

# Application of a Kalman filter and a Deterministic Annealing filter for track reconstruction in the HADES experiment

Masterarbeit von Erik Krebs

vorgelegt am  
Fachbereich Physik  
der Goethe-Universität  
in Frankfurt am Main

durchgeführt am  
GSI Helmholtzzentrum für Schwerionenforschung,  
Darmstadt

Frankfurt am Main, den 28.03.2012

## Abstract

The **H**igh-**A**cceptance **D**i**E**lectron **S**pectrometer (HADES) is installed at the Helmholtzzentrum für Schwerionenforschung in Darmstadt, Germany. The physics program of HADES is focused on the properties of the light vector mesons  $\rho$ ,  $\omega$  und  $\Phi$  in nuclear matter. These mesons decay with a small branching ratio into dileptons. The momentum reconstruction of these leptons allows to determine some properties of the vector mesons.

The Kalman filter is a set of mathematical equations that allows to estimate the internal state of a dynamic system perturbed by process noise. It is an optimal estimator in a sense that it minimizes the variance of the estimation error. Since its discovery in 1960 the Kalman filter has become a popular state estimator and has been successfully applied in many different areas like satellite navigation, weather forecast or estimation of battery charge.

An environment with multiple, ambiguous and fake measurements poses a challenge to the Kalman filter as it does not include a mechanism to handle competing measurements or reject measurements. The Deterministic Annealing filter is an extension of the Kalman filter that allows competition between measurements and rejection of fake measurements.

A Kalman filter and Deterministic Annealing filter have been implemented for future use in the track reconstruction software of the HADES experiment. The performance of the algorithms has been studied using simulated Au+Au reactions and compared to the current algorithm for momentum reconstruction.

Chapter 2 gives an overview of the HADES detector. The basics of the Kalman filter and Deterministic Annealing filter are presented in chapter 4. The adaptation of these methods for HADES is described in chapter 6. Details about the software implementation can be found in chapter 7. The results of the simulation studies are presented in chapter 8.

# Contents

<b>Contents</b>	<b>1</b>
<b>List of Figures</b>	<b>3</b>
<b>List of Tables</b>	<b>6</b>
<b>List of Symbols</b>	<b>7</b>
<b>1 Introduction</b>	<b>9</b>
1.1 The Physics Program of HADES . . . . .	9
1.2 Track Reconstruction . . . . .	10
<b>2 The HADES Detector</b>	<b>11</b>
2.1 START and VETO Detectors . . . . .	11
2.2 Particle Identification . . . . .	13
2.2.1 The RICH Detector . . . . .	13
2.2.2 The META Detector . . . . .	14
2.3 The Tracking System . . . . .	15
2.3.1 The Magnet . . . . .	16
2.3.2 The Multi-Wire Drift Chambers . . . . .	17
2.4 The Forward Wall Hodoscope . . . . .	20
<b>3 Track Reconstruction in HADES</b>	<b>21</b>
<b>4 State Estimation with the Kalman Filter</b>	<b>24</b>
4.1 Definitions . . . . .	24
4.2 The Kalman Filter . . . . .	26
4.2.1 The Linear Kalman Filter . . . . .	26
4.2.2 The Extended Kalman Filter . . . . .	29
4.2.3 Goodness of the Estimate . . . . .	30
4.2.4 Alternative Formulations of the Kalman Filter . . . . .	32
4.2.5 Yet More Kalman Filter and Alternatives . . . . .	33
4.3 The Deterministic Annealing Filter . . . . .	33
<b>5 Runge-Kutta Methods</b>	<b>36</b>

<b>6</b>	<b>Adaptation of the Kalman Filter for HADES</b>	<b>39</b>
6.1	Track Representation . . . . .	40
6.2	Propagation through the Magnetic Field . . . . .	41
6.3	Measurement Vector and Projector Matrix . . . . .	45
6.3.1	Reconstructed Segment Hits . . . . .	45
6.3.2	Drift Chamber Measurements . . . . .	46
6.4	Energy Loss and Multiple Scattering . . . . .	56
<b>7</b>	<b>Implementation in HYDRA</b>	<b>63</b>
7.1	The HYDRA Framework for Track Reconstruction . . . . .	63
7.2	The Classes in the Kalman Filter Library . . . . .	64
7.3	Using the Kalman Filter Library . . . . .	67
7.3.1	Initialization . . . . .	67
<b>8</b>	<b>Results</b>	<b>71</b>
8.1	Tests with the Kalman Filter . . . . .	73
8.1.1	Parameters and Settings . . . . .	73
8.1.2	Covariance Matrix and Roundoff Errors . . . . .	75
8.1.3	Energy Loss and Multiple Scattering . . . . .	76
8.1.4	Dependencies of Track Reconstruction Performance on the Polar Angle . . . . .	81
8.2	Effect of Non-Normal Error Distributions . . . . .	81
8.3	Parameters Settings for the Deterministic Annealing Filter . . . . .	87
8.4	Comparison of Methods . . . . .	89
<b>9</b>	<b>Summary and Outlook</b>	<b>102</b>
<b>A</b>	<b>Settings for the Kalman Filter and Retrieving Output Data</b>	<b>104</b>
A.1	Settings for the Kalman Filter . . . . .	104
A.2	Retrieving Output Data . . . . .	107
<b>B</b>	<b>Additional Plots</b>	<b>112</b>
	<b>Bibliography</b>	<b>123</b>

# List of Figures

2.1	A 3-D explosion view and a side view schematic of the HADES detector. . . . .	12
2.2	Schematics of the RICH detector [Tar10]. . . . .	13
2.3	Side and back view of the magnet. . . . .	16
2.4	The magnetic field along and perpendicular to the beam axis. . .	17
2.5	The HADES tracking system. . . . .	18
2.6	Wire orientations (a) of an MDC plane and layout of the geometry of a drift cell (b). . . . .	18
3.1	Cluster search for fired wires. . . . .	22
3.2	Principle of the track candidate search finding an inner and outer track segment. Only one MDC layer per module is shown [HAD09].	23
4.1	Basic operation of the Kalman filter. . . . .	27
4.2	$\chi^2$ probability density function and distributions with different degrees of freedom $\nu$ . . . . .	32
4.3	Measurement weight as a function of standardized distance for different annealing factors. . . . .	35
6.1	Example track traversing the detector [Gal12]. . . . .	40
6.2	The track parameterization in HADES [Rus06]. . . . .	41
6.3	Simulated signals of a drift chamber. Both pictures were taken from [Mar05]. . . . .	47
6.4	Track parameterization of a particle traversing a drift cell. . . .	48
6.5	Time-of-flight residuals. . . . .	49
6.6	The error for time-of-flight correction if the individual values for each layer are taken instead of the weighted mean in figure 6.6. .	50
6.7	The wire coordinate system. . . . .	51
6.8	A particle passed through two neighbouring drift cells. . . . .	52
6.9	A virtual layer is created that is perpendicular on the predicted track state and the sense wires. . . . .	53
6.10	The minimum distances are projected onto the virtual layer. . . .	54
6.11	The projections of the minimum distances are transformed into a common coordinate system. . . . .	55

6.12	The dimensions of a drift cell in the different MDC modules [Mar05].	58
6.13	Relative contributions of the MDC different materials to the radiation length. . . . .	59
6.14	Energy loss by radiation as a function of particle energy. . . . .	61
7.1	Class diagram for the Kalman filter library. . . . .	65
7.2	The components of the HKalSystem class. . . . .	66
7.3	The message exchange between the participating classes during a typical track reconstruction procedure is shown in this sequence diagram. . . . .	67
8.1	The momentum and polar angle distributions of the different particle species in the test sample. Included are only tracks that have been reconstructed by the segment fit. . . . .	72
8.2	Momentum resolution of the spline fit. . . . .	73
8.3	The momentum resolution of the Kalman filter compared to the input from the spline fit. . . . .	74
8.4	A comparison of the momentum resolution for forward and backward iteration. . . . .	74
8.5	The $\chi^2$ over the momentum when including and excluding the effects of multiple scattering. . . . .	78
8.6	The efficiency plotted against the momentum with and without including the effects of multiple scattering. . . . .	78
8.7	Energy loss by ionization from the Kalman filter and the simulation.	80
8.8	The energy loss of positrons as calculated by the Kalman filter. .	81
8.9	The momentum resolution as a function of the momentum for different polar angle $\Theta$ ranges. . . . .	82
8.10	The polar angle $\Theta$ as a function of the azimuthal angle $\phi$ for tracks that could not be reconstructed or were cut due to high $\chi^2$ . . . . .	82
8.11	The momentum resolution plotted over the momentum for different polar angle ranges. . . . .	83
8.12	The momentum estimates from the spline fit of proton tracks with ideal and real tracking. . . . .	84
8.13	Probability distributions of the Kalman filter for proton tracks. .	86
8.14	The momentum resolution and efficiency of the DAF for different annealing schemes. . . . .	88
8.15	The assignment probabilities for the case that there are two competing measurements in an MDC layer belonging to the same track. . . . .	88
8.16	The momentum resolution and efficiency of the DAF for different cut-off parameters. . . . .	89
8.17	Comparison of the momentum resolution for different $\Theta$ ranges for positron tracks. . . . .	93
8.18	Probability distributions for positron tracks. . . . .	94
8.19	The polar angle $\Theta$ over the azimuthal angle $\phi$ for positron tracks that could not be reconstructed or were cut due to high $\chi^2$ . . . . .	95

8.20	Efficiency over momentum for positron tracks. . . . .	96
8.21	Efficiency over the polar angle $\Theta$ for positron tracks. . . . .	97
8.22	Comparison of the momentum resolution for different $\Theta$ ranges for proton tracks. . . . .	98
8.23	The polar angle $\Theta$ over the azimuthal angle $\phi$ for proton tracks that could not be reconstructed or were cut due to high $\chi^2$ . . . . .	99
8.24	Efficiency over momentum for proton tracks. . . . .	100
8.25	Efficiency over the polar angle $\Theta$ for proton tracks. . . . .	101
B.1	Comparison of the momentum resolution for positron tracks. . . . .	113
B.2	Comparison of $\chi^2$ distributions for positron tracks. . . . .	114
B.3	Comparison of the momentum resolution for electron tracks. . . . .	115
B.4	Comparison of the momentum resolution for different $\Theta$ ranges for electrons. . . . .	116
B.5	The polar angle $\Theta$ over the azimuthal angle $\phi$ for electron tracks that could not be reconstructed or were cut due to high $\chi^2$ . . . . .	117
B.6	Efficiency over momentum for electron tracks. . . . .	118
B.7	Efficiency over the polar angle $\Theta$ for electron tracks. . . . .	119
B.8	Comparison of the momentum resolution for proton tracks. . . . .	120
B.9	Comparison of $\chi^2$ distributions for proton tracks. . . . .	121
B.10	Probability distributions for proton tracks. . . . .	122

# List of Tables

6.1	Materials and diameters of the different wire types [Mün10]. . . . .	57
6.2	Effective material properties of the drift chambers. . . . .	58
8.1	Comparison of the momentum resolution without and with smoothing. . . . .	73
8.2	The Kalman filter with different amount of iterations. . . . .	75
8.3	The effect of different initial covariance matrices and numerically more stable formulations of the Kalman filter equations. . . . .	77
8.4	Difference between ideal and real tracking for the spline fit. . . . .	85
8.5	Effects of different input methods on the Kalman filter. . . . .	85
8.6	Difference between ideal and real tracking for the global fit. . . . .	85
8.7	Comparison of the momentum resolution for proton tracks with different input methods for the initial momentum. . . . .	86
8.8	Tested annealing schemes. . . . .	89
8.9	Comparison of the momentum resolution for positron tracks. . . . .	91
8.10	Comparison of the momentum resolution for electron tracks. . . . .	91
8.11	Comparison of the momentum resolution for proton tracks. . . . .	91



# List of Symbols

$C$	State estimation error covariance matrix, 28
$F$	State propagation matrix, 24
$f$	State propagation function, 30
$G$	Input matrix in the linear Kalman filter., 24
$\Gamma$	The gamma function [PTVF07, chapt. 6.1]., 31
$\gamma$	Incomplete gamma function [PTVF07, chapt. 6.2]., 31
$H$	Projection matrix., 24
$h$	Projection function that extracts a measurement vector from a state., 30
$K$	Kalman gain matrix, 28
$k$	Time index or index of measurement point., 25
$m$	Measurement vector., 24
$N$	Number of measurements., 26
$n$	Number of competing hits at a measurement site, 34
$\nu$	Number of degrees of freedom, 31
$P$	Regularized gamma function [PTVF07, chapt. 6.2]., 31
$Q$	Process noise covariance matrix., 25
$r$	Dimension of measurement vector., 24
$s$	Dimension of state vector., 24
$T$	Annealing factor., 34

$\mathbf{u}$  Process noise vector., 24  
 $V$  Measurement error covariance matrix., 25  
 $\mathbf{v}$  Measurement noise vector., 24  
 $\mathbf{w}$  Process noise vector., 24  
 $\mathbf{x}$  State vector., 24

# Chapter 1

## Introduction

### 1.1 The Physics Program of HADES

The HADES detector is installed at the heavy-ion synchrotron SIS18<sup>1</sup> at the GSI Helmholtzzentrum für Schwerionenforschung Darmstadt. Its physics program is mainly focused on research about the properties of hadrons when embedded into hot and dense nuclear matter. Additional research topics include the studies of vector meson-nucleon coupling strengths, the electromagnetic form factor of the nucleon in the time-like region and the validity of the Vector Meson Dominance model [Gal09].

Heavy-ion collisions like C+C or Au+Au at energies of 1–2A GeV produce a reaction volume of high density for the short time period of up to 10 fm/c. Theoretical models based on Quantum Chromo Dynamics predict modifications in the properties of hadrons under such conditions. The light vector mesons  $\rho$ ,  $\omega$  und  $\Phi$  are especially suited for such investigations. The lifetime of these mesons is comparable to that of the duration of the compression phase produced during heavy ion collisions. They decay with a small branching ratio into dileptons which are not subject to the strong force and can thus leave the reaction zone unobstructed. Reconstructing the momentum of these leptons allows to determine some properties of the vector mesons like for example their invariant mass distribution in the hot, dense nuclear matter.

The branching ratio of the decays into dileptons is merely  $10^{-4}$  to  $10^{-5}$ . In order to do dilepton spectroscopy a detector should have:

- a high geometric acceptance to indentify  $e^+e^-$  pairs,
- the capability to process high data rates at high beam intensities,
- a high invariant-mass resolution of a few percent in the mass range of the  $\rho$  and  $\omega$  mesons is required to distinguish in-medium effects [Mar05, p.8].

The HADES detector has been designed to meet these requirements.

---

<sup>1</sup>Schwerionen Synchrotron (Heavy Ion Synchrotron)

## 1.2 Track Reconstruction

The track reconstruction at HADES currently consists of the following steps:

1. Track candidate search and cluster finding
2. Segment fitting
3. Momentum reconstruction
4. Particle identification.

More details about the track reconstruction procedure can be found in chapter 3. The current algorithm for the final momentum reconstruction has been in use since 2004. The goal of this work is to study if the Kalman filter and Deterministic Annealing Filter can further improve the momentum reconstruction. A disadvantage of the current reconstruction method is that the segment fit neglects the residual magnetic field in the multi-wire drift chambers (MDCs) and uses a straight line model which is not accurate for particles with low momenta.

The Kalman filter is suitable for efficient track reconstruction at high event rates and is a promising method to improve the aforementioned weaknesses in the current tracking software. In addition to the magnetic field present in the MDCs it takes multiple scattering and energy loss into account which should result in an improved momentum resolution for particles with momenta of less than  $250 \text{ MeV}/c$ . An additional goal is to obtain a better  $\chi^2$  normalization, so that the  $\chi^2$  value is a better measurement for reconstruction quality. No large matrices have to be inverted in the Kalman filter and the computational effort increases only linearly with the number of measurements. The anticipated gain in computing time is also an important aspect in regard to the planned analysis of heavy-ion collisions like Au+Au and the occurring high particle multiplicities.

The Kalman filter optimizes the variances of the estimation error for linear systems. In the case of nonlinear systems like the movement of a charged particle in a magnetic field the system model has to be linearized. This is a possible drawback. However, the linearization needs to be valid over short ranges only.

To avoid systematics from the segment fitter the Kalman filter has been adapted to work with the drift time information from the drift chambers directly. To fake hits and competition between measurements resolve an extension of the Kalman filter, the Deterministic Annealing Filter, is used.

## Chapter 2

# The HADES Detector

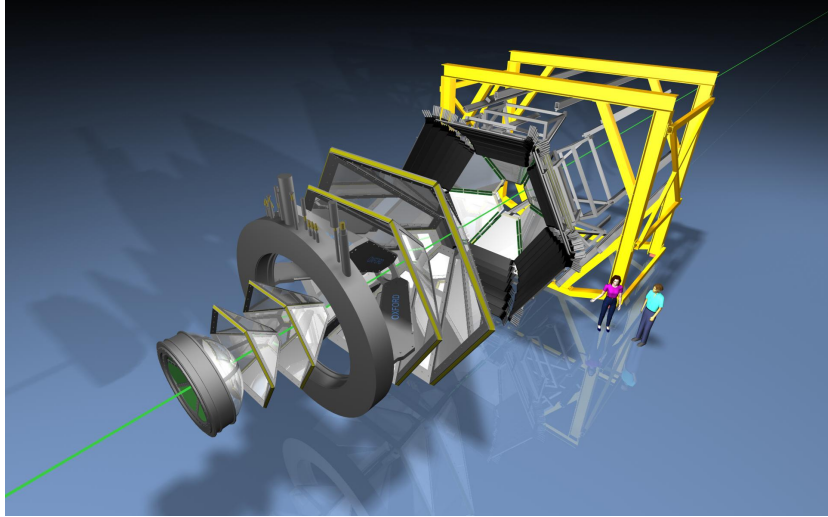
The HADES detector is divided into six equal sectors of trapezoidal shape aligned symmetrically around the beam axis. It covers an acceptance of  $18^\circ$ - $85^\circ$  over the polar angle. The azimuthal acceptance is limited to 85% due to the magnetic coils. The detector is depicted in figure 2.1. It consists of these components [HAD09, B<sup>+</sup>09]:

1. START and VETO detectors for reaction time measurements,
2. RICH and META detectors for particle identification,
3. a tracking system with multi-wire drift chambers (MDCs) and magnetic coils and
4. a forward wall hodoscope to detect spectator protons and characterize events.

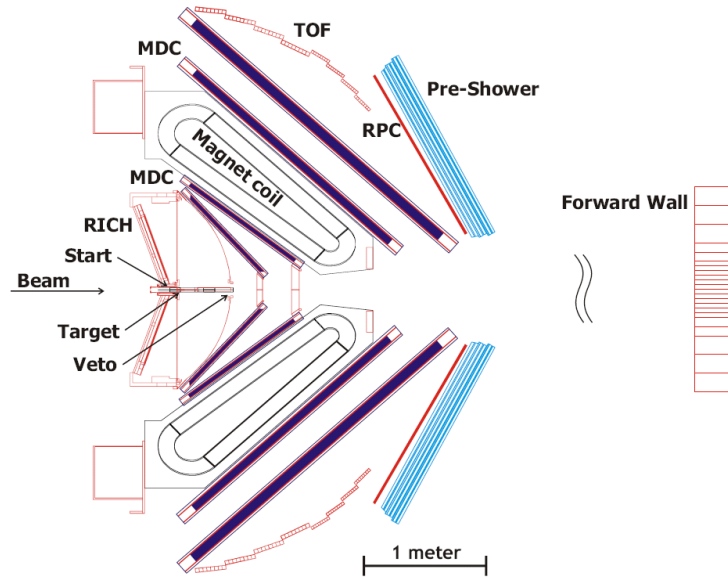
### 2.1 START and VETO Detectors

START and VETO each consist of 8 diamond strips detectors. With a thickness of  $100\ \mu m$ , multiple scattering and production of secondary particle are kept low. The detectors are placed in front of and behind the target, respectively. They provide a start time for the time of flight measurement of the particles between the target and TOF detectors. Additionally, the VETO detector helps to reject events.

Using the START detector is not possible for high-intensity ( $\geq 10^7\ Hz$  proton beams) since the induced secondary particles prevent a stable RICH operation. Hence, there is no start time reference for tracks in the same event and the difference in time of flight with respect to the fastest particle is measured instead.



