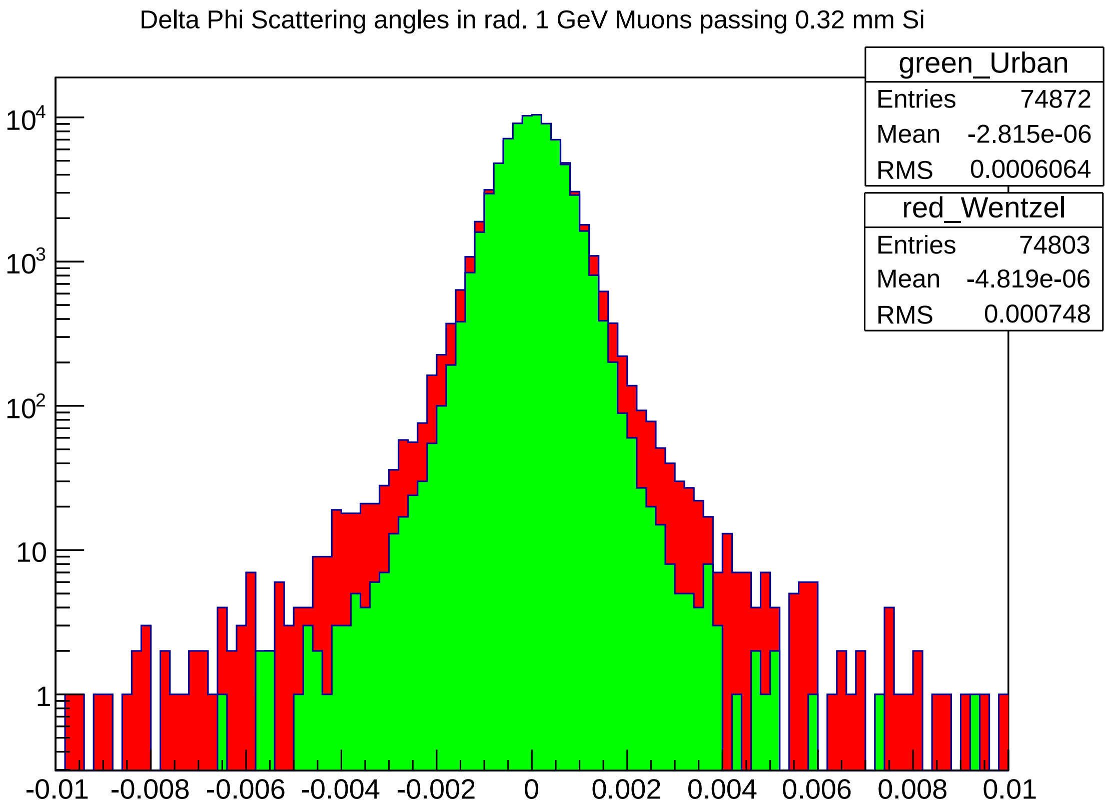


Figure 5.1: Comparison of two different MSC models implemented and used in GEANT4. One can see that the Wentzel model produces heavier tails in the angle distribution than the Urban model.

produces longer tails in the scattering angle distribution of muons passing through one of the 320 micro m Si layers of the SVD detector. The difference is deliberately shown for 1 GeV tracks, as it was particularly large at that particle momentum.

As a consequence, all results shown in this thesis use the Urban multiple scattering model enabling consistent electromagnetic scattering behaviour of all charged particles.

Validation of the smearing of the hits is done by the TrackFitChecker module. It calculates the residuals of the smeared hits with respect to the TrueHits as well as their standard scores, separately for every layer. This makes sure that measurement smearing – also called digitization – works correctly and the a priory covariances of the hits are correctly passed to GENFIT.

As TrueHits are only available for PXD and SVD hits, CDC hits are skipped in these tests. Table 5.1 shows an excerpt of the output of the digitised vs. TrueHit tests produced by TrackFitChecker. Only the output for the PXD layer 1 is shown to keep the table small enough. The simulation producing table 5.1 used Gaussian smearing of the PXD/SVD TrueHits with $\sigma = pitch/\sqrt{12}$. It is therefore no surprise that the standard deviations of the u and v residuals reproduce the values from table 5.2. The

layer			1	
test variable	mean	std	robust mean	robust std
res(u)/cm	−0.0000	0.0014	−0.0000	0.0014
res(v)/cm	0.0000	0.0016	−0.0000	0.0015
pull u	−0.0091	0.9945	−0.0032	1.0038
pull v	0.0111	0.9953	−0.0015	0.9986
χ^2	1.9798	1.9706	1.9273	1.8205

Table 5.1: Mean and standard deviations of the Gaussian-smeared PXD Hits with respect to their TrueHits in layer 1. Robust means truncated mean/std in case of the residuals and χ^2 and median/scaled MAD in case of the pulls. The sample size was 10360. The truncation ratio was 0.005.

standard normal distributed pulls show that the $\sigma = pitch/\sqrt{12}$ for every sensor is written correctly into RecoHits objects defined in GENFIT. The χ^2 values have two degrees of freedom corresponding to the dimensionality of the PXD hits.

5.2 Kalman Filter validation performance

While the Kalman filter is not likely to be used in production, as the implementation in GENFIT does not offer any means of robustification, it is nevertheless important to test it thoroughly, as its code is used by the DAF. Also, the correctness of the track propagation code and the treatment of material effects are easier to check with the Kalman filter than with the DAF. This is because in case of perfect input data and correct implementation the standard test statistics, p-values and pulls, must exactly follow a uniform and a standard normal distribution, respectively. As mentioned in section 2.2.6, this not the case for the DAF.

To find possible shortcomings in the current tracking code, the validation starts with tracks that are as perfect as possible and then turns to more and more realistic ones.

The initial tests are done with tracks from a particle gun, found by the MCTrackFinder (perfect track finding) and using only Gaussian smearing in the tracking detectors. MCTrackFinder was configured to resolve the left right ambiguity of CDC hits so that this information was available to the Kalman filter. In addition, tracks with too few hits were not passed to the fitter, as a meaningful momentum estimation is not possible in this case. The limit was set to an accumulated hit dimensionality ($\sum_i \dim(\boldsymbol{m}_i)$) of 5. Table 5.2 shows the approximate standard deviations used to smear the TrueHits produced by the PXD and SVD simulation to create RecoHits for the track fitting.

The options for CDC digitization followed the recommendations of the CDC group: a Gaussian smearing of the drift length with $\sigma_{\mathrm{DL}} = 130\,\mu\mathrm{m}$. The translation of the measured drift time – the actual output of a CDC measurement – into a drift length adds a small non-Gaussian error to the drift length. This error can be approximated

Layer	R/mm	σ_u/µm	σ_v/µm	t/mm
1	14	14	16	0.42
2	22	14	25	0.42
3	38	15	45	0.32
4	80	22	69	0.32
5	105	22	69	0.32
6	135	22	69	0.32

Table 5.2: Parameters of the Belle II Inner Tracker as currently implemented when using "TrueHits" instead of realistic clusters in basf2. From left to right: layer number, radius R, resolution in u, resolution in v (u and v are orthogonal to each other, v is parallel to the beam direction) and the thickness t of the silicon sensors. σ_u and σ_u are calculated by basf2 using $pitch/\sqrt{12}$. Actually, the pitch size is slightly different in some sensors of a layer, but the values in the table are a good approximation.

by $\sigma_{\text{TR}} = 40/\sqrt{12}\,\mu\text{m}$; therefore the CDC hit variance given to GENFIT is $\sigma^2_{\text{CDCHit}} = \sigma^2_{\text{DL}} + \sigma^2_{\text{TR}}$.

A full digitiser such as the one for the VXD does not exist for the CDC at the moment. The CDC supports smearing of the drift length with one or two Gaussian errors of different width. As mentioned above, the CDC group recommends a single Gaussian smearing of 130 µm. Therefore this was used in all studies presented in this thesis, not just for the validation of the implementation.

The magnetic field in the tracker is $\boldsymbol{B} = (0, 0, 1.5)\,\text{T}$. Again, as the final magnetic field maps are not yet fully developed, this constant field was used in the all results shown in this thesis. There was no misalignment of the tracking detectors, the simulation and the track fitter always used the same fully detailed geometry , the correct particle type was always known to the fitter, and the true track parameters were used as the seed values for the fit.

5.2.1 High-momentum tracks

First track fit results of different particles types at high momenta were tested by shooting 10000 of them with randomly chosen sign, an initial momentum of 1 GeV, a uniform ϕ angle distribution and a constant $\theta = 90°$ through the full simulated Belle II detector. including all tracking detectors and all passive material, such as the beam pipe, support structures, etc.. The origin of the particle gun tracks was always set to $(0, 0, 0)$. For all successfully fitted tracks the distribution of the p-values was filled into a histogram. Then a cut on the p-value of 0.0005 was applied. The remaining tracks were propagated the point of closest approach to their true origin, where the impact parameters (res(x), res(y), res(z)), momentum and transverse momentum residuals (res(p), res(p_{T})) and reduced transverse momentum residual (res($p_{\text{T}}/p_{\text{T,t}}$) $= (p_{\text{T}} - p_{\text{T,t}})/p_{\text{T,t}}$ as well as the pulls of the x, y, z coordinates and p and p_{T} were calculated. The table accompanying the p-value histogram shows the mean and standard deviation of all these quantities, as well as the truncated mean and standard deviation in case of the residuals, and median and scaled MAD in case of the pulls. The truncation cuts away the the largest and smallest values, in total 0.5 % of sample. For the MAD, the scaling factor for a Gaussian distribution was used. “Outlier” is defined as a value further away from the median than 4 times the scaled MAD.

Figure and table 5.2 show the results of this test for $\mu^\pm$ particles. Muons are easier to fit correctly than other particles for three reasons: they do not undergo hadronic interactions; they have a long life time and are unlikely to decay in the tracker; and material interaction via bremsstrahlung can be neglected for the muon momenta occurring in Belle II. Indeed, the p-value distribution is nearly perfectly flat. Only 32 tracks have a p-value below 0.0005 (approximately 5 are expected in a perfectly uniform p-value distribution). The scaled MAD of the pull distributions show that the core follows a standard normal distribution. 139 tracks are identified as outliers in the momentum pull distribution at the origin.

The next figure and table 5.3 show the result of a test with identical conditions to figure and table 5.2, apart from a change of the particle type from muons to kaons. 88 tracks of the 10000 tracks could not be fitted. One fit failed, and 87 tracks had too few hits.

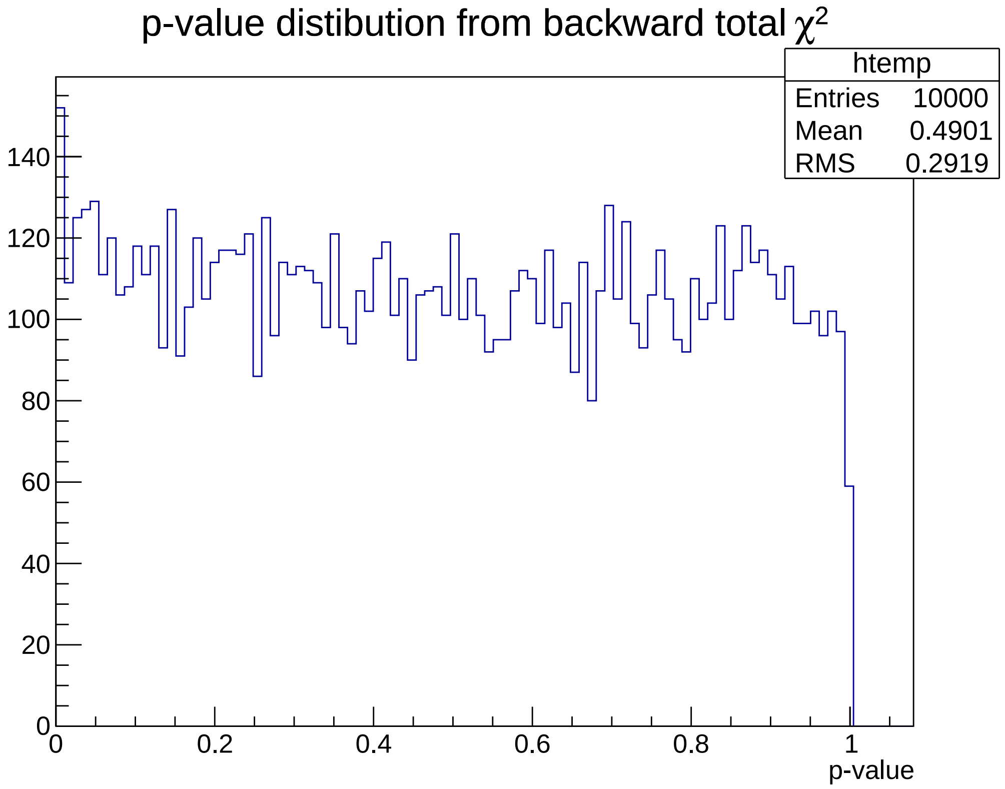
The p-value distribution is as flat as in the muon test case shown in figure 5.2, but has 94 instead of 32 tracks with a p-value under 0.0005. This is due hadronic interactions causing extreme scattering angles. As hadronic interactions are neglected in the statistical treatment of material effects, they necessarily lead to bad p-values when they occur.

As mentioned in section 2.3.2, electrons and positrons are a special case in track fitting, because in their case bremsstrahlung is the most important material effect. Figure and table 5.4 show the test results of 10000 electrons. Again everything was kept the same except the particle type. 88 tracks were not fitted, one because it did not have enough hits, and 87 fits failed. Most of the failures occurred because the estimated momentum fell below a lower limit, set to 4 MeV in GENFIT. A closer inspection of these tracks with the event display revealed that they showed an extreme increase of their curvature due to the emission of a bremsstrahlung photon carrying away most of their energy.

The p-value distribution is clearly not uniform, which is to be expected as one cannot model the highly non-Gaussian distribution of bremsstrahlung in such a way that a least-squares estimator produces perfect p-values distributions. The same fact is visible in the width of the momentum pull distribution. The standard deviation is much larger than 1, while the scaled MAD representing the width of the core is much smaller than 1.

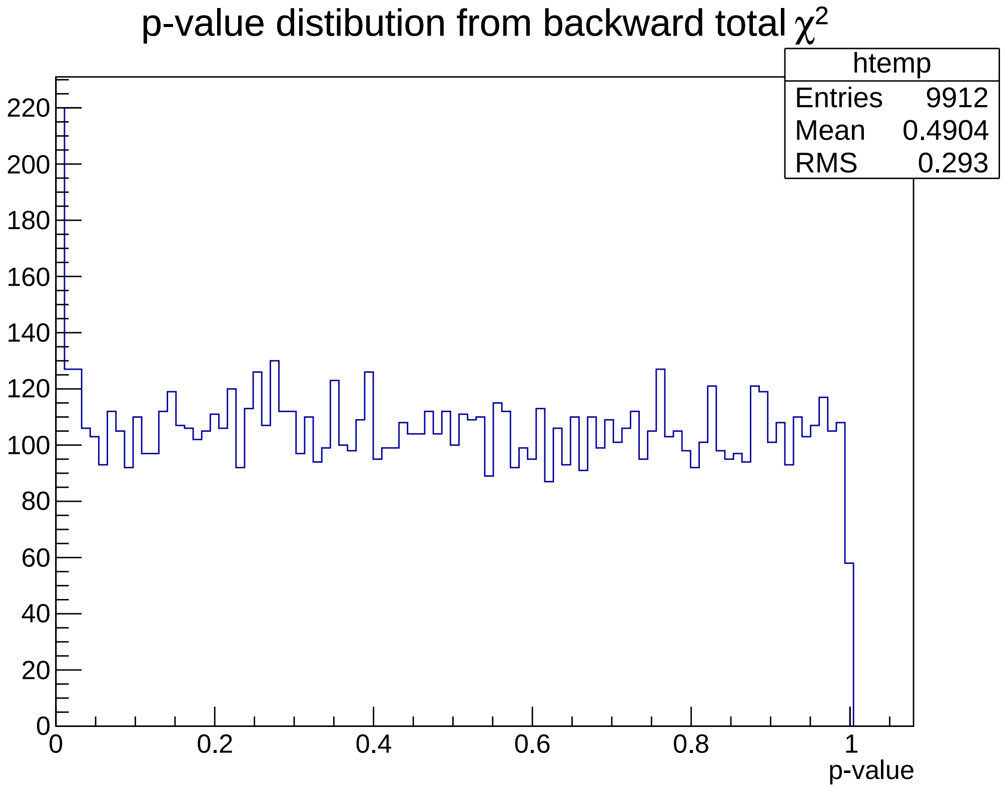
It is important to note that the Kalman filter is still able to estimate all track parameters of electrons without bias. While the p_T and p pulls seem to have an bias in figure and table 5.4 – its direction depending on whether one looks at the standard deviation or the MAD – this is merely an artefact of the outliers in the reconstructed momentum and the fact that fitted momentum is always non-Gaussian, but much more so in the case of electrons. An inspection of the reconstructed curvature and the pulls of the momentum components (pull p_y, pull p_y, pull p_x) reveals an unbiased distribution.

GENFIT currently uses a slightly modified version of equation (2.42) to calculate the



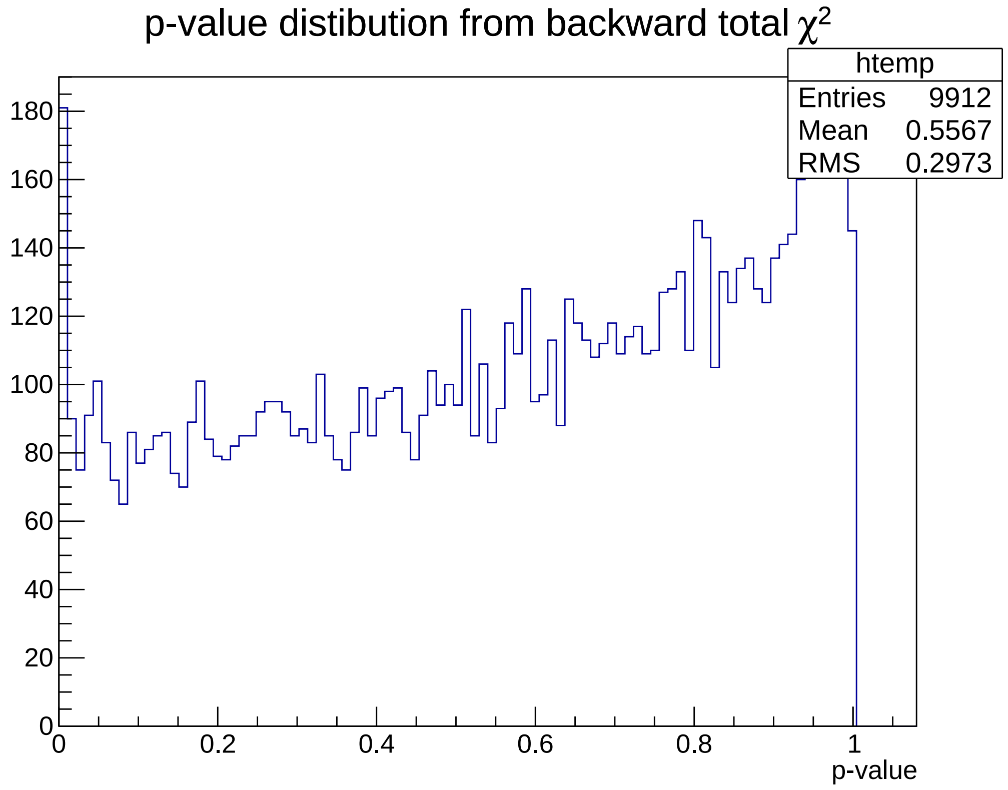
test variable	mean	std	trunc. mean	trunc. std	
res(x)/cm	0.0000	0.0017	0.0000	0.0016	
res(y)/cm	0.0000	0.0017	0.0000	0.0015	
res(z)/cm	0.0000	0.0026	0.0000	0.0025	
res(p_{T})/GeV	0.0003	0.0296	0.0001	0.0022	
res(p)/GeV	0.0003	0.0296	0.0001	0.0022	
res($p_{\mathrm{T}}/p_{\mathrm{T,t}}$)	0.0000	0.0026	0.0000	0.0025	
res(r)/cm	0.0018	0.0015	0.0018	0.0014	
test variable	mean	std	median	scaled MAD	outlier
pull x	−0.0005	1.1197	0.0132	1.0269	29
pull y	0.0149	1.1270	0.0138	1.0257	27
pull z	0.0004	1.0888	0.0102	1.0174	24
pull p_{T}	0.0178	1.5546	0.1473	1.0438	139
pull p	0.0185	1.5545	0.1478	1.0435	139

Figure 5.2: Fit results of 10000 1 GeV $\mu^{\pm}$ generated by the particle gun. See 5.2.1 for details of the setup and the explanation of the quantities shown in the table. 32 tracks had a p-value below 0.0005 and were not used in the table. The truncation ratio was 0.005.



test variable	mean	std	trunc. mean	trunc. std	
res(x)/cm	0.0000	0.0049	0.0000	0.0017	
res(y)/cm	−0.0000	0.0046	0.0000	0.0017	
res(z)/cm	−0.0000	0.0065	0.0000	0.0027	
res(p_T)/GeV	0.0108	0.4532	0.0002	0.0207	
res(p)/GeV	−0.0059	0.4889	−0.0003	0.0166	
res$(p_\mathrm{T}/p_{\mathrm{T},\mathrm{t}})$	−0.0000	0.0065	0.0000	0.0027	
res(r)/cm	0.0021	0.0064	0.0019	0.0015	
test variable	**mean**	**std**	**median**	**scaled MAD**	**outlier**
pull x	0.0108	2.9172	−0.0027	1.0086	40
pull y	−0.0272	2.8127	−0.0047	1.0088	38
pull z	−0.0114	2.7659	−0.0096	1.0494	36
pull p_T	−0.1336	1.4791	−0.1084	1.0821	18
pull p	−0.1218	1.2969	−0.1054	1.0822	15

Figure 5.3: Fit results of 9912 1 GeV K$^\pm$ generated by the particle gun. See 5.2.1 for details of the setup and the explanation of the quantities shown in the table. 94 tracks had a p-value below 0.0005 and were not used in the table.



test variable	mean	std	trunc. mean	trunc. std	
res(x)/cm	−0.0001	0.0024	−0.0000	0.0019	
res(y)/cm	−0.0000	0.0024	−0.0000	0.0020	
res(z)/cm	−0.0000	0.0027	−0.0000	0.0026	
res(p_T)/GeV	0.0041	0.0897	0.0056	0.0800	
res(p)/GeV	0.0041	0.0897	0.0056	0.0800	
res($p_\mathrm{T}/p_{\mathrm{T,t}}$)	−0.0000	0.0027	−0.0000	0.0026	
res(r)/cm	0.0023	0.0025	0.0022	0.0018	
test variable	mean	std	median	scaled MAD	outlier
pull x	−0.0200	1.0730	−0.0079	0.9866	35
pull y	−0.0241	1.0562	−0.0221	0.9847	32
pull z	−0.0071	1.0848	−0.0042	1.0369	16
pull p_T	−0.1325	2.3091	0.1246	0.4224	313
pull p	−0.1325	2.3091	0.1246	0.4224	313

Figure 5.4: Fit results of 9912 1 GeV $e^\pm$ generated by the particle gun. See 5.2.1 for details of the setup and the explanation of the quantities shown in the table. 83 tracks had a p-value below 0.0005 and were not used in the table. The truncation ratio was 0.005.

increase of the momentum variance. The difference is an additional factor of 1.44. Tests showed that it hardly makes a difference whether this factor is used or not. It was probably introduced in one of the projects GENFIT drew code from, to manually tune the bremsstrahlung variance in order to achieve a certain value in a test.