(a)



(b)

Figure 2.1: A 3-D explosion view and a side view schematic of the HADES detector.

## 2.2 Particle Identification

### 2.2.1 The RICH Detector

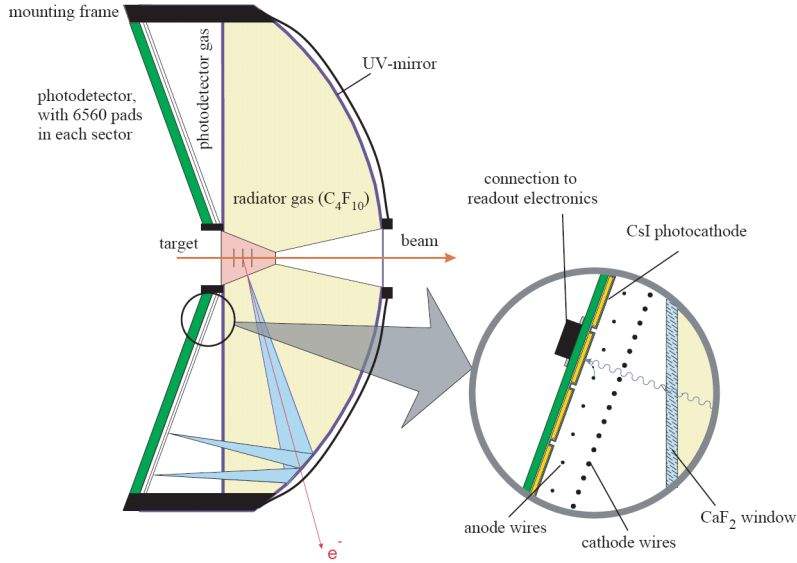


Figure 2.2: Schematics of the RICH detector [Tar10].

The particles produced at the target during a heavy-ion collision first reach the **R**ing **I**maging **C**herenkov (RICH, fig. 2.2) detector. They pass through a radiator gas with refraction index  $n$ . If the particle's velocity  $\beta$  is larger than the speed of light  $c/n$  in that medium then it emits Cherekov light. The light cone is centred around the particle's trajectory and has an opening angle of  $\cos \Theta_C = 1/(\beta \cdot n)$ . The photons are reflected onto a photosensitive detector by a spherical carbon fibre mirror and focused to rings.

The radiator gas used is perfluorobutane ( $C_4F_{10}$ ) which has a refraction index of  $n = 1.00151$ . Only particles with a relativistic velocity of

$$\beta \geq 1/n \simeq 0.9985$$

which corresponds to a Lorentz factor of

$$\gamma \geq \sqrt{\frac{n^2}{n^2 - 1}} \simeq 18.2$$

can be detected. Hence, the opening angle  $\Theta_C$  of the light cone is nearly constant. The electrons and positrons produced during heavy-ion collisions at the relevant kinetic energies of 1–2  $AGeV$  exceed the threshold while the hadrons stay below the threshold of the RICH detector reaching only up to  $\beta = 0.95$ . This allows a discrimination for hadrons and electrons/positrons [Z<sup>+</sup>99].

### 2.2.2 The META Detector

The **M**ultiplicity and **E**lectron **T**rigger **A**rray (META) is located behind the drift chambers as seen from the target and consists of **t**ime-**o**f-flight scintillator bars (TOF), the **R**esistive **P**late **C**hamber (RPC) wall and electromagnetic PreShower detector. The main tasks of the META detector are:

- Providing charged particle multiplicity for the first level trigger.
- Measuring the time-of-flight and energy loss which is used for particle identification.
- Reconstructing the position of charged particles which are correlated with reconstructed tracks and can be spatially correlated with the RICH rings and META hits.

#### The TOF Wall

The TOF wall covers the large polar angles between  $44^\circ$  and  $88^\circ$ . Each of the six sectors contains 8 modules with a set of 8 scintillator rods, respectively. Charged particles traversing the detector may interact with the scintillating material that then absorbs part of the particles' energy. The energy is reemitted in the form of photons which are then collected by photo-multiplier tubes (PMTs) from both sides of a scintillator rod. Electric signals are generated in the PMTs allowing measurements of the arrival time and the amplitude of the light signal. Let  $a_{left}$  and  $a_{right}$  denote the the light signal's amplitudes at each end of a scintillator rod. Furthermore, let  $t_{left}$  and  $t_{right}$  denote the difference between the time given by the START detector and the arrival time of the light signals in the TOF rod. With these measurements it is possible to extract the particle position  $x$  along the rod, its time-of-flight  $t_{tof}$  and deposited energy  $\Delta E$ :

$$x = \frac{1}{2} (t_{left} - t_{right}) \cdot V_g \quad (2.1)$$

$$t_{tof} = \frac{1}{2} \left( t_{left} + t_{right} - \frac{L}{V_g} \right) \quad (2.2)$$

$$\Delta E \propto \sqrt{a_{left} \cdot a_{right} \cdot e^{\frac{L}{\lambda_{at}}}} \quad (2.3)$$

where  $V_g$  is the group velocity of light in the scintillator rod,  $\lambda_{at}$  its attenuation length and  $L$  the length of the rod.

Signals from the PMTs are processed by a constant fraction discriminator (CFD). By performing an analog sum of the CFD signals it is possible to trigger on the charged-particle multiplicity. [A<sup>+</sup>02]

#### The RPC wall

A Resistive Plate Counter is a gas detector that consists of two parallel electrode plates. A electric field is applied between the plates. The gap between the



plates is filled with a counter gas that provides electrons by ionization and a quencher gas that absorbs photons. Charged particles trigger an electron avalanche by the same process as in drift chambers described in chapter 2.3.2. This causes a discharge. Due to the high resistivity of the electrodes and the photon absorbing quencher gas the discharge can't spread through the whole gas. Thus, the electric field is switched off in a limited area where the discharge occurred [SC81].

In multi-gap RPCs the gas gap is divided by additional anode-cathode plates. This combines the good time resolution of a narrow gap with the higher rate capabilities and lower power dissipation of a wide gap [Z<sup>+</sup>96].

The HADES detector utilizes a multi-gap RPC wall [FSW00] that covers the small polar angles between 18° and 44°. In 2009, the RPC wall has replaced the TOFINO detector for time-of-flight measurements since the multiplicity in systems like Au+Au called for a detector with higher granularity [B<sup>+</sup>09].

### The PreShower Detector

The momenta of the hadrons and the hit densities in the detectors are higher at low polar angles making the discrimination of leptons and hadrons more difficult. The PreShower detector is located behind the RPC wall. The positrons and electrons tracked with HADES loose energy mainly due to radiation when passing through material. If the emitted photons have an energy of at least  $2m_e = 1022 \text{ keV}/c^2$  than  $e^+/e^-$  pairs can be generated which again loose energy due to radiation. The result is a cascading electromagnetic shower. Radiation loss plays little role for hadrons in the SIS18 energy regime. Therefore the PreShower detector offers another method to distinguish hadrons from positrons and electrons in addition to the RICH detector [B<sup>+</sup>04].

## 2.3 The Tracking System

A magnetic field deflects charged particles, so that their momentum can be reconstructed from the curvature of their trajectory. The momentum transfer  $\Delta\mathbf{p}_t$  transverse to the particle's trajectory only depends on the magnetic field  $\mathbf{B}$  it traverses:

$$\Delta\mathbf{p}_t = \int d\mathbf{p} = \int \mathbf{F}_L dt = \int q \cdot (\mathbf{v} \times \mathbf{B}) dt = q \cdot \int d\mathbf{s} \times \mathbf{B}. \quad (2.4)$$

The momentum and momentum transfer vectors are added. The resulting deflection angle is larger for particles with low momenta than for particles with high momenta. In order to calculate the momentum transfer, information about the magnetic field and the particles' position and direction in front of and behind the magnet is needed.

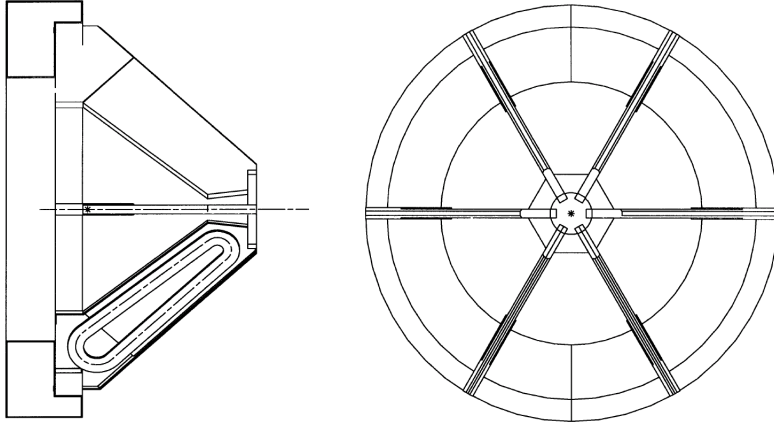


Figure 2.3: Left: Side view of the magnet. The lower half shows a coil inside its chamber. Right: Back view of the magnet. The coil chambers are located at the sector borders [Tar10].

### 2.3.1 The Magnet

The purpose of the magnet (fig. 2.3) is to provide a sufficient momentum transfer to allow momentum reconstruction with a resolution of  $\sigma_p/p = 1.5\text{--}2\%$  for electrons. Particles with momenta between  $0.1\text{ GeV}/c$  and  $2\text{ GeV}/c$  should be accepted over a large polar angle range of  $18^\circ\text{--}85^\circ$  with nearly full azimuthal coverage. This calls for a compact design of the magnetic coils.

The detector has been designed to keep the detector material mostly outside of the magnetic field thereby minimizing the effects of multiple scattering and achieving a better momentum resolution. A weak deflection of positrons and electrons in the radiator gas of the RICH ensures that the emitted photons are projected onto rings. Additionally, the time resolution of the photo multipliers used in the META detector would deteriorate in a magnetic field. A high magnetic field can also lead to problems inside the MDCs if the drifting electrons and ions are moving on highly curved trajectories. Therefore, the magnetic field strength needs to be high between the outer and inner MDCs to deflect the particles and be kept low in the detector regions.

The HADES magnet consists of six superconducting coils aligned symmetrically around the beam axis. Each coil is separately contained in its individual coil case. The compact construction leads to little loss in geometrical acceptance. Figure 2.4 shows the highly inhomogeneous magnetic field at azimuthal angles of  $\Phi = 0^\circ$  (the plane at the sector border dividing the coils) and  $30^\circ$  (the mid plane of a sector). The maximum value is seen at the centre of a sector ( $\Phi = 30^\circ$ ) and small polar angles. The magnet with its toroidal field geometry has been designed so that particles are mainly deflected in polar directions [M<sup>+</sup>04].

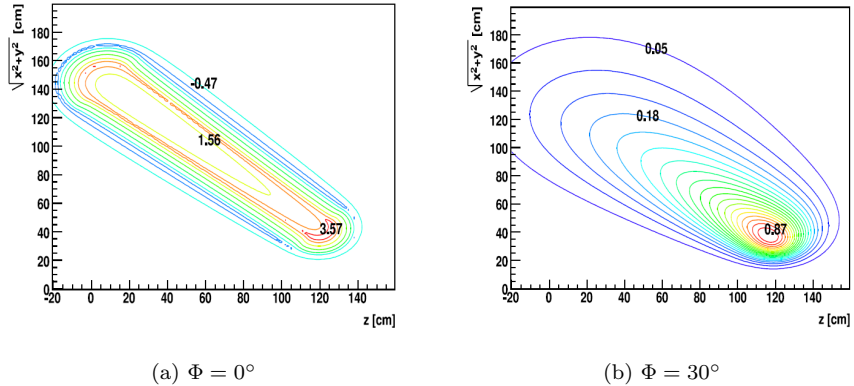


Figure 2.4: The magnetic field along the beam axis ( $z$ -coordinate) and perpendicular to the beam axis at the border of a sector ( $\Phi = 0^\circ$ ) and middle of a sector ( $\Phi = 30^\circ$ ) [HAD09].

### 2.3.2 The Multi-Wire Drift Chambers

The position of the particles are tracked via four planes of multi-wire drift chambers (MDCs I-IV) in each sector. Two MDC planes (MDC planes I and II) are placed in front of and two behind (planes III and MDC IV) the magnetic field in order to determine the direction of the particle at these points (fig. 2.5). Each plane is made of six trapezoidal layers. The signal and field wires in each layer are oriented in different angles ( $\pm 0^\circ, \pm 20^\circ, \pm 40^\circ$  as can be seen in figure 2.6a) to cope with left-and-right ambiguities in the high-multiplicity environment of a heavy-ion collision.

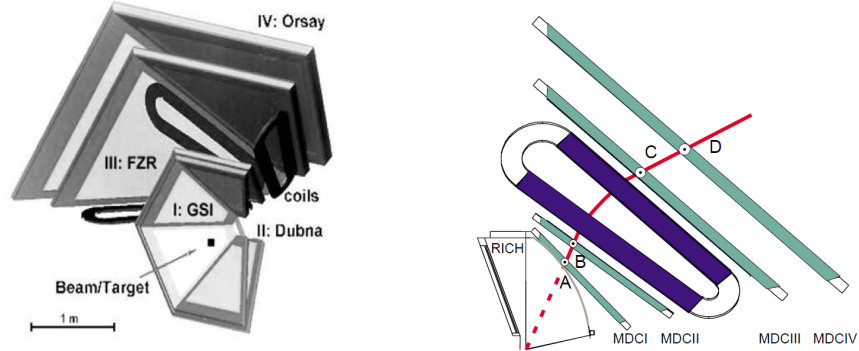
The wires inside the chambers form elongated drift cells. Such a drift cell is comprised of several cathode and field wires plus a sense wire in its centre (fig. 2.6b). Each layer consists of about 1100 such cells. Their size varies with each layer to keep the granularity constant. In the innermost chamber (MDC I), the size of the cells is  $5 \times 5 \text{ mm}^2$  while in MDC IV it is  $14 \times 10 \text{ mm}^2$ .

A drift chamber has a spatial resolution between 70 and  $110 \mu\text{m}$ . It is higher in polar direction which is also the main direction of the momentum transfer [Mar05].

#### Drift Chamber Physics

A drift chamber is a gas detector with an electric field present. When charged particles enter an MDC there are three important processes taking place:

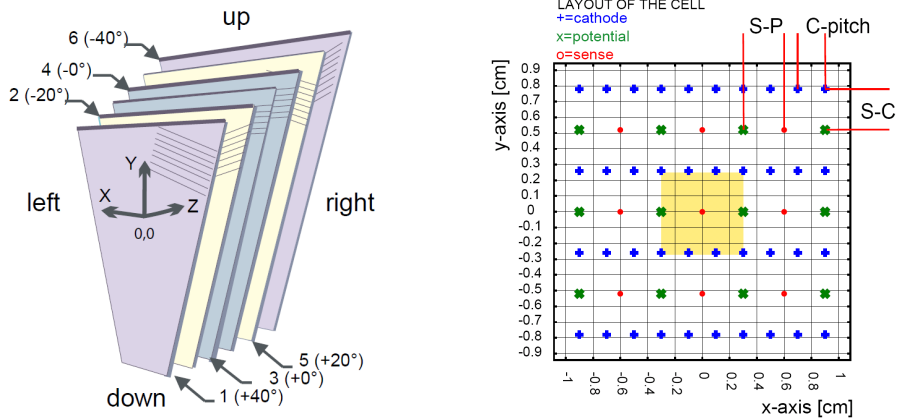
1. Ionization of the gas molecules.
2. Drift of the ions and electrons in the electric field.



(a) Schematic view of the position of the four MDC planes with respect to the magnetic coils. The names of the institutes where the chambers were fabricated are indicated [HAD09].

(b) Knowing the positions in front of (points A and B) and behind the magnetic field (points C and D) the deflection angle and momentum transfer of the particle due to the magnet field can be determined [C.L00].

Figure 2.5: The HADES tracking system.



(a) The different wire orientations in the MDC layers allow for the reconstruction of space points from fired wire clusters (see chapter 3). The wires of the two  $\pm 0^\circ$  layers are shifted by half of a drift cell width [Mar05]. The stereo angles were chosen for maximum spatial resolution in the direction of the particle deflection [HAD09].

(b) Shown is the layout of a drift cell marked by the yellow square in MDC II. The cathode (blue) and field wires (green) form the borders. A sense wire (red dot) is located in the centre of the drift cell [Mar05].

Figure 2.6: Wire orientations (a) of an MDC plane and layout of the geometry of a drift cell (b).

### 3. Gas amplification (electron avalanche) near the sense wires.

Charged particles passing through a gas may interact electromagnetically with the gas molecules and loose energy during this process. The important form of energy loss for drift chambers is by ionization where the particle interacts with the electrons of the gas molecules and transfers enough of its kinetic energy so that one or more electrons are no longer bound to the gas molecules. This is called primary ionization. If these electrons have sufficient energy they can ionize more molecules (secondary ionization). The ionization is statistically distributed along the particle's path resulting in clusters of electron and ion clouds.

The cathode and field wires enclose the sense wire in an electric field. The sense wire is on a positive potential. The electrons are accelerated along the electric field lines towards the sense wire in the drift cell while the ions drift towards the field and cathode wires. Since an electron's mass is much smaller and its mean free path length larger compared to an ion the electrons have a higher velocity. Due to collisions with gas molecules the electrons are decelerated. The drift velocity is therefore constantly changing. However, on a macroscopic scale a mean drift velocity can be observed.

The electric field is strongest near the sense wire where the amplification sets in. The number of free electrons  $N$  grows exponentially in this avalanche by

$$N = N_0 \cdot e^{\alpha x} \tag{2.5}$$

where  $N_0$  is the amount of electrons from the primary ionization,  $\alpha$  is the Townsend coefficient and  $x$  is the distance the avalanche has progressed. The Townsend coefficient is the inverse of the mean free path length  $\lambda$  of an electron until it ionizes a gas molecule:

$$\alpha \equiv \frac{1}{\lambda}. \tag{2.6}$$

Its value depends on the strength of the electric field and the utilized gases.

The electron clouds generate a signal on the sense wire which starts a time measurement. The stop signal is deduced from an external detector e.g. a delayed trigger of the START detector. Since this signal is the same for all drift cells it is also called a common stop signal. An 8 channel **A**mplifier **S**haper **D**iscriminator (ASD8) processes the measurements of eight drift cells. The ASD8 first amplifies the signal. The slower ion clouds induce a signal on the cathode and field wires that also contribute to the measurement. Since the ions are much slower than the electrons this results in a long tail in the signal. The shaper eliminates these ion tails. Finally, the discriminator generates a nearly rectangular logical signal of a width equivalent to the time that the signal exceeded a set threshold voltage and transfers this signal to a **T**ime to **D**igital **C**onverter (TDC). The TDC then measures the time between the logical signal of the ASD8 and the common stop signal.

The choice of a gas mixture for the MDC's is a non-trivial task. The gas mixture must allow a stable operation of the MDC's with a high efficiency, provide a good time resolution, have a large radiation length to reduce multiple scattering and minimize ageing of the chambers.

If the free electrons recombine with ions or attach to gas molecules then they are lost before they can be amplified. Noble gases like Argon and Helium are chemically inert and have a low electronegativity. Electron capture can thus be avoided with these gases. Argon is superior to Helium regarding primary ionization while Helium with its lower mass and higher radiation length doesn't cause as much multiple scattering. The concentration of highly electronegative gases like oxygen or water vapour needs to be kept low because oxygen can easily capture electrons in low electrical fields and water vapour in high electrical fields. Water vapour also influences the drift velocity of the electrons which negatively affects the position resolution of the drift chamber.

Photons produced by recombination processes can cause ionization and avalanches further away from the original particle's trajectory. To avoid such a delocalization of the signal a quencher gas is added that absorbs the photons. Typical quenchers are organic compounds like Methane  $\text{CH}_4$ , Ethane  $\text{C}_2\text{H}_6$  or iso-Butane  $\text{C}_4\text{H}_{10}$ . These gases transform the energy of the absorbed photons only in rotation and vibration energy. While they reduce the gas amplification they are needed for a stable operation of the drift chamber.

MDC plane I is currently filled with a 70 : 30 mixture of Argon and carbon dioxide. The other planes contain 84% Argon and 16% iso-Butane.

## 2.4 The Forward Wall Hodoscope

The forward wall consists of 287 scintillator cells connected to photo-multipliers. During the deuteron+proton beam experiment in May 2007, the forward wall has been successfully used to register spectator protons which allowed for a discrimination between the p+p and the n+p collisions. The forward wall was located 7 metres downstream behind the target and covered polar angles between  $0.33^\circ$  and  $7.17^\circ$  [L<sup>+</sup>09].

## Chapter 3

# Track Reconstruction in HADES

Track reconstruction in HADES is done in the following steps [HAD09]:

**Track candidate search:** On its own, a measurement of a single wire in a drift cell does not yield a position of track. Only when using wire measurements of several layers with different wire orientations information about track position and direction can be inferred. The wires that generated a signal are projected onto a common plane. For the inner chambers, the plane is chosen so that the drift cells projected on this plane are roughly of equal size. In case of the outer chambers with coplanar geometry, the centre plane of two outer chambers serves as the projection plane. The fired wires projected onto the planes may cross on the plane forming a local maximum as shown in fig. 3.1. If enough wires contributed to the maximum then it is accepted as a wire cluster. High thresholds for the required number of wires in a maximum reduce the efficiency of the algorithm finding the wire clusters. On the other hand, low thresholds result in more fake clusters due to randomly crossing wires. The threshold is computed dynamically for each event depending on the amount of fired wires.

The target position and the location of the maximum in the inner projection plane then define a straight track segment in space. The deflection of the particle in the magnetic field is approximated by a momentum kick on a virtual plane. The intersection of the inner track segment with the kick plane and a wire cluster on the outer projection plane form another track segment, the outer track segment. This matching of inner and outer segments depicted in fig 3.2 define a track candidate.

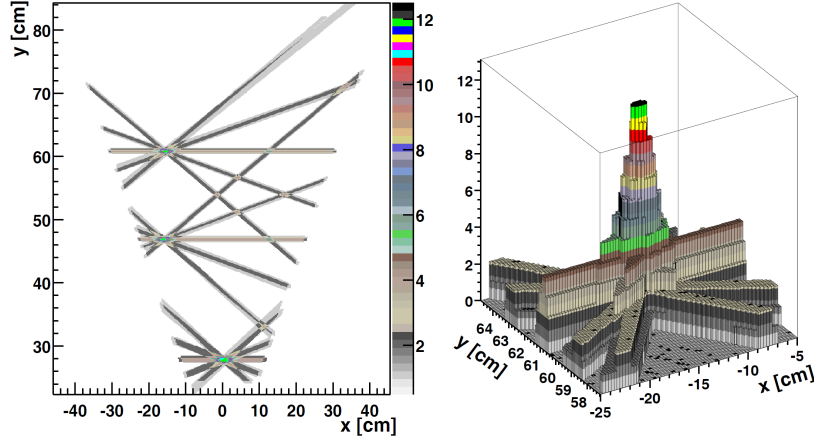


Figure 3.1: Left: The fired wires are projected onto a common plane and form local maxima at the intersection points. Right: A cluster is defined by a sufficient amount of crossing wires [HAD09].

The spatial resolution of the candidate search for the inner segment is  $\sigma_x = 0.5 \text{ mm}$  and  $\sigma_y = 0.16 \text{ mm}$ . In case of the outer segment it is  $\sigma_x = 1.6 \text{ mm}$  and  $\sigma_y = 0.46 \text{ mm}$  [Mar11a].

**Segment fit:** The precision of the candidates' position is further improved by the segment fit. This requires converting the measured drift times into a distance to the sense wires in a drift cell (also see chapter 6.3.2). The space coordinates of the track are then fitted to a track model. The segment fitter computes four space points each located on a virtual plane in the middle of a module. The segment fit improves the position resolution to  $0.2 \text{ mm}$  in x-direction and  $0.1 \text{ mm}$  in y-direction.

**Particle identification:** Inner track segments are matched with rings in the RICH. Outer track segments are matched with the META detectors. Matched rings in the RICH detector are used to identify leptons.

**Momentum estimation using a spline:** The four points computed by the segment fit are interpolated with a cubic spline. The goal of a spline is to calculate continuous interpolation functions over intervals. The intervals are defined by a set of supporting points. Higher order splines additionally require continuous derivatives of higher order. For example, cubic splines additionally require continuous first and second order derivatives, both within the intervals and at its borders. In the HADES reconstruction software, a fifth order spline is used. This method provides a first estimate of the particle momentum using the matched inner and outer track segments. The relative momentum resolution  $\sigma_{p/p}$  of the method obtained



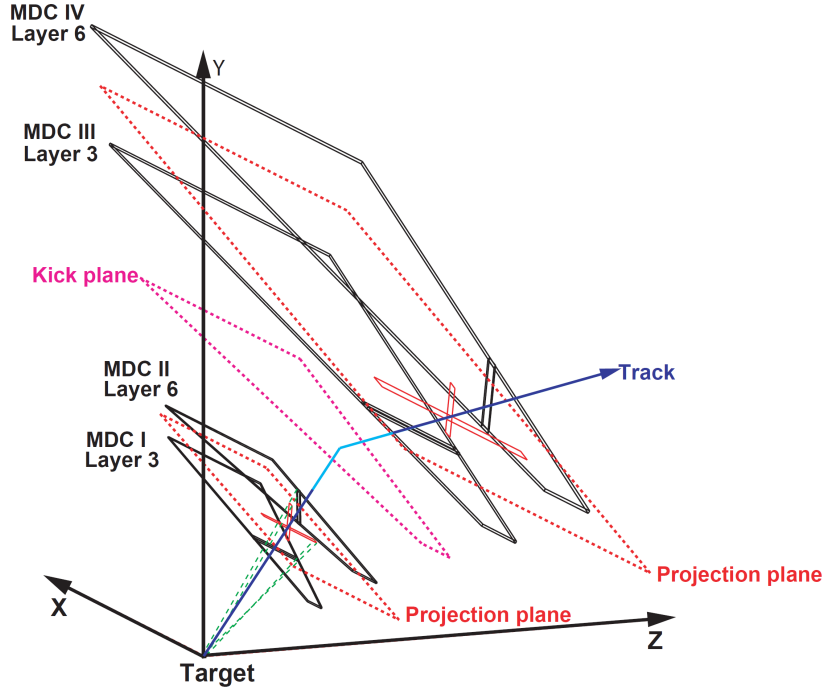


Figure 3.2: Principle of the track candidate search finding an inner and outer track segment. Only one MDC layer per module is shown [HAD09].

from simulations ranges from 1.5% to 4.5% for  $0.15 \text{ GeV}/c$  electrons in the  $\Theta$  region of  $20^\circ$  to  $80^\circ$  and decreases to values ranging from 1% to 2.8% for  $1.4 \text{ GeV}/c$  electrons in the given angular region [HAD09].

**Momentum reconstruction via a Runge-Kutta method:** This step iteratively optimizes the position and direction estimates from the segment fitter and the momentum estimation from the spline method. A fourth order Runge-Kutta scheme is used to numerically solve the equation of motion of a charged particle in the inhomogeneous magnetic field.

**Particle identification:** Time-of-flight measurements in the TOF/RPC and energy loss in the MDCs supplement particle identification. Leptons are identified by rings in the RICH and an electromagnetic shower in the Pre-Shower detector.

## Chapter 4

# State Estimation with the Kalman Filter

### 4.1 Definitions

Many processes can be described by state-space systems. Given a mathematical model of the process and the internal state of the system at a time it is possible to predict how the system will evolve over time. A continuous-time, deterministic linear system can be defined by the following two equations:

$$\dot{\mathbf{x}}(t) = F\mathbf{x}(t) + G\mathbf{u}(t) + \mathbf{w}(t) \quad (4.1)$$

$$\mathbf{m}(t) = H\mathbf{x}(t) + \mathbf{v}(t). \quad (4.2)$$

The system equation 4.1 specifies the evolution of the state vector  $\mathbf{x}(t)$  over time  $t$ . The state vector is a collection of  $s$  state variables  $x_1, \dots, x_s$  that fully describes the internal state of the system at a time  $t$ . The  $s$ -dimensional domain of the state vector is called the state space of the system. Since the state is a function of time it can be referred to as a trajectory. The known vector  $\mathbf{u}$  is called the control or input vector, the matrix  $F$  is the system or propagator matrix and  $G$  the input matrix. The system may be subject to random process noise  $\mathbf{w}$ .

While the full state of the system is usually hidden from observers one gets measurable output  $\mathbf{m}$ . The output matrix  $H$  is also called the projector matrix since it projects the unknown state variables  $\mathbf{x}$  onto a measurement vector. The measurement equation 4.2 describes how a measurement is extracted from the state vector. The process noise  $\mathbf{w}$  and measurement noise  $\mathbf{v}$  are known statistically. Let  $r$  be the dimension of the measurement vector and  $s$  the dimension of the state vector then  $H$  must be an  $r \times s$ -matrix.

For example when tracking a charged particle in a magnetic field then the position and momentum of the particle would fully describe the state of the particle. The magnetic field that the particle traverses is the controllable input.

Detectors give position information at different times. With this knowledge it is possible to estimate the particle momentum that was hidden from direct observation.

Often, only the knowledge of the system state at a discrete set of times is relevant. Equations 4.1 and 4.2 for a continuous-time system can be transformed into a discrete-time system:

$$\mathbf{x}_k = F_{k-1}\mathbf{x}_{k-1} + G_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \quad (4.3)$$

$$\mathbf{m}_k = H_k\mathbf{x}_k + \mathbf{v}_k \quad (4.4)$$

where  $k$  is the index of the time.

The motion of a charged particle in a magnetic field is a nonlinear process like most real processes. Nonlinear systems can be approximated as linear systems taking advantage of the well-understood mathematics for linear systems.

Only the variances  $\sigma^2$  of the process noise  $\mathbf{w}$  and the measurement noise  $\mathbf{v}$  are known. The covariance of two random scalar variables  $x_1$  and  $x_2$  with mean values  $\bar{x}_1$  and  $\bar{x}_2$  is defined by the expectation value

$$\begin{aligned} C_{x_1x_2} &\equiv \sigma_{x_1x_2}^2 \\ &\equiv E[(x_1 - \bar{x}_1)(x_2 - \bar{x}_2)]. \end{aligned} \quad (4.5)$$

In case of a vector  $\mathbf{x} = (x_1, \dots, x_s)^T$  of random variables the covariance matrix is

$$C \equiv E[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T]. \quad (4.6)$$

Since  $C_{x_1x_2} = C_{x_2x_1}$  a covariance matrix is symmetric. Additionally it is positive semi-definite since for any  $s$ -dimensional column vector  $\mathbf{z}$  we have

$$\begin{aligned} \mathbf{z}^T C \mathbf{z} &= \mathbf{z}^T E[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T] \mathbf{z} \\ &= E[\mathbf{z}^T (\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{z}] \\ &= E[(\mathbf{z}^T (\mathbf{x} - \bar{\mathbf{x}}))^2] \\ &\geq 0 \end{aligned} \quad (4.7)$$

The diagonal elements of the covariance matrix are the variances  $\sigma_{x_i}^2$  of the random variable  $x_i$ . The off-diagonal elements describe correlations between the components of the vector  $\mathbf{x}$ . The correlation of two random variables  $x_1$  and  $x_2$  is

$$R_{x_1x_2} = E(x_1x_2). \quad (4.8)$$

The two variables are called uncorrelated if  $R_{x_1x_2} = E(x_1)E(x_2)$ , i.e. the element  $C_{x_1x_2}$  of the covariance matrix is zero. A random variable may change with an independent variable  $t$  which could be time, for example. Noise is called

white if  $x(t_1)$  and  $x(t_2)$  are uncorrelated for all  $t_1 \neq t_2$ . Otherwise it is called coloured noise.

The Kalman filter is optimal for Gaussian, zero-mean, uncorrelated and white process noise  $\mathbf{w}$  and measurement noise  $\mathbf{v}$ . The Kalman filter may still be used or modified for different kind of noise [Sim06, chapters 5.2 and 7]. The covariance matrices are named  $Q$  for the process noise and  $V$  for the measurement noise:

$$Q_k \equiv \mathbf{w}_k \mathbf{w}_k^T \quad (4.9)$$

$$V_k \equiv \mathbf{v}_k \mathbf{v}_k^T \quad (4.10)$$

## 4.2 The Kalman Filter

The Kalman filter in its original form [Kal60] was proposed for linear systems. For its first application in the NASA Apollo program [MS85] the Kalman filter was adapted for nonlinear systems. The principle of the Kalman filter is outlined in figure 4.1. An estimate for the initial state vector  $\mathbf{x}_0$  needs to be made. The initial covariance matrix  $C_0$  represents the estimation error of  $\mathbf{x}_0$ . The covariance matrix should be filled with appropriate values. Assuming too small errors will not give the Kalman filter enough freedom to update the state vectors while too large errors can lead to numerical instability and a divergence of the filter.

The core of the Kalman filter is the interplay between the prediction and filter steps. The prediction propagates the track state and covariance matrix according to the system equation into the future. The prediction is then corrected in the filter step by processing the real measurement. This process is iteratively repeated for all measurements  $k = \{1 \dots N\}$ . The smoother goes back through all filtered states and further improves them by using all the information collected in the previous prediction and filter phases.

Thus, the prediction makes an estimate of the future state of the system. The filter step estimates the system state in the present while the smoother allows to estimate the state of the system in the past.

### 4.2.1 The Linear Kalman Filter

A linear discrete-time system is given by equations 4.3 and 4.4. The system equation 4.3 is used to propagate the estimate of the mean state vector  $\mathbf{x}_{k-1}$  from time  $k-1$  to time  $k$ . The expected state vector  $\mathbf{x}_k^{k-1}$  at time  $k$  then is

$$\mathbf{x}_k^{k-1} = F_{k-1} \mathbf{x}_{k-1} + G_{k-1} \mathbf{u}_{k-1}. \quad (4.11)$$

The subindex denotes that  $\mathbf{x}_k^{k-1}$  is the state estimate at time  $k$ . The superindex denotes that  $k-1$  measurements have been processed so far. For simplicity  $\mathbf{x}_k^k \equiv \mathbf{x}_k$  is written. Unlike 4.3 there is no process noise in this equation since

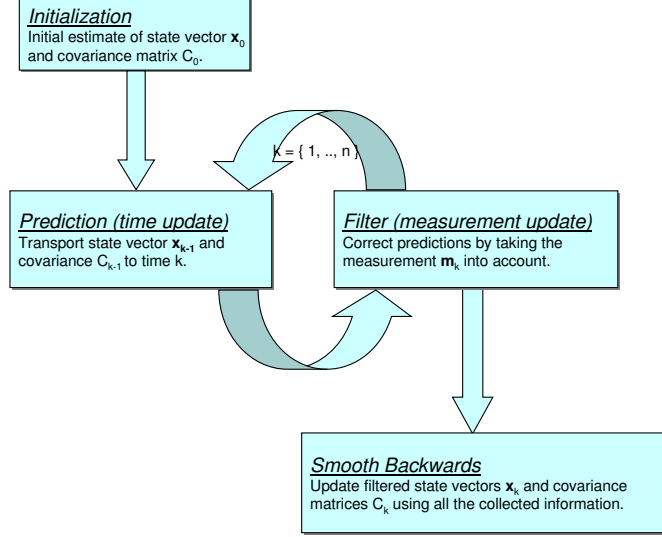


Figure 4.1: Starting with initial values the Kalman filter propagates the state vector  $\mathbf{x}$  through time and then improves the prediction by processing the measurement in the filter step. This is repeated for all measurements. The smoother goes back through all filtered states and further improves them by using all the collected information.

only mean values of the state are propagated. This transports the state vector from time  $k - 1$  to  $k$ . The covariance matrix at time  $k$  would be

$$C_k^{k-1} = E \left[ (\tilde{\mathbf{x}}_k - \mathbf{x}_k^{k-1}) (\tilde{\mathbf{x}}_k - \mathbf{x}_k^{k-1})^T \right].$$

The true state  $\tilde{\mathbf{x}}_k$  is propagated with the system equation 4.3:

$$C_k^{k-1} = E \left[ (F_{k-1} \tilde{\mathbf{x}}_{k-1} + G_{k-1} \mathbf{u}_{k-1} + \mathbf{w}_{k-1} - \mathbf{x}_k^{k-1}) (\dots)^T \right].$$

Substituting the expected state  $\mathbf{x}_k^{k-1}$  with 4.11 yields

$$\begin{aligned} C_k^{k-1} &= E \left[ (F_{k-1} (\tilde{\mathbf{x}}_{k-1} - \mathbf{x}_{k-1}) + \mathbf{w}_{k-1}) (\dots)^T \right] \\ &= E \left[ F_{k-1} (\tilde{\mathbf{x}}_{k-1} - \mathbf{x}_{k-1}) (\tilde{\mathbf{x}}_{k-1} - \mathbf{x}_{k-1})^T F_{k-1}^T + \mathbf{w}_{k-1} \mathbf{w}_{k-1}^T + \right. \\ &\quad \left. F_{k-1} (\tilde{\mathbf{x}}_{k-1} - \mathbf{x}_{k-1}) \mathbf{w}_{k-1}^T + \mathbf{w}_{k-1} (\tilde{\mathbf{x}}_{k-1} - \mathbf{x}_{k-1}) F_{k-1}^T \right]. \end{aligned} \quad (4.12)$$

The expectation value of  $(\tilde{\mathbf{x}}_{k-1} - \mathbf{x}_{k-1}) (\dots)^T$  is the covariance at the previous time  $k - 1$ . Since the process noise  $\mathbf{w}_{k-1}$  is uncorrelated with  $(\tilde{\mathbf{x}}_{k-1} - \mathbf{x}_{k-1})$

and zero-mean the resulting covariance is:

$$C_k^{k-1} = F_{k-1} C_{k-1} F_{k-1}^T + Q_{k-1}. \quad (4.13)$$

The two equations 4.11 and 4.13 are also called the time-update equations. The update of the state in the filter step is done with

$$\mathbf{x}_k = \mathbf{x}_k^{k-1} + K_k (\mathbf{m}_k - H_k \mathbf{x}_k^{k-1}). \quad (4.14)$$

The quantity  $(\mathbf{m}_k - H_k \mathbf{x}_k^{k-1})$  is called the innovation or correction term. It is the residual of the real measurement  $\mathbf{m}_k$  and the expected measurement  $H_k \mathbf{x}_k^{k-1}$  which is obtained from the measurement equation 4.4. The matrix  $K_k$  is called the Kalman gain matrix and represents a measurement by how much the innovation improves the expected state vector. If either the innovation or the gain matrix are zero then the measurement has no effect on the predicted state.

The gain matrix is chosen so that it is optimal in the sense that the sum  $J_k$  of the variances of the estimation error is minimized:

$$\begin{aligned} J_k &= E \left[ (\tilde{x}_1 - x_{k,1})^2 \right] + \dots + E \left[ (\tilde{x}_s - x_{k,s})^2 \right] \\ &= E \left[ \|\tilde{\mathbf{x}} - \mathbf{x}_k\|_2^2 \right] \\ &= E \left[ (\tilde{\mathbf{x}} - \mathbf{x}_k)^T (\tilde{\mathbf{x}} - \mathbf{x}_k) \right] \\ &= E \left[ \text{Tr} (\tilde{\mathbf{x}} - \mathbf{x}_k) \right] \\ &= \text{Tr} C_k \end{aligned}$$

The covariance matrix  $C_k$  of the filtered state can be derived by first using equation 4.14 in the estimation error  $\tilde{\mathbf{x}} - \mathbf{x}_k$  and then calculate its expectation value. Together with the definition 4.6 of a covariance a formula for  $C_k$  can be obtained. Finally, solving  $\partial J_k / \partial K_k$  yields the solution of the optimal gain matrix  $K_k$ . Detailed derivations can be found in [Sim06, chapters 3 and 5] and [Fuj]. One formulation for the covariance  $C_k$  and Kalman gain  $K_k$  is:

$$K_k = C_k^{k-1} H_k^T (H_k C_k^{k-1} H_k^T + V_k)^{-1} \quad (4.15a)$$

$$C_k = (I - K_k H_k) C_k^{k-1} \quad (4.15b)$$

As the estimation error from the prediction  $C_k^{k-1}$  and thereby also the gain matrix  $K_k$  approaches zero the Kalman filter will weight the predicted state  $\mathbf{x}_k^{k-1}$  more heavily than the innovation. On the other hand the filter will weight the innovation more heavily if the measurement error covariance  $V_k$  approaches zero since the Kalman gain then becomes

$$\lim_{V \rightarrow 0} K_k \rightarrow H_k^{-1}.$$

## Smoothing

Due to the recursive nature of the Kalman filter approach the computed state vector  $\mathbf{x}_k$  is based on the  $k$  measurements collected so far. It is unaffected by subsequent estimates. A smoother allows to further improve this estimate using information from any subsequent measurements as well. Described here is the smoother by **Rauch Tung and Striebel** (RTS smoother) [RTS65]. This algorithm is a fixed-interval type smoother. It is useful when a fixed amount of measurements  $\mathbf{m}_1, \dots, \mathbf{m}_n$  is available and an estimate is desired for all times  $k \in \{1, \dots, n\}$ .