One has to decide what is more important: a p-value distribution having a mean of 0.5, or a pull width at the interaction point close to 1. It is evident from figure and table 5.4 that one cannot have both using an unmodified Kalman filter. A factor 0.25 in the variance leads to a p-value distribution with mean 0.5 for the 1 GeV electrons, but leads to even wider momentum pulls. Additionally this tuning would require momentum depended factors.

5.2.2 Low-momentum and curling tracks

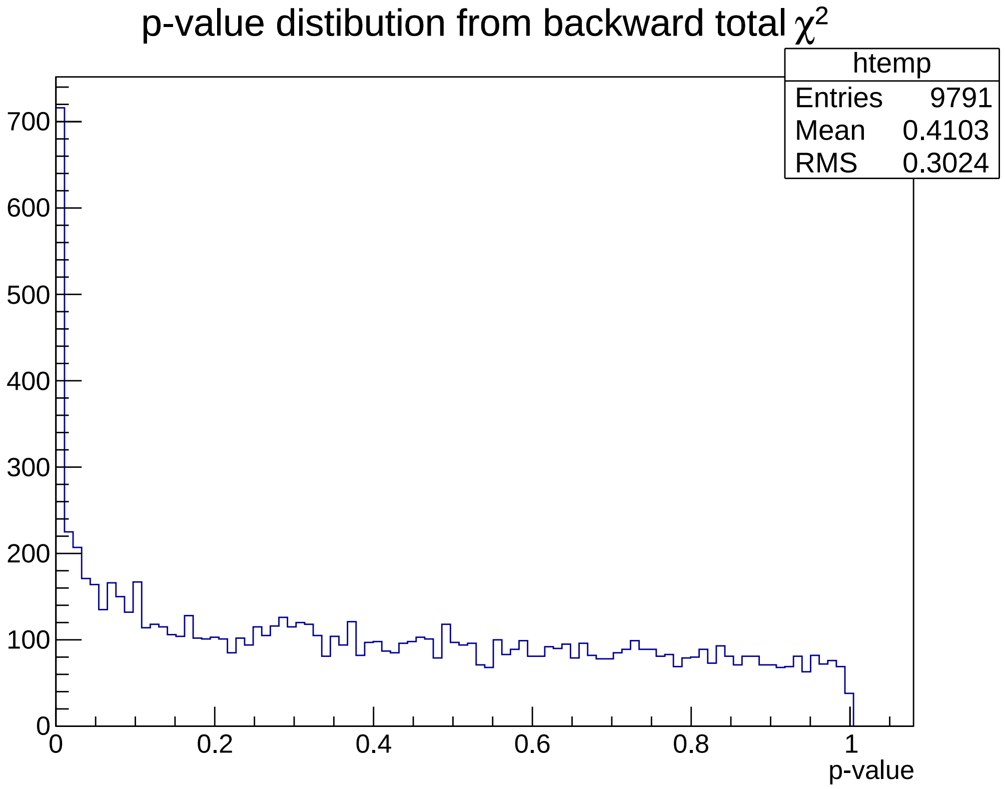
At low momenta electromagnetic multiple scattering and energy loss become so strong that some particles of all types exhibit the properties that makes the fitting of electrons difficult. Both effects are non-Gaussian distributed, and therefore the p-value distribution cannot in principle be perfectly flat. At higher momenta, the absolute deviations from the mean energy loss or the Gaussian distribution of the scattering angle are so small that this fact stays hidden under the measurement error, leading to the flat p-value distribution shown in figure 5.2. At sufficiently momentum, some tracks suffer such a strong material interaction that the track will be an outlier in terms of the p-value; in extreme cases the fit fails completely.

Figure 5.5 shows the fit results of 10000 particle gun generated tracks with 100 MeV initial momentum and a fixed θ angle of 85°. All other parameters were the same as in section 5.2.1. These tracks do not leave the CDC; instead after generating hits there, curl back into the VXD detector, creating additional hits there.

Three of the 10000 tracks in figure and table 5.5 had not enough hits and were not passed to the fitter. 206 fits failed, mostly because the estimated momentum fell below the minimum momentum threshold of 4 MeV, which is hard-coded in GENFIT. As in the case of the 1 GeV electrons, most of the “momentum too low” errors are caused by extremely strong changes in the curvature of the true particle trajectory.

Because most of the hits of the 100 MeV tracks are still in the CDC, the shape of the p-value distribution in figure and table 5.5 is dominated by the CDC contribution to the track fit. On average the p-values are too small, indicating an underestimation of the errors. However, this effect is not strong enough to influence the core width of the pull distributions as shown by the MAD of the pulls in figure and table 5.5. Again all estimated track parameters are completely unbiased. This was verified by inspecting the distributions of the pulls of the momentum components p_x, p_y and p_z.

In order to illustrate the strong material effects alluded to above, two tracks from the 100 MeV muon sample are shown in figures 5.6 and 5.7. The track in figure 5.6 it typical for the 100 MeV sample. It generates 6 hits in the VXD, many hits in the CDC, and then re-enters into the SVD where it generates further hits before it decays or is absorbed in the material. An entirely different picture is shown by the track in figure 5.7, which curls several times trough all tracking detector rapidly losing energy. The p-value



test variable	mean	std	trunc. mean	trunc. std	
res(x)/cm	−0.0003	0.0196	−0.0002	0.0134	
res(y)/cm	−0.0001	0.0214	0.0002	0.0132	
res(z)/cm	−0.0000	0.0220	−0.0000	0.0190	
res(p_T)/GeV	−0.0005	0.0008	−0.0005	0.0007	
res(p)/GeV	−0.0005	0.0007	−0.0005	0.0006	
res($p_\mathrm{T}/p_{\mathrm{T,t}}$)	−0.0001	0.2206	−0.0002	0.1910	
res(r)/cm	0.0158	0.0243	0.0148	0.0122	
test variable	mean	std	median	scaled MAD	outlier
pull x	−0.0403	2.0323	−0.0169	1.0374	83
pull y	−0.0150	1.8968	0.0065	1.0400	89
pull z	−0.0012	1.3565	−0.0048	1.0453	71
pull p_T	−0.7717	1.1449	−0.6657	1.0495	33
pull p	−0.7603	1.1181	−0.6461	1.0398	28

Figure 5.5: Fit results of 9791 100 MeV $\mu^\pm$ with $\theta = 85°$ generated by the particle gun. See 5.2.2 for details of the setup and the explanation of the quantities shown in the table. 310 tracks had a p-value below 0.0005 and were not used in the table. The truncation ratio was 0.005.

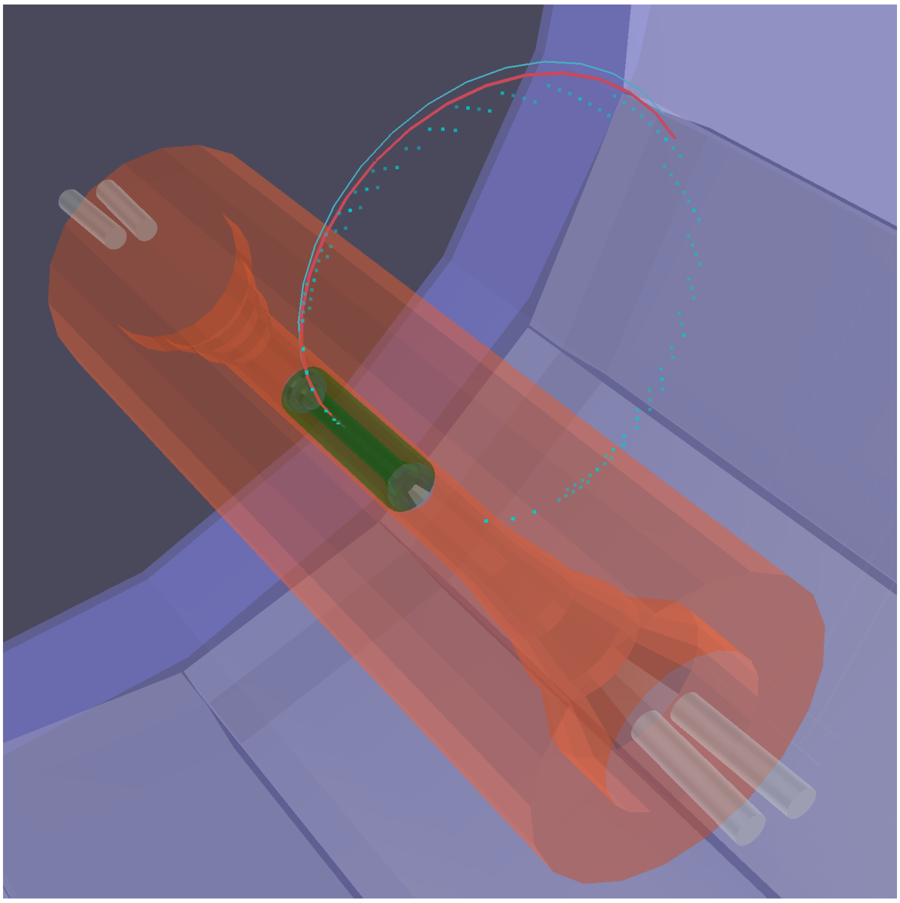


Figure 5.6: A typical track from the 100 MeV muon tracks sample. The PXD is coloured green, the SVD orange. The wires of the CDC are not shown. The SimHits created by GEANT4 are turquoise.

of this track is obviously very bad, as the average momentum loss calculated by the Bethe–Bloch formula is nowhere near the actual energy loss.

At around a transverse momentum of 60 MeV a track will not make hits in the CDC any more. Of course there are a few exceptions as a track can be scattered so strongly in the direction of the CDC that it does enter it. To investigate how well track parameters can be reconstructed using only the 6 VXD layers, a sample of 10000 60 MeV $\mu^{\pm}$ with $\theta = 80°$ was generated, while all other parameters stayed the same.

Figure and table 5.8 show the fit results of this sample. Three tracks had too few hits and 484 fits failed. Again, most of them failed because the estimated momentum fell below the threshold in GENFIT. There are several noteworthy aspects of the low-momentum tracking illustrated by the fitted 60 MeV sample. First one can see an overestimation of the errors of the fitted track parameters. This is visible in the p-values, which are too large on average, and in the momentum pull distributions, which are too narrow. Besides the fact that the widths are also too narrow when looking at the non robust standard deviation, this effect looks similar to the 1 GeV electron test sample.

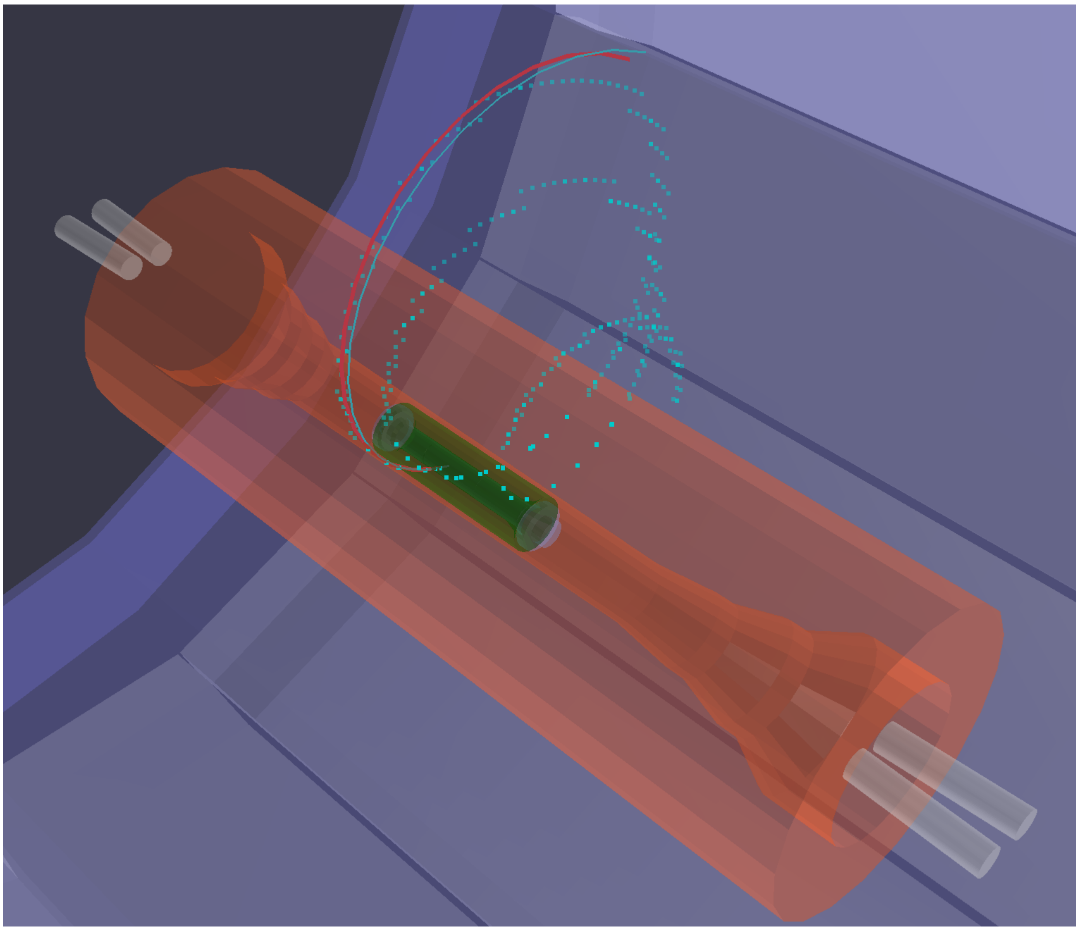
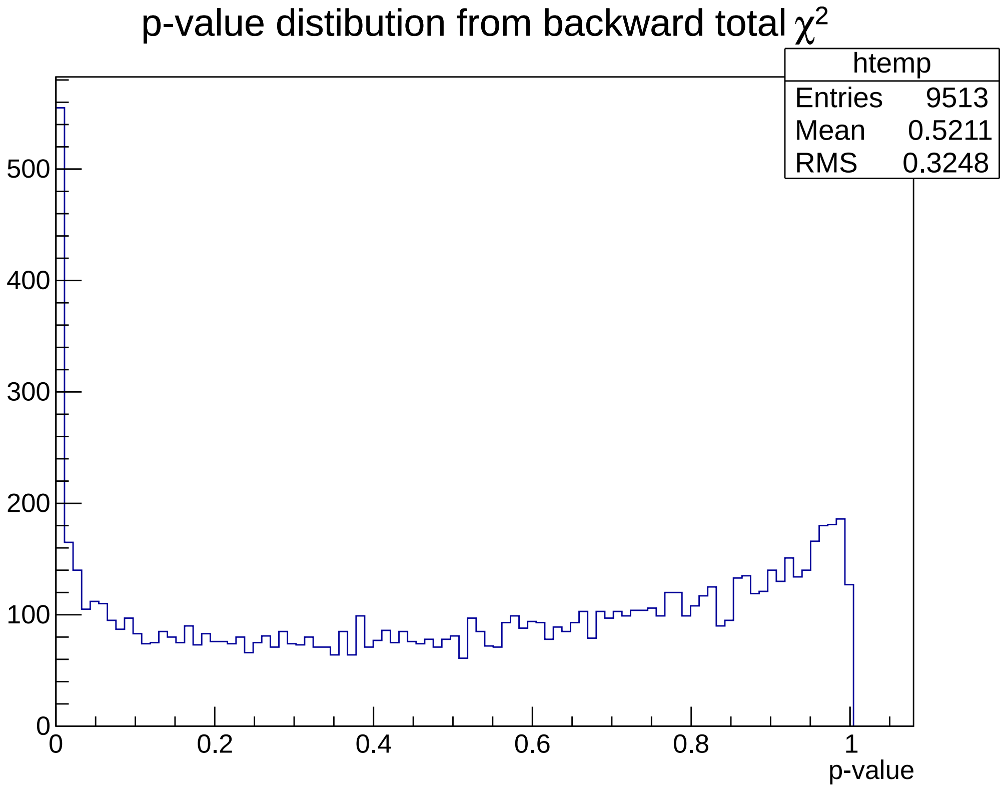


Figure 5.7: A track from the 100 MeV muon tracks sample suffering by chance from very strong material interactions. See the caption of figure 5.6 for the meaning of the colours.

Further tests, however, showed this is very likely not caused by the non-Gaussian distribution of the material effects. The overestimation of tracking errors at low momenta in the VXD was investigated in detail, and it was concluded that the most like cause is a problem in the tracking software. Section 5.5 documents the attempts to solve this problem. Despite this issue in error estimation, the means of the pull distributions of p_x, p_y, p_z were unbiased.

The other noteworthy aspect is the fact that there are basically no VXD only curling tracks. Virtually all 60 MeV tracks are stopped either before reaching the CDC or before re-entering the SVD. It makes therefore no sense to search for curling tracks in this momentum range. Indeed the exclusion of the few additional hits some very low momentum tracks have after penetrating the last SVD layer, improves the track fit result.

This fact is illustrated by figure and table 5.9. The initial track parameters in this sample were kept identical to the test case shown in figure and table 5.8, but the MTrackFinder was configured to exclude CDC hits, and the magnetic field was set to 0 outside



test variable	mean	std	trunc. mean	trunc. std	
res(x)/cm	0.0003	0.0378	0.0003	0.0315	
res(y)/cm	0.0003	0.0382	0.0002	0.0315	
res(z)/cm	0.0013	0.0471	0.0011	0.0430	
res(p_{T})/GeV	−0.0004	0.0026	−0.0004	0.0024	
res(p)/GeV	−0.0003	0.0025	−0.0003	0.0024	
res($p_{\mathrm{T}}/p_{\mathrm{T,t}}$)	0.0214	0.7968	0.0193	0.7284	
res(r)/cm	0.0374	0.0386	0.0358	0.0275	
test variable	mean	std	median	scaled MAD	outlier
pull x	0.0056	1.2926	0.0091	1.0120	58
pull y	0.0047	1.3045	0.0023	1.0014	54
pull z	0.0234	1.2246	0.0278	1.0071	52
pull p_{T}	−0.1090	0.6361	−0.1184	0.5821	29
pull p	−0.0970	0.6241	−0.1100	0.5710	28

Figure 5.8: Fit results of 9513 60 MeV $\mu^{\pm}$ generated by the particle gun. See 5.2.2 for details of the setup and the explanation of the quantities shown in the table. 253 tracks had a p-value below 0.0005 and were not used in the table. The truncation ratio was 0.005.

the SVD. In this test case 1 track had to few hits, and 271 fits failed, mostly because the estimated momentum was too low. Compared to the muon sample without the magnetic field cut, there are fewer failed tracks and fewer tracks with very small p-values. This does not mean that GENFIT has a problem with fitting curling tracks, but rather that the few additional hits in the sample without magnetic field cut lie far away from a helix extrapolation with deterministic energy loss because of the strong random material effects. As it would be difficult to find these additional hits with a realistic track finder, the results in figure and table 5.9 are actually more representative for this momentum range.

5.2.3 Tuning of the detector plane size in GENFIT

When GENFIT propagates to the next hit in the track candidate, this hit is by default defined on an infinite plane, in order not to miss the measurement plane. To make fitting of curling tracks possible, the measurement planes on which the VXD hits are defined must have finite size; otherwise the propagation could hit the measurement plane from the wrong side in case of curling tracks. If, on the other hand, the measurement plane in GENFIT would have just the physical size of the sensor, the propagation would sometimes miss the plane, and the fit would fail.

GENFIT offers the possibility to set finite measurement planes with an an arbitrary “overlap size” around the physical sensor size. This overlap has to be tuned such that the total number of failed fits, either because of missing the plane or because of hitting the plane from the wrong side, is minimised.

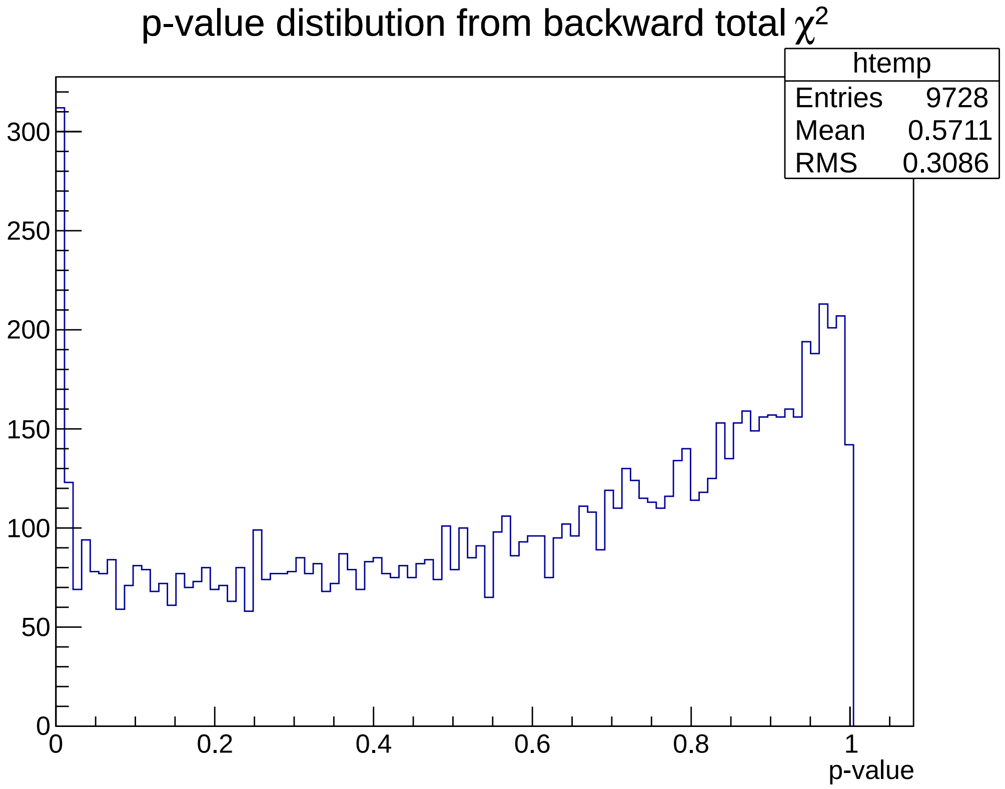
This tuning should be done under the most realistic conditions, but as there is currently a very high rate of failing track fits when using the basf2 track finders in combination with the the current GENFIT (more about this in section 5.4), the author decided to use EvtGen generated tracks “found” by the MCTrackFinder for this tuning.

A fixed sample of simulated tracks from EvtGen was fitted with systematic altered overlap sizes. A relative large overlap of 20 cm was selected and uploaded to the basf2 software repository. To yield optimal results this study should be repeated in the future, when the combination of track finders and the track fitter yields more stable results.

5.2.4 Tracks from simulated electron positron collisions

In basf2, the program EvtGen is used to simulate the decays of the $\Upsilon(4s)$ resonance. This simulation leads to the following particle types leaving hits in the tracking detectors: 71.6 % $\pi^{\pm}$, 14.5 % $K^{\pm}$, 7.0 % $e^{\pm}$, 4.6 % $\mu^{\pm}$ 2.3 % p and p^{-}, and finally a very low number – under 0.05 % – of “other” particles, mainly the ions of H and He isotopes, and unstable baryons like Ξ or Σ.