First the Kalman filter is run on all  $n$  measurements. The final estimate  $x_n$  from the Kalman filter coincides with smoothing. Then the estimates with  $k < n$  are smoothed in the reverse order using the following equations:

$$\mathbf{x}_k^n = \mathbf{x}_k + A_k (\mathbf{x}_{k+1}^n - \mathbf{x}_{k+1}^k) \quad (4.16a)$$

$$A_k = C_k F_k^T (C_{k+1}^k)^{-1} \quad (4.16b)$$

$$C_k^n = C_k + A_k (C_{k+1}^n - C_{k+1}^k) A_k^T. \quad (4.16c)$$

The smoothed state  $\mathbf{x}_{k+1}^n$  and its covariance  $C_{k+1}^n$  at time  $k+1$  are known from the previous iteration. The propagator matrix  $F_k$  and state  $\mathbf{x}_{k+1}^k$  with the respective covariance  $C_{k+1}^k$  are known from the prediction from  $k$  to the subsequent time point  $k+1$ . The state to improve is the estimate  $\mathbf{x}_k$  of the Kalman filter at time  $k$ .

The derivation of the RTS smoother can be found in [Fuj] and [Sim06, chapter 9.4.2.].

## Inverse Kalman Filter

After the Kalman filter and smoother is done it is possible to exclude a measurement in retrospect. The inverse Kalman filter allows to efficiently remove the measurement  $\mathbf{m}_k$  from the smoothed estimate  $\mathbf{x}_k^n$ . This can be used to find outliers or to check detector alignment [Frü87].

$$\mathbf{x}_k^{n*} = \mathbf{x}_k^n + K_k^{n*} (\mathbf{m}_k - H_k \mathbf{x}_k^n) \quad (4.17a)$$

$$K_k^{n*} = C_k^n H_k^T (-V_k + H_k C_k^n H_k^T)^{-1} \quad (4.17b)$$

$$\begin{aligned} C_k^{n*} &= (I - K_k^{n*} H_k) C_k^n \\ &= \left[ (C_k^n)^{-1} - H_k^T (V_k)^{-1} H_k \right]^{-1}. \end{aligned} \quad (4.17c)$$

## 4.2.2 The Extended Kalman Filter

The Kalman filter described in 4.2.1 is optimal only for linear systems. The extended Kalman filter is a modification for estimation of nonlinear systems. It is no longer optimal, but provides good results if the system is approximately

linear over short ranges. In general a nonlinear system can be described by the following system and measurement equations:

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_k \quad (4.18)$$

$$\mathbf{m}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k \quad (4.19)$$

To derive the extended Kalman filter the above equations are linearized around a nominal trajectory by a Taylor expansion. The Jacobians of  $\mathbf{f}_k$  and  $\mathbf{h}_k$  are used as the propagator matrix  $F_k$  and projection matrix  $H_k$ . The result is called the linearized Kalman filter. To keep the linearization error small the nominal trajectory has to be close to the real trajectory. For example, when estimating the flight of an aircraft the nominal trajectory could be the planned flight path. When tracking charged particles through a magnetic field a different solution has to be found: The idea of the extended Kalman filter is to use the state estimate as the nominal trajectory. The equations for the prediction step then are [Sim06, chapter 13]

$$\mathbf{x}_k^{k-1} = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \quad (4.20a)$$

$$C_k^{k-1} = F_{k-1} C_{k-1} F_{k-1}^T + Q_{k-1} \quad (4.20b)$$

$$F_{k-1} = \left. \frac{\partial \mathbf{f}_{k-1}}{\partial \mathbf{x}} \right|_{\mathbf{x}_{k-1}}$$

and for filter step

$$\mathbf{x}_k = \mathbf{x}_k^{k-1} + K_k (\mathbf{m}_k - \mathbf{h}_k(\mathbf{x}_k^{k-1})) \quad (4.21a)$$

$$K_k = C_k^{k-1} H_k^T (H_k C_k^{k-1} H_k^T + V_k)^{-1} \quad (4.21b)$$

$$C_k = (I - K_k H_k) C_k^{k-1} \quad (4.21c)$$

$$H_k = \left. \frac{\partial \mathbf{h}_k}{\partial \mathbf{x}} \right|_{\mathbf{x}_k}$$

### 4.2.3 Goodness of the Estimate

The Kalman filter has its origins in least squares methods. The least squares method is an approach to find an approximate solution of an overdetermined or inexact system of equations. One application is to estimate a set of parameters for a model that fit a given amount of data points. This is accomplished by minimizing the sum of the squared residuals of the observed and expected values. To reflect that some of the data points may be more reliable than others they may be weighted by their inverse covariances. If the measurement errors are independent and normally distributed.

In case of the Kalman filter there are two residuals to consider: The first one is the residual between the expected state vector  $\mathbf{x}_k^{k-1}$  after propagation from time  $k-1$  to  $k$  and the state vector  $\mathbf{x}_k$  after filtering. The second one is the residual of the actual measurement  $\mathbf{m}_k$  and the measurement expected



from the filtered estimate  $\mathbf{h}_k(\mathbf{x}_k)$ . The  $\chi_k^2$  after processing the measurement at time  $k$  then is [Fu]j

$$\chi_k^2 = \chi_{k-1}^2 + (\mathbf{x}_k - \mathbf{x}_k^{k-1})^T (C_k^{k-1})^{-1} (\mathbf{x}_k - \mathbf{x}_k^{k-1}) + (\mathbf{m}_k - \mathbf{h}_k(\mathbf{x}_k))^T (V_k)^{-1} (\mathbf{m}_k - \mathbf{h}_k(\mathbf{x}_k)). \quad (4.22)$$

In statistics the number degrees of freedom is the number of independent observations minus the number of parameters that are estimated [Ore82]. If there are  $N$  measurements where each measurement vector has the dimension  $r$  and  $s$  state parameters to estimate then the number degrees of freedoms  $\nu$  is

$$\nu = N \cdot r - s. \quad (4.23)$$

The probability density function of the sum of squared residuals with  $\nu$  degrees of freedom is [BSGM06]

$$f(\nu, t) = \begin{cases} \frac{1}{2^{\frac{\nu}{2}} \Gamma(\frac{\nu}{2})} t^{\frac{\nu}{2}-1} e^{-\frac{t}{2}} & x > 0 \\ 0 & x \leq 0 \end{cases} \quad (4.24)$$

where the gamma function  $\Gamma(a)$  is defined as

$$\Gamma(a) \equiv \int_0^{\infty} e^{-t} t^{a-1} dt. \quad (4.25)$$

If the parameter  $a$  is an integer the gamma function becomes the faculty function

$$\Gamma(n+1) = n! \quad n \in \mathbb{Z}. \quad (4.26)$$

The respective distribution function is obtained by integration of  $f(\nu, t)$ :

$$\begin{aligned} P(\nu, \chi^2 \leq x) &= \int_{-\infty}^x f(\nu, t) dt \\ &= \begin{cases} \frac{\gamma(\frac{1}{2}\nu, \frac{1}{2}x)}{\Gamma(\frac{1}{2}\nu)} \equiv \frac{1}{2^{\frac{\nu}{2}} \Gamma(\frac{\nu}{2})} \int_0^x t^{\frac{\nu}{2}-1} e^{-\frac{t}{2}} dt & x > 0 \\ 0 & x \leq 0 \end{cases} \end{aligned} \quad (4.27)$$

The function  $\gamma(a, x)$  is called the incomplete gamma function. The  $\chi^2$  probability density and distribution functions with different degrees of freedom are plotted in figure 4.2.

The value  $P(\nu, \chi^2 \leq x)$  describes the probability that a  $\chi^2 \leq x$  would be observed by chance alone. Likewise  $1 - P(\nu, \chi^2 \leq x)$  is the probability that the  $\chi^2$  is at least  $x$ .

A high amount of very low probabilities  $1 - P$  for a particular data set are an indication that either the system model is wrong, the variances of the errors are incorrect or the errors are not normally distributed. On the other hand, an accumulation of probabilities close to one, that means an abnormally large amount of seemingly good  $\chi^2$  values near zero have been observed, may caused by overestimating the errors [PTVF07, chap. 15].

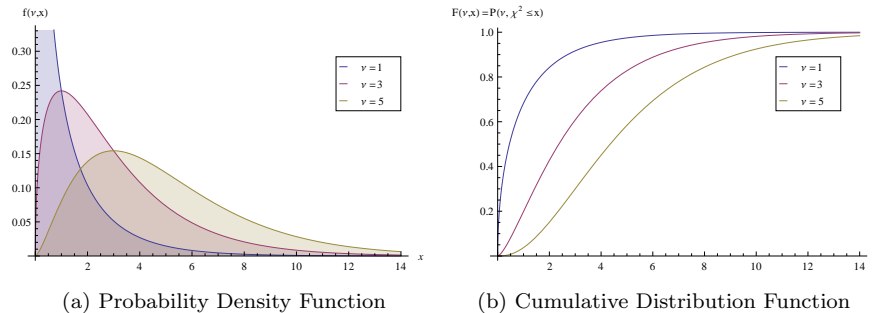


Figure 4.2:  $\chi^2$  probability density function and distributions with different degrees of freedom  $\nu$ .

#### 4.2.4 Alternative Formulations of the Kalman Filter

There are many different ways to write equations 4.21b and 4.21c that calculate the gain and update the covariance matrix. These will be referred to as the conventional formulation as they were originally proposed by Kalman [Kal60]. While mathematically equivalent these may have advantages concerning computational efficiency or numerical stability.

The formulation by *Swerling* [Swe59] calculates the covariance before the gain matrix:

$$C_k = \left[ (C_k^{k-1})^{-1} + H_k^T (V_k)^{-1} H_k \right]^{-1} \quad (4.28a)$$

$$K_k = C_k H_k^T (V_k)^{-1} \quad (4.28b)$$

It requires the inversion of three matrices, two of which have the dimension of the state vector. Therefore the dimension of the state vector should be small if this formulation is used.

The *Joseph stabilized* [BJ68] formulation is more complex than formula 4.21c for the calculation of the covariance matrix, but it is more robust and numerically stable.

$$C_k = (I - K_k H_k) C_k^{k-1} (I - K_k H_k)^T + K_k V_k K_k^T \quad (4.29)$$

Unlike 4.21c it guarantees that  $C_k$  remains symmetric and positive definite as long as  $C_k^{k-1}$  is symmetric and positive definite as well.

The *square root* and *UD*-filter perform a Cholesky- and  $UDU^T$ -decomposition respectively of the covariance matrix. The Kalman filter equations are reformulated so that only the Cholesky- and  $UD$ -factors are propagated. This increases numerical stability at the cost of additional computational effort. Algorithms for these methods can be found in [GA08, chapter 6].

The *sequential filter* [Sim06, chapter 6] avoids matrix inversions and treats each component of the measurement vector separately. This can be beneficial

for embedded systems. Let  $r$  be the dimension of the measurement vector  $\mathbf{m}_k$ . When processing the measurement  $\mathbf{m}_k$  initialize with  $\mathbf{x}_{0,k} = \mathbf{x}_k^{k-1}$  and  $C_{0,k} = C_k^{k-1}$ . The Kalman gain  $K_{i,k}$ , covariance matrix  $C_{i,k}$  and state vector  $\mathbf{x}_{i,k}$  are then updated iteratively for each  $i \in \{1 \dots r\}$ :

$$K_{i,k} = \frac{C_{i,k} H_{i,k}^T}{V_{i,k}} \quad (4.30a)$$

$$C_{i,k} = (I - K_{i,k} H_{i,k}) C_{i-1,k} \quad (4.30b)$$

$$\mathbf{x}_{i,k} = \mathbf{x}_{i-1,k} + K_{i,k} (\mathbf{m}_{i,k} - H_{i,k} \mathbf{x}_{i-1,k}) \quad (4.30c)$$

where  $H_{i,k}$  is the  $i$ -th row of the projector matrix and  $V_{i,k}$  is the variance of the  $i$ -th component of the measurement vector.

#### 4.2.5 Yet More Kalman Filter and Alternatives

Many extensions of the Kalman filter and adaptations to certain problems exist. For example, the Kalman filter can be modified to work with coloured noise or correlated process and measurement noise. Several approaches like the unscented Kalman filter exist that reduce the linearization error of the extended Kalman filter. For highly non-linear systems the particle filter may be an alternative to the Kalman filter. There are approaches that improve stability for non-Gaussian noise. Information filtering only propagates the inverse covariance matrix and can improve numerical stability with large initial estimation uncertainties. The  $H_\infty$  filter can be applied when the system model is unknown or varying. The books of Dan Simon [Sim06] and Grewal and Andrews [GA08] extensively deal with such topics.

Some extensions of the Kalman filter belong to the class referred to as adaptive filters and are suitable in case of fake measurements or multiple measurements per detector layer. An overview of adaptive methods can be found in [FS99] and [FS06]. One such method is the Deterministic Annealing Filter.

### 4.3 The Deterministic Annealing Filter

During track reconstruction (chapter 3) track candidates may be assigned wrong measurements. Furthermore, MDC wire measurements only yield a drift distance. It is ambiguous on which side of the wire the particle passed by. Additionally, it is possible that a track passes two neighbouring cells in an MDC layer. Such an environment with multiple, ambiguous and fake measurements poses a challenge to the Kalman filter as it does not include a mechanism to handle competing measurements or reject measurements.

The adaptive methods incorporate mechanisms that can handle competing measurements in a layer and optimize hit assignment. An overview of adaptive methods can be found in [FS99] and [FS06]. The method implemented for HADES is the **Deterministic Annealing Filter (DAF)** which is an extension of the Kalman filter. Measurements are now assigned weights which is the

estimated probability that the measurement belongs to the track. Competing measurements  $\mathbf{m}_k^i$  for the same time (detector layer)  $k$  are combined into an effective measurement  $\bar{\mathbf{m}}_k$ :

$$\bar{\mathbf{m}}_k = \bar{V}_k \left( \sum_{i=1}^n p_k^i (V_k^i)^{-1} \mathbf{m}_k^i \right) \quad (4.31)$$

$$\bar{V}_k = \left( \sum_{i=1}^n p_k^i (V_k^i)^{-1} \right)^{-1} \quad (4.32)$$

The measurements are weighted by their respective error covariances  $V_k^i$ . The index  $i = \{1 \dots n\}$  runs over all competing measurements. The effective measurement  $\bar{\mathbf{m}}_k$  and effective measurement covariance  $\bar{V}_k$  can then be used in the Kalman filter equations 4.21a and 4.21b instead of  $\mathbf{m}_k$  and  $V_k$ . The DAF algorithm consists of two steps: First, a Kalman filter and smoother is run on all measurements using their respective weights. Then the weights are recalculated according to the distance of the real measurement to the estimated measurement from the smoother [Fle06]:

$$p_k^i = \frac{\Phi_k^i}{\sum_{j=1}^n \Lambda_k^j + \Phi_k^j} \quad (4.33)$$

$$\Phi_k^i = \frac{1}{(2\pi)^{\frac{r}{2}} \sqrt{T \cdot |V|}} \cdot \exp \left( -\frac{1}{2T} (\mathbf{m}_k^i - \mathbf{h}_k(\mathbf{x}_k^n))^T (V_k^i)^{-1} (\dots) \right) \quad (4.34)$$

$$\Lambda_k^i = \frac{1}{(2\pi)^{\frac{r}{2}} \sqrt{T \cdot |V|}} \cdot \exp \left( -\frac{\chi_{\text{cut}}^2}{2T} \right) \quad (4.35)$$

with  $r \equiv \dim \mathbf{m}_k^i$  and  $|V|$  is the determinant of  $V$ .

There are two parameters in the DAF: the cut-off parameter  $\chi_{\text{cut}}^2$  and annealing factor  $T$ . The cut-off parameter is equivalent to a  $\chi^2$  cut for low  $T$ . The annealing factor basically inflates the measurement errors. The effect of these two parameters is shown in figure 4.3 for a single measurement without competitors. The distance has little effect on the assignment probability for very high annealing factors so that even measurements that are far away are still taken into account. The lower the annealing factor the higher the weight is affected by the distance and the DAF is forced to make a decision whether to accept or reject the measurement.

A problem when optimizing measurement weights is to avoid local optima since information in the initial phases of the filter may be insufficient. Therefore an annealing process is introduced where the aforementioned two steps (Kalman filter and recalculating the weights) are iterated several times and the annealing factor is lowered in each iteration. Since the measurements now contribute according to their weights the number of degrees of freedom for the DAF then

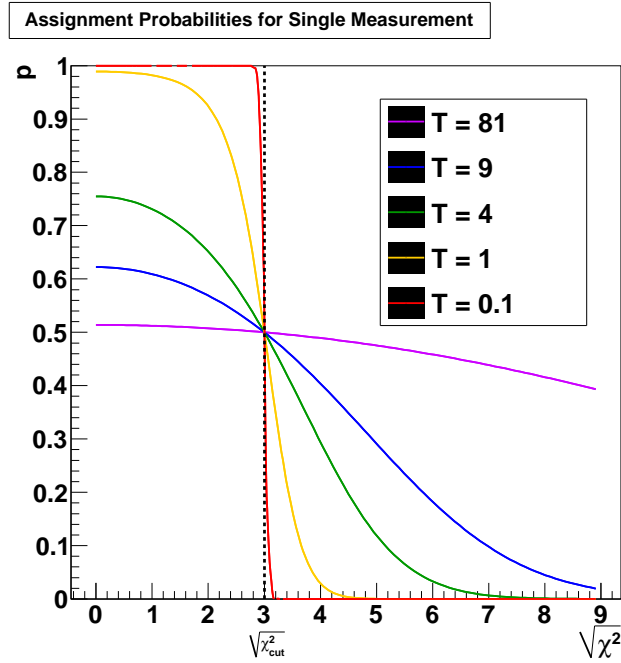


Figure 4.3: The weight  $p$  of a measurement without competitors as a function of the standardized distance  $\chi_{\text{cut}}^2$  for different annealing factors  $T$ . The distance has less effect on the weight for high annealing factors.

is [Win02]:

$$\nu = \left( \sum_k \sum_i p_k^i \cdot r \right) - s. \quad (4.36)$$

## Chapter 5

# Runge-Kutta Methods

The Kalman filter requires a method to propagate charged particles through the magnetic field. In a homogeneous the particle follows a helical trajectory. However, in inhomogeneous magnetic fields as it is the case for the HADES detector the equation of motion can no longer be solved analytically making a numerical approach necessary. The class of Runge-Kutta schemes are numerical methods to integrate differential equations. Finding a solution  $x = x(t_N)$  of an ordinary differential equation

$$\frac{dx}{dt} = f(x, t) \quad (5.1)$$

at the point  $t_N$  with the known value

$$x(t_0) = x_0. \quad (5.2)$$

at some starting point  $t_0$  is called an initial value problem. A solution  $x(t)$  would be

$$x(t_N) = x_0 + \int_{t_0}^{t_N} f(x, t) dt. \quad (5.3)$$

If the primitive function  $x(t)$  is unknown this integral has to be solved numerically. One possibility is to replace the unknown integral by a known increment function  $\Phi_n$ . The interval  $[t_0, t_N]$  is partitioned into  $N$  subintervals. Let  $h_n = t_{n+1} - t_n$  be the length of the  $n$ -th subinterval or the step size. The solution  $x(t)$  can then be recursively approximated for each  $n \in 1, \dots, N$  by

$$x_{n+1} = x_n + h_n \cdot \Phi_n(h_n, t_n, x_n, x_{n+1}) + \mathcal{O}(h_n^{p+1}). \quad (5.4)$$

This is called a one-step method. Each step is treated as if it were the first one since the value of  $x_{n+1}$  only depends on the result  $x_n$  of the previous step. The local discretization error in each step is of the order  $p + 1$ . The global error for the whole integration is of one order lower. If the increment function does

not depend on  $x_{n+1}$  then the scheme is called explicit, otherwise it is called implicit. Implicit schemes require solving nonlinear algebraic equations, but are numerically more stable and less susceptible to round-off errors [DB08, chap. 6] [Klo11]. The simplest approach is the Euler method where the derivative  $f(x, t)$  is used as the increment function in each step. The explicit Euler method is

$$\begin{aligned} k_1^{(n)} &= f(x_n, t_n) \\ x_{n+1} &= x_n + h_n \cdot k_1^{(n)} + \mathcal{O}(h_n^2) \end{aligned} \quad (5.5)$$

and the implicit form would be

$$\begin{aligned} k_1^{(n)} &= f(x_{n+1}, t_{n+1}) \\ x_{n+1} &= x_n + h_n \cdot k_1^{(n)} + \mathcal{O}(h_n^2). \end{aligned} \quad (5.6)$$

Both are one-step methods of order  $p = 1$  since the error in each step is of second order  $\mathcal{O}(h_n^2)$ . The Euler method is neither very accurate nor stable [PTVF07]. Only one evaluation point  $k_1^{(n)}$  is used in each step. The accuracy can be improved by adding more evaluation points. For example, the midpoint rule as proposed by Carl Runge ([Run95]) improves the Euler scheme by using a second point in the middle of each subinterval:

$$\begin{aligned} k_1^{(n)} &= f(x_n, t_n) \\ k_2^{(n)} &= f\left(x_n + \frac{1}{2}h_n k_1^{(n)}, t_n + \frac{1}{2}h_n\right) \\ x_{n+1} &= x_n + h_n \cdot k_2^{(n)} + \mathcal{O}(h_n^3). \end{aligned} \quad (5.7)$$

In general a Runge-Kutta method is a derivative free one-step method with  $s$  evaluation points:

$$\begin{aligned} k_i^{(n)} &= f\left(x_n + h_n \sum_{j=1}^s a_{ij} k_j^{(n)}, t_n + c_i \cdot h_n\right), \quad i = 1, \dots, s \\ x_{n+1} &= x_n + h_n \cdot \underbrace{\sum_{i=1}^s b_i k_i^{(n)}}_{\Phi_n(h_n, t_n, x_n, x_{n+1})} + \mathcal{O}(h_n^{p+1}) \end{aligned} \quad (5.8)$$

where  $0 \leq c_1 < \dots < c_s \leq 1$ . The parameters may not be chosen arbitrarily, but may be subject to constraints. An important constraint is consistency as inconsistent methods do not converge [DB08]. A Runge-Kutta method is consistent if and only if

$$\sum_{i=1}^s b_i = 0. \quad (5.9)$$

Other possible constraints include:

- invariance to autonomisation  $\Leftrightarrow c_i = \sum_{j=1}^s a_{ij}$ ,  $i = 1 \dots s$  and the scheme is consistent,
- explicit  $\Leftrightarrow a_{ij} = 0$ ,  $\forall j \geq i$  with  $i, j = 1, \dots, s$ ,
- further order constraints.

Thus, the Euler method (5.5) would be a Runge-Kutta method of  $1^{st}$  order with  $s = 1$  evaluation points while the midpoint method a  $2^{nd}$  order with 2 evaluation points. Generally, the order  $p$  of the scheme and number of evaluation points  $s$  do not have to be the same. An often used Runge-Kutta method with  $p = s = 4$  is

$$\begin{aligned}
 k_1^{(n)} &= f(x_n, t_n) \\
 k_2^{(n)} &= f\left(x_n + \frac{1}{2}h_n k_1^{(n)}, t_n + \frac{1}{2}h_n\right) \\
 k_3^{(n)} &= f\left(x_n + \frac{1}{2}h_n k_2^{(n)}, t_n + \frac{1}{2}h_n\right) \\
 k_4^{(n)} &= f\left(x_n + h_n k_3^{(n)}, t_n + h_n\right) \\
 x_{n+1} &= x_n + \frac{1}{6}h_n \cdot \left(k_1^{(n)} + 2k_2^{(n)} + 2k_3^{(n)} + k_4^{(n)}\right) + \mathcal{O}(h_n^5). \quad (5.10)
 \end{aligned}$$



## Chapter 6

# Adaptation of the Kalman Filter for HADES

In order to use the Kalman filter in track reconstruction the following quantities need to be defined:

- the parameters of the state vector  $\mathbf{x}_k$  of the trajectory at a measurement layer,
- initial values for the state vector and covariance matrix (described in chapter 8,
- the propagation function  $\mathbf{f}_k(\mathbf{x}_k)$  to extrapolate the track state through the magnetic field to the next measurement layer and the propagator matrix  $F_k$ ,
- the projection function  $\mathbf{h}_k(\mathbf{x}_k)$  and the respective projector matrix  $H_k$ ,
- the process noise covariance matrix  $Q_k$ .

The complexity of the Kalman filter depends on the dimension of the state vector as several matrices depend on this dimension. Higher matrix dimensions lead to more operations and may introduce roundoff errors. Therefore, the state vector should have few parameters for an efficient track reconstruction. Correlations between the state parameters should be avoided. Both the projection and propagation function should be close to linear to minimize the linearization error of the extended Kalman filter.

The Kalman filter works with either reconstructed segment points. The track is iteratively propagated from one drift chamber to the next one and the estimate then further improved by taking the measurement into account. The magnetic field is strongest between MDC-II and MDC-III. This step mainly determines the quality of the estimate from the Kalman filter. The Kalman filter is further adapted to work with the drift time information and an annealing process is added to deal with competing and fake hits. The projection function differs

vastly for these approaches. Figure 6.1 shows an example track of a particle traversing the detector. The clusters of fired wires can be seen and the drift chambers are sketched out.

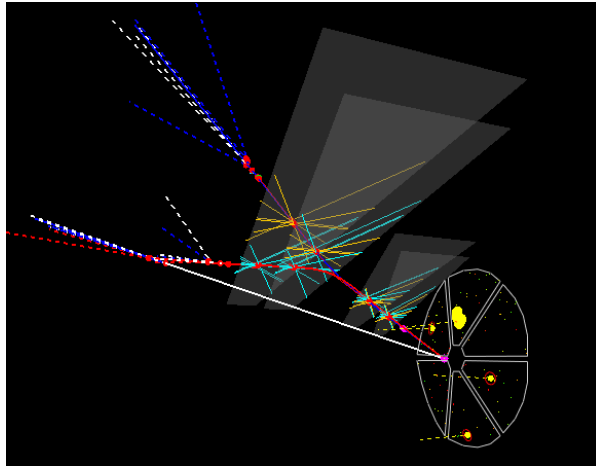


Figure 6.1: Example track traversing the detector [Gal12].

## 6.1 Track Representation

In the HADES specific coordinate system called the sector coordinate system a track is parameterized by four variables  $(\Theta, \phi, \rho, z)$  as depicted in figure 6.2. The variable  $\Theta$  is the polar angle, i.e. the angle between the positive z-axis and the track direction vector. The azimuthal angle  $\phi$  is the angle between the positive x-axis and the track direction. The parameter  $\rho$  is the minimum distance of the track to the z-axis. The final parameter  $z$  is the coordinate on the z-axis which points in beam direction. The angle  $\phi$  can assume values between  $60^\circ$  and  $120^\circ$ . It is rotated with respect to the lab system by  $(n + 1) \cdot 60^\circ \bmod 360^\circ$  where  $n \in \{0, \dots, 5\}$  is the sector number.

The state of a particle is fully described by its position and momentum. The chosen state vector for the Kalman filter is the same as in [Man99, GK06]:

$$\mathbf{x} = \left( x, y, t_x, t_y, \frac{q}{p} \right). \quad (6.1)$$

The parameters  $x$  and  $y$  are two cartesian coordinates in the sector coordinate system. Since track states are always located on a planar measurement layer it is sufficient to store only two position coordinates. The track direction is

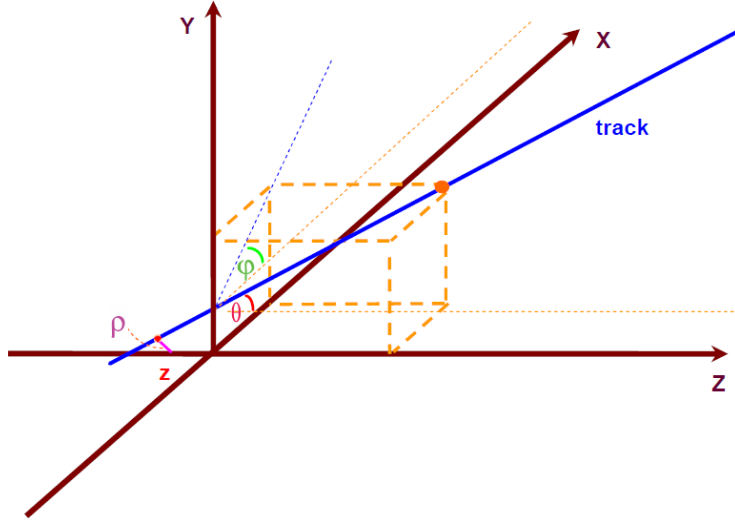


Figure 6.2: The track parameterization in HADES [Rus06].

represented by the tangents of two angles:

$$t_x = \tan \phi = \frac{p_x}{p_z} \quad (6.2)$$

$$t_y = \tan \Theta = \frac{p_y}{p_z} \quad (6.3)$$

The final parameter  $q/p$  is the ratio of the particle charge  $q$  in units of elementary charges and the magnitude of the momentum  $p$ .

## 6.2 Propagation through the Magnetic Field

The equation of motion of a charged particle traversing an inhomogeneous magnetic field can not be solved analytically. A numerical method to solve this problem are Runge-Kutta methods. A general description of Runge-Kutta schemes can be found in chapter 5. The following Runge-Kutta method offers a good compromise between accuracy, computation costs and numerical stability:

$$\begin{aligned}
k_1^{(n)} &= f(x_n, t_n) \\
k_2^{(n)} &= f\left(x_n + \frac{1}{2}h_n k_1^{(n)}, t_n + \frac{1}{2}h_n\right) \\
k_3^{(n)} &= f\left(x_n + \frac{1}{2}h_n k_2^{(n)}, t_n + \frac{1}{2}h_n\right) \\
k_4^{(n)} &= f\left(x_n + h_n k_3^{(n)}, t_n + h_n\right) \\
x_{n+1} &= x_n + \frac{1}{6}h_n \cdot \left(k_1^{(n)} + 2k_2^{(n)} + 2k_3^{(n)} + k_4^{(n)}\right) + \mathcal{O}(h_n^5). \quad (6.4)
\end{aligned}$$

This method is of order four and the function  $f(x, t)$  is evaluated at four points. The differential equation that needs to be solved is the equation of motion of a charged particle in a magnetic field given by the Lorentz force:

$$\frac{d\mathbf{p}}{dt} = \mathbf{F}_l = \kappa \cdot q \cdot \mathbf{v} \times \mathbf{B} \quad (6.5)$$

with the momentum  $\mathbf{p}$  given in  $MeV/c$ , the charge  $q$  in units of the elementary charge, the velocity  $\mathbf{v}$  in  $mm/s$  and the magnetic field in Tesla. The coefficient  $\kappa$  is necessary to convert the Lorentz force in units of  $MeV/c \cdot s$ . Its value then is

$$\kappa = 0.299792458 \frac{MeV}{c \cdot mm \cdot T}. \quad (6.6)$$

Since the Lorentz force always acts perpendicular to the direction of motion the magnitude of the particle's velocity and momentum does not change. The time can be replaced by the path length:  $dt = ds/|\mathbf{v}|$ . Equation 6.5 then becomes:

$$d\mathbf{p} = \kappa \cdot q \cdot \frac{\mathbf{v}}{|\mathbf{v}|} \times \mathbf{B} ds. \quad (6.7)$$

Substituting the normalized direction vector  $\mathbf{e} = \mathbf{v}/|\mathbf{v}| = \mathbf{p}/|\mathbf{p}|$  the above equation is

$$d\mathbf{e} = \begin{pmatrix} de_x \\ de_y \\ de_z \end{pmatrix} = \kappa \cdot \frac{q}{p} \cdot \mathbf{e} \times \mathbf{B} ds = \kappa \cdot \frac{q}{p} \cdot \begin{pmatrix} e_y B_z - e_z B_y \\ e_z B_x - e_x B_z \\ e_x B_y - e_y B_x \end{pmatrix} \cdot ds. \quad (6.8)$$

The particle is propagated in  $z$ -direction which is aligned along the beam direction by using the Runge-Kutta method shown in 6.4. The differentials of the two state parameters  $t_x = e_x/e_z$  and  $t_y = e_y/e_z$  that describe the track direction

then are:

$$\begin{aligned}
dt_x &= (e_z \cdot de_x - e_x \cdot de_z) / e_z^2 \\
&= \kappa \cdot \frac{q}{p} \cdot (e_y e_z B_z - e_z^2 B_y - e_x^2 B_y + e_x e_y B_x) / e_z^2 \cdot ds \\
&= \kappa \cdot \frac{q}{p} \cdot (t_y B_z - (1 + t_x^2) B_y + t_x t_y B_x) \cdot ds \tag{6.9}
\end{aligned}$$

$$\begin{aligned}
dt_y &= (e_z \cdot de_y - e_y \cdot de_z) / e_z^2 \\
&= \dots \\
&= \kappa \cdot \frac{q}{p} \cdot (-t_x B_z - (1 + t_y^2) B_x - t_x t_y B_y) \cdot ds. \tag{6.10}
\end{aligned}$$

Replacing the path length  $s$  by the  $z$ -coordinate with

$$ds = \sqrt{t_x^2 + t_y^2 + 1} \cdot dz,$$

the differential equation 6.8 can be rewritten:

$$\begin{aligned}
x' &\equiv \frac{\partial x}{\partial z} = t_x \\
y' &\equiv \frac{\partial y}{\partial z} = t_y \\
t'_x &\equiv \frac{\partial t_x}{\partial z} \\
&= \kappa \cdot \frac{q}{p} \cdot \sqrt{t_x^2 + t_y^2 + 1} \cdot (t_y B_z - (1 + t_x^2) B_y + t_x t_y B_x) \tag{6.11} \\
t'_y &\equiv \partial t_y / \partial z \\
&= \kappa \cdot \frac{q}{p} \cdot \sqrt{t_x^2 + t_y^2 + 1} \cdot (-t_x B_z - (1 + t_y^2) B_x - t_x t_y B_y) \\
\left(\frac{q}{p}\right)' &\equiv \frac{\partial q}{\partial z} = 0.
\end{aligned}$$

The derivation of the state vector  $\mathbf{x}$  by the step direction  $z$  is:

$$\frac{\partial \mathbf{x}}{\partial z} = \begin{pmatrix} \partial x / \partial z \\ \partial y / \partial z \\ \partial t_x / \partial z \\ \partial t_y / \partial z \\ \partial(q/p) / \partial z \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \\ t'_x \\ t'_y \\ 0 \end{pmatrix} \equiv \mathbf{f}(\mathbf{x}, z). \tag{6.12}$$

This is the propagation function  $\mathbf{f}(\mathbf{x}, z)$  in equation 4.18. The propagation of the state vector  $\mathbf{x}$  by a Runge-Kutta step from  $z_0$  to  $z_e$  is done by solving the above equation. Let  $h = z_e - z_0$  be the step size. The Runge-Kutta method numerically computes the change in the track state  $\Delta \mathbf{x} = \mathbf{x}_{z_e} - \mathbf{x}_{z_0}$  from  $\mathbf{x}_{z_0}$  to  $\mathbf{x}_{z_e}$  by evaluating the propagation function  $\mathbf{f}$  at four points with different

z-coordinates as in 6.4:

$$\begin{aligned}
\Delta \mathbf{x}_1 &= h \cdot \mathbf{f}(\mathbf{x}(z_0), z_0) \\
\Delta \mathbf{x}_2 &= h \cdot \mathbf{f}\left(\mathbf{x}(z_0) + \frac{1}{2}\Delta \mathbf{x}_1, z_0 + \frac{1}{2}h\right) \\
\Delta \mathbf{x}_3 &= h \cdot \mathbf{f}\left(\mathbf{x}(z_0) + \frac{1}{2}\Delta \mathbf{x}_2, z_0 + \frac{1}{2}h\right) \\
\Delta \mathbf{x}_4 &= h \cdot \mathbf{f}(\mathbf{x}(z_0) + \Delta \mathbf{x}_3, z_0 + h).
\end{aligned} \tag{6.13}$$

The predicted state vector then is:

$$\mathbf{x}_{z_e} = \mathbf{x}_{z_0} + \frac{1}{6}\Delta \mathbf{x}_1 + \frac{1}{3}\Delta \mathbf{x}_2 + \frac{1}{3}\Delta \mathbf{x}_3 + \frac{1}{6}\Delta \mathbf{x}_4 + \mathcal{O}(h^5). \tag{6.14}$$

The state vector can be approximated in each intermediate position

$$z_i = \left\{ z_0, z_0 + \frac{1}{2}h, z_0 + \frac{1}{2}h, z_e \right\} \tag{6.15}$$

by

$$\mathbf{x}(z_i) = \mathbf{x}_{z_0} + \frac{\Delta \mathbf{x}_{z_i}}{h} \cdot (z_i - z_0). \tag{6.16}$$

The propagator matrix  $F$  can be extrapolated by

$$F \equiv \frac{d\mathbf{x}_{z_e}}{d\mathbf{x}_{z_0}} = I + \frac{1}{6}F_1 + \frac{1}{3}F_2 + \frac{1}{3}F_3 + \frac{1}{6}F_4. \tag{6.17}$$

The required intermediate  $F_i, i \in \{1, \dots, 4\}$  are computed as follows:

$$\begin{aligned}
F_i &\equiv \frac{d\Delta \mathbf{x}_{z_i}}{d\mathbf{x}_{z_0}} \\
&= h \cdot \frac{d\mathbf{f}(\mathbf{x}_{z_i}, z_i)}{d\mathbf{x}_{z_i}} \cdot \frac{d\mathbf{x}_{z_i}}{d\mathbf{x}_{z_0}} \\
&= h \cdot \frac{d\mathbf{f}(\mathbf{x}_{z_i}, z_i)}{d\mathbf{x}_{z_i}} \cdot \left( I + F_{i-1} \cdot \frac{z_i - z_0}{h} \right)
\end{aligned} \tag{6.18}$$

with the Jacobi matrices

$$\begin{aligned}
\frac{d\mathbf{f}(\mathbf{x}, z_i)}{\mathbf{x}} &= \begin{pmatrix} \frac{\partial x'}{\partial x} & \frac{\partial x'}{\partial y} & \frac{\partial x'}{\partial t_x} & \frac{\partial x'}{\partial t_y} & \frac{\partial x'}{\partial(\frac{q}{p})} \\ \frac{\partial y'}{\partial x} & \frac{\partial y'}{\partial y} & \frac{\partial y'}{\partial t_x} & \frac{\partial y'}{\partial t_y} & \frac{\partial y'}{\partial(\frac{q}{p})} \\ \frac{\partial t'_x}{\partial x} & \frac{\partial t'_x}{\partial y} & \frac{\partial t'_x}{\partial t_x} & \frac{\partial t'_x}{\partial t_y} & \frac{\partial t'_x}{\partial(\frac{q}{p})} \\ \frac{\partial t'_y}{\partial x} & \frac{\partial t'_y}{\partial y} & \frac{\partial t'_y}{\partial t_x} & \frac{\partial t'_y}{\partial t_y} & \frac{\partial t'_y}{\partial(\frac{q}{p})} \\ \frac{\partial(\frac{q}{p})'}{\partial x} & \frac{\partial(\frac{q}{p})'}{\partial y} & \frac{\partial(\frac{q}{p})'}{\partial t_x} & \frac{\partial(\frac{q}{p})'}{\partial t_y} & \frac{\partial(\frac{q}{p})'}{\partial(\frac{q}{p})} \end{pmatrix} \\
&= \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \partial t'_x / \partial t_x & \partial t'_x / \partial t_y & \partial t'_x / \partial(\frac{q}{p}) \\ 0 & 0 & \partial t'_y / \partial t_x & \partial t'_y / \partial t_y & \partial t'_y / \partial(\frac{q}{p}) \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (6.19)
\end{aligned}$$

These formulas allow to propagate the state vector and calculate the Jacobian matrix needed by the Kalman filter [GK06].