A sample of 6000 events was generated, using EvtGen and the full basf2 detector simulation. The tracks were then reconstructed with the MCTrackFinder and the Kalman filter, using the same settings and under the same conditions as in 5.2.1. This means that the fitter still gets optimal information such as perfect seed values and information how



test variable	mean	std	trunc. mean	trunc. std	
res(x)/cm	0.0003	0.0363	0.0003	0.0316	
res(y)/cm	−0.0001	0.0390	−0.0003	0.0317	
res(z)/cm	0.0013	0.0471	0.0013	0.0435	
res(p_T)/GeV	−0.0002	0.0027	−0.0002	0.0026	
res(p)/GeV	−0.0001	0.0027	−0.0001	0.0026	
res($p_\mathrm{T}/p_{\mathrm{T,t}}$)	0.0225	0.7970	0.0213	0.7361	
res(r)/cm	0.0377	0.0377	0.0361	0.0276	
test variable	**mean**	**std**	**median**	**scaled MAD**	**outlier**
pull x	0.0120	1.2504	0.0144	1.0265	55
pull y	−0.0074	1.3390	−0.0295	1.0098	66
pull z	0.0259	1.2223	0.0295	1.0448	61
pull p_T	−0.0587	0.6628	−0.0800	0.6217	19
pull p	−0.0471	0.6553	−0.0741	0.6163	21

Figure 5.9: Fit results of 9513 60 MeV $\mu^\pm$ generated by the particle gun. See 5.2.2 for details of the setup and the explanation of the quantities shown in the table. 154 tracks had a p-value below 0.0005 and were not used in the table. The truncation ratio was 0.005.

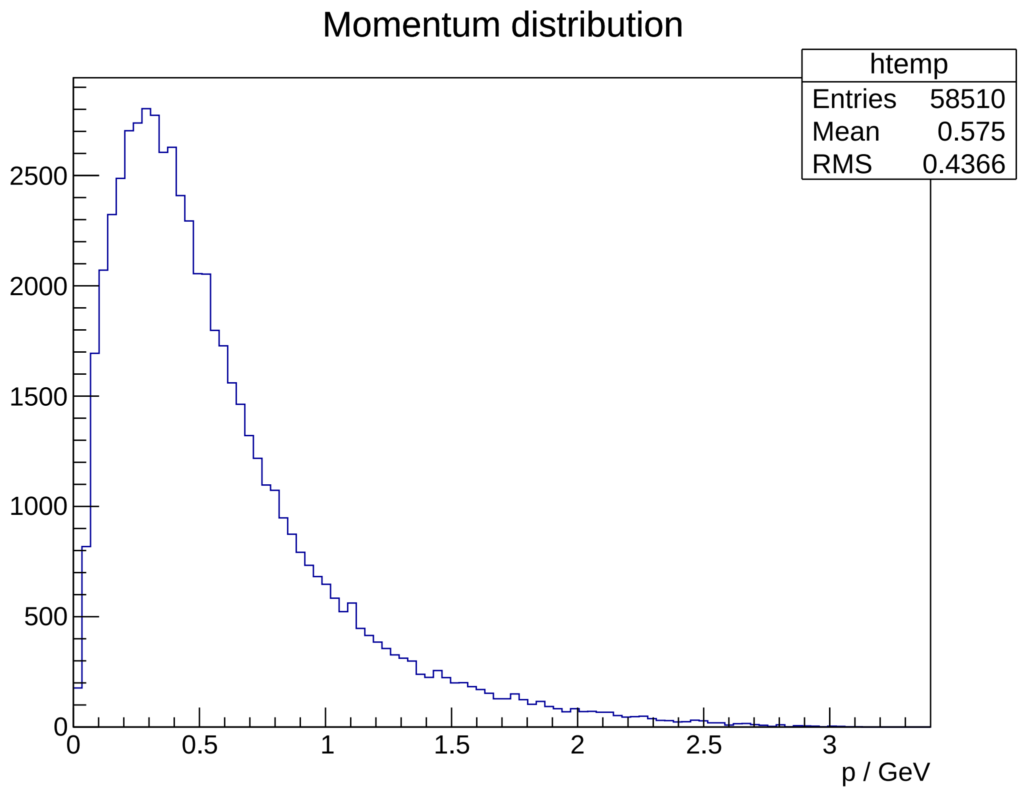
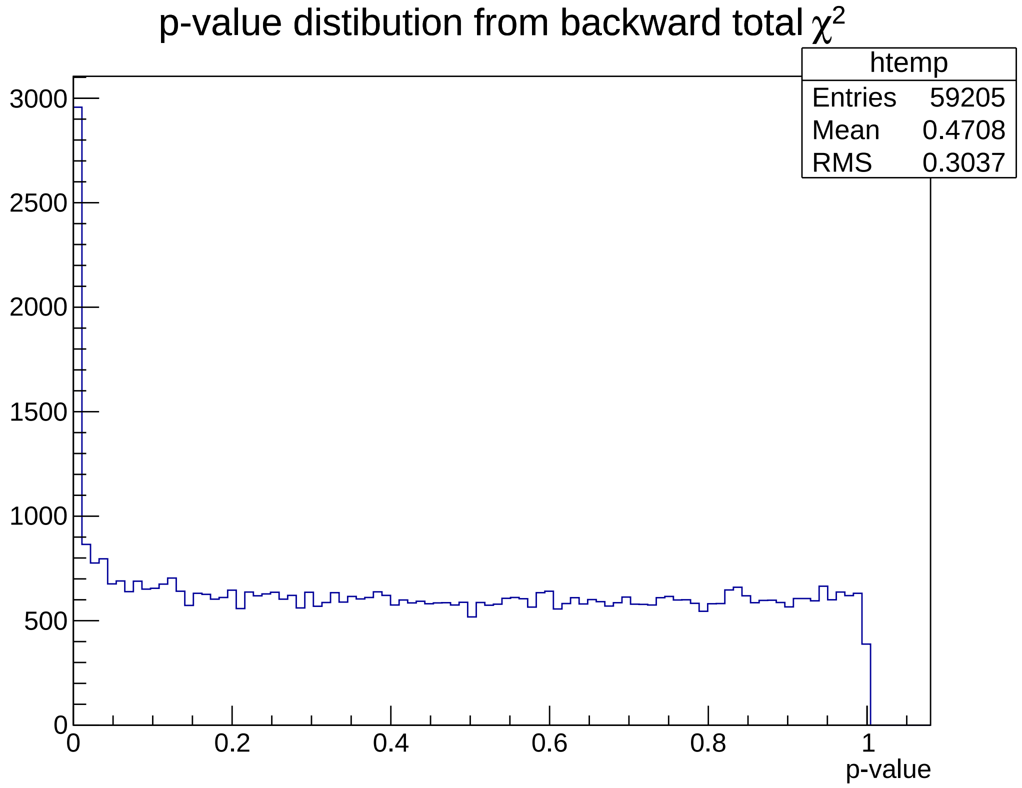


Figure 5.10: The momentum distribution of charged particles created by 6000 EvtGen events reaching at least the PXD detector.

to resolve the wire hit ambiguity. The MCTrackFinder created 60402 tracks candidates from the 6000 events.

The particles in this sample have a wide range of initial momenta and directions. The momentum distribution of the particles generated by EvtGen and reaching at least the PXD detector is shown in figure 5.10.

Figure and table 5.11 shows how the Kalman filter performs when fitting these 60402 tracks. While now all particle types and initial momenta and angles are present in the track sample, all other conditions were kept perfect as described in section 5.2.1. 1195 fits failed. The overall p-value distribution is flat with the exception of the large peak in the first bin. This peak comes as no surprise, as it also present in the tests of low-momentum tracks generated by the particle gun. Indeed, the momentum distribution of the EvtGen generated tracks in figure 5.10 shows the large portion of low-momentum tracks in the sample. The residuals or impact parameters shown in Figure and table 5.11 have relatively heavy tails besides a p-value cut of 0.01 applied to the sample before creating the table. The scaled MAD is a measure of the width of the core of the residuals if it were Gaussian. The large differences between the scaled MAD, the 10 % truncated standard deviation and the normal standard deviation of the residuals illustrates their heavy tails.



test variable	mean	std	trunc. mean	trunc. std	scaled MAD
res(x)/cm	0.0012	0.4884	−0.0000	0.0044	0.0037
res(y)/cm	−0.0004	0.3028	0.0000	0.0044	0.0037
res(z)/cm	0.0104	0.7965	0.0001	0.0043	0.0041
res(p_T)/GeV	0.0029	0.2614	−0.0000	0.0019	0.0016
res(p)/GeV	0.0009	0.3038	−0.0001	0.0022	0.0016
res$(p_\mathrm{T}/p_\mathrm{T,t})$	0.1532	8.2633	0.0004	0.0204	0.0100
res(r)/cm	0.0208	0.4961	0.0011	0.1352	0.0037
test variable	**mean**	**std**	**median**	**scaled MAD**	**outlier**
pull x	−0.0049	4.8093	−0.0068	1.0365	433
pull y	0.0113	6.5889	0.0001	1.0336	452
pull z	0.0417	5.2475	0.0057	1.0420	476
pull p_T	−0.1331	3.2588	−0.0523	1.0266	743
pull p	−0.2398	2.2897	−0.0788	1.0413	736

Figure 5.11: Results of 59205 successful track track fits generated by EvtGen, using Gaussian smearing in the VXD. The total number of track candidates was 60402. See 5.2.4 for details of the setup and the explanation of the quantities shown in the table. 2884 tracks had a p-value below 0.01 and were not used in the table. The truncation ratio was 0.1.

In the next test case Gaussian smearing of the true hits in PXD and SVD was replaced by realistic digitization and clustering of pixels and strips. The results are shown in figure and table 5.12. The same sample of simulated tracks as before was used – just the hit digitization was changed. This change involves only a minority of the hits, as the CDC digitization stayed the same. Consequently the differences between figure and table 5.12 and 5.11 are small. The ones that are visible are easily explained by the characteristics of the realistic VXD hit digitization.

Realistic variances of the measurement error were not yet available in the digitiser. The VXD software group is currently working on error estimates which may even depend on the local incident angle of the track, as given by the predicted Kalman filter state at a VXD sensor.

As a consequence the author had to make his own approximate assessment of the cluster measurement errors. When only VXD clusters were used for fitting, $\sigma = 1/\sqrt{2} \cdot pitch/\sqrt{12}$ resulted in a p-value distribution with a mean relatively close to 0.5. This value was therefore used in all test cases using VXD cluster hits.

The realistic digitization produces heavier tails than a Gaussian smearing, so even if the core width is much smaller there will be more outliers. This is the reason for the higher number of failed fits – 2647 out of 60483 compared to 1195 out of of 60402 – and more outliers in the pull distribution in figure and table 5.12. The slightly larger total number of tracks in the sample (60483 instead of 60402) is caused by the slightly higher number of hits the realistic digitization produces compared to the number of TrueHits.

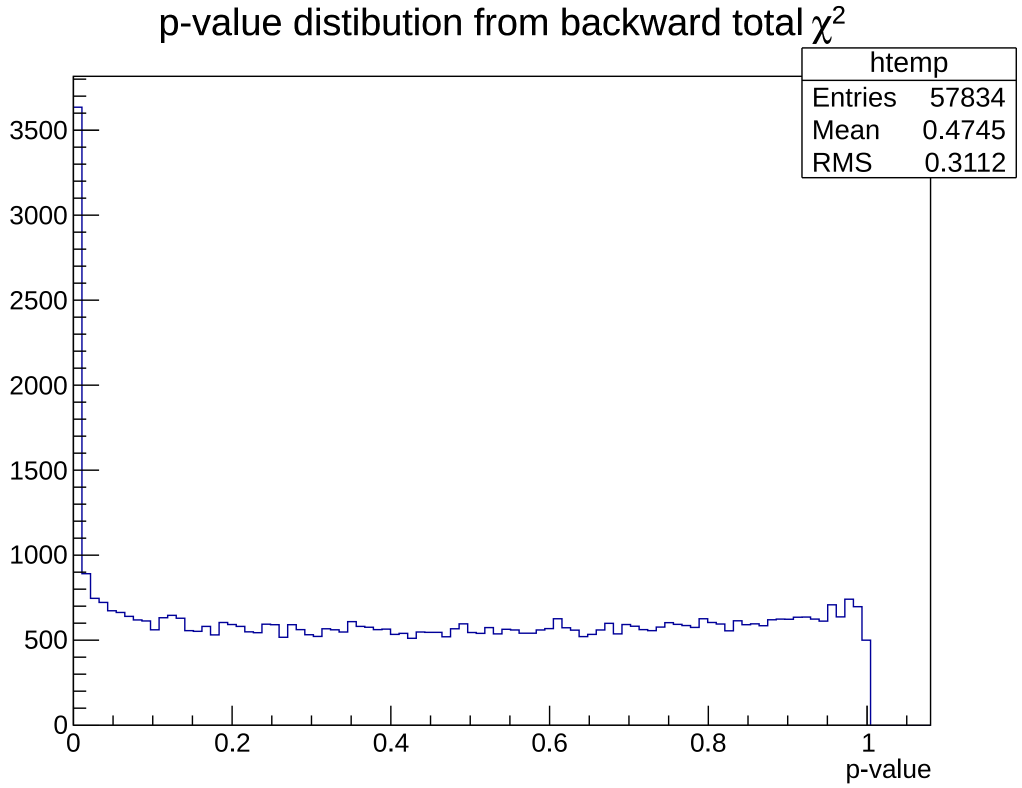
Most importantly, the the Kalman filter still delivers unbiased track parameters. This is evident by the fact that all pull distributions have a mean and median close to 0; again, while p and p_T do not show this, p_x, p_y and p_z do.

5.3 DAF validation and performance

Like the Kalman filter, the DAF was first extensively tested in a controlled environment, using the particle gun and the VXDSimpleBackground module for generating artificially positioned background hits. The results of this study were presented by the author at the CHEP 2012 conference and can be obtained from the proceedings [29] free of charge. This study shows the detection of outlier hits works very well in both cases in which outliers are passed to the track fit: either signal and background hits compete with each other in a measurement layer, or the track candidate contains only an outlier hit in a layer.

The GENFIT implementation of the DAF also offers the possibility to resolve the left/right ambiguity of wire hits, by duplicating the hits internally with opposite signs and let DAF decide which sign fits the track better. This feature was also tested with particle gun tracks were DAF chose correctly with over 99 % accuracy.

The DAF is designed to automatically identify and suppress hits not belonging to the fitted track. If all hits in a track candidate really belong to one track and do not suffer from some exceptionally large measurement error, the DAF yields the same result as the Kalman filter. This means that tracks suffering from very strong material effects



test variable	mean	std	trunc. mean	trunc. std	scaled MAD
res(x)/cm	0.0012	0.3416	−0.0000	0.0042	0.0034
res(y)/cm	−0.0015	0.2499	0.0000	0.0042	0.0034
res(z)/cm	0.0109	1.0615	0.0002	0.0040	0.0037
res(p_T)/GeV	0.0013	0.0794	0.0000	0.0019	0.0016
res(p)/GeV	0.0008	0.1101	−0.0001	0.0022	0.0016
res($p_\mathrm{T}/p_\mathrm{T,t}$)	0.1134	9.3091	0.0007	0.0192	0.0091
res(r)/cm	0.0201	0.4161	0.0015	0.0291	0.0035
test variable	**mean**	**std**	**median**	**scaled MAD**	**outlier**
pull x	−0.0005	5.0917	−0.0102	1.0468	481
pull y	0.0065	7.0044	0.0026	1.0492	494
pull z	0.1117	5.8733	0.0648	1.0223	530
pull p_T	−0.1148	3.5006	−0.0445	1.0324	739
pull p	−0.2291	2.3176	−0.0702	1.0349	748

Figure 5.12: Results of 57829 successful Kalman fits. 60483 fits were attempted. Tracks generated by EvtGen, using realistic digitization in the VXD. See 5.2.4 for details of the setup and the explanation of the quantities shown in the table. 3568 tracks had a p-value below 0.01 and were not used in the table. The truncation ratio was 0.1.

are reconstructed as well (or badly) as by the Kalman filter. However, in some special cases the DAF can give better reconstructed track parameters, even if track finding is perfect. For example, a sudden strong change in direction due to multiple scattering in between the very last few hits on the track should lead to the down-weighting of these hits as they appear to the algorithms as outliers and therefore not worsening the result.

Knowing this, it is not surprising that figure and table 5.13 shows similar results as figure and table 5.12, because all settings were the same in both cases, apart from changing the fitting algorithm from the Kalman filter to the DAF. The values for the core width of the residuals and the 10 % truncated std are essentially the same. The pull distributions are centered around 0, and the scaled MAD is 1, as it should be.

No annealing scheme was used and the only temperature value was set to 1. This causes the DAF to iterate over the hits until the weights are settled. The probability cut (α in equation 2.30) was set to 0.001. The DAF was instructed to resolve the wire hit ambiguities, but already got the correct sign for initialisation of the wire hits weights from the MCTrackFinder.

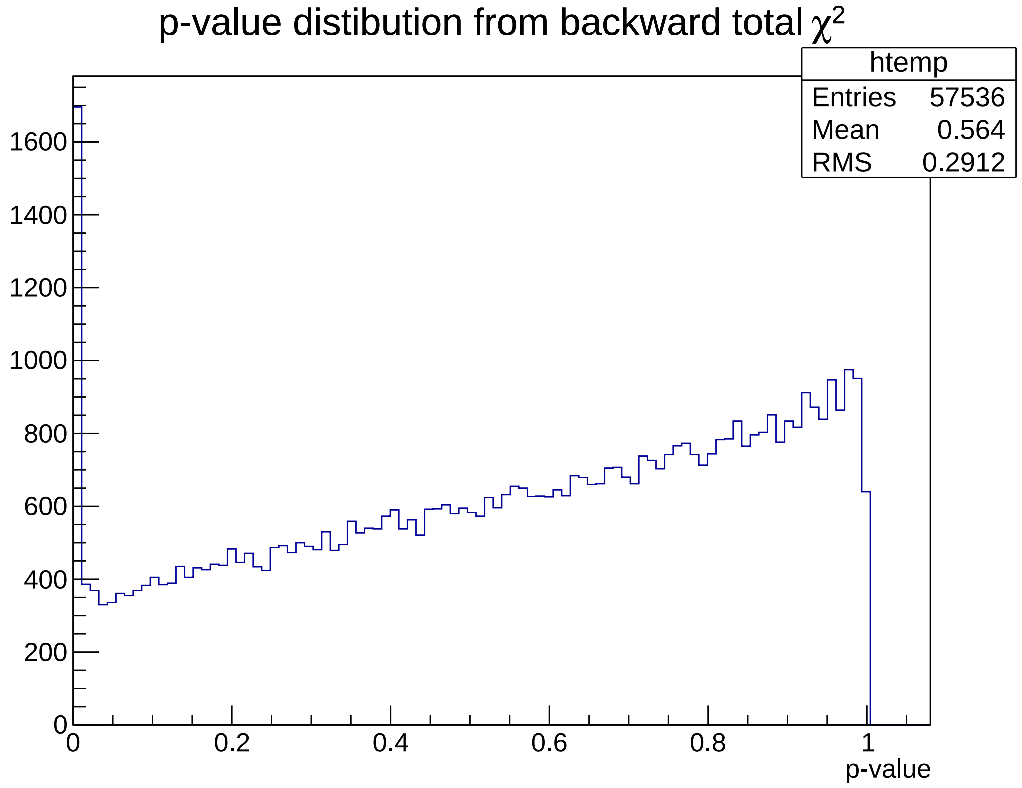
Of course there are several differences that need to be addressed. First the number of successful fits is slightly lower in the DAF case. Because the current GENFIT implementation quits with an error if tracks where the first view hits lead the prediction away form the true track are fitted, an increase in the number of filter iterations can cause a higher number of failed fits as the “dangers” hit combination is processed again with slightly different stating values. The DAF needs 3 fitter iterations for the weights to settle (counting forward plus backward filter as one iterations), while the Kalman filter was set to two iterations.

Second, the p-value distribution looks very different. The lower number of tracks falling into the first bin is likely caused by the better reconstruction of tracks with large material effects in the very first few or very last few hits, as mentioned above. The reason for the excess of high p-values is not fully understood. Some further remarks on this issue can be found in section 5.5.

5.4 Combined track finding and fitting performance

The tests in this section do not use the MCTrackFinder any more, but real track finders. This has two consequences: first, the track candidates passed to the fitter may include hits not belonging to the track; second, the seed values are not the true track parameters any more, but the best ones the track finders are able to produce at the moment.

Currently there are several CDC track finders and the VXD only track finder (VXDTF). There is, however, no “global” track finder yet that combines the information from all tracking detectors. In order to be able to conduct tracking studies using all tracking detectors without having to resort to the MCTrackFinder, the author has written a module called MCTrackCandCombiner for combining CDC and VXD track candidates, using some information from the simulated truth. The version of VXDTF currently available has a bug in the clean-up process of the track candidates: it can end up with multiple track candidates sharing the same hits. The detection of curling tracks is currently still



test variable	mean	std	trunc. mean	trunc. std	scaled MAD
res(x)/cm	0.0003	0.3060	−0.0000	0.0043	0.0034
res(y)/cm	−0.0015	0.2316	0.0000	0.0042	0.0034
res(z)/cm	0.0106	0.7890	0.0002	0.0040	0.0037
res(p_T)/GeV	0.0018	0.0946	−0.0000	0.0019	0.0016
res(p)/GeV	0.0008	0.1502	−0.0001	0.0022	0.0016
res($p_\mathrm{T}/p_{\mathrm{T,t}}$)	0.1225	6.9226	0.0007	0.0193	0.0090
res(r)/cm	0.0217	0.3757	0.0015	0.0261	0.0035
test variable	mean	std	median	scaled MAD	outlier
pull x	−0.0028	6.0165	−0.0106	1.0489	601
pull y	0.0025	7.4070	0.0020	1.0517	614
pull z	0.1343	6.7274	0.0643	1.0243	652
pull p_T	−0.0937	4.0177	−0.0501	1.0389	860
pull p	−0.2354	2.4951	−0.0783	1.0447	796

Figure 5.13: Results of 57536 successful DAF fits. 60483 fits were attempted. Tracks generated by EvtGen, using realistic digitization in the VXD. See 5.3 for details of the setup and the explanation of the quantities shown in the table. 1664 tracks had a p-value below 0.01 and were not used in the table. The truncation ratio was 0.1.

under development; at the time being, such tracks create several track candidates, one for each half circle.

TRASAN – the CDC track finder ported from Belle I – which was chosen for CDC track finding in this thesis, also lacks the ability to find curling tracks, producing multiple track candidates instead. This feature is missing because it was not ported to basf2.

Because of these issues in the track finders – which are clearly only temporary, as developers are currently working on them – the MCTrackCandCombiner also performs a clean-up of the track candidates. If a curling track is found several times, only the candidate with the largest number of hits created by one and the same particle is kept; the other ones are discarded. Also all track candidates having less than 60 % of their hits belonging to the same track are deleted.