## 6.3 Measurement Vector and Projector Matrix

### 6.3.1 Reconstructed Segment Hits

In a first step the Kalman filter was adapted to use the reconstructed segment points as “measurement” vectors. As described in chapter 3 the segment fit provides space coordinates with errors for up to four points that are assigned to a track. The segment hits are stored in the coordinate system of the respective segment plane. Since these planes are not parallel the coordinate and error vectors are transformed into a common coordinate system, the sector coordinate system which is described in section 6.1. The transformed vectors are passed to the Kalman filter as measurement  $\mathbf{m}_k$  and error vectors  $\boldsymbol{\epsilon}_k$ . Only the  $x$  and  $y$  coordinates are stored. The third coordinate is given by the constraint that the measurement is located on the segment plane.

The measurement equation or projection function  $\mathbf{h}_k(\mathbf{x}_k)$  extracts a measurement from a state vector. The chosen track state representation contains the  $x$  and  $y$  coordinates resulting in a simple measurement equation:

$$\mathbf{h}_k(\mathbf{x}_k = (x, y, t_x, t_y, q/p)^T) = (x, y)^T = \mathbf{m}_k. \quad (6.20)$$

The projector matrix is the Jacobian of  $\mathbf{h}$ :

$$\begin{aligned}
H_k &\equiv \left. \frac{\partial \mathbf{h}_k}{\partial \mathbf{x}} \right|_{\mathbf{x}_k} \\
&= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad (6.21)
\end{aligned}$$

### 6.3.2 Drift Chamber Measurements

In a second step the Kalman filter for HADES was adjusted to use the MDC time measurements directly. The relevant component of this measurement is the drift time, i.e. the time it took the primary electrons to trigger a signal on the wire. First of all this drift time has to be extracted from the MDC time measurement. The track is propagated to the next MDC layer. With these information the minimum distance of the track to the sense wire can be obtained. This minimum distance is used as the measurement vector in the Kalman filter. There are two additional problems:

- There may be fake measurements that in reality belong to a different track.
- A particle may traverse two neighbouring drift cells and trigger two measurements in the same MDC layer.

The Kalman filter cannot handle fake or competing measurements. One solution is to use the Deterministic Annealing Filter introduced in 4.3. A common reference system for all the competing measurements in an MDC layer is needed in order to combine them into an effective measurement.

#### Extracting the Drift Time

Figure 6.3a shows a simulated signal of a sense wire. A logical signal is generated with a width that is comparable to the time the amplified and shaped signal exceeded the threshold. A **T**ime to **D**igital **C**onverter (TDC) then measures *time1* ( $t_1$ ) and *time2* ( $t_2$ ) where  $t_1$  is the time between the leading edge of the logical ASD8 signal and the common stop signal. Likewise,  $t_2$  is the time between the trailing edge and the common stop. The difference  $t_2 - t_1$  is the **T**ime **o**ver **T**hreshold (ToT).

The TDC measurements after proper TDC calibration are the sum of the real drift time  $t_{drift}$ , the propagation time  $t_{wire}$  of the signal along the sense wire to the read-out electronics and the time-of-flight  $t_{tof}$  of the particle to the drift cell relative to the stop signal derived from the trigger. The time-of-flight also includes deviations that are common to all cells [HAD09, Mar05].

After wire clusters have been found during the track candidate search (explained in chapter 3) the position  $x_{wire}$  where the signal was generated on the wire is known. The signal speed on a sense wire obtained from the *MdcDigitPar* container in HYDRA<sup>1</sup> is  $v_{wire} = 250 \text{ mm/ns}$  [Mar11b], roughly  $5/6$  of the speed of light. The propagation time on the wire then simply is

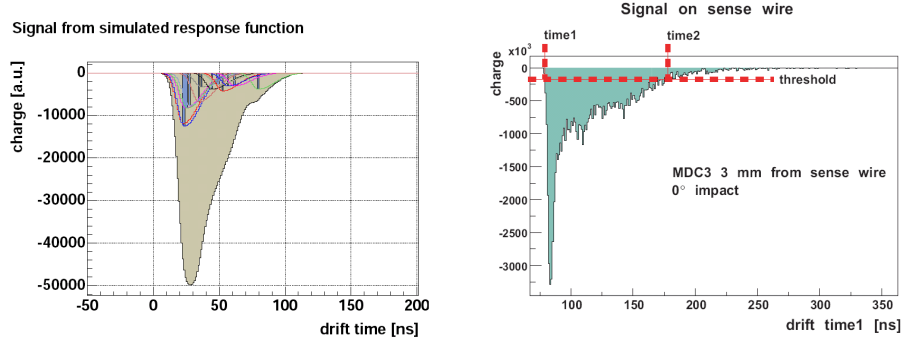
$$t_{wire} = \frac{x_{wire}}{v_{wire}}. \quad (6.22)$$

The corrected time  $t_1 - t_{wire}$  still contains the time-of-flight offset. The track candidate search computes two points of the trajectory in the inner and outer

---

<sup>1</sup>HADES **sY**stem for **D**ata **R**eduction **A**nalysis - The analysis software framework for HADES.





(a) A drift chamber signal is composed of the ASD8 response functions of single electrons. The signal has been simulated with GARFIELD [CER00]. (b) The values  $time1$  is obtained when the leading edge of the ASD8 signal first exceeded the threshold. When the trailing edge falls below the threshold again  $time2$  is known.

Figure 6.3: Simulated signals of a drift chamber. Both pictures were taken from [Mar05].

segment, respectively. Assuming a straight line track model in each segment, the class *HMdcSizesCellsLayer* can calculate the minimum distance and impact angle of the trajectory. Figure 6.4 shows how these values are defined for a particle trajectory. Using these estimations for the impact parameters the container *MdcCal2ParSim* can calculate the expected drift time  $t_{drift,sim}$  and error  $\Delta t_{1,sim}$  of  $time1$ . In the MDC layer  $l$  the time-of-flight correction would then be

$$t_{tof}^l = t_1^l - t_{wire}^l - t_{drift,sim}^l. \quad (6.23)$$

Since this is based on inaccurate estimations of the impact parameters a weighted mean value over all layers  $l$  in a module is used instead. The individual time-of-flight corrections are weighted by the variances  $(\Delta t_{1,sim})^2$  of  $time1$ , i.e. measurements with a small error are weighted higher than those with a large error [Sim06, chap. 3.2]:

$$\bar{t}_{tof} = \left[ \sum_{l=0}^5 \frac{1}{(\Delta t_{1,sim}^l)^2} \right]^{-1} \cdot \sum_{l=0}^5 \frac{t_1^l - t_{wire}^l - t_{drift,sim}^l}{(\Delta t_{1,sim}^l)^2}. \quad (6.24)$$

The superscript  $l$  denotes the index of a layer in a drift chamber module. With these corrections the measured drift time then is

$$t_{drift} = t_1 - t_{wire} - \bar{t}_{tof} \quad (6.25)$$

The uncertainty in the time-of-flight calculations contributes to the drift time error. To estimate this error data from Au+Au reactions at 1.5A GeV simulated

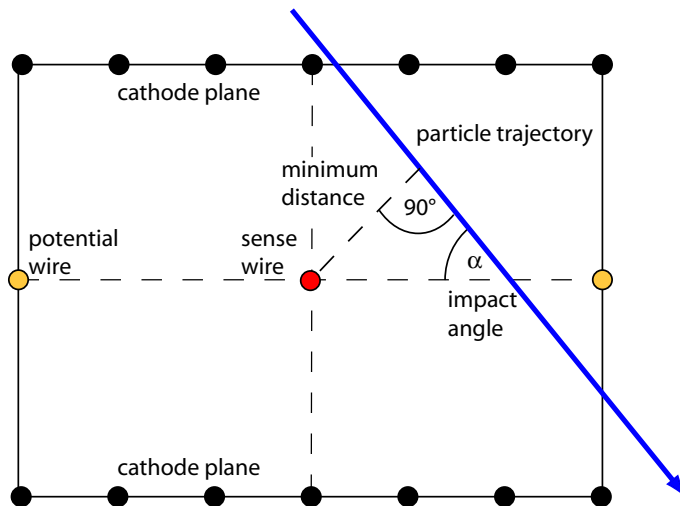


Figure 6.4: The trajectory of a particle traversing a drift cell is parameterized by its minimum distance to the sense wire and impact angle  $\alpha$ . An impact angle of  $90^\circ$  corresponds to an impact perpendicular to the wire plane [Mar05].

with HGeant<sup>2</sup> were used. The weighted mean values  $\bar{t}_{tof}$  for the time-of-flight were calculated as previously described and compared with the time-of-flight in the HGeant simulation obtained from the container *HMdcCal1Sim*. The Kalman filter uses the RMS of the distributions shown in figure 6.5 as the additional error due to the time-of-flight correction.

For comparison the time-of-flight from *HMdcCal1Sim* minus the corrections for the individual layers are plotted in figure 6.6. The results are similar to the approach using the weighted mean value, except for the first MDC module which shows a systematic shift in some of the time-of-flight calculations. This second peak is gone when working with the mean values. The reasons for this peculiar structure are not known and should be investigated in the future.

### Reconstructing the Minimum Distance

There is a non-trivial correlation between the drift time and the minimum distance of the particle to the sense wire [Mar05]. The class *HMdcSizesCellsLayer* is used to convert the time to a distance and calculate the impact angle of the trajectory. It needs two space points as input that are obtained from the expected state vector. Since the magnetic field is weak in the MDCs and the two space points lie close together the trajectory can be approximated as a straight line in the vicinity of the drift cell. The container *MdcCal2Par* uses the drift time and the expected impact angle to calculate the minimum distance error.

<sup>2</sup>The GEANT [CER93] based simulation package used in the HADES software HYDRA.

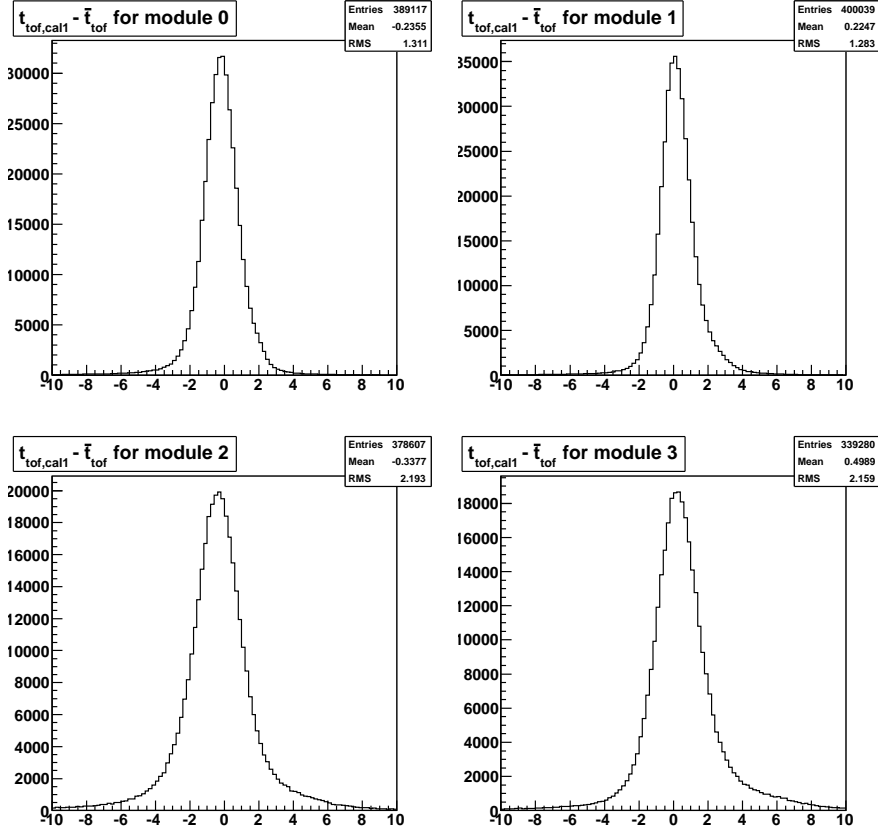


Figure 6.5: The plots show the difference of the time-of-flight obtained from the simulation data  $t_{tof,cal1}$  and the weighted mean value for the time-of-flight  $\bar{t}_{tof}$ . The RMS values are used as errors for the time-of-flight corrections.

### Projection Function for Single Wire Measurements

A projection function is needed that extracts a measurement vector from the track state  $\mathbf{x} = (x, y, t_x, t_y, q/p)$ . A vector is transformed from the sector coordinate system into the wire coordinate system of a drift chamber by a translation and rotation. Let  $R$  be the matrix that describes this transformation,  $u$  the coordinate of the track state on the wire,  $v$  the coordinate on the wire plane of the drift chamber,  $w$  the coordinate perpendicular to the wire plane (see fig. 6.7) and  $\theta = 90^\circ - \alpha$  where  $\alpha$  is the impact angle of the track as defined in figure 6.4. The angle  $\tan \theta = p_v/p_w$  is obtained by transforming the non-normalized direction vector  $(t_x, t_y, 1)$  into the wire coordinate system:

$$\tan \theta(t_x, t_y) = R_{21} \cdot t_x + R_{22} \cdot t_y + R_{23}. \quad (6.26)$$

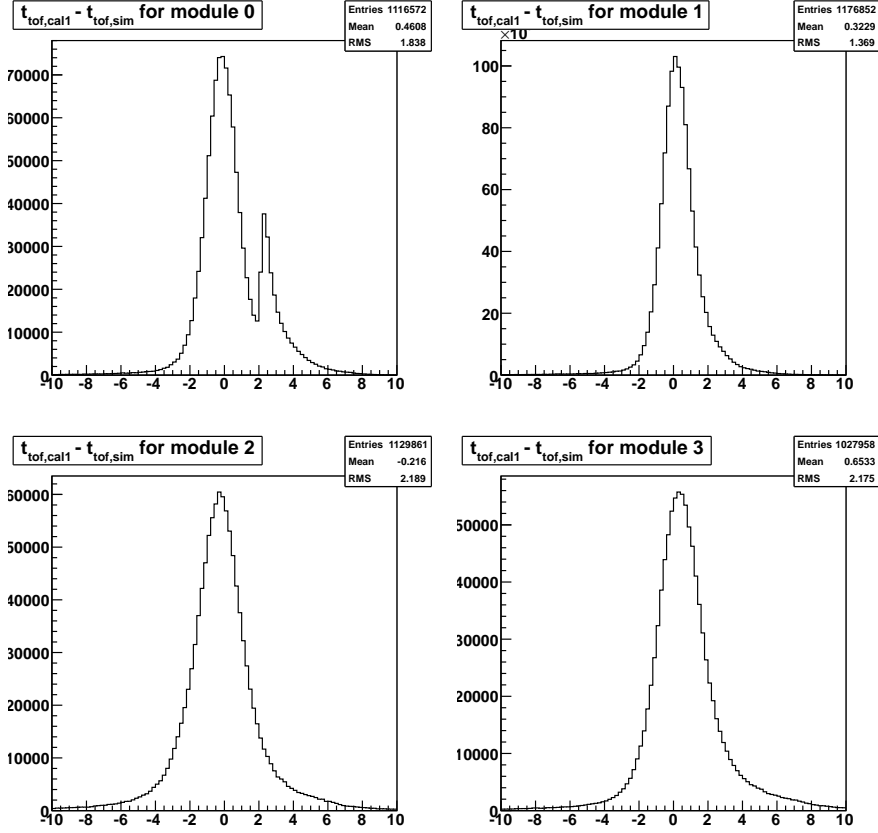


Figure 6.6: The error for time-of-flight correction if the individual values for each layer are taken instead of the weighted mean in figure 6.6.

The projection function that computes the minimum distance of the track to the wire is

$$\mathbf{h}(x) = (R_{21} \cdot x + R_{22} \cdot y + R_{23} \cdot z(x, y)) \cdot \cos \theta(t_x, t_y) \quad (6.27)$$

$$= (R_{21} \cdot x + R_{22} \cdot y + R_{23} \cdot z(x, y)) \cdot \frac{1}{\sqrt{1 + (\tan \theta(t_x, t_y))^2}}. \quad (6.28)$$

Since track states are always defined on a plane the  $z$ -coordinate is dependent on the  $x$  and  $y$  positions. With this the projector matrix

$$H_k = \left. \frac{\partial \mathbf{h}_k}{\partial \mathbf{x}} \right|_{\mathbf{x}_k}$$

can be calculated. The minimum distance still carries an ambiguity on which side of the sense wire the particle passed by. If this is ignored and the exp-

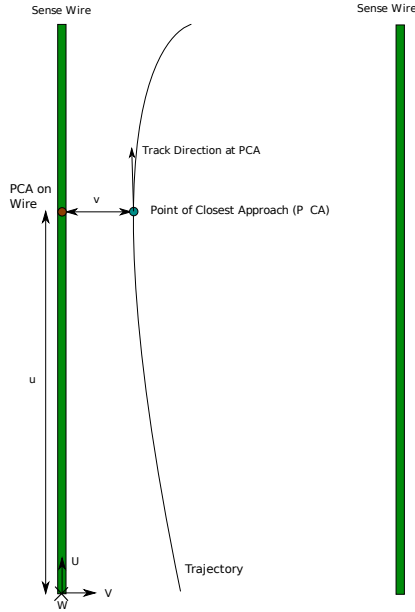


Figure 6.7: The wire coordinate system.

tected and true track state lie on opposite sides of the wire then this may pull the estimate into the wrong direction [Man95]. This can be resolved with the Deterministic Annealing Filter by treating the two possibilities as competing measurements.

When there are competing wire measurements from different cells at the same drift chamber layer then a common reference frame for all measurements is needed.

### Projection Function for Competing Wire Measurements

The reconstructed minimum distance is passed to the Kalman filter as the measurement vector. Because particles may have traversed two neighbouring drift cells there can be up to two measurements in an MDC layer (fig. 6.8). The competing measurements are combined into an effective measurement  $\bar{\mathbf{m}}_k$ . To calculate the innovation  $\mathbf{h}_k(\bar{\mathbf{x}}_k) - \bar{\mathbf{m}}_k$  the projection function is needed  $\mathbf{h}_k(\bar{\mathbf{x}}_k)$ .

The two measurements need to be transformed in a common reference frame that still allows for a simple projection function. The DAF propagates the trajectory to the MDC layer. A virtual layer is created that serves as the common reference frame. One of the plane's axis ( $\mathbf{U}$ ) points along the wire, the other one ( $\mathbf{V}$ ) from the point of closest approach (PCA) on the wire to the PCA on the trajectory (see fig 6.7). The expected track position is the origin of the plane's coordinate system.