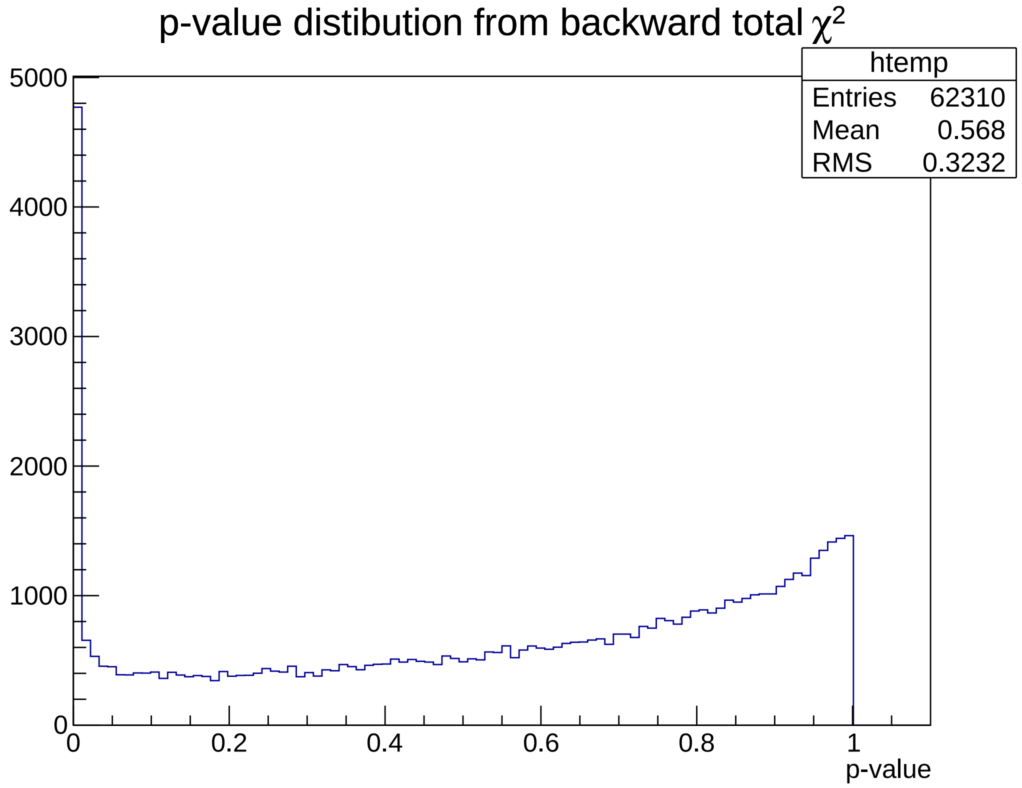
Finally, because the information exchange of the calorimeter and particle identification detectors and the tracking detectors is still in a rudimentary state, MCTrackCandCombiner inserts the mass hypothesis matching the majority of the hits in a track candidate.

The GENFIT version used in this thesis cannot follow a track where the first few or the last few hits are aligned in such a way that the extrapolated track does not point to the next measurement plane, or that the momentum estimate bases on these hits lies below a hard-coded threshold. A robust estimation method like the DAF cannot overcome this problem, as forward and backward filter have to run at least once until the down-weighting of outlier hits can begin. Using realistic finders obvious worsens this problem, and indeed one can see that in the results presented in figure and table 5.14 only 88 % of the track candidates, created by the MCTrackCandCombiner in the way mentioned above, could be successfully fitted. For this test the annealing scheme of the DAF was changed from 1 to 81,9,4,1.

Several choices were made to prevent an even higher rate of failed fits. Although TRASAN is not tuned for the Belle II CDC and does not find curling tracks, it creates three times fewer failed fits that the Legendre finder, which is intended to be the primary CDC track finder in Belle II. This problem is known to the CDC tracking group, but solutions are not ready yet. Although the GENFIT implementation of the DAF is able to resolve the left/right ambiguity of wire hits, this feature had to be deactivated, as it leads to an increase of the fit failure rate if the wire hit weights are not already initialised according to the correct sign. This will of course also be addressed in the upcoming release 2 of GENFIT.

The results with realistic track finding are shown in figure and table 5.13. The core width (scaled MAD) of the position residuals increases by 77 % compared to the test case with perfect track finding (see figure and table 5.14). The strength of the tails of the position residuals increases even more, as indicated by an almost fourfold increase of the 10 % truncated standard deviation. The momentum residuals are much less affected. The pull distributions now have 3 times more outliers after the 0.01 p-value cut, whereas the their core width remains correct, as the scaled MAD is still close to 1.

The large increase of the position residuals compared to the momentum residual indicates a potential gain in fit quality after the implementation of a feature envisaged in the VXDTF, namely to pass competing hits to the DAF, where the final decision whether they really belong to the track or not should be possible with higher accuracy



test variable	mean	std	trunc. mean	trunc. std	scaled MAD
res(x)/cm	−0.0363	5.7092	−0.0002	0.0165	0.0056
res(y)/cm	0.0210	5.8711	0.0000	0.0165	0.0056
res(z)/cm	0.1856	6.5477	0.0009	0.0303	0.0059
res(p_T)/GeV	0.0191	1.5716	0.0001	0.0029	0.0019
res(p)/GeV	−0.0274	1.6842	−0.0003	0.0037	0.0019
res($p_\mathrm{T}/p_{\mathrm{T,t}}$)	1.2360	46.7517	0.0028	0.1035	0.0174
res(r)/cm	−0.3042	4.6855	−0.4609	4.6715	0.0055
test variable	mean	std	median	scaled MAD	outlier
pull x	−0.7116	163.3308	−0.0716	1.0709	2395
pull y	0.4177	167.2054	−0.0444	1.0670	2381
pull z	0.6288	37.6981	0.0407	1.0484	2430
pull p_T	−0.6817	14.6206	−0.0359	1.0648	2272
pull p	−10.6903	97.7903	−0.1198	1.0942	2834

Figure 5.14: Results of 62310 successful DAF fits. 70421 tracks were found by the MCTrackCandCombiner. See 5.4 for details of the setup and the explanation of the quantities shown in the table. 4702 tracks had a p-value below 0.01 and were not used in the table. The truncation cuts away 10 % of the sample after the p-value cut.

than in the track finder.

Figure and table 5.15 show the results of test case with a single modification with respect to the previous one. However, it is an important one: the inclusion of realistic background.

The addition of background hits is done by the background mixer module. It reads special ROOT files, created by the respective sub-detector groups, which contain background hits distributed according to the physics processes relevant in the sub-detector. The additional hits look identical to the digitisers the as the hits created by the GEANT4 simulation. Even background hits having energy below the digitiser threshold are mixed in, as they can create a signal if they coincide with another low energy hit (either background or created by GEANT4).

The background files currently include background created by the following processes: Coulomb scattering of beam particles with the beam pipe gas, the Touschek effect, and radiative Bhabha scattering. Figure 5.16 gives a rough impression of the distribution of the background hits, highlighted by their white colour. On the one hand, there are more background hits in the CDC than signal hits; on the other hand, most of them show a very different pattern (lines parallel to z) from signal hits. The SVD background is hardly visible in this image. Its distribution is very uneven, as many event have very little SVD background hits, while a few are flooded with background hits outnumbering the SVD signal hits by far.

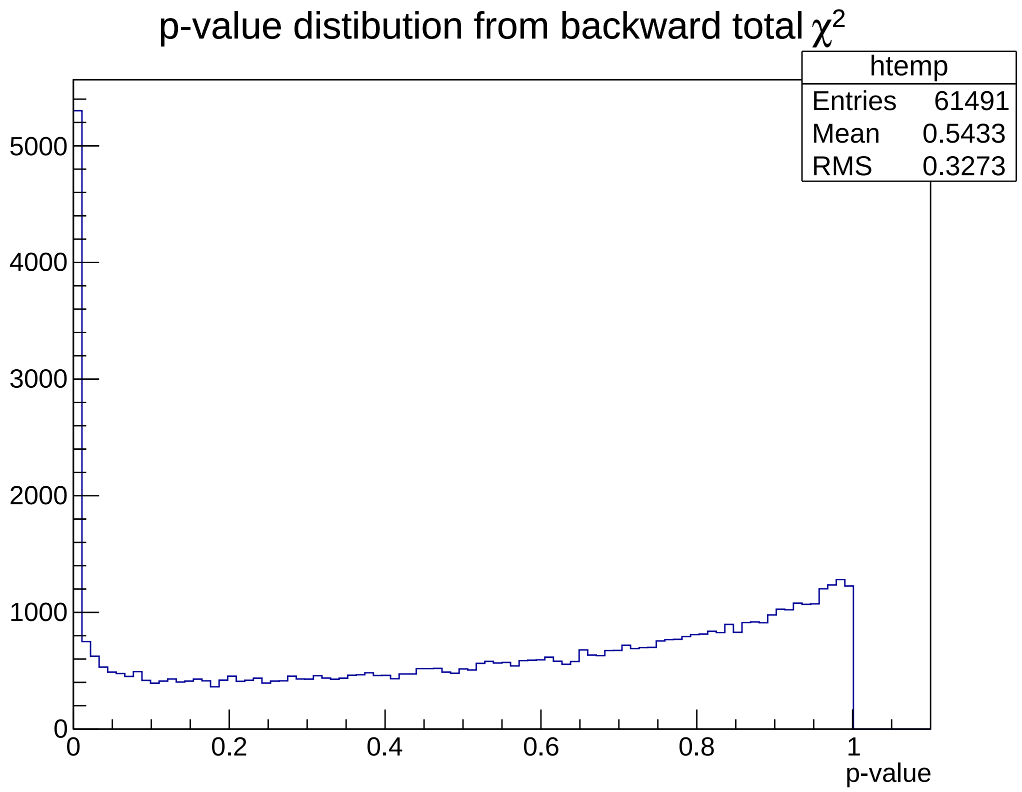
The PXD background could not be included because the current version of the VXDTF cannot deal with it properly. The cuts and filters of the VXDTF were tuned to give optimal results when dealing only with SVD hits. As the PXD suffers from much more background, application of the SVD tuning creates a large combinatorial overhead that cannot be resolved in a reasonable amount of time. Further development work to address this is currently in progress.

Of course, simply omitting the PXD background makes it too easy for the VXDTF. In chapter 6 a test including the PXD hits, but no background, is compared to a test case completely ignoring all PXD hits. The results of an improved VXDTF dealing with realistic background in the PXD will likely lie somewhere in between.

The comparison of figure and table 5.15 (including SVD and CDC background) and figure and table 5.14 (no background) is encouraging, as the difference in track reconstructed performance is relatively small. This means that the current track finding and the fitting algorithms can already deal well with SVD and CDC background.

5.5 Conclusion and open issues in basf2 tracking

The most important property of an estimator is the delivery of unbiased results. In all tests that were conduced both the Kalman filter and the DAF always produced unbiased residual and pull distribution. More precisely, both the median and the truncated mean of the x, y, z, p_x, p_y, p_z residual and pull distributions at the point of closest approach to the true track vertex were always very close to 0. The scaled MAD of the pulls was 1 in the the different test cases of the 6000 EvtGen event in section 5.2.4 to section 5.4.



test variable	mean	std	trunc. mean	trunc. std	scaled MAD
res(x)/cm	−0.0180	5.8540	−0.0004	0.0175	0.0057
res(y)/cm	0.0179	5.8354	0.0002	0.0176	0.0057
res(z)/cm	0.1601	6.5354	0.0011	0.0317	0.0060
res(p_T)/GeV	8.1621	1895.3343	0.0001	0.0030	0.0019
res(p)/GeV	−8.3288	1925.0026	−0.0003	0.0038	0.0020
res($p_\mathrm{T}/p_\mathrm{T,t}$)	0.9817	44.5168	0.0036	0.1067	0.0175
res(r)/cm	−0.3030	4.8308	−0.4775	4.8259	0.0055
test variable	mean	std	median	scaled MAD	outlier
pull x	−0.3107	155.3938	−0.0762	1.0927	2549
pull y	−0.0386	148.4887	−0.0627	1.0874	2548
pull z	0.5738	28.5868	0.0526	1.0651	2592
pull p_T	−0.6418	15.8670	−0.0450	1.0790	2278
pull p	−10.3694	94.6791	−0.1158	1.1025	2814

Figure 5.15: Fit results of 61491 found by the MCTrackCandCombiner. Realistic background is used in SVD and CDC. 8903 of 70633 fits failed. See 5.4 for details of the setup and the explanation of the quantities shown in the table. 5214 tracks had a p-value below 0.01 and were not used in the table. The truncation cuts away 10 % of the sample after the p-value cut.

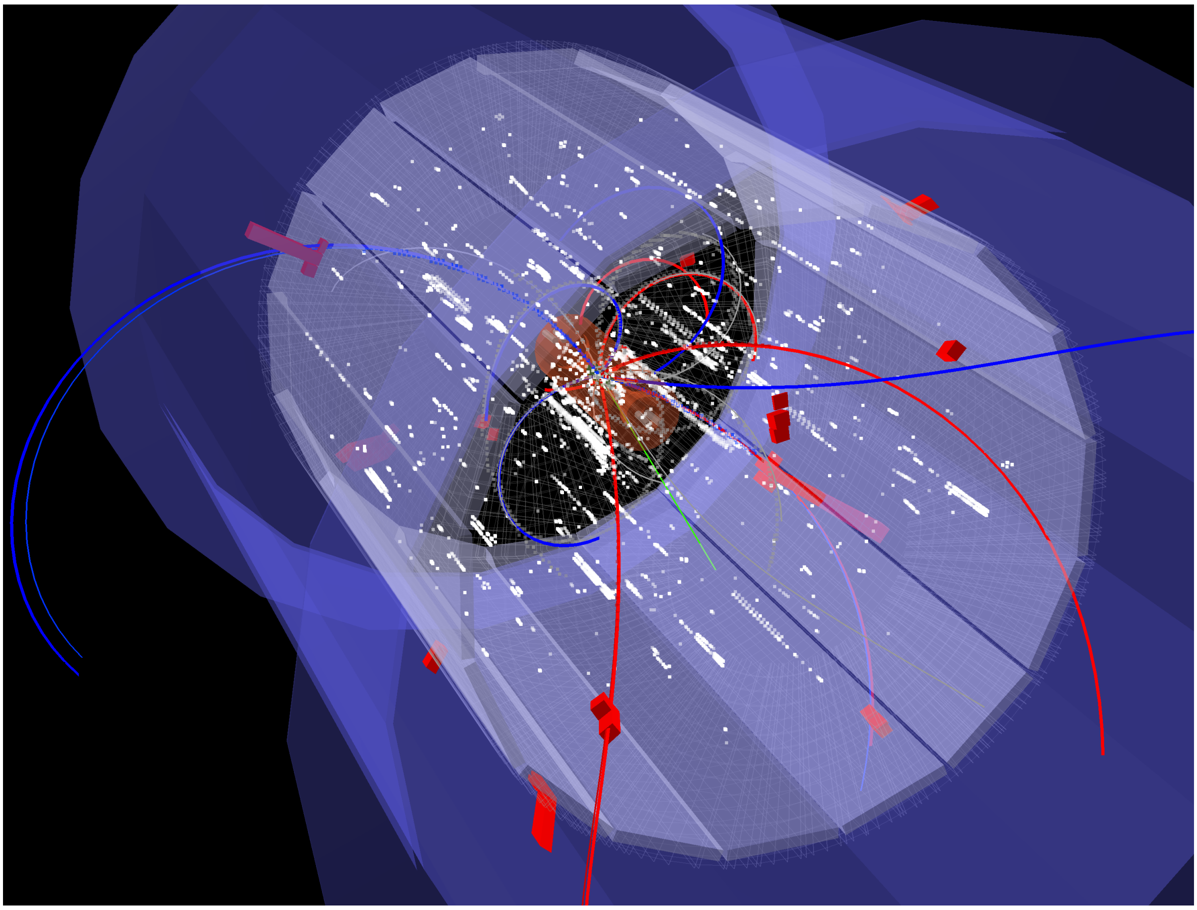


Figure 5.16: A random event from the 6000 EvtGen generated sample with background hits (white) in CDC and SVD.

This shows that the vast majority of tracks fits provide a correct error estimate.

It has to be admitted, however, that there are still missing features, peculiarities and likely some bugs in the tracking software. The following discussion will point out some of these shortcomings.

Low fit efficiency under realistic conditions

The most pressing issue in the tracking software is the high rate of fit failures when the input is coming from the real track finders. This problem is well known in the Belle II tracking group, and its solution is one of the primary goals for the next major release of GENFIT. Also the output of the track finders has to be further improved. The fact that the Legendre finder causes 3 times more fit failures than TRASAN clearly shows room for improvement on this side of the tracking software, too. The VXD track finder also has some known issues which are currently worked on.

The introduction of reference tracks in GENFIT 2 will also increase the fit efficiency of low momentum tracks under perfect fitting conditions. As explained in section 5.2.2, most track fit failures of low-momentum tracks occur when the fitted momentum falls under the minimum momentum threshold of 4 MeV, which is hard-coded in GENFIT. Of course the real momentum is nowhere near this threshold. But when the first few or last few hits are by chance aligned in a way to produce the corresponding curvature, currently GENFIT simply quits, even when the following hits would get the momentum estimation back to a reasonable value.

Error estimation

In figure and table 5.5 one can see that low-momentum curling tracks suffer from a light underestimation of the error when tracked by the CDC. In contrast, figure and table 5.9 show a clear overestimation of the errors of tracks fitted with VXD hits. This effect visible not only in the p-value distribution, but also in the pull distribution. Note that this has nothing to do with the missing accurate error estimations for the realistic cluster hits, as Gaussian-smeared TrueHits were used in this test.

There are also two other cases not shown so far were the error estimations are not entirely correct. One concerns the fitting of alpha particles, where the errors of the q/p (charge over momentum) parameter are too small in the forward filter and too large in the backward filter. As the mean of the q/p pulls is still 0, it is likely a bug in the function that calculates the variance increase of q/p caused by energy loss by ionisation (Bethe–Bloch formula). This was only uncovered too recently to be fixed before handing in this thesis, but will likely be fixed shortly after.

The other case concerns tracks with θ angles far away from $90°$, for example $\theta = 25°$. If only the VXD hits are used for tracking, the errors are overestimated, even at relatively high momenta such as 1 GeV. The p-values look very similar in this case to the 60 MeV VXD hits only setup. Indeed it is very likely that they are symptoms of the same problem.

As the overestimation of the error at shallow θ angle and low momentum only shows itself in VXD tracking, the author spent considerable time to find a solution, being responsible for the tracking performance of low-momentum tracks. While several bugs or simplifications in GENFIT potentially connected to this issue where found and fixed, none of them proved to be the reason of this strange behaviour.

First of all it could be established that the problem is connected to multiple scattering. If multiple scattering is switched off in GEANT4 and in GENFIT. the overestimation disappears, yet when all material effects but multiple scattering are switched off in GEANT4 and GENFIT, the problem remains.

Next the multiple scattering angle distributions produced by the Urban model in GEANT4 were studied systemically for different momenta, and it was shown that their width is in good agreement with the Highland formula 2.38. Therefore the overestimation of the error is not caused by a difference in the multiple scattering models of GEANT4 and GENFIT.

In order to increase the accuracy of the multiple scattering model in GENFIT, it was modified to calculate not only the variances and covariances of the direction coordinates, but also the position variances and covariances. This was done by calculating and implementing equations 4.3 to 4.21. As the PXD and SVD sensors are very thin, the correct treatment of the position variances and covariances had very little effect and did not solve the overestimation problem.

Eventually a general mistreatment of material effect in GENFIT was found. The material effects covariance matrix was not added to the track parameter covariance matrix were the material effects occurred, but instead half of it was added at the measurement plane where the propagation started and the other half were it ended. This was reported to the current main developer of GENFIT and subsequently fixed by him. Unfortunately, while this led to some improvement in other test cases, it did not solve the overestimation issue.

Another change in GENFIT intended to solve this problem was the projection of the material effects covariance matrix onto the measurement plane. While this projection was already done for the propagated covariance matrix before transforming it into the local 5D coordinate system, it was missing in case of the material effects matrix, therefore the projection matrix could be reused for the material effects matrix. This extension of the projection also did not solve the overestimation issue. To some extent this is plausible, as the projection mainly affects the position variances. Yet they are so small that their inclusion in the multiple scattering covariance matrix has hardly any effect, so correcting the projection does not make a big difference either.

Additional test were conducted that showed that the crucial quantity responsible for the overestimation was the incident angle between the measurement plane and the track. The smaller the incident angle is, the larger is the overestimation of the error. It does not matter if this is achieved by changing the θ angle of the particle gun or by rotating the detector. Rotating the detector can be done by manipulating the XML file that contains the alignment parameters. This fact also connects the θ angle problem to the low momentum problem, besides the fact that they are only there when multiple scattering is simulated. At low momenta the ϕ component of the incident angle is always clearly not 90°.

To conclude, something is wrong with the error estimates when the incident angle between the track and a VXD sensor is not 90° and multiple scattering is active. Yet neither the Highland formula nor the transformation of the Highland variance into the multiple scattering covariance matrix (equations 4.3 to 4.21) take the incident angle between measurement plane and track into account. In addition, while the projection of the material effects covariance matrix onto the measurement plane takes the incident angle into account, it turned out to be irrelevant whether or not this projection is applied to the material effects covariance matrix. The author was not able to resolve this contradiction.

Too large p-values of the DAF

As stated in section 5.3, the DAF produces a p-value distribution with an unexpected excess of large p-values. This is not the case when fitting VXD only tracks. There the p-value distribution is flat with a slight depletion in the first bin for high momentum tracks, and the usual excess in the first bin for very low momentum tracks. CDC only tracking shows a tendency for too large p-values when the DAF is used that is not present with the Kalman filter. When the DAF is enabled to resolve the left/right ambiguity of the CDC wire hits, the p-value distribution get very slanted towards high values and has a mean of 0.6. But this does not explain the p-value distributions shown in figure 5.14 and 5.15, as the left/right ambiguity resolution was switched off in these test cases.

Too many outliers in pull distributions

All the pull distributions shown in the tables in this chapter show a relatively high number of outliers, especially when one considers the fact that a p-value cut was always applied to the tracks before calculating the pulls. As the the pulls are calculated at the true origin of the track, which is almost always inside the beam pipe, a strong material effect between the track origin and the first hit in the tracking detector can result in a pull outlier track having a normal p-value. To test if a material interaction between the first PXD layer and the collision point can explain the pull outliers, two test cases were simulated. Particles from 1000 EvtGen events propagating through the whole detector and another 1000 EvtGen generated events where all material in the interaction region was deleted. This can be done very easily by commenting out all lines in the Belle2.xml file containing the letters "IR". All other parameters relevant for tracking were the same as in 5.11.

The results of this test case including the interaction region is shown in figure and table 5.17 and should be compared to the second test case shown in figure and table 5.18, where all material was deleted from the interaction region.

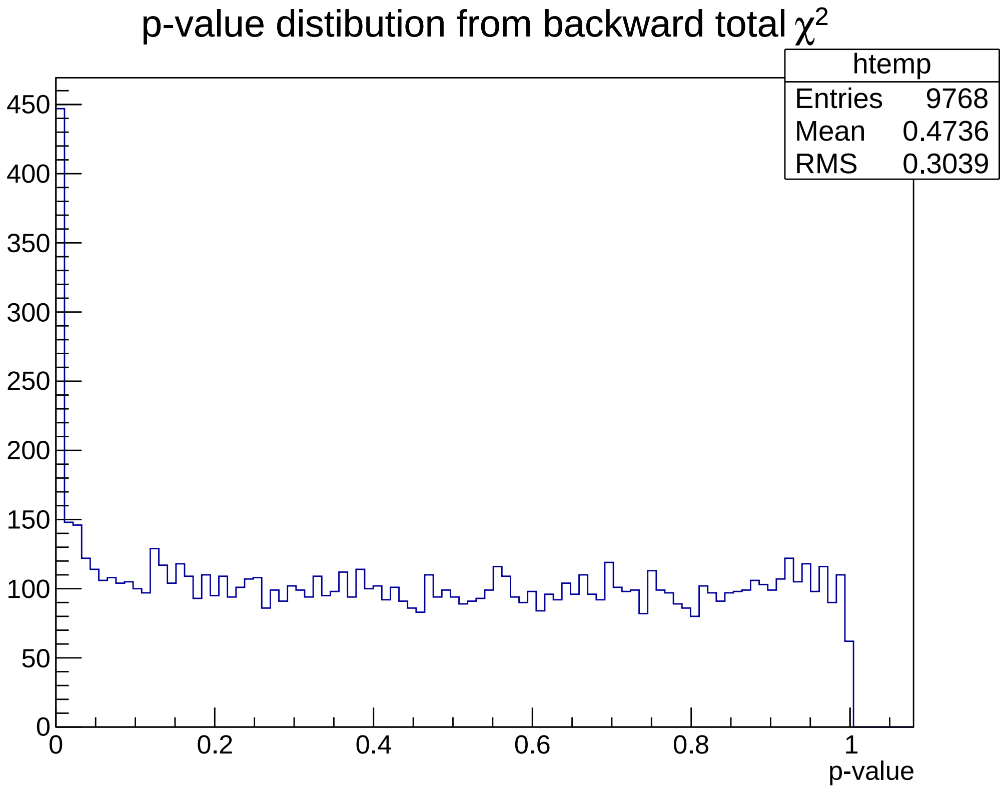
The comparison shows that the number of outliers clearly drops when excluding the interaction region material, but on the other hand many more data points in the pull distribution remain outside of a distance equal to 4 times the scaled MAD as one would expect in a Gaussian-distributed sample of this size. It should be recalled that an absolute value larger than 4 times the scaled MAD is the definition of an outlier in all tables used in the thesis.

Therefore approximately half of the pull outliers can be attributed to material effects in the interaction region and are therefore understood, while the rest is not.

Additionally the distributions of the true initial θ angles and momenta of the pull outlier tracks were inspected after applying a p-value cut of 0.01. This showed no correlation between the outliers and specific p or θ values. It is also noteworthy that most outliers in the different pull distributions (x, y, z, p_{T}, p) come from different tracks instead while only the minority of tracks creates outliers across all pull distributions. As an unexpectedly high number of pull outliers can indicate issues in the track fitting code, it further investigations are clearly required.

5.6 Performance of RAVE

To test both the interface between basf2 and the RAVE vertex fitting library and the most basic fitting algorithms RAVE offers, a sample of 10000 events was generated, each



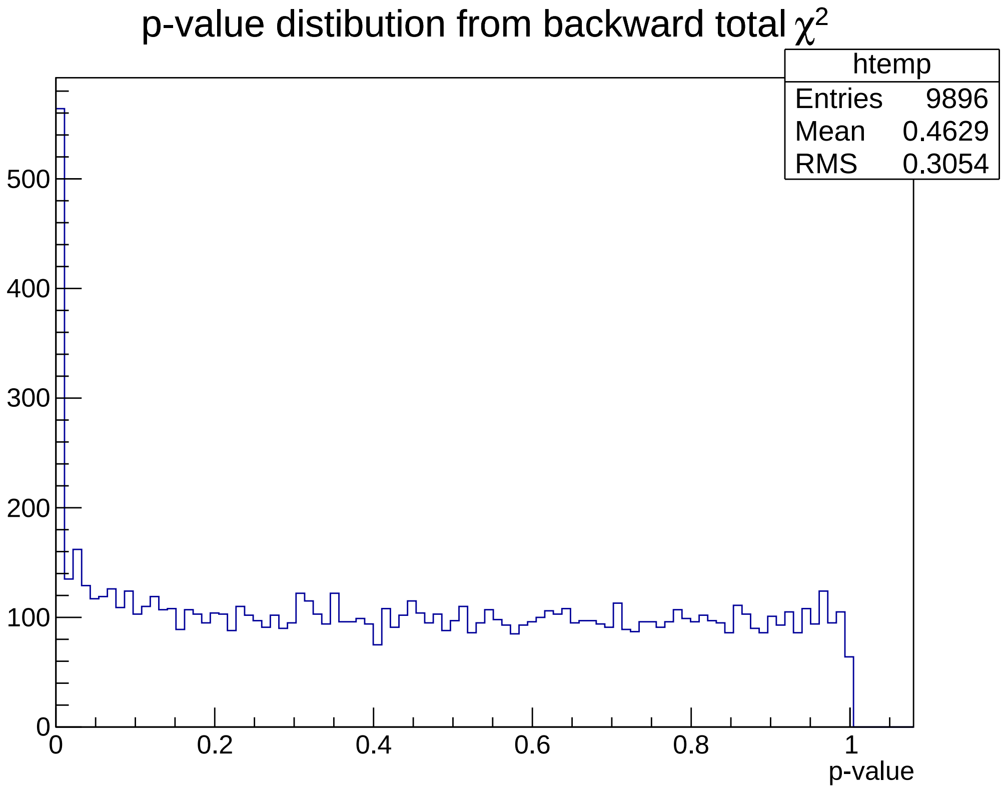
test variable	mean	std	median	scaled MAD	outlier
pull x	−0.0641	3.8639	−0.0078	1.0371	83
pull y	0.0488	4.9381	0.0132	1.0265	83
pull z	0.0611	5.5237	0.0059	1.0452	82
pull p_{T}	−0.1636	3.6092	−0.0748	1.0410	140
pull p	−0.2542	2.2790	−0.0945	1.0375	129

Figure 5.17: Fit results of 9768 tracks created by EvtGen. Before creating the table 431 tracks were discarded because of a p-value below 0.01.

with 3 muon tracks from the particle gun, originating at the same vertex. This tests whether the fitted track parameters are correctly converted to the RAVE input format, magnetic field information is passed on correctly, and the fitted vertex position and covariance matrix can be read back to basf2 without error.

The $\mu^{\pm}$ all had a momentum of 1 GeV, a uniformly distributed $\phi \in [0°, 360°]$ and a uniformly distributed $\theta \in [60°, 120°]$. The other properties were the same as described in 5.2.1, so the full detector was simulated. The fitting algorithm selected in RAVE was the Kalman filter.

The p-value distribution of all 30000 $\mu^{\pm}$ tracks is shown in 5.19. One track fit failed. It comes as no surprise that this distribution looks essentially the same as the one shown in figure and table 5.2. Before the tracks were passed to RAVE, they were propagated into the beam pipe, so that RAVE could use its vacuum propagator without any accuracy



test variable	mean	std	median	scaled MAD	outlier
pull x	−0.0235	7.1608	−0.0090	0.9835	37
pull y	0.0039	2.8914	0.0056	0.9819	37
pull z	0.0806	7.1943	0.0079	0.9870	37
pull p_{T}	−0.1403	3.1644	−0.0734	1.0393	103
pull p	−0.1961	1.4612	−0.0641	1.0555	105

Figure 5.18: Fit results of 9896 tracks created by EvtGen. Before creating the table 549 tracks were discarded because of a p-value below 0.01. There was no material between PXD layer 1 and the collision point in this simulation.

penalties.

The p-values, vertex position residuals (or impact parameters) and vertex position pulls of the 10000 fitted vertices are shown in figure and table 5.20. Most of the excess of low p-values in the first bin is a direct consequence of about 170 tracks with too low p-values from the Kalman filter. The standard deviations and MADs of the vertex position pulls are essentially the same. This indicates that no further outliers exists in the fitted vertices sample after discarding the 121 vertices with p-values below 0.0005.

The successful fit of these 10000 vertices proves that the RAVE/basf2 interfaces are working correctly.

Figure and table 5.21 illustrate the performance of one of the robust vertex fit algorithms in RAVE. The same sample of 10000 times 3 muon tracks used for Figure and

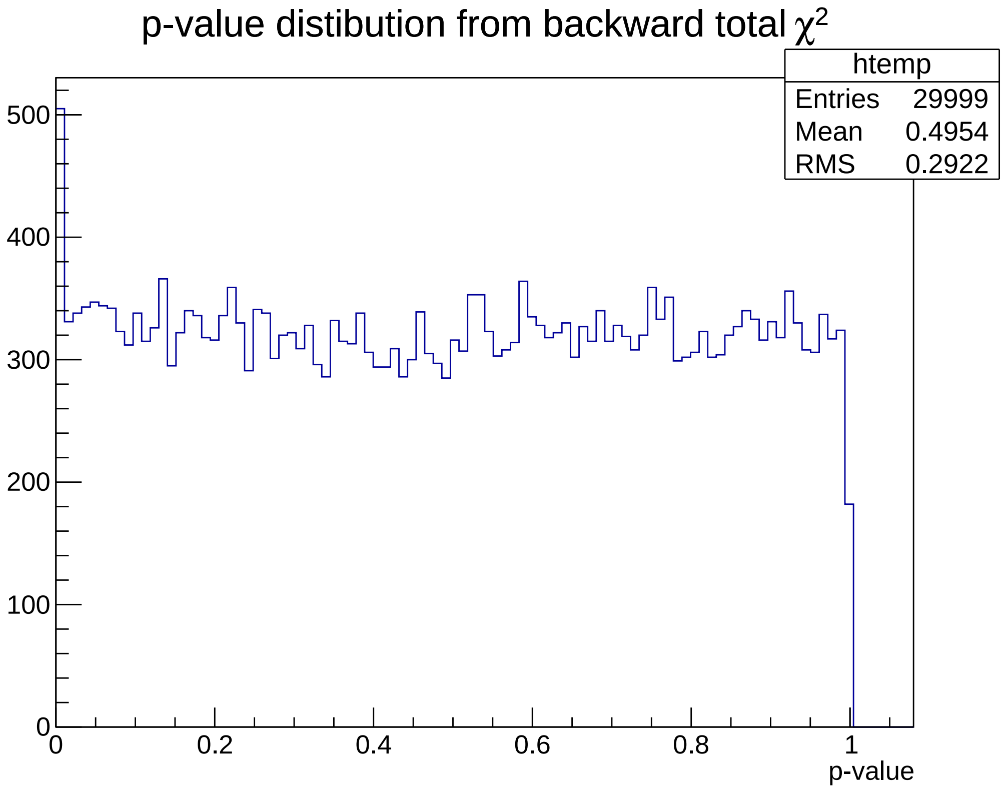
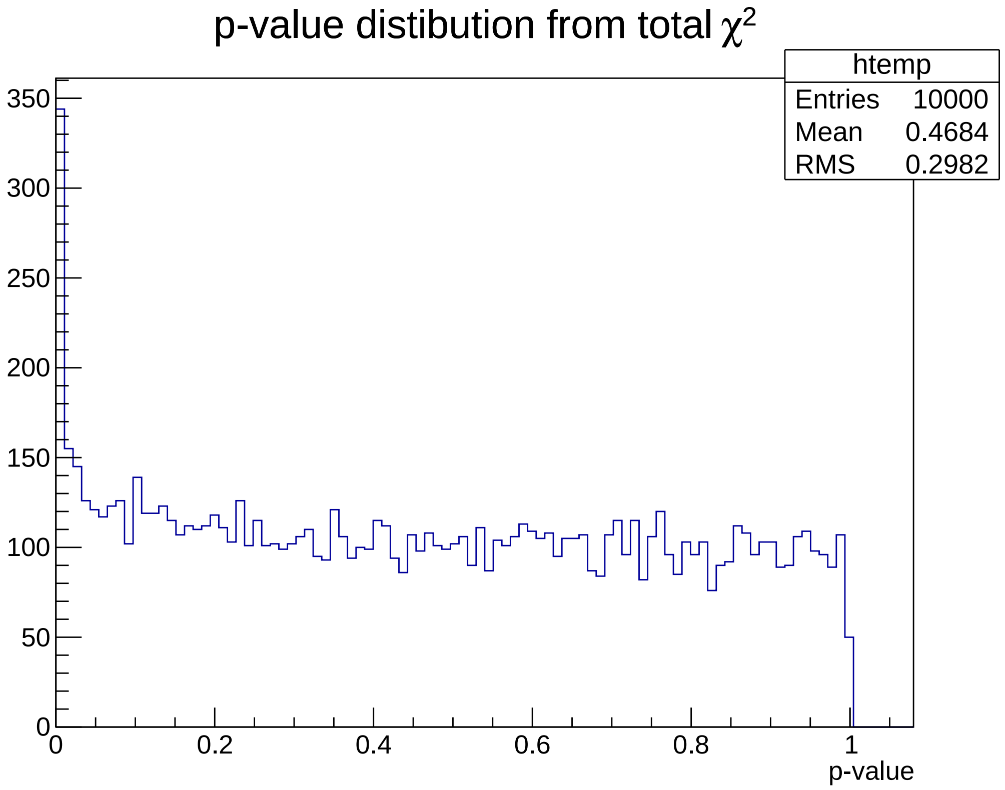


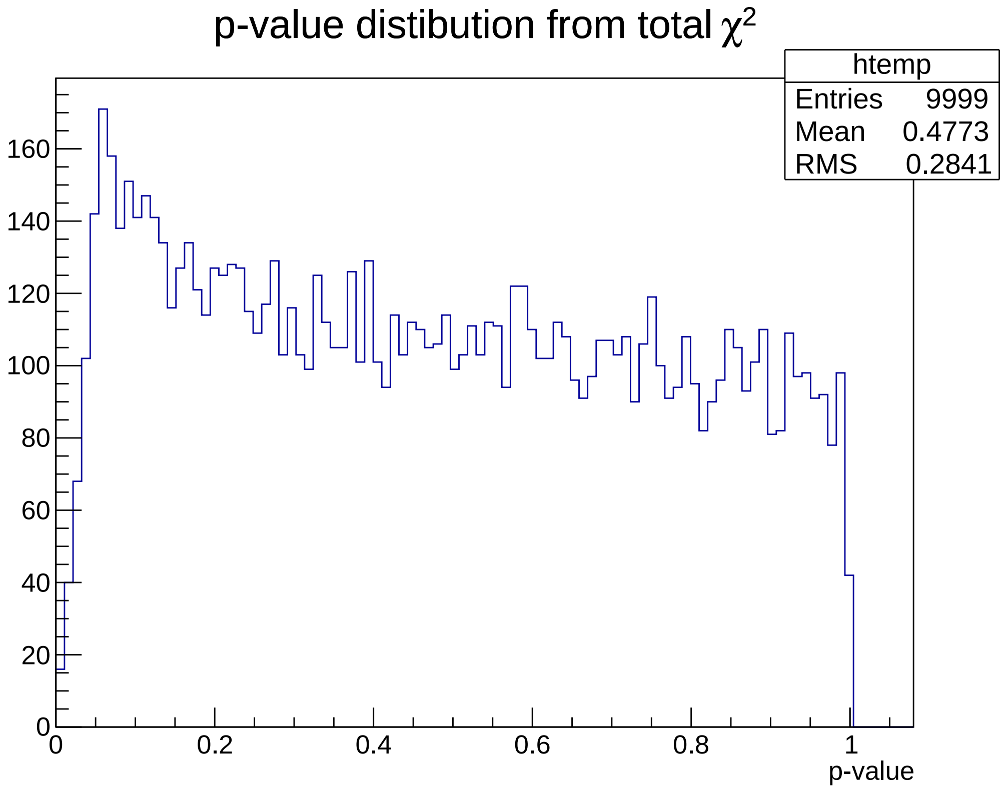
Figure 5.19: p-value distribution of tracks given to RAVE to test the basf2 to RAVE interfaces. The fitted vertices are shown in figure and table 5.20.

table 5.20 were fitted again with the “Adaptive Vertex Fitter” (AVF) in RAVE. The 170 or so outlier tracks in the muon track sample now do not affect the vertex fits any more, as they are successfully down-weighted. Even without a p-value cut on the fitted vertices the standard deviations and MADs of the vertex position pulls are the same. The default parameters of the AVF were used in this test case: initial temperature of 256, annealing ratio of 0.25, and a σ_{cut} of 3.



test variable	mean	std	trunc. mean	trunc. std	
res(x)/cm	0.0000	0.0025	0.0000	0.0023	
res(y)/cm	0.0000	0.0025	0.0000	0.0023	
res(z)/cm	−0.0000	0.0017	0.0000	0.0016	
test variable	**mean**	**std**	**median**	**scaled MAD**	**outlier**
pull x	0.0065	1.0765	0.0005	1.0363	14
pull y	0.0118	1.0928	0.0001	1.0669	8
pull z	0.0012	1.0542	−0.0034	1.0365	4

Figure 5.20: Result of 10000 vertex fits, each consisting of 3 1 GeV muon tracks and using the Kalman vertex fitter in RAVE. For the details of the setup see section 5.6. 121 vertices had a p-value below 0.0005 and were not used in the table.



test variable	mean	std	trunc. mean	trunc. std	
res(x)/cm	0.0000	0.0027	0.0000	0.0023	
res(y)/cm	0.0000	0.0031	0.0000	0.0024	
res(z)/cm	0.0000	0.0020	0.0000	0.0016	
test variable	mean	std	median	scaled MAD	outlier
pull x	0.0098	1.0630	0.0015	1.0232	15
pull y	0.0137	1.0979	0.0017	1.0498	19
pull z	0.0043	1.0537	−0.0018	1.0318	8

Figure 5.21: Result of 10000 vertex fits, each consisting of 3 1 GeV muon tracks and using the AVF fitter in RAVE. For the details of the setup see section 5.6. No vertex had a p-value below 0.0005, so the table uses the full sample as shown in the figure.

6 Performance of track and vertex reconstruction examples relevant for physics analysis

6.1 Reconstruction of slow pions from charged D star decays

The $\mathrm{D}^{*\pm}$ is a very common intermediate state in the decay chain of the B mesons in Belle II. The most likely (67.7 %) decay mode for the D^{*+} is $\mathrm{D}^{*+} \longrightarrow \mathrm{D}^0 + \pi^+$, where the pion has a very low average energy of about 160 MeV. The following discussion refers both to this process and to the charge-conjugate process. Figure 6.1 and figure 6.2 show the p and p_T distributions of the slow pions that reach the VXD. The low-energy pion is indeed characteristic for the decay and is used as an identifier that $\mathrm{D}^{*+} \longrightarrow \mathrm{D}^0 + \pi^+$ (or its charge-conjugate process) is what actually happened in the event.

The identification and reconstruction of these slow pions is one of the major motivations for the development of a VXD only track finder and for the rigorous testing of the track fitting software at very low particle energies. It is therefore a natural choice for measuring the performance of the track reconstruction software in the low momentum regime.

The tracking strategy in Belle I led to a fast drop in the reconstruction efficiency of the slow pions below a momentum of 100 MeV. Figure 6.3 illustrates this problem.

The test case was constructed by creating 10000 event with EvtGen. A user decay file forced one of the B mesons into the $\mathrm{D}^{*+} \longrightarrow \mathrm{D}^0 + \pi^+$ decay mode, while the other B decay was simulated without modifications. Of course the full detector as currently implemented in basf2 was simulated.

To test the fitting of the slow pions under perfect conditions, TrueHit digitization was selected and the MCTrackFinder was configured to extract the slow pions from the event and pass it to the fitter module. This resulted in 11946 track candidates with an accumulated measurement ndf of at least 5 per track. 11415 of the 11946 fits were successful (95.6 %). The failed fits typically quit with the GENFIT error message "momentum too low". The fitting under these perfect "finding" conditions are shown in figure and table 6.4. As there are no extraneous hits that do not belong to the pion track in this setup, the Kalman filter with 2 iterations was used.

The results are perfectly in line with the material presented earlier in this thesis, especially the results of fitting of 100 MeV muons in figure and table 5.5. The scaled MAD of the pulls is 1 throughout, and there is no bias in the residuals or the pulls, at

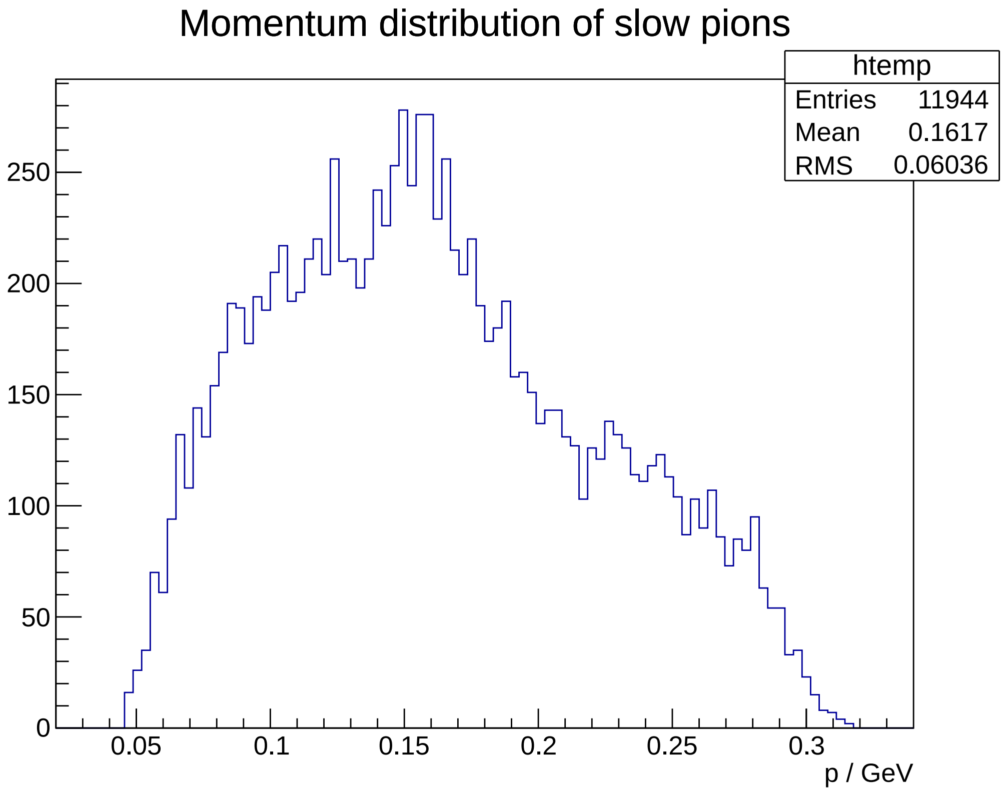


Figure 6.1: The momentum distribution of slow pions from $D^{*\pm}$ decays with at least 3 hits in the tracking detectors.

least when p_x,p_x and p_x are inspected instead of the non-Gaussian distributed p and p_T. The higher number of p-value outliers compared to the 100 MeV muon test case is due to the hadronic interactions that muons lack.

Figure and table 6.5 show the correlation between true and fitted track momenta. As expected from the residual table, the estimated momentum is centered around the true momentum. Figure and table 6.5 also show the fitting efficiency for different momenta. As mentioned above, it is 95.6 % for the entire pion sample. Approximately half of the failed fits quit with the “momentum too low” message which should not occur in GENFIT 2. One can therefore expect that GENFIT 2 will cut the failure rate by a factor of two.

The next step is to investigate the reconstruction performance under realistic conditions. This implies the presence of background in the SVD and in the CDC, and track finding with the VXDTF and TRASAN. Again, because no full track finder exists yet, the combination of track candidates from the individual finders is done with the MCTrackCandCombiner module and therefore with help from the simulated truth.