Let  $\mathbf{p}_0$  be the expected position of the track. A direction vector  $\mathbf{t}_0$  can be

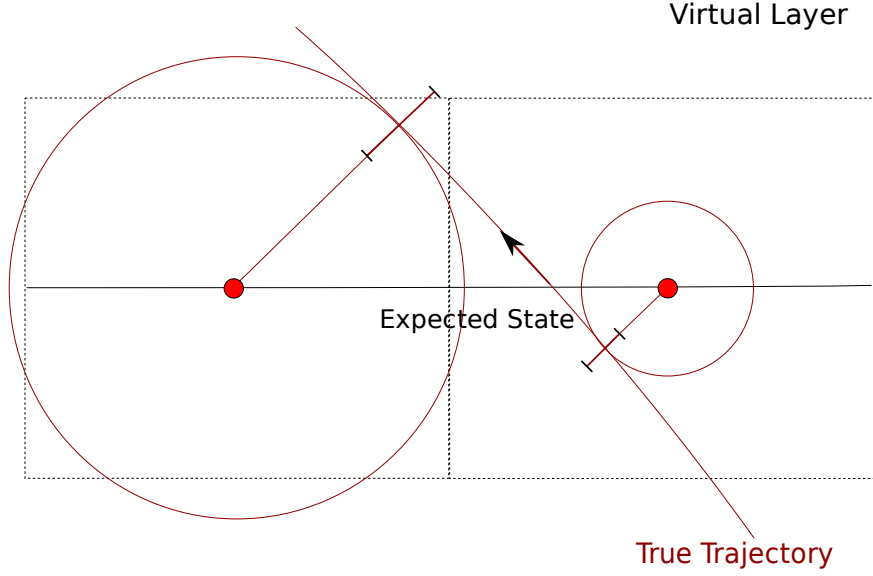


Figure 6.8: A particle passed through two neighbouring drift cells.

obtained from the track state parameters  $t_x$  and  $t_y$ :

$$\mathbf{t}_0 = \frac{1}{\sqrt{t_x^2 + t_y^2 + 1}} \begin{pmatrix} t_x \\ t_y \\ 1 \end{pmatrix}. \quad (6.29)$$

The trajectory is approximated as a straight line in the vicinity of the MDC layer:

$$\mathbf{p}(t) = \mathbf{p}_0 + t \cdot \mathbf{t}_0. \quad (6.30)$$

The wire end points  $\mathbf{w}_0$  and  $\mathbf{w}_1$  define a line segment:

$$\mathbf{w}(s) = \mathbf{w}_0 + s \cdot \mathbf{s}_0 \quad \text{with } \mathbf{s}_0 \equiv \mathbf{w}_1 - \mathbf{w}_0 \text{ and } s \in [0, \dots, 1]. \quad (6.31)$$

If there are two competing hits in an MDC layer it doesn't matter which of the two sense wires that generated a signal is used because all sense wires of a layer are parallel. We look for the parameters  $t_c$  and  $s_c$  so that the distance of the trajectory to the wire  $\|\mathbf{p}(t_c) - \mathbf{w}(s_c)\|$  is minimal. The PCA of the track to the sense wire then is  $\mathbf{p}(t_c)$  and the PCA on the wire is  $\mathbf{w}(s_c)$ . The solutions are [Ebe01]:

$$s_c = \frac{be - cd}{ac - b^2} \quad (6.32)$$

$$t_c = \frac{ae - bd}{ac - b^2} \quad (6.33)$$

with

$$a = \mathbf{s}_0 \cdot \mathbf{s}_0, b = \mathbf{s}_0 \cdot \mathbf{t}_0, c = \mathbf{t}_0 \cdot \mathbf{t}_0, d = \mathbf{s}_0 \cdot (\mathbf{w}_0 - \mathbf{p}_0) \text{ and } e = \mathbf{t}_0 \cdot (\mathbf{w}_0 - \mathbf{p}_0).$$

If  $s_c$  is smaller than zero or greater than one then the PCA is outside of the wire. If  $ac - b^2$  is equal zero then the trajectory and wire are parallel.

The axis ( $\mathbf{U}$ ) of the virtual plane is a unit vector along the direction of the wire

$$\mathbf{U} = \frac{1}{\|\mathbf{w}_1 - \mathbf{w}_0\|} \cdot (\mathbf{w}_1 - \mathbf{w}_0) \quad (6.34)$$

and the second axis  $\mathbf{U}$  points from the PCA on the wire to the PCA on the track

$$\mathbf{V} = \frac{1}{\|\mathbf{p}(t_c) - \mathbf{w}(s_c)\|} (\mathbf{p}(t_c) - \mathbf{w}(s_c)). \quad (6.35)$$

The origin  $\mathbf{O}$  of the virtual layer's coordinate system is the predicted track position (fig. 6.9). The resulting virtual layer is perpendicular on the predicted track state and the plane with the sense wires.

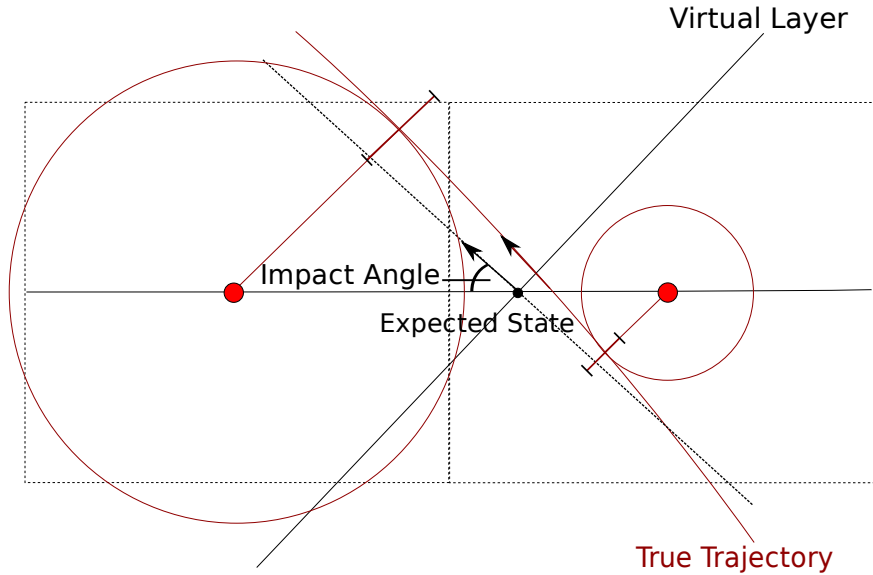


Figure 6.9: A virtual layer is created that is perpendicular on the predicted track state and the sense wires.

The minimum distances are projected onto the virtual layer (fig. 6.10). Then the projected measurements are converted in regard to the centre of the virtual

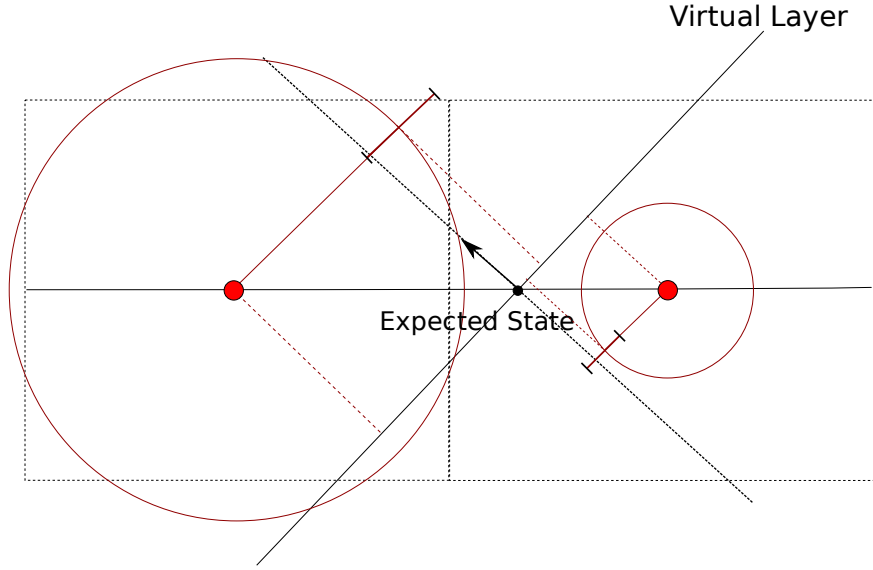


Figure 6.10: The minimum distances are projected onto the virtual layer.

layer (fig. 6.11). Finally the effective measurement and measurement error are calculated with formulas 4.31 and 4.32.

To easily calculate the innovation the track state vector is transformed into the virtual layer coordinate system as well:

$$\underbrace{\mathbf{x} = \left( x, y, t_x, t_y, \frac{q}{p} \right)^T}_{\text{state in sector coordinates}} \rightarrow \underbrace{\mathbf{x}' = \left( u, v, t_u, t_v, \frac{q}{p} \right)^T}_{\text{state in virt. layer coordinates}} .$$

Let  $\mathbf{W}$  be the normal vector of the virtual plane. The transformed state parameters then are:

$$\begin{aligned} u &= (\mathbf{p}_0 - \mathbf{O}) \cdot \mathbf{U} \\ v &= (\mathbf{p}_0 - \mathbf{O}) \cdot \mathbf{V} \\ t_u &= \frac{\mathbf{t}_0 \cdot \mathbf{U}}{\mathbf{t}_0 \cdot \mathbf{W}} \\ t_v &= \frac{\mathbf{t}_0 \cdot \mathbf{V}}{\mathbf{t}_0 \cdot \mathbf{W}} . \end{aligned} \quad (6.36)$$

In this new coordinate system the parameter  $u$  corresponds to the position on the sense wire and  $v$  to the minimum distance to the wire. The projection



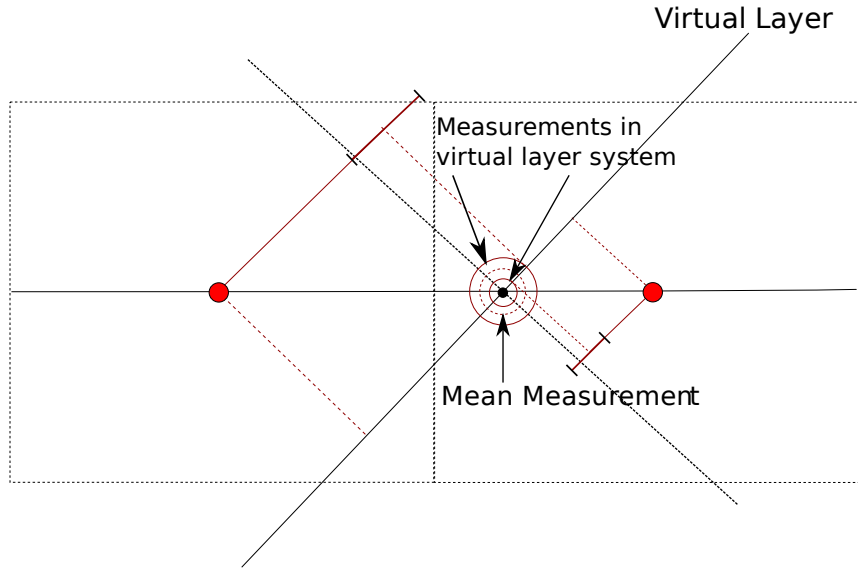


Figure 6.11: The projections of the minimum distances are transformed into a common coordinate system.

function  $\mathbf{h}$  and projector matrix  $H$  assume the simple forms

$$\mathbf{h}(\mathbf{x}') = v \quad (6.37)$$

$$H = (0 \ 1 \ 0 \ 0 \ 0). \quad (6.38)$$

Since the Kalman gain matrix  $K$  depends on the covariance matrix  $C$  it has to be transformed into the virtual layer coordinate system as well. The transformation matrix is the Jacobian

$$J_{Sec \rightarrow Lay} = \frac{\partial (u, v, t_u, t_v, q/p)^T}{\partial (x, y, t_x, t_y, q/p)^T}. \quad (6.39)$$

Then the covariance in the new coordinate system is [EPIM97]

$$C' = J_{Sec \rightarrow Lay} \cdot C \cdot J_{Sec \rightarrow Lay}^T. \quad (6.40)$$

The inverse transformation back to the sector coordinate system is done via

$$\begin{aligned} \begin{pmatrix} x \\ y \\ z \end{pmatrix} &= \mathbf{O} + u \cdot \mathbf{U} + v \cdot \mathbf{V} \\ t_x &= \frac{t'_x}{t'_z} \\ t_y &= \frac{t'_y}{t'_z} \end{aligned} \tag{6.41}$$

with the direction vector

$$\mathbf{t}' = \frac{1}{\|t_u \cdot \mathbf{U} + t_v \cdot \mathbf{V} + \mathbf{W}\|} (t_u \cdot \mathbf{U} + t_v \cdot \mathbf{V} + \mathbf{W}).$$

The transformation of the covariance matrix is carried out with the Jacobian

$$J_{Lay \rightarrow Sec} = \frac{\partial (x, y, t_x, t_y, q/p)^T}{\partial (u, v, t_u, t_v, q/p)^T}. \tag{6.42}$$

## 6.4 Energy Loss and Multiple Scattering

Interactions between the particles and detector materials can randomly perturb the particle's trajectory. This results in a reduction of the particle's energy. Multiple scattering can also cause a change in direction. Energy loss and multiple scattering are taken into account by the Kalman filter. During the propagation of the particle the properties of the traversed materials are calculated after each Runge-Kutta step. Then the particle momentum and the process noise matrix can be updated.

### Material Budget of the MDC's

Taking multiple scattering and energy loss into account requires knowledge about the properties of the passed materials. An MDC module consists of the following components in the active area:

- 12  $\mu m$  wide, aluminized mylar foils that form the inlet and outlet windows of an MDC.
- Cathode and field wires forming the boundary of the drift cell.
- A sense wire in the centre of each cell.
- A mixture of a counter and quencher gas. MDC plane I is filled with a 70 : 30 mixture of Argon and carbon dioxide. The other planes contain 84% Argon and 16% iso-Butane.

	Module	Material	Diameter [ $\mu m$ ]
Cathode wire	MDC I	Copper/Berillium 98%/2%	76
	MDC II-III	Aluminum	80
	MDC II-III	Aluminum	100
Field wire	MDC I-II	Aluminum	80
	MDC III-IV	Aluminum	100
Sense wire	MDC I-III	Gold plated tungsten	20
	MDC IV	Gold plated tungsten	30

Table 6.1: Materials and diameters of the different wire types [Mün10].

The materials and diameters of the wires are listed in table 6.1. The dimension of a drift cell and the distances (pitch) between the wires can be found in figure 6.12.

It is not useful to model single wires in the Kalman filter due to precision and performance reasons. Instead, a material mixture is calculated by comparing the volume occupancies of the various components in a drift cell [Mün10]. If the component  $i$  occupies the volume  $v_i$  and has the density  $\rho_i$  then its proportion by weight  $p_i$  is:

$$p_i = \frac{m_i}{\sum_i m_i} = \frac{v_i \rho_i}{\sum_i v_i \rho_i} \quad (6.43)$$

To get the effective atomic mass  $A_{eff}$ , atomic number  $Z_{eff}$  and radiation length  $X_0$  of a mixture the properties of the components can be added [CER93, chapter CONS110]:

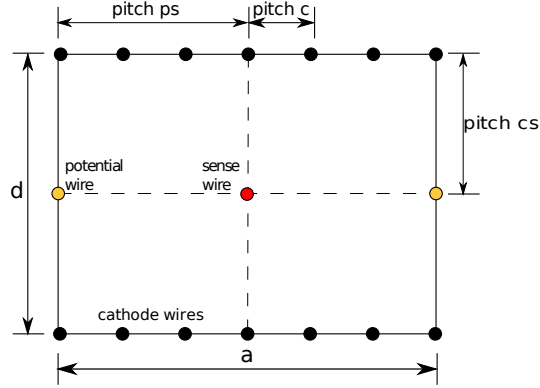
$$A_{eff} = \sum_i p_i \cdot A_i \quad (6.44)$$

$$Z_{eff} = \sum_i p_i \cdot Z_i \quad (6.45)$$

$$\frac{1}{\rho X_0} = \sum_i \frac{p_i}{\rho_i X_{0,i}} \quad (6.46)$$

Similarly, the density of a mixture is:

$$\begin{aligned} \frac{1}{\rho} &= \frac{V}{M} \\ &= \sum_i \frac{V_i}{M} \\ &= \sum_i \frac{V_i}{m_i} \frac{m_i}{M} \\ &= \frac{p_i}{\rho_i}. \end{aligned} \quad (6.47)$$



Module	pitch ps [mm]	pitch cs [mm]	pitch c [mm]	drift cell area $a \times d$ [mm $\times$ mm]
I	2.5	2.5	2	$5 \times 5$
II	2.5	3	2	$5 \times 6$
III	4	6	2	$8 \times 12$
IV	5	7	4	$10 \times 14$

Figure 6.12: The dimensions of a drift cell in the different MDC modules [Mar05].

Module	A	Z	$\rho$ [g/cm <sup>3</sup> ]	$X_0$ [cm]	I [eV]
I	38.87	17.52	$3.47012 \cdot 10^{-3}$	5836.09	172.24
II	41.07	19.02	$3.5668 \cdot 10^{-3}$	5767.87	156.01
III	31.85	15.18	$6.74459 \cdot 10^{-3}$	3402.59	157.47
IV	36.65	17.25	$3.90731 \cdot 10^{-3}$	5583.12	153.00

Table 6.2: Effective material properties of the drift chambers.

The mean excitation energy  $I$  of a compound can be approximated by another additivity rule [SB82]:

$$I = \exp\left(\frac{\sum_i p_i \frac{Z_i}{A_i} \cdot \ln I_i}{\frac{Z}{A}}\right) \quad (6.48)$$

with

$$\frac{Z}{A} = \sum_i \frac{Z_i}{A_i}. \quad (6.49)$$

The effective properties in the MDCs calculated with the above formulas are listed in Table 6.2. Figure 6.13 shows the relative contribution of the different MDC components to the effective radiation length.

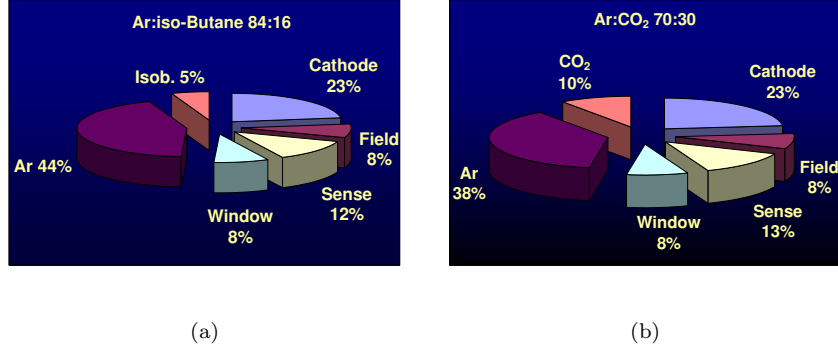


Figure 6.13: Relative contributions of the different materials in MDC I to the effective radiation length for (a) Argon/iso-Butane (84 %/16 %) and (b) Argon/CO<sub>2</sub> (70 %/30 %) gas mixtures [Mün10].

### Multiple Scattering

Particles with a charge  $z$  can undergo elastic scattering in the Coulomb field of the nuclei of the detector material. For hadronic particles the strong interaction also contributes. Multiple scattering randomly changes the direction of the particle. Its momentum  $p$  remains unaffected since the nuclei are much more massive. The width of the deflection angle due to multiple scattering is [N<sup>+</sup>10]

$$\Theta_{MS} = \frac{13.6 \text{ MeV}}{\beta pc} z \sqrt{t} (1 + 0.0038 \ln t) \quad (6.50)$$

and the variance is [Man04]

$$C_{MS} = \left( \frac{13.6 \text{ MeV}}{\beta pc} \right)^2 t (1 + 0.0038 \ln t)^2 \quad (6.51)$$

where  $t = l/X_0$  is the track length  $l$  inside the medium in units of radiation lengths  $X_0$ .

### Energy Loss due to Ionization

When traversing the detector material charged particles interact with the electrons of the nuclei. The nuclei are excited or ionized and the particle loses energy in the process. The mean energy loss for a spinless particle with charge  $z$  is described by the Bethe-Bloch formula [N<sup>+</sup>10]:

$$-\left\langle \frac{dE}{dx} \right\rangle = K z^2 \frac{Z}{A} \frac{1}{\beta^2} \left( \frac{1}{2} \ln \left( \frac{2m_e c^2 \beta^2 \gamma^2 T_{max}}{I^2} \right) - \beta^2 \right) \quad (6.52)$$

$$K = 4\pi N_{Ar} r_e^2 m_e c^2 = 0.307075 \frac{\text{MeV} \cdot \text{cm}^2}{g} \quad (6.53)$$

Here  $T_{max}$  is the maximum kinetic energy which a particle with mass  $M$  may be transfer onto an electron:

$$T_{max} = \frac{2m_e c^2 \beta^2 \gamma^2}{1 + 2\gamma \frac{m_e}{M} + \left(\frac{m_e}{M}\right)^2}. \quad (6.54)$$

The formulas for electrons and positrons differ because of spin, kinematics and identity of the incident electron with the electrons which it ionizes. For relativistic electrons ( $z = 1, \beta \rightarrow 1$ ) the mean ionization loss is [SRF94]

$$-\left\langle \frac{dE}{dx} \right\rangle = \frac{1}{2} K \frac{Z}{A} \left( 2 \ln \frac{2m_e c^2}{I} + 3 \ln \gamma - 1.95 \right). \quad (6.55)$$

The corresponding formula for relativistic positrons is slightly different due to the different free electron cross section that must be applied in the calculation:

$$-\left\langle \frac{dE}{dx} \right\rangle = \frac{1}{2} K \frac{Z}{A} \left( 2 \ln \frac{2m_e c^2}{I} + 4 \ln \gamma - 2 \right). \quad (6.56)$$

The above formulas are valid for moderate energies with  $0.1 \lesssim \beta\gamma \lesssim 1000$  in intermediate-Z materials. For protons this corresponds to momenta roughly between  $94 \text{ MeV}/c$  and  $30,000 \text{ MeV}/c$ . This is within the momenta range which can be expected for the hadronic background in HADES. In the case of electrons or positrons however, radiation loss is more important than ionization loss.