The MCTrackCandCombiner inspects the track candidates and passes them to the fitter only if they were really created by the pion from the $D^{*\pm} \longrightarrow D^0 + \pi^\pm$ decay, and combines them to a single track candidate if they are coming from the same slow

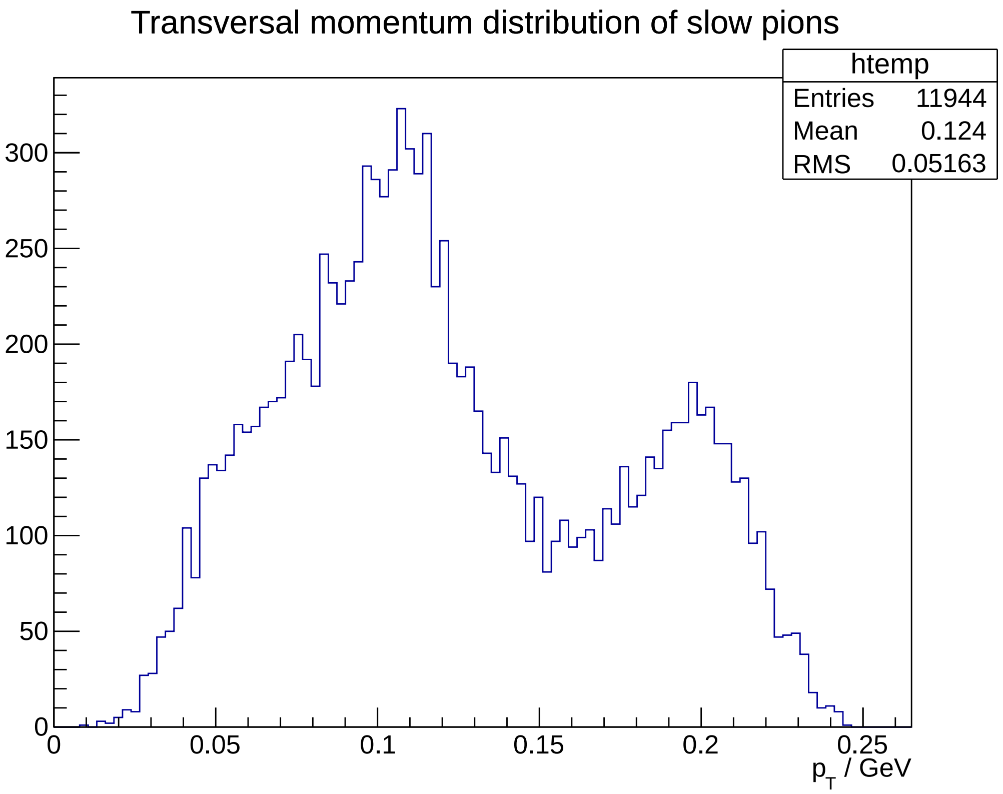


Figure 6.2: The transverse momentum distribution of slow pions from $D^{*\pm}$ decays with at least 3 hits in the tracking detectors.

pion. This still guarantees that only slow pions that really were found without simulated truth information will be declared as “found”. “Found” in this context means that more than 60 % of the track hits are really from the slow pion. So the actual “cheating” consists in using simulation information to discard ghost tracks or to discard multiple track candidates that are segments of the same curling track.

With this configuration the MCTrackCandCombiner finds 11272 slow pion tracks, which is only a few less than the 11974 from the MCTrackFinder. This is a finding rate of 94 %. Of the 11272 pion track candidates, 3213 were found exclusively by the VXDTF, 520 were found exclusively by TRASAN, and 7539 pions were found by both VXDTF and TRASAN and therefore merged by the MCTrackCandCombiner. As in the last chapter, the seed values for the fitter were taken from the real track finders.

The fitting results of this test case are shown in figure and table 6.6. The main difference is that the successful fit rate drops significantly from 95.6 % of all tracks to 76.1 % of all found tracks. As mentioned before, the high fail rate in GENFIT is currently addressed by the GENFIT developers. The distribution reconstructed momentum is shown in figure 6.8. It can be compared to the true momentum distribution in figure 6.1.

The other differences to the test case with perfect track finding and seed values shown in figure and table 6.4 are relatively small. There are about 200 additional p-value

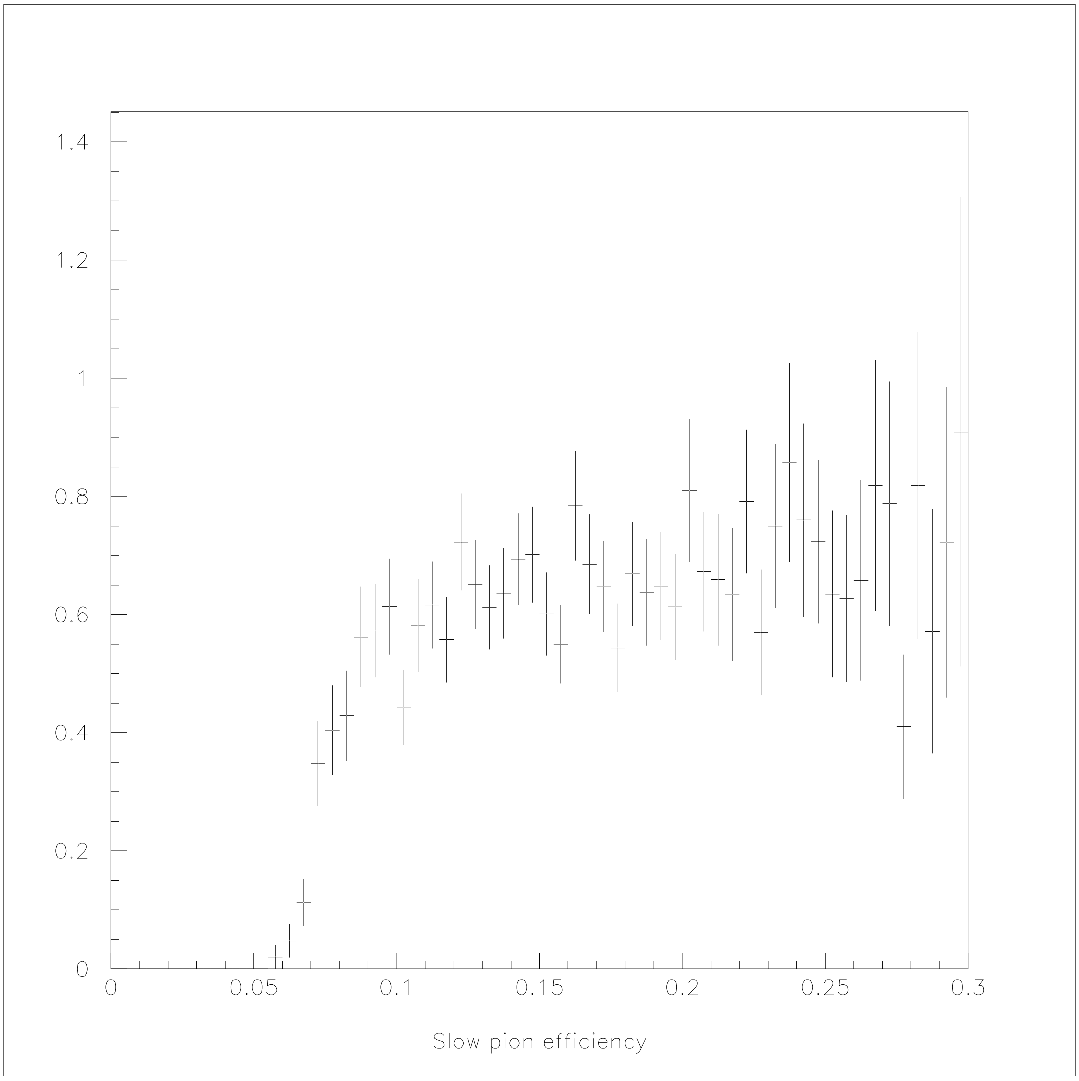
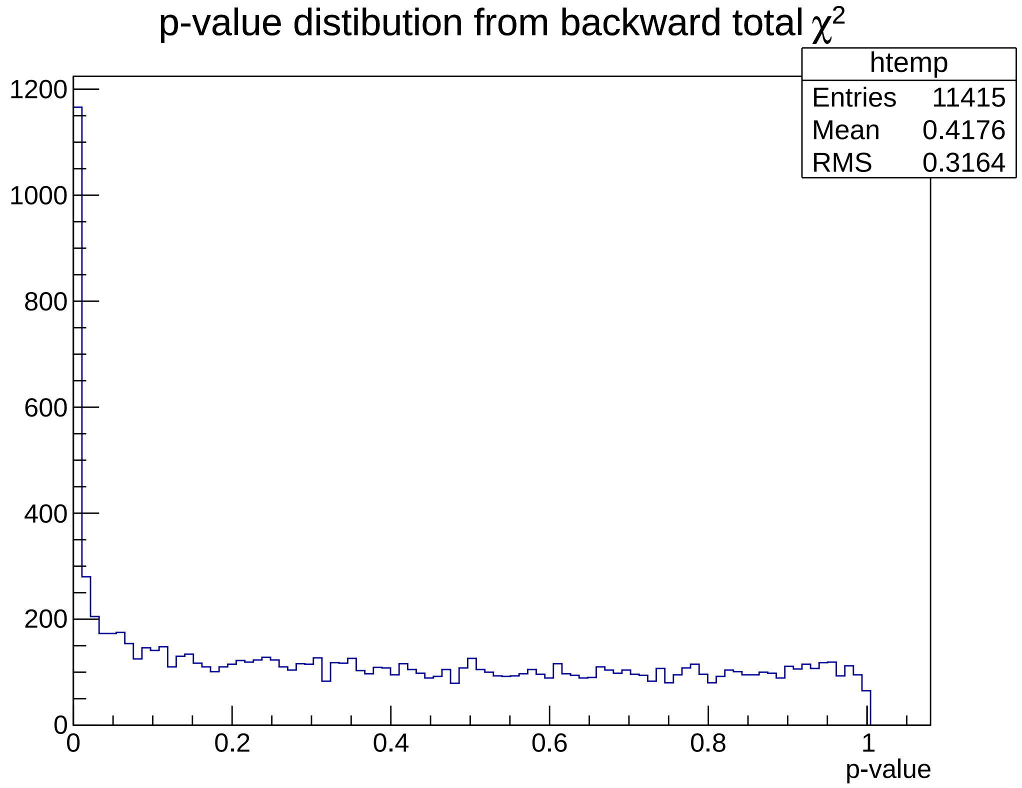


Figure 6.3: Reconstruction efficiency of slow pions from D$^{*\pm}$ decays in Belle I

.

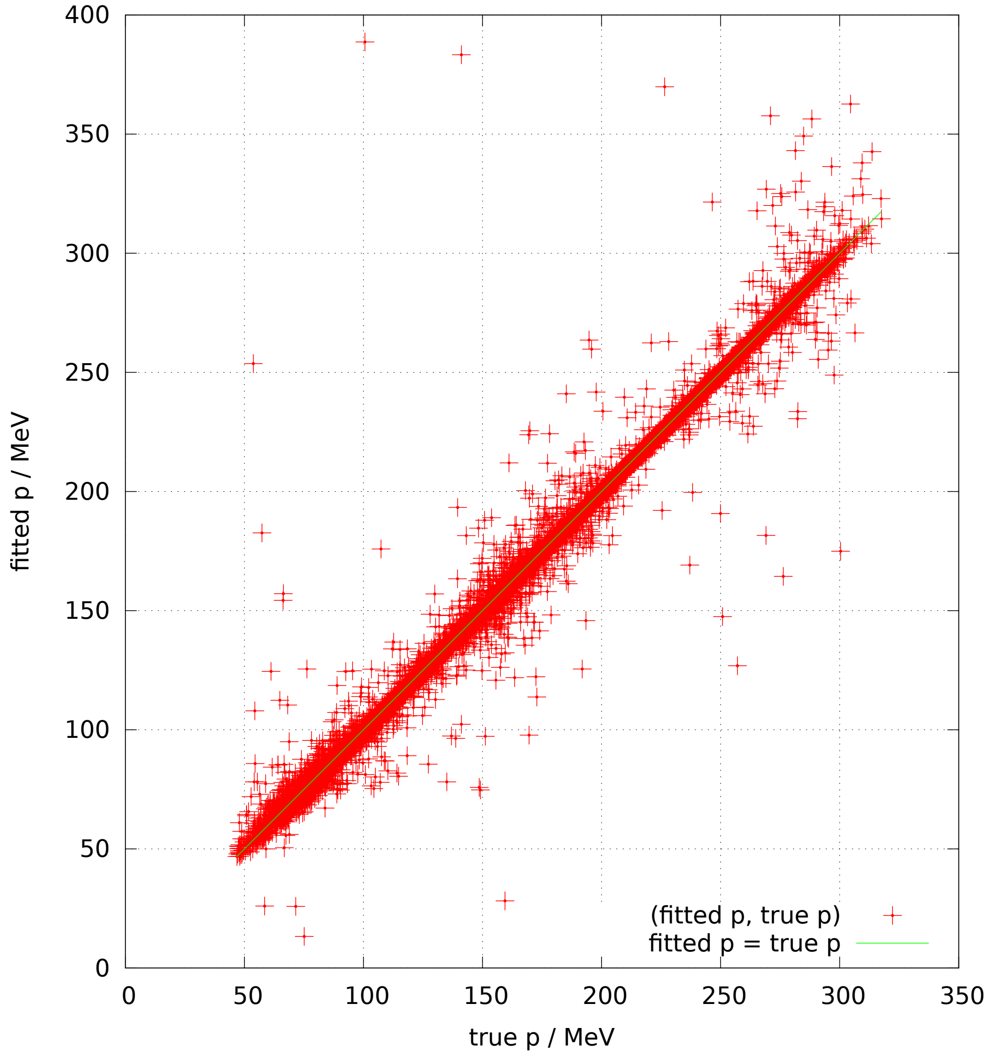
outliers, and the number of outliers in the pull distributions after the p-value cut approximately doubles. However, the core width of the pull distributions remains close to 1, so that the error estimates are still well matched with the actual spatial and momentum resolution.

Figure and table 6.7 show the the correlation between true and fitted track momenta and the momentum dependent finding and fitting efficiency. Of course the average distance to the green line is larger than in in figure 6.5, which was already obvious from the larger residual widths in figure and table 6.6. The finding rate is still 82 % for track momenta under 100 MeV. Although GENFIT currently has the mentioned strong problems if fit efficiency, the combined pion reconstruction efficiency is 60 % which will surly increase with with GENFIT 2. As shown in figure 6.3, the Belle I reconstruction efficiency below 100 MeV is not larger than 40 %. The better low momtum reconstruction



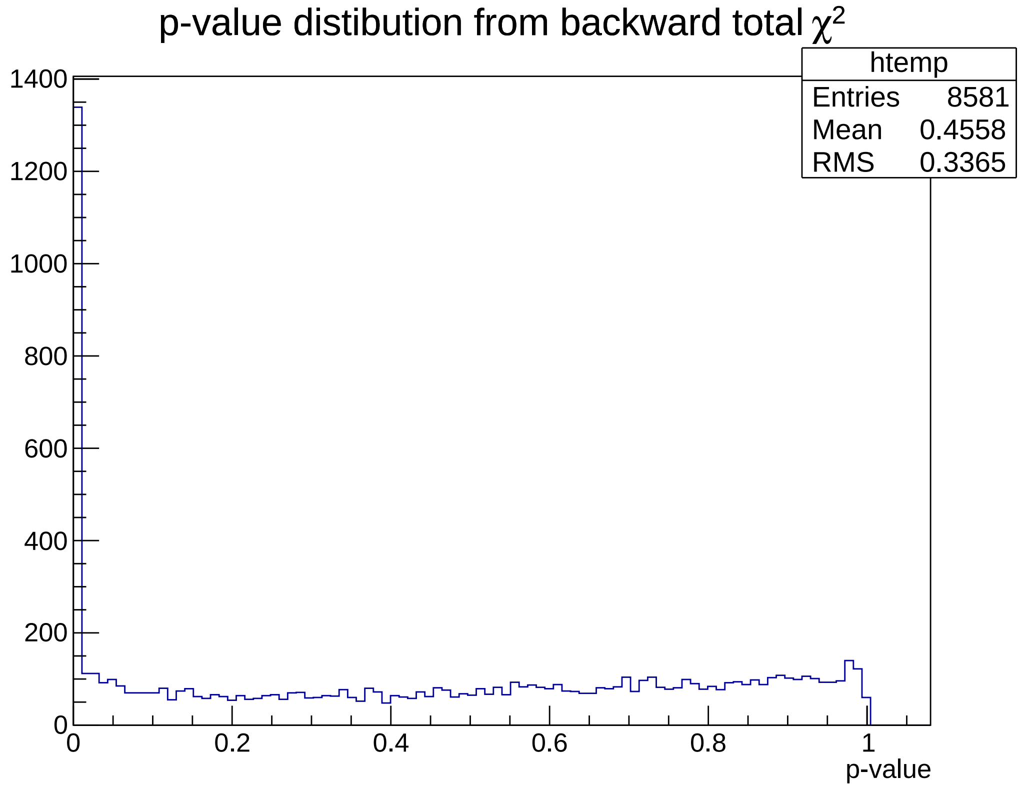
test variable	mean	std	trunc. mean	trunc. std	scaled MAD
res(x)/cm	−0.0018	0.1848	0.0001	0.0131	0.0121
res(y)/cm	0.0020	0.2365	0.0001	0.0132	0.0120
res(z)/cm	0.0132	0.4686	0.0003	0.0121	0.0125
res(p_{T})/GeV	0.0003	0.0073	−0.0002	0.0013	0.0013
res(p)/GeV	−0.0005	0.0104	−0.0003	0.0011	0.0010
res($p_{\mathrm{T}}/p_{\mathrm{T,t}}$)	0.3420	10.7099	0.0037	0.1278	0.1000
res(r)/cm	0.0433	0.2965	0.0142	0.0129	0.0131
test variable	**mean**	**std**	**median**	**scaled MAD**	**outlier**
pull x	−0.0301	6.8239	−0.0263	1.0565	113
pull y	0.0326	6.2315	0.0053	1.0502	115
pull z	0.0925	6.8508	−0.0121	1.0626	127
pull p_{T}	−0.2245	3.9101	−0.2445	1.0286	118
pull p	−0.6198	9.8471	−0.3223	1.0193	106

Figure 6.4: Fitting results of slow pions found by the MCTrackFinder. 11946 tracks were in the simulated sample. 11415 were fitted successfully. 1143 of the fitted tracks had a p-value below 0.01 and were not included in the table. The truncation cuts away 10 % of the sample after the p-value cut.



p_{true}/MeV	0 – 100	100 – 150	150 – 200	200 – 250	250 – inf	all p
trackable $\pi^{\pm}$	2075	3424	3278	1940	1226	11943
found ratio	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
fitted ratio	0.8978	0.9828	0.9622	0.9639	0.9478	0.9557

Figure 6.5: Scatter plot of fitted and true momentum of the slow pions. The y-distance of a point to the green line is the momentum residual of this track. The table shows the finding and fitting efficiency for different momentum ranges. As the MCTrackFinder was used in this test case, the finding efficiency is always 100 %.



test variable	mean	std	trunc. mean	trunc. std	scaled MAD
res(x)/cm	0.0025	0.5791	−0.0004	0.0191	0.0140
res(y)/cm	0.0015	0.5970	−0.0001	0.0189	0.0139
res(z)/cm	0.0846	2.3717	0.0011	0.0173	0.0138
res(p_T)/GeV	0.0006	0.0237	−0.0003	0.0019	0.0016
res(p)/GeV	−0.0035	0.0424	−0.0005	0.0020	0.0013
res($p_\mathrm{T}/p_\mathrm{T,t}$)	1.0007	22.1320	0.0135	0.1897	0.1103
res(r)/cm	0.1107	0.8239	0.0194	0.0218	0.0159
test variable	mean	std	median	scaled MAD	outlier
pull x	0.0197	8.3545	−0.0227	1.0978	254
pull y	−0.1113	20.4635	−0.1028	1.0763	260
pull z	0.4221	10.7242	0.0230	1.0869	265
pull p_T	−0.2691	4.9082	−0.1897	1.0650	201
pull p	−3.5910	39.5658	−0.3121	1.0666	189

Figure 6.6: Fitting results of slow pions found by the MCTrackCandCombiner. 11272 slow pions were found. 8581 were fitted successfully. 1321 of the fitting tracks had a p-value below 0.01 and were not included in the table. The truncation cuts away 10 % of the sample after the p-value cut.

is also evident when comparing figure 6.8 and 6.9. Because of the different sample sizes used in the two figure it the higher relative number of reconstructed pions in 6.8 it is a bit difficult to spot the different in first 100 MeV.

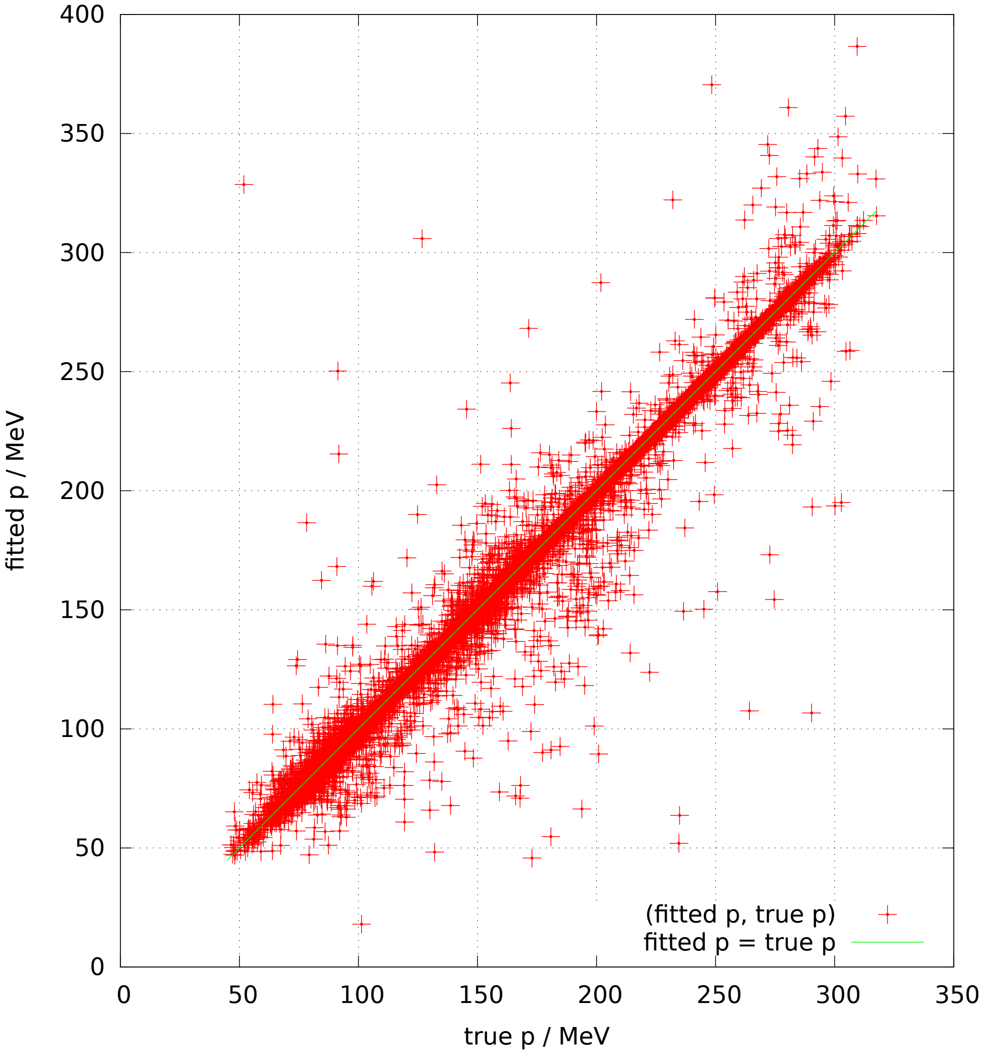
As an additional cross check for the track finder quality, the simulated sample with 10000 events was reprocessed with the realistic finders, but the minimum ratio of hits really coming from the slow pion in a found track was raised to 80 % in the MCTrackCandCombiner. This resulted in 11168 track candidates, of which 3357 were only found by VXDTF, 813 only by TRASAN, and 6998 by both finders. The fact that these numbers are very similar to the case where only 60 % of hits in a track candidate needed to come from the pion to be declared as "found", strengthens the trust in the track finders.

Besides the use of simulation information for combining and discarding of track candidates and the lack of background in the PXD, the presented finding ratios in this section are not unrealistic, as the current weaknesses of the track reconstruction software are well known and are currently being resolved.

To have some kind of worst case assessment of the influence of realistic PXD background, the slow pion reconstruction test case was repeated, but the VXDTF parameters were changed so that the PXD hits would be ignored completely, essentially reducing the PXD detector to some additional non-sensitive material. The only other change in setting was the increase of the track purity to 80 % in the MCTrackCandCombiner, so only track candidates with more than 80 % of the hits coming from the slow pion would be passed on to the fitting module. The same sample of 10000 slow pion events was used.

The results of this test case are presented in figure and table 6.10 and 6.11. Of course the number of reconstructed pions drops, as there are less hits and the required purity of the track candidates was increased to 80 %. Ignoring the PXD hits has a severe impact on the x, y, and z residuals, more than doubling their width by both measures: the truncated standard deviation and the scaled MAD. The momentum residuals stay basically the same, which comes as no surprise, as the PXD contributes very little to the curvature measurement.

When ignoring hits from the PXD, the slow pion finding efficiency changes significantly only in the first column (momentum less than 100 MeV) of figure and table 6.11, but is still at 57 %. The main conclusion from this test is therefore that no matter how difficult it will be to deal with the realistic PXD background, it cannot get worse than that.



p_{true}/MeV	0 – 100	100 – 150	150 – 200	200 – 250	250 – inf	all p
trackable $\pi^{\pm}$	2101	3433	3297	1940	1234	12005
found ratio	0.8153	0.9700	0.9472	0.9835	0.9668	0.9385
fitted ratio	0.5997	0.6662	0.7458	0.8031	0.8241	0.7148

Figure 6.7: Scatter plot of fitted and true momentum of the slow pions found by MCTrackCandCombiner. The y-distance of a point to the green line is the momentum residual of this track. The table shows the finding and fitting efficiency for different momenta ranges. The successful fit ratio is calculated with respect to the trackable $\pi^{\pm}$ not the found $\pi^{\pm}$. Very few extreme outlier tracks lie outside of the plot.

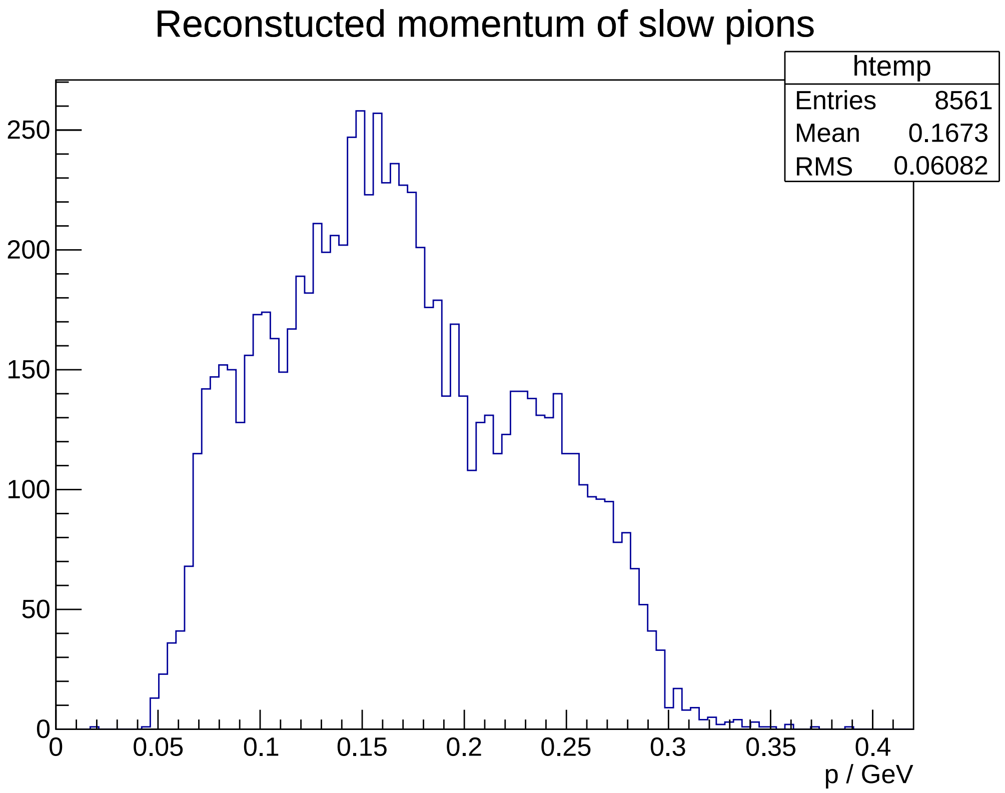


Figure 6.8: The momentum distribution of reconstructed slow pions from $D^{*\pm}$ decays.

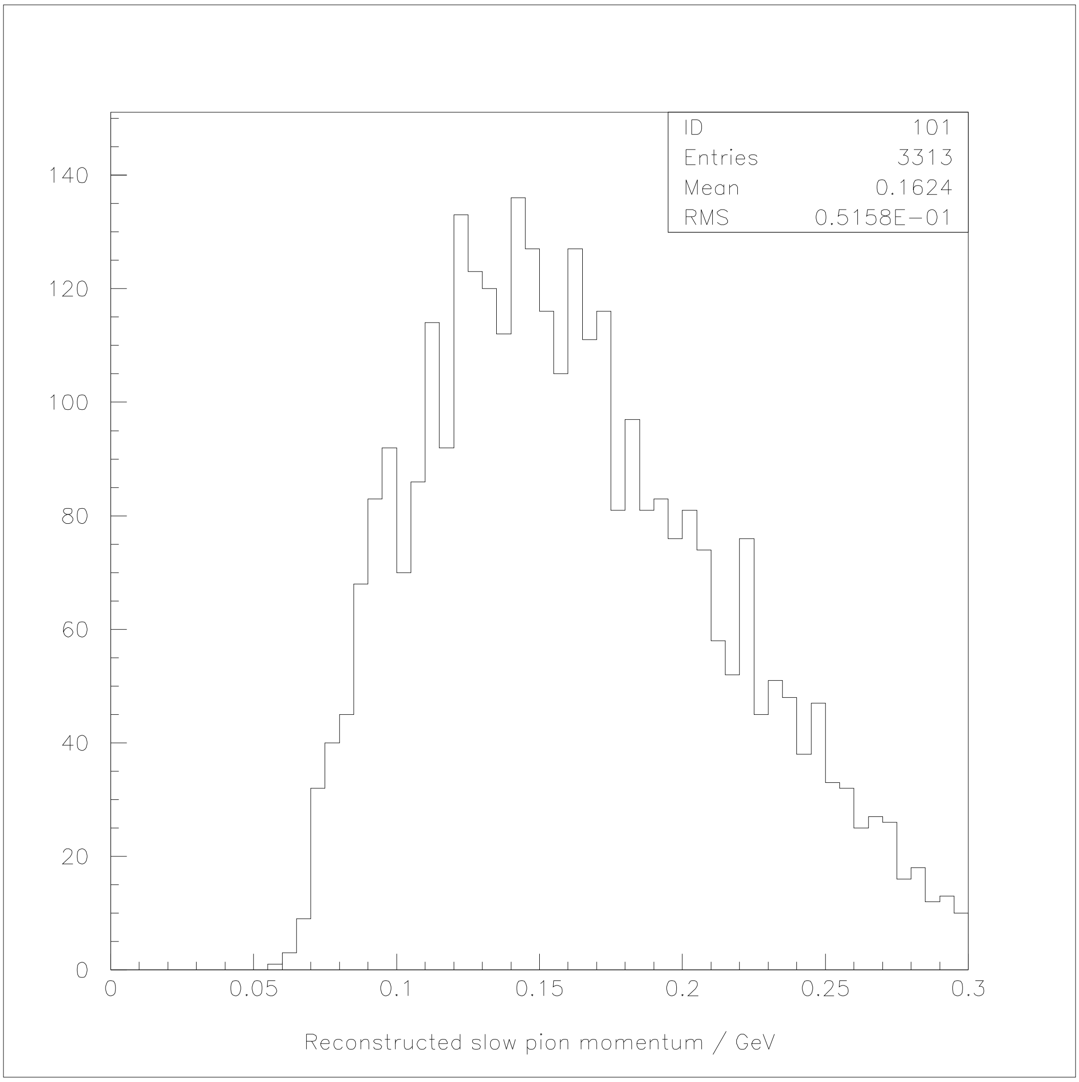
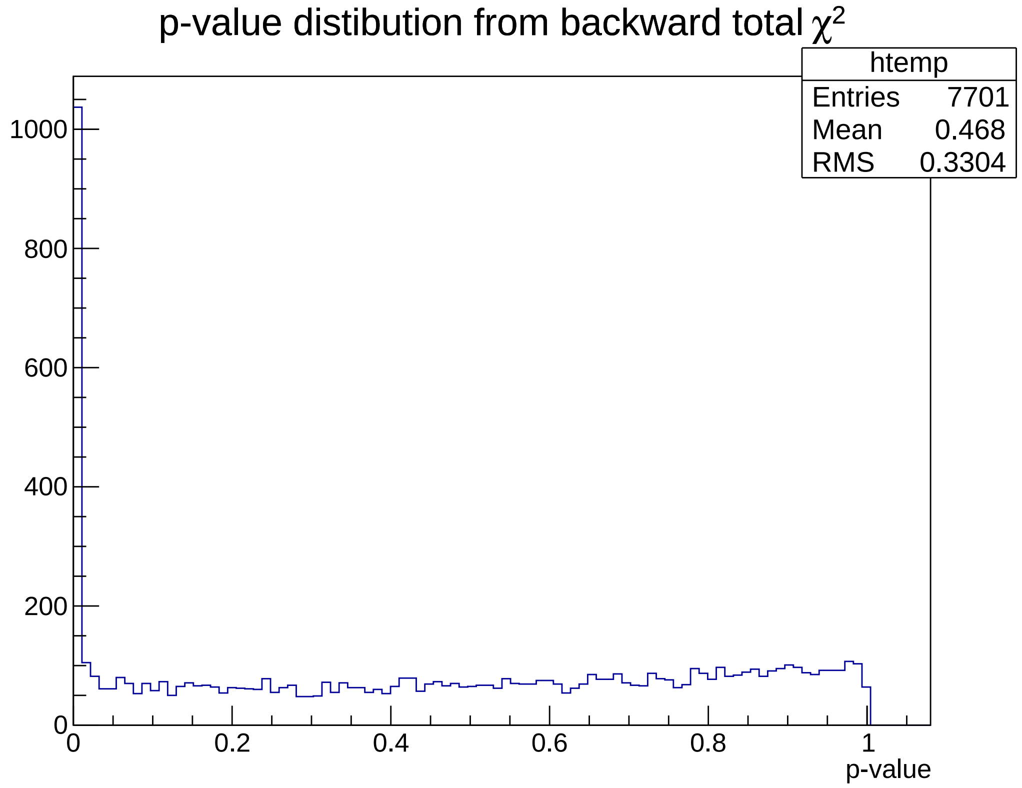
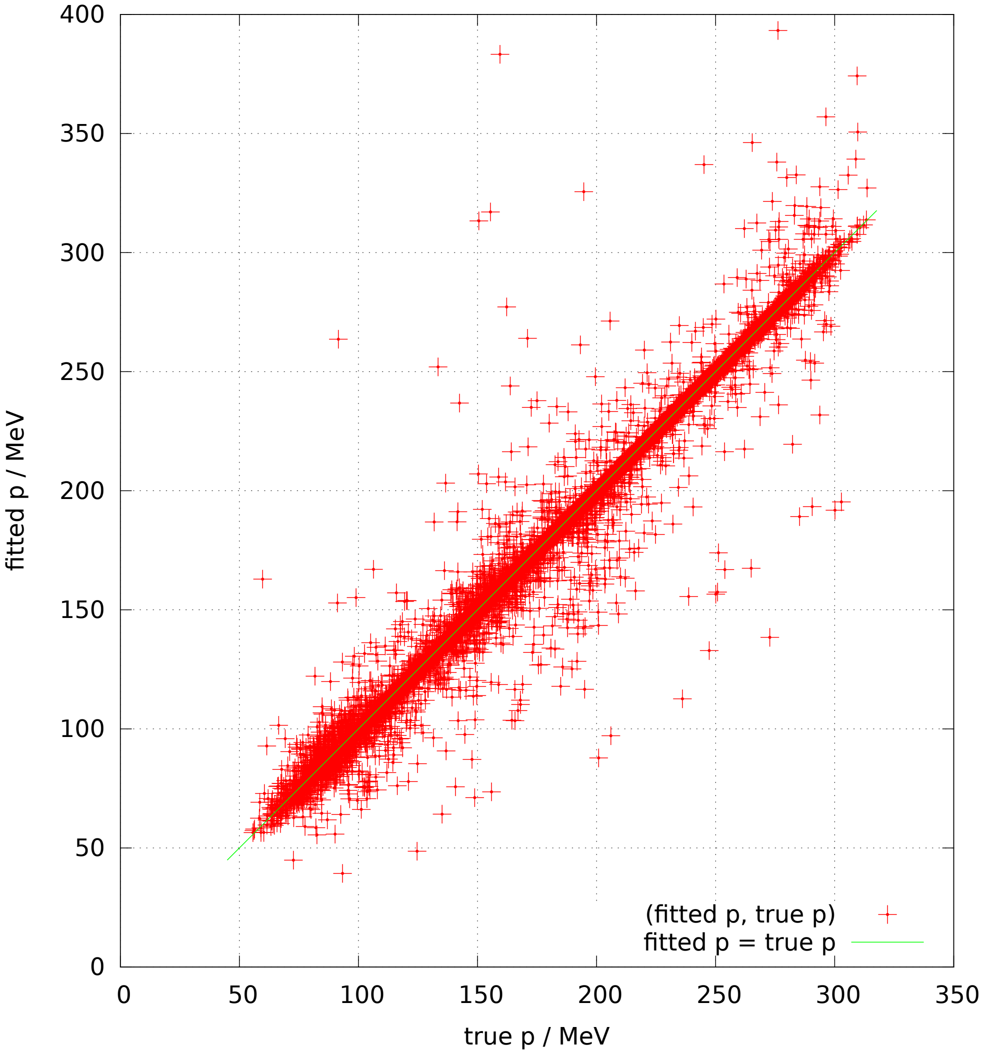


Figure 6.9: The momentum distribution of slow pions from D$^{*\pm}$ decays reconstructed in Belle I.



test variable	mean	std	trunc. mean	trunc. std	scaled MAD
res(x)/cm	−0.0246	1.1483	0.0004	0.0488	0.0361
res(y)/cm	0.0161	0.9644	0.0007	0.0493	0.0356
res(z)/cm	0.2239	3.9009	0.0020	0.0535	0.0369
res(p_T)/GeV	0.0004	0.0122	−0.0002	0.0020	0.0017
res(p)/GeV	−0.0072	0.0512	−0.0005	0.0020	0.0012
res($p_\mathrm{T}/p_\mathrm{T,t}$)	2.4064	47.1492	0.0175	0.4784	0.2737
res(r)/cm	0.2181	1.4834	0.0544	0.0551	0.0415
test variable	**mean**	**std**	**median**	**scaled MAD**	**outlier**
pull x	−0.3329	11.2846	−0.0157	1.0010	274
pull y	0.0805	12.8399	−0.0326	0.9989	269
pull z	0.8499	15.8671	0.0005	0.9986	330
pull p_T	−0.2470	4.4333	−0.2059	1.0754	217
pull p	−6.6213	50.8367	−0.3383	1.1316	263

Figure 6.10: Fitting results of slow pions found by the MCTrackCandCombiner without considering PXD hits. 10477 slow pions were found. 7701 were fitted successfully. 1031 of the fitted tracks had a p-value below 0.01 and were not included in the table. The truncation cuts away 10 % of the sample after the p-value cut.



p_{true}/MeV	0 – 100	100 – 150	150 – 200	200 – 250	250 – inf	all p
trackable $\pi^{\pm}$	2103	3433	3296	1941	1235	12008
found ratio	0.5739	0.9263	0.9163	0.9737	0.9538	0.8723
fitted ratio	0.4080	0.6082	0.6757	0.7877	0.8089	0.6413

Figure 6.11: Scatter plot of fitted and true momentum of the slow pions found by MCTrackCandCombiner without considering PXD hits. The y-distance of a point to the green line is the momentum residual of this track. The table shows the finding and fitting efficiency for different momenta ranges. The successful fit ratio is calculated with respect to the trackable $\pi^{\pm}$, not the found $\pi^{\pm}$. Very few extreme outlier tracks lie outside of the plot.

6.2 Robust reconstruction of B decay vertex position

CP violation studies will be a major part of the physics program in Belle II, as they have been in Belle. The measurement of the z-component of the distance of the decay vertices of the two B mesons created in a collision is a key ingredient to CP studies in the B meson system. To obtain this value, the positions of the B decay vertices have to be reconstructed with the best possible precision.

A common strategy in Belle was to fully reconstruct only one of the B decay chains, the one with an easy to identify and dominant decay mode, while the vertex positions of the daughters of the other B would not be reconstructed. Instead all remaining tracks would be fitted to one vertex after applying some very lose cuts beforehand. Whiles this strategy seems to be a perfect use case for a robust vertex fitter, this option was not available in Belle. There was a scheme to iterate over vertex fits and to identify and remove tracks causing the largest χ^2 increment in the Kalman filter based KFitter library. While KFitter was ported from Belle to Belle II, this rough robustification of KFitter was not; therefore a direct comparison of the Adaptive Vertex Fitter (AVF) in RAVE with this robustification is not possible in basf2.

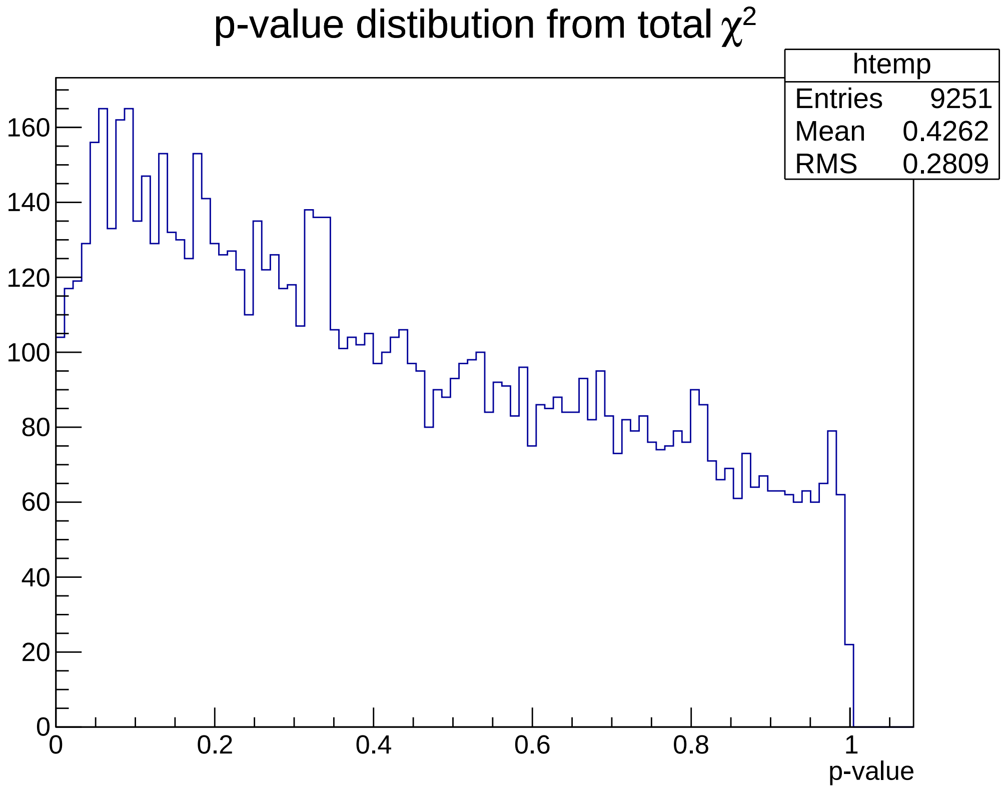
Instead the following scheme was conceived for the comparison: A set of tracks is fitted both with the AVF algorithm of RAVE and with the class VertexFitKFit. VertexFitKFit is KFitter's geometric vertex fitter. The weights assigned by the AVF are used to decide which tracks do not belong to the common vertex, and these are removed from the sample. The remaining set is then refitted with VertexFitKFit. If RAVE identifies the outlier tracks correctly, VertexFitKFit should yield good results when the outlier tracks are removed from the track sample.

The test was implemented by creating 10000 EvtGen events modified with a user DEC file. The DEC files forces the Υ(4s) to decay in B_0 and $\bar{\mathrm{B}}_0$, then one of the B_0 has to decay to J/Ψ plus K_s. The other B_0 decay is not modified. After the Geant4 simulation and track reconstruction the simulated truth information is used to remove all tracks originating from the B → J/Ψ side. Tracks with a point of closest approach too far from (0,0,0) are also deleted from the sample. All remaining reconstructed tacks are then passed to RAVE and KFitter. Finally, as mentioned above, the vertices are fitted again with KFitter after the removal of all tracks not belonging to the vertex according to RAVE.

This procedure was done twice: Once the tracks were reconstructed under perfect track finding conditions (MCTrackFinder) and once with SVD and CDC background and realistic track finders with help from the MCTrackCandCombiner module, as described in section 5.4.

The AVF algorithm was initialised with its default parameters, the same that were used in section 5.6: initial temperature of 256, annealing ratio of 0.25, and a σ_{cut} of 3. In both test cases all tracks were extrapolated to their point of closest approach to $(0,0,0)$ and only passed to the vertex fitter when the condition $|x| < 1/\sqrt{2}\,\text{cm} \wedge |y| < 1/\sqrt{2}\,\text{cm} \wedge |z| < 5\,\text{cm}$ was met at that point.

Figure and table 6.12 shows how the B decay vertices were reconstructed by RAVE with the MCTrackFinder. Only 9608 of the 10000 possible vertex fits were attempted

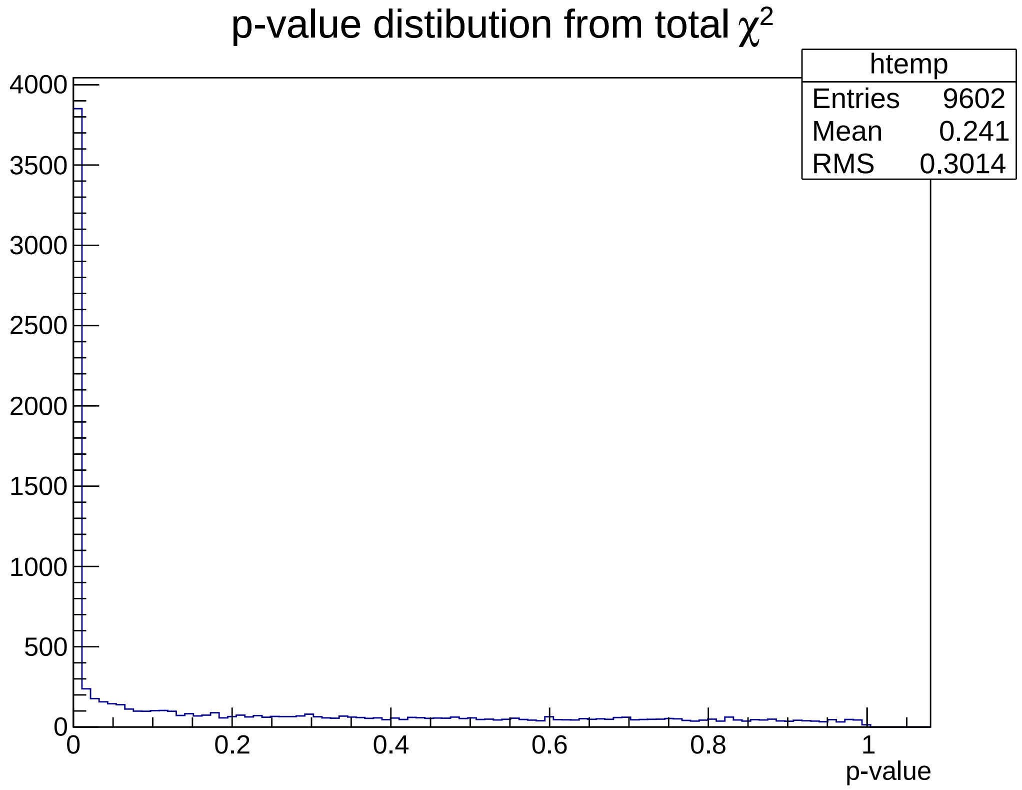


test variable	mean	std	trunc. mean	trunc. std	scaled MAD
res(x)/cm	0.0007	0.0304	0.0006	0.0104	0.0046
res(y)/cm	0.0006	0.0402	0.0002	0.0106	0.0045
res(z)/cm	0.0039	0.0445	0.0026	0.0111	0.0045
test variable	mean	std	median	scaled MAD	outlier
pull x	0.1780	10.1030	0.0514	1.4323	340
pull y	0.2569	10.4151	0.0095	1.3848	378
pull z	0.9854	9.3690	0.3564	1.5111	405

Figure 6.12: Result of 9251 successful RAVE vertex fits. 9608 fits were attempted. 43 vertices had a p-value below 0.005 and are not included in the table. The input tracks for the vertex fit were reconstructed under ideal conditions. The truncation ratio was 0.01.

because sometimes there were less than two tracks, which is not enough for a vertex fit in the absence of prior information about the interaction point. The p-value diagram shows no outliers at all, something one would expect from a robust fitting method. About 350 vertex fits failed, mostly because two tracks from different vertices were fed to the AVF. In this case, both tracks get a small weight and no unique vertex can be determined.

Figure and table 6.13 show essentially what is known from the construction of the test case: Many events contain tracks from different vertices, therefore a simple Kalman filter cannot correctly reconstruct the vertex in this case. While the residuals in the table are



test variable	mean	std	trunc. mean	trunc. std	scaled MAD
res(x)/cm	0.0006	0.0130	0.0004	0.0070	0.0041
res(y)/cm	0.0003	0.0133	0.0002	0.0068	0.0039
res(z)/cm	0.0017	0.0136	0.0015	0.0074	0.0038
test variable	mean	std	median	scaled MAD	outlier
pull x	0.1114	1.9429	0.0810	1.3494	94
pull y	0.0164	1.9109	0.0140	1.3066	101
pull z	0.4723	2.1253	0.3277	1.3899	123

Figure 6.13: Result of 9602 successful KFitter vertex fits. 9608 fits were attempted. 3610 vertices had a p-value below 0.005 and were not included in the table. The input tracks for the vertex fit were reconstructed under ideal conditions. The truncation ratio was 0.01.

even a bit better than the ones in the previous figure and table, this is only achieved by discarding over one third of the fitted vertices before calculating the residuals.

Most of the failed rave vertex fits occurred, when only 2 track survived the track reconstruction and the cut on the distance to $(0, 0, 0)$. It is very likely they did not originate at the same vertex. The in contrast to common Kalman vertex fit implementations, the AVF will refuse to fit a vertex if the minimal distance between tracks highly incompatible with the track error.

Figure and table 6.14 show that the number of bad vertex fits drops nearly to 0, if

KFitter gets the tracks that were identified as part of the common vertex by RAVE. The fitting of those RAVE selected tracks by a completely different fitter implementation without bad p-values, shows that the decisions of the AVF are correct, even in this extreme case where half of the tracks in an event are fitted to a single vertex. The AVF also works in a fully automatic fashion, without any tuning or interaction required from the user.

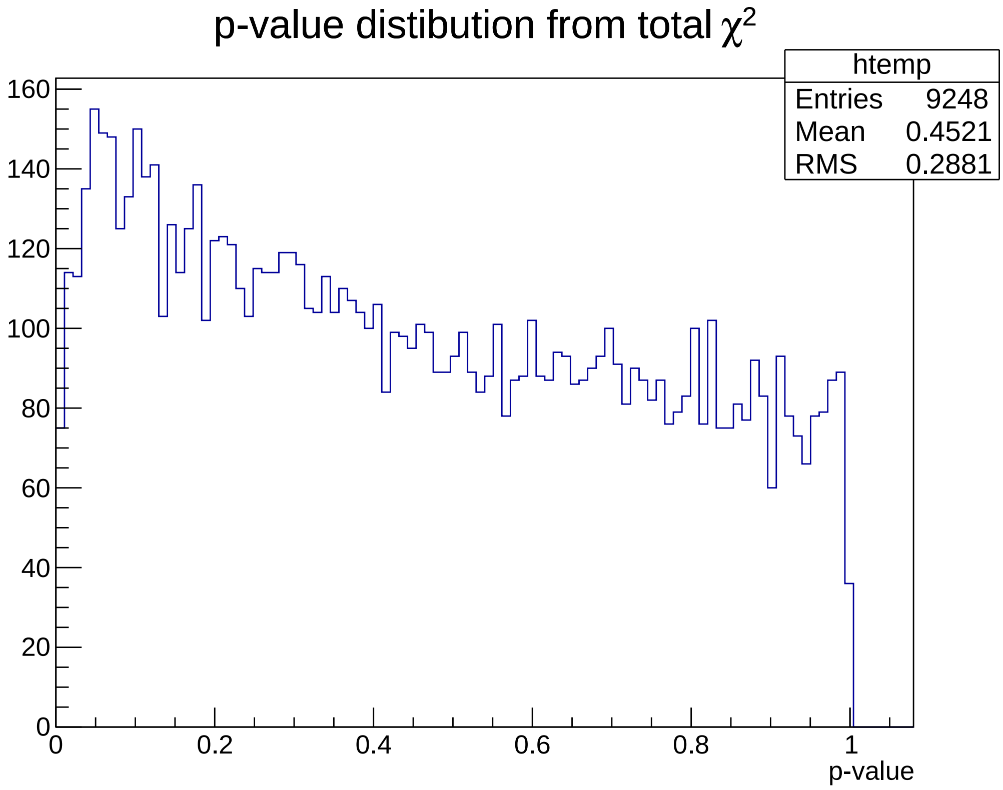
It is noteworthy that the KFitter pulls in figure and table 6.14 are clearly worse than the ones of the AVF, although KFitter now extracts its information from the same tracks as the AVF. This can be explained by the fact that tracks that are not rejected as outliers, but nevertheless do not fit perfectly to the the vertex are somewhat down-weighted by the AVF, whereas they have full weight in KFitter.

Figure and table 6.15 to figure and table 6.17 show the second test case, in which the same vertex fitting scheme was applied, but with realistic conditions in the track fitting stage. While the residual distributions obviously get wider and have heavier tails, the general picture stays the same, meaning the estimation of the B decay vertex works very well by just passing all the tracks from the non-tag side to the AVF.

Again KFitter produces pull distributions with broader core width and more outliers as RAVE, which can be seen by comparing 6.15 with 6.17. Also the truncated standard deviation of the residuals is larger for KFitter.

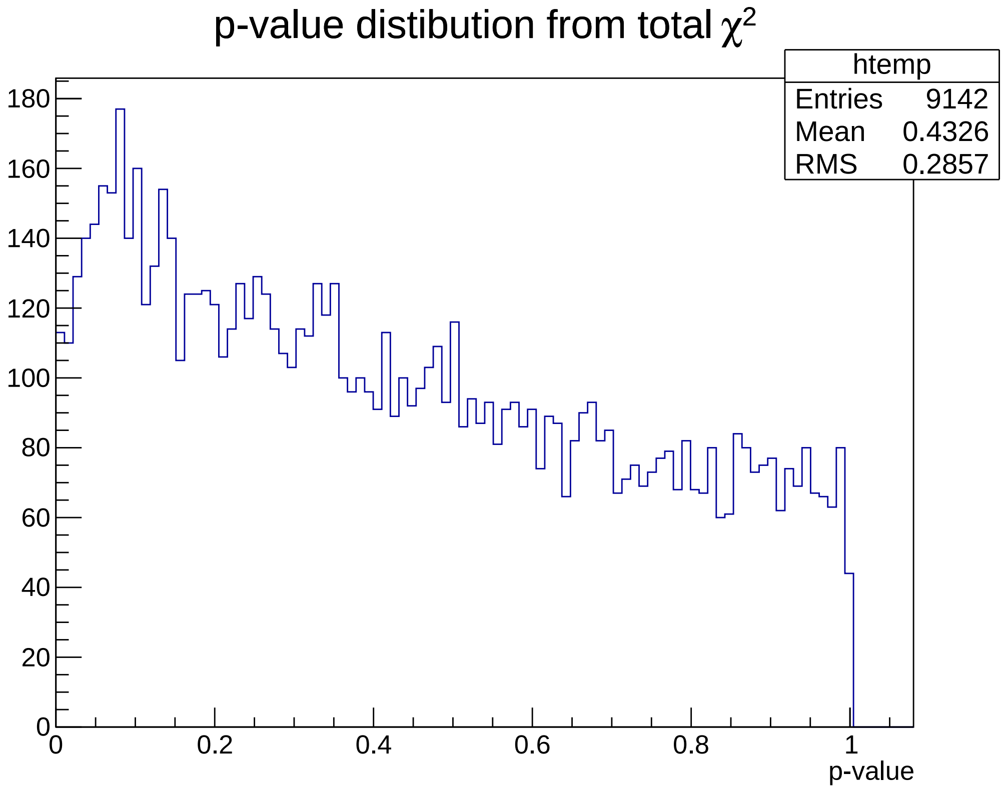
The p-value of the fitted vertices and the vertex pull distribution generally show an underestimation of the error. As the test in chapter 5 showed, the fitting of EvtGen events results in a high number of pull distribution outliers, even after a p-value cut. The track p-value outliers together with the pull distribution outliers are sufficiently frequent to suspect that the majority of vertex fits will a least use one track with a significantly underestimated error. This at least contributes to the too small p-value and too wide pull distribution of the fitted vertices. If the track sample has perfect pulls and p-values, RAVE produces perfect pulls and p-values, too, as established in section 5.6.

All tables in this section show a bias in the z coordinate of the fitted vertices. The fits are done in the laboratory frame, and in this frame the particles have on average a larger momentum component in the positive z direction than in the opposite direction. Reconstructing the vertices in the centre of mass frame should therefore make this bias vanish.



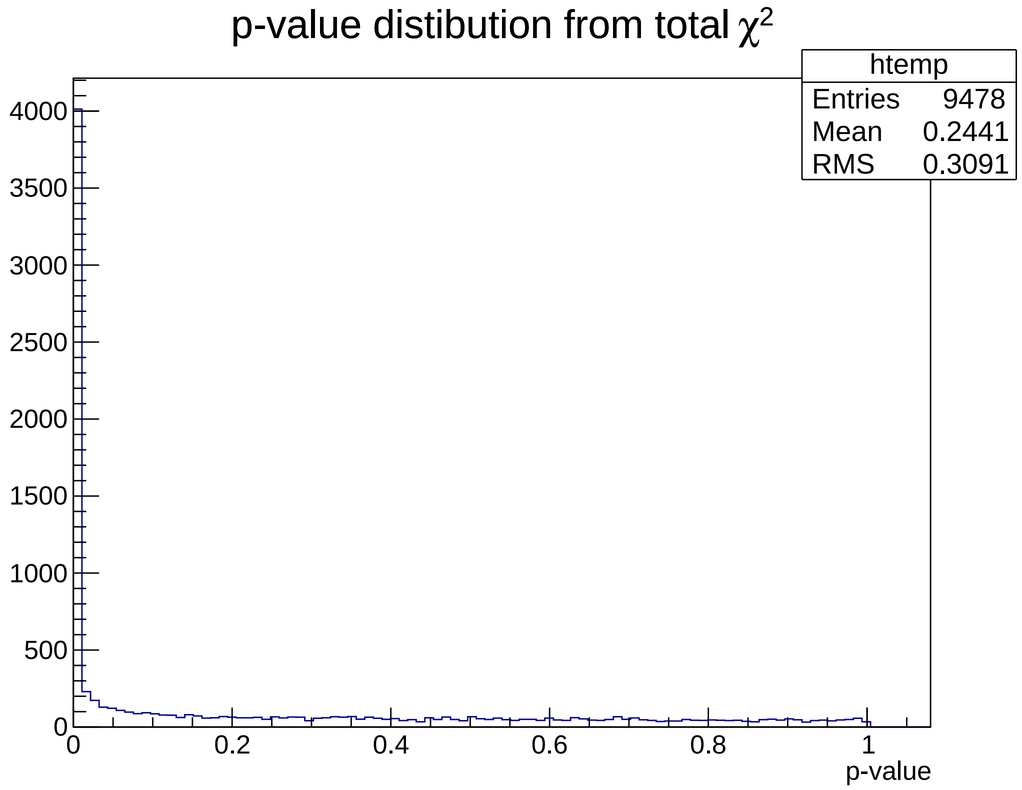
test variable	mean	std	trunc. mean	trunc. std	scaled MAD
res(x)/cm	0.0008	0.0309	0.0007	0.0105	0.0047
res(y)/cm	0.0006	0.0403	0.0002	0.0108	0.0046
res(z)/cm	0.0039	0.0445	0.0026	0.0112	0.0045
test variable	mean	std	median	scaled MAD	outlier
pull x	0.2057	7.1398	0.0655	1.6000	527
pull y	0.1113	8.7799	0.0140	1.5521	544
pull z	1.1890	9.6024	0.3793	1.6623	593

Figure 6.14: Result of 9248 successful KFitter vertex fits. 9251 fits were attempted after. 16 vertices had a p-value below 0.005 and were not included in the table. The input tracks for the vertex fit were reconstructed under ideal conditions. The truncation ratio was 0.01.



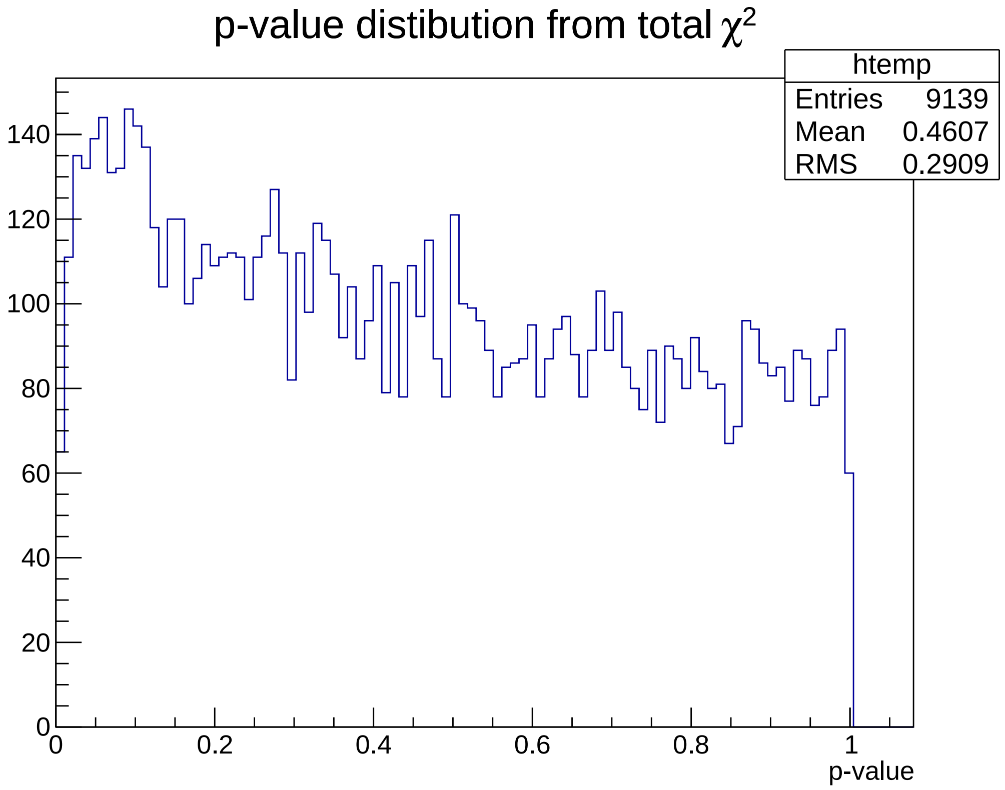
test variable	mean	std	trunc. mean	trunc. std	scaled MAD
res(x)/cm	0.0003	0.0667	0.0004	0.0219	0.0053
res(y)/cm	−0.0007	0.0868	−0.0001	0.0212	0.0050
res(z)/cm	0.0038	0.1174	0.0027	0.0214	0.0050
test variable	mean	std	median	scaled MAD	outlier
pull x	0.1075	5.1869	0.0401	1.3633	266
pull y	0.0425	4.7919	0.0312	1.3228	287
pull z	0.7482	4.0011	0.4187	1.4317	324

Figure 6.15: Result of 9142 successful RAVE vertex fits. 9481 fits were attempted. 56 vertices had a p-value below 0.005 and were not included in the table. The input tracks for the vertex fit were reconstructed under realistic conditions. The truncation ratio was 0.01.



test variable	mean	std	trunc. mean	trunc. std	scaled MAD
res(x)/cm	0.0012	0.0699	0.0001	0.0151	0.0047
res(y)/cm	−0.0010	0.0942	−0.0002	0.0153	0.0046
res(z)/cm	0.0033	0.1305	0.0020	0.0166	0.0046
test variable	mean	std	median	scaled MAD	outlier
pull x	0.0632	2.2023	0.0391	1.3107	94
pull y	0.0241	1.9323	0.0366	1.3025	97
pull z	0.5755	2.2970	0.4285	1.3863	126

Figure 6.16: Result of 9478 successful KFitter vertex fits. 9481 fits were attempted. 3815 vertices had a p-value below 0.005 and were not included in the table. The input tracks for the vertex fit were reconstructed under realistic conditions. The truncation ratio was 0.01.



test variable	mean	std	trunc. mean	trunc. std	scaled MAD
res(x)/cm	0.0003	0.0691	0.0004	0.0231	0.0054
res(y)/cm	−0.0006	0.0876	−0.0000	0.0222	0.0051
res(z)/cm	0.0040	0.1179	0.0027	0.0226	0.0051
test variable	**mean**	**std**	**median**	**scaled MAD**	**outlier**
pull x	0.0769	10.8194	0.0395	1.5289	497
pull y	0.0468	11.6889	0.0312	1.4822	513
pull z	0.7677	12.5850	0.4466	1.5983	535

Figure 6.17: Result of 9139 successful KFitter vertex fits. 9142 fits were attempted after. 13 vertices had a p-value below 0.005 and were not included in the table. The input tracks for the vertex fit were reconstructed under realistic conditions. The truncation ratio was 0.01.

Bibliography

[1] Fabjan C W and Schopper H 2011 *Detectors for Particles and Radiation. Part 1: Principles and Methods* (*Landolt-Börnstein - Group I Elementary Particles, Nuclei and Atoms* vol 21B1) (Springer Berlin Heidelberg)

[2] Lettenbichler J 2012 *Pattern recognition in the Silicon Vertex Detector of the Belle II experiment* Master's thesis University of Vienna

[3] Frühwirth R and Strandlie A 1999 *Computer Physics Communications* **120** 197–214 ISSN 0010-4655 URL http://www.sciencedirect.com/science/article/pii/S0010465599002313

[4] Highland V 1975 *Nuclear Instruments and Methods* **129** 497–499

[5] Nakamura, K and Particle Data Group 2010 *Journal of Physics G: Nuclear and Particle Physics* **37** 075021 URL http://stacks.iop.org/0954-3899/37/i=7A/a=075021

[6] Adam W, Frühwirth R, Strandlie A and Todorov T 2005 *Journal of Physics G: Nuclear and Particle Physics* **31** N9

[7] Waltenberger W 2004 *Development of Vertex Finding and Vertex Fitting Algorithms for CMS* Ph.D. thesis Vienna University of Technology

[8] Waltenberger W 2011 *IEEE Transactions on Nuclear Science* **58** 434–444

[9] Nadler M and Frühwirth R 2011 *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **648** 246–253 ISSN 0168-9002 URL http://www.sciencedirect.com/science/article/pii/S0168900211010321

[10] Dempster A, Laird N and Rubin D 1977 *Journal of the Royal Statistical Society. Series B (Methodological)* 1–38

[11] Frühwirth R, Regler M, Bock R, Grote H and Notz D 2000 *Data analysis techniques for high-energy physics* vol 11 (Cambridge University Press)

[12] Blobel V 2006 *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **566** 14–17

[13] Frühwirth R 1987 *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **262** 444–450

[14] Littlefield B and Hanselman D 2004 *Mastering Matlab 7* (Prentice Hall)

[15] Chatrchyan S, Hmayakyan G, Khachatryan V, Sirunyan A, Adam W, Bauer T, Bergauer T, Bergauer H, Dragicevic M, Erö J *et al.* 2008 *Journal of Instrumentation* **3** S08004 URL http://iopscience.iop.org/1748-0221/3/08/S08004

[16] Gjersdal H, Frühwirth R, Nadler M and Strandlie A 2013 *Journal of Instrumentation* **8** P01009 URL http://stacks.iop.org/1748-0221/8/i=01/a=P01009

[17] Grenier P, Alimonti G, Barbero M, Bates R, Bolle E, Borri M, Boscardin M, Buttar C, Capua M, Cavalli-Sforza M *et al.* 2011 *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **638** 33–40

[18] Gjersdal H personal communication.

[19] Abe T, Adachi I, Adamczyk K, Ahn S, Aihara H, Akai K, Aloi M, Andricek L, Aoki K, Arai Y *et al.* 2010 Belle II technical design report Tech. rep.

[20] Aushev T, Bartel W, Bondar A, Brodzicka J, Browder T *et al.* 2010 (*Preprint* 1002.5012)

[21] Allison J, Amako K, Apostolakis J, Araujo H, Dubois P *et al.* 2006 *IEEE Transactions on Nuclear Science* **53** 270

[22] Lange D 2001 *Nuclear Instruments and Methods* **A462** 152–155

[23] Brun R and Rademakers F 1997 *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **389** 81–86 ISSN 0168-9002 URL http://www.sciencedirect.com/science/article/pii/S016890029700048X

[24] Moll A 2011 *Journal of Physics: Conference Series* **331** 032024 URL http://stacks.iop.org/1742-6596/331/i=3/a=032024

[25] Glattauer R, Mitaroff W, Frühwirth R, Lettenbichler J and Nadler M *To be published in Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*

[26] Lutz O 2012 *Search for $B \to h^* \nu \bar{\nu}$ decays at Belle and development of track finding for Belle II* Ph.D. thesis Karlsruhe Institute of Technology

[27] Frost O 2013 *A Local Tracking Algorithm for the Central Drift Chamber of Belle II* Master's thesis Karlsruhe Institute of Technology

[28] Höppner C, Neubert S, Ketzer B and Paul S 2010 *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **620** 518–525 ISSN 0168-9002 URL http://www.sciencedirect.com/science/article/pii/S0168900210007473

[29] Nadler M and Frühwirth R 2012 *Journal of Physics: Conference Series* **396** 022037 URL http://stacks.iop.org/1742-6596/396/i=2/a=022037