### Energy Loss due to Radiation

Figure 6.14 shows the different contributions of energy loss for electrons in lead depending on the particle energy. Radiation loss already starts to become dominant after a few tens of MeV.

Energy loss of electrons and positrons due to bremsstrahlung is described by the Bethe-Heitler equation [Man04]:

$$\left\langle \frac{dE}{dx} \right\rangle = -\frac{E}{X_0}. \quad (6.57)$$

Integrating yields the energy  $E'$  that a particle with initial energy  $E_0$  has after traversing the distance  $t = l/X_0$  in the material:

$$\left\langle \frac{E'}{E_0} \right\rangle = e^{-t}. \quad (6.58)$$

As can be seen radiation loss is proportional to the particle energy while ionization loss only rises logarithmically. The critical energy  $E_c$  is sometimes defined as the energy where both loss rates are equal. For gases this is [N<sup>+</sup>10]:

$$E_c = \frac{710 \text{ MeV}}{Z + 0.92}. \quad (6.59)$$

For example, in the case of MDC module I with an effective atomic number  $Z \approx 17.5$  the critical energy would be around  $38 \text{ MeV}$ .

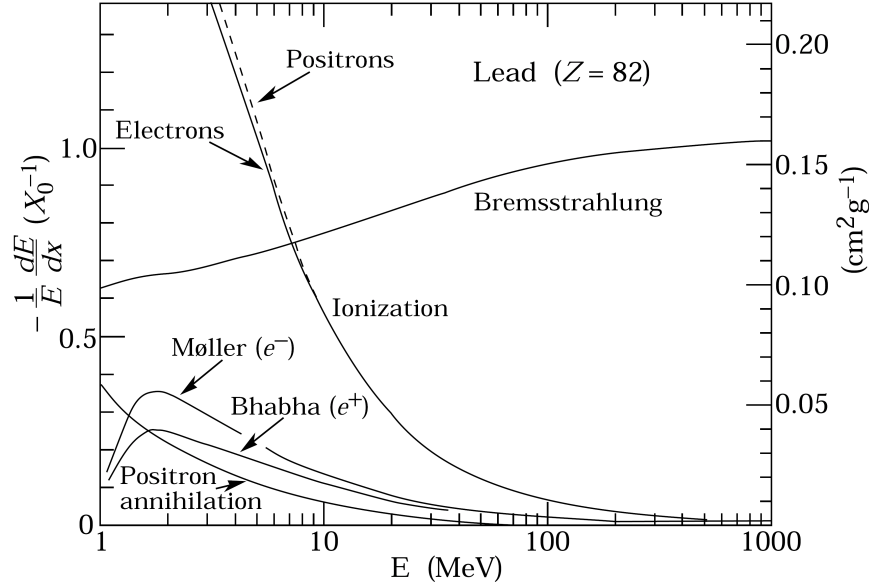


Figure 6.14: The energy loss per radiation length as a function of energy for electrons and positrons in lead [N+10]. Radiation loss starts to dominate after a few tens of MeV.

### Integration into the Kalman Filter

After each Runge-Kutta step the properties of the traversed materials are calculated. Inside an MDC chamber a material mixture is used as described in 6.4, outside the chamber it is air. A hypothesis for the particle type is needed, especially the distinction between electrons/positrons and hadrons is necessary. The energy of the particle and therefore the track parameter  $q/p$  is corrected by the mean energy loss.

The variances of multiple scattering and energy loss are added to the process noise matrix  $Q$ . The contribution of multiple scattering for thin scatterers to the process noise matrix  $Q$  is [Man04]:

$$\begin{aligned}
 Q(t_x, t_x) &= (1 + t_x^2) \cdot (1 + t_x^2 + t_y^2) \cdot C_{MS} \\
 Q(t_y, t_y) &= (1 + t_y^2) \cdot (1 + t_x^2 + t_y^2) \cdot C_{MS} \\
 Q(t_x, t_y) &= Q(t_y, t_x) = t_x t_y \cdot (1 + t_x^2 + t_y^2) \cdot C_{MS}.
 \end{aligned}
 \tag{6.60}$$

Ionization loss is mostly relevant at low energies where the resolution is mainly affected by multiple scattering [Man99]. Therefore its effects on the momentum uncertainty is neglected. Radiation loss affects the momentum estimation

error [Man04, SRF94]:

$$Q\left(\frac{q}{p}, \frac{q}{p}\right) = \left(\frac{q}{p}\right)^2 \left(e^{-t\frac{\ln 3}{\ln 2}} - e^{-2t}\right) \quad (6.61)$$



## Chapter 7

# Implementation in HYDRA

### 7.1 The HYDRA Framework for Track Reconstruction

The software framework for the analysis in the HADES experiment is called HYDRA [sg11] (**H**ADES **s**Ystem for **D**ata **R**eduction **A**nalysis). The analysis is done per event. An event is the record of all physical interactions in the detector resulting from the reaction between a beam particle and the target. Events can be real or simulated. During the reconstruction process the following steps are executed for each event:

1. Read information about the current event.
2. Execute the task set for the event.
3. Write event data in a ROOT tree.
4. Clear information from the current event.

A task in an algorithm used in the reconstruction process. Such a task must inherit from the abstract class `HTask`. A task is executed by calling its member function `HTask *next(Int_t &errCode)` which also returns a possible error code and a pointer to the task it is connected to after it is finished. The functions `Bool_t init()` and `Bool_t reinit()` are called before the first call of the `next()` function and each time a new input file is read, respectively. They essentially retrieve pointers to needed parameters and categories needed and perform the specific initialization of the task.

A special task is the task set (class `HTaskSet`) which groups together several tasks. Tasks may arbitrarily be added and connected among each other. For each event the global `Hades` class executes such a task set that is set up by the user. This allows for a flexible system to choose the algorithms and their execution sequence during the track reconstruction.

The `HReconstructor` class is another special task representing an algorithm that transforms data. It provides the pure virtual `Int_t execute()` function that has to be implemented in a derived class.

## 7.2 The Classes in the Kalman Filter Library

For the integration into the HYDRA framework a task for the Kalman filter is needed. The `HKalFilterTask` class is derived from `HReconstructor` and fulfills this purpose. The class layout of the Kalman filter library is shown as UML diagrams in 7.1 and 7.2.

The `HKalFilterTask` has two important data members of classes `HKalInput` and `HKalSystem`. The `HKalInput` class looks for the next track candidate in the current event by looping over META-detector matches and extracts the measurement information. The Kalman filter can process time measurements from drift chambers or reconstructed segment hits as “measurements”. The input generates objects of class `HKalMdcHit` that store a measurement vector along with its error. When the Kalman filter works with reconstructed segments then the measurement vector contains the x- and y-positions of a segment hit in the sector coordinate system. For wire hits the drift time and error are stored. The drift time is later converted to a minimum drift distance (see chapter 6.3.2) which is then treated as the “measurement” by the Kalman filter.

Hits also have a pointer to the measurement layer where the hit has been registered. The class `HKalMdcMeasLayer` contains information about the geometry and material of an MDC related detector component. The measurement layers are modelled as planes with a certain thickness. The plane either represents the sense wire layer of an MDC or the virtual segment plane. The measurement layers themselves are stored in the class `HKalDetCradle` that serves as a container.

`HKalFilterTask` uses the spline fit and segment points to generate an estimate for the initial track state. The main Kalman filter and DAF algorithms are implemented in the `HKalSystem` class. Only the track propagation is done by a separate class `HKalRungeKutta` that needs access to the HADES field map (`HMdcTrackGField`). In the future, part of the algorithms collected in `HKalSystem` will be distributed over several separate classes to increase maintainability and make this class less prone to programming errors.

Figure 7.2 shows the class layout of `HKalSystem` that processes the measurements retrieved from the `HKalInput` class. For each observation point a `HKalTrackSite` is created that stores information about the results of the Kalman filter and the measurements. MDC measurement information is stored in `HKalMdcHit` just as they have been generated by the input class. A site may contain several hits in the case of competing measurements. The class `HKalTrackState` stores the results of the Kalman filter: the state vectors, covariance, process noise and propagation matrices. Every site has a track state for each of the various Kalman filter steps: the prediction, filtering, smoothing and possibly inverse filtering.

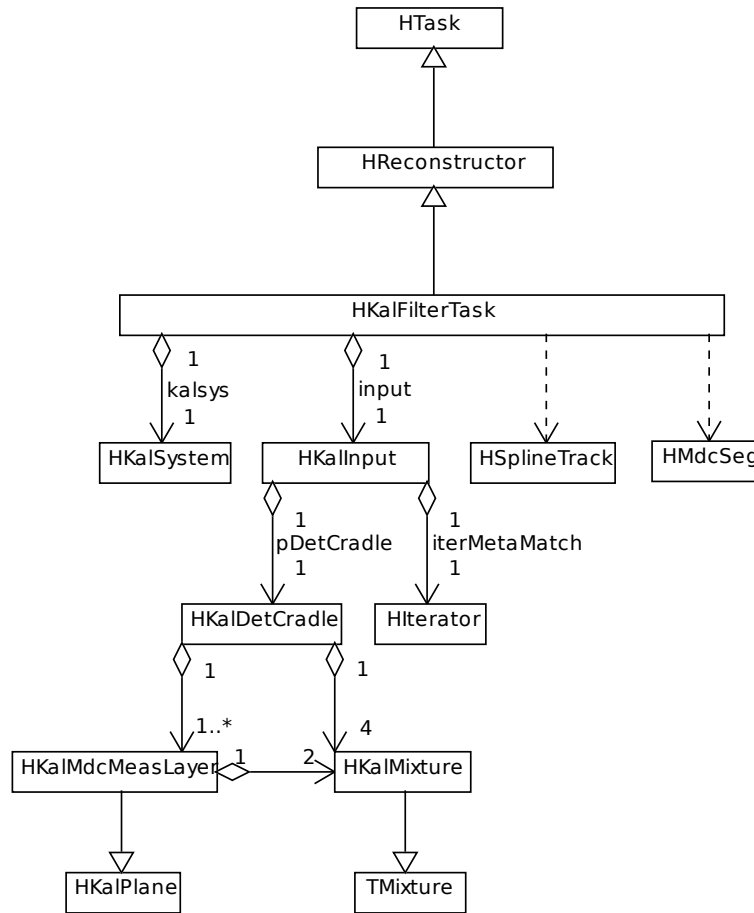


Figure 7.1: Class diagram for the Kalman filter library. The structure of the HKaSystem class is shown separately in figure 7.2.

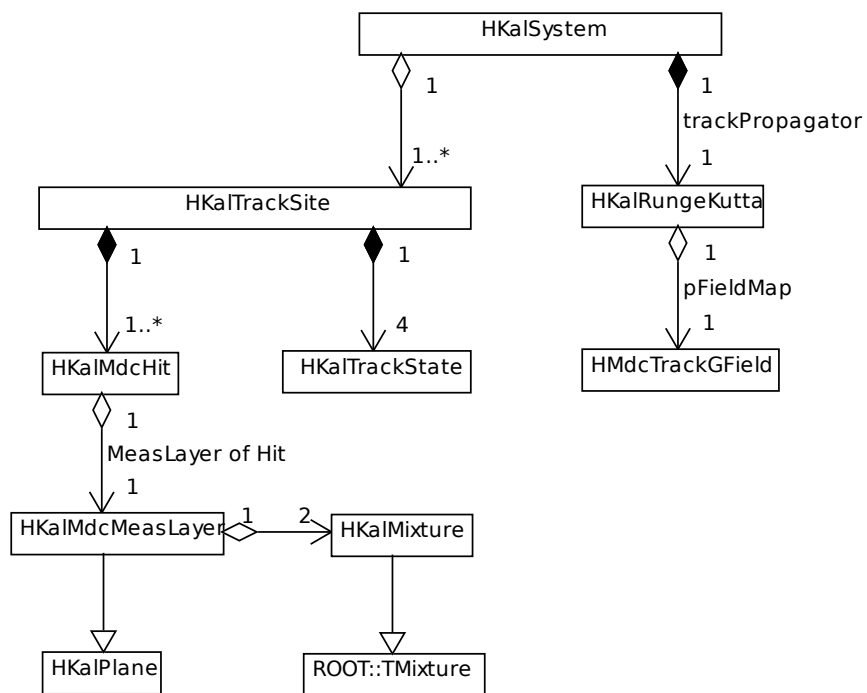


Figure 7.2: The components of the HKalSystem class.

## 7.3 Using the Kalman Filter Library

### 7.3.1 Initialization

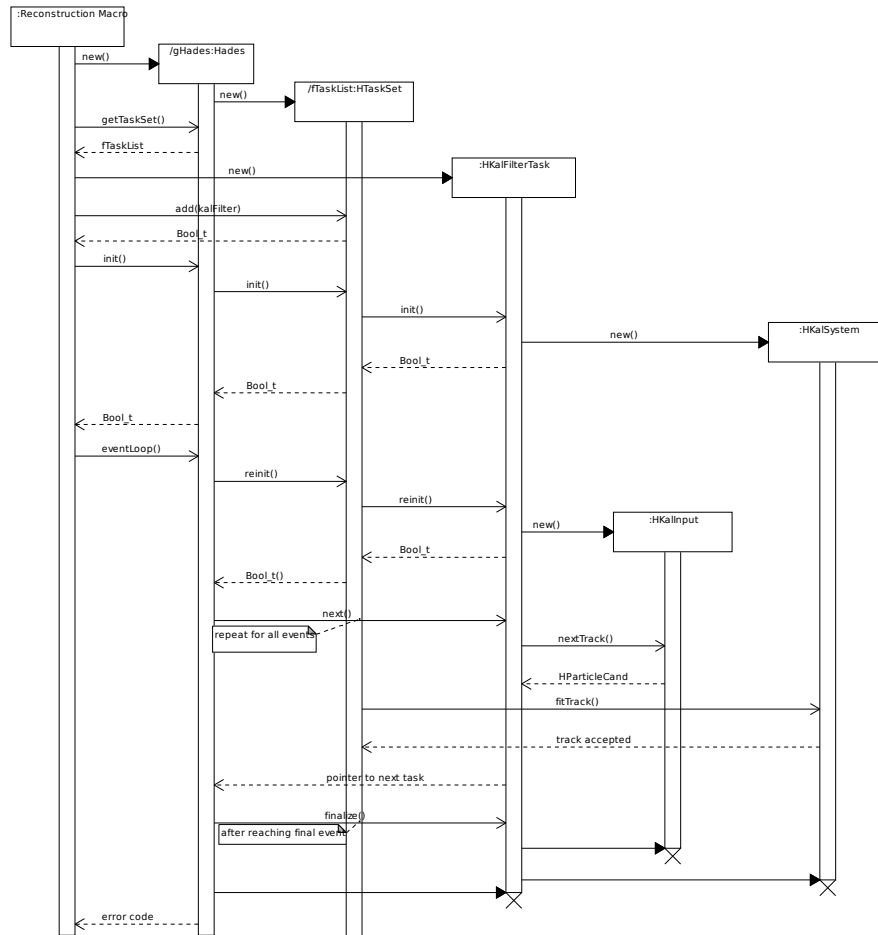


Figure 7.3: The message exchange between the participating classes during a typical track reconstruction procedure is shown in this sequence diagram.

Of the classes in the Kalman library only `HKalFilterTask` needs to be accessed directly by the user. Figure 7.3 shows the interactions relevant to the Kalman filter. First the global `Hades` object has to be created. The necessary initializations include setting up the spectrometer, runtime database, parameter sources and task sets. An example program for this procedure can be found in the HYDRA manual [sg11, chap. 2.6]. In order to add the Kalman filter a task set needs to be created and added to the task list of the global `Hades`

object. What needs to be changed in the example program is to replace the Runge-Kutta task with the Kalman task:

```

Bool_t createHades(Int_t datasource ,
    TString inputDir ,
    TString inputFile ,
    Int_t refId ,
    TString eventbuilder ,
    TString paramSource ,
    TString asciiParFile ,
    TString rootParFile) {
...

    // Create spline task set. No need to add the Runge-Kutta task
    here anymore.
    HSplineTaskSet *splineTaskSet = new HSplineTaskSet("","");
    HTask *splineTasks = splineTaskSet->make("","spline");

    // Create Kalman filter task.
    // Input is simulation or real data.
    Bool_t sim = kTRUE;
    // kTRUE: filter wire hits.
    // kFALSE: filter segment hits.
    Bool_t fitWires = kTRUE;
    HKalFilterTask *kalFilter = new HKalFilterTask(sim, fitWireHits ,
        refId);

    // Add to the task set of the global HADES object.
    HTaskSet *masterTaskSet = gHades->getTaskSet("simulation");
    masterTaskSet->add(splineTasks);
    masterTaskSet->add(kalFilter);

...

```

After the initialization is finished and `gHades->eventLoop()` has been called in the main program the Kalman filter will be used as a reconstructor task instead of the Runge-Kutta algorithm.

This example macro demonstrates how to read the output from the Kalman filter and fill some simple histograms:

```

// Example macro that reads output of the Kalman filter task
// and fills some simple histograms.

void histKalman(TString dataFiles = "reco*.root") {

    // HLoop is a helper class to allow for fast looping
    // of HADES dsts.
    // kTRUE: create Hades (needed to work with standard
    // eventstructure)
    HLoop *loop = new HLoop(kTRUE);
    loop->addFiles(dataFiles);

    // Read categories relevant to the Kalman filter.
    if(!loop->setInput("-*"+HKalTrack,+HKalSite,+HKalHitWire,+
        HKalHit2d")) {
        cout<<"Could not retrieve pointer to categories."<<endl;
        exit(1);
    }
}

```

```

}

// Make an iterator to loop over HKalTrack objects.
TIterator *iterCand = 0;
if (loop->getCategory("HKalTrack"))
    iterCand = loop->getCategory("HKalTrack")->MakeIterator();

// Get category pointers.
HCategory* catCand =
    HCategoryManager::getCategory(catKalTrack);
if(!catCand) {
    cout<<"Category catKalTrack not found."<<endl;
    exit(1);
}

HCategory *catCandSite =
    HCategoryManager::getCategory(catKalSite);
// Correct hit category is determined later.
HCategory *catCandHit = NULL;

// Make some example histograms.
// -----
TH2F *hPhiTheta = new TH2F("hphitheta",
    "#Theta : #Phi for not reconstructed
    tracks",
    60, 60.F, 120.F, 90, 0.F, 90.F);
hPhiTheta->GetXaxis()->SetTitle("#Phi [#circ]");
hPhiTheta->GetYaxis()->SetTitle("#Theta [#circ]");

TH2F *hchimom = new TH2F("hchimom", "#chi^{2} : p",
    100, 0.F, 2000.F, 60, 0.F, 100.F);
hchimom->GetXaxis()->SetTitle("p [MeV/c]");
hchimom->GetYaxis()->SetTitle("#chi^{2}");

// Loop over events.
// -----
Int_t entries = loop->getEntries();

for (Int_t i = 0; i < entries; i++) {
    // Get next event. Categories will be cleared before.
    nbytes = loop->nextEvent(i);
    if(nbytes <= 0) {
        cout<<"Error reading next event. nbytes = "
            <<nbytes<<endl;
        break;
    } // last event reached

    // Loop over track candidates.
    // -----
    iterCand->Reset();
    HKalTrack *track;
    while((track = (HKalTrack*)iterCand->Next()) != 0) {

        // Segment and wire hits have different categories.
        if(track->getIsSegHit()) {
            catCandHit = HCategoryManager::getCategory(
                catKalHit2d);
        }
    }
}

```

```

    } else {
        catCandHit = HCategoryManager::getCategory(
            catKalHitWire);
    }

    // Get indices of the first and last measurement sites.
    Int_t fromSite = track->getIdxFirst();
    Int_t toSite   = track->getIdxLast();

    if(fromSite < 0) {
        cout<<"Index to start site is not set. Skipping
            track."<<endl;
        continue;
    }

    // Reconstruction has been successful or unsuccessful.
    Bool_t trackAccepted = track->getAccepted();

    // Retrieve data and fill histograms.
    // -----
    if(trackAccepted) {
        Float_t chi2   = track->getChi2();
        Float_t ndf    = track->getNdf();
        // Reconstructed momentum.
        Float_t momReco = firstSite->getMomSmoo();

        hChi2Mom->Fill(chi2 / ndf, momReco);
    } else {
        Float_t phi    = track->getPhi();
        Float_t theta  = track->getTheta();

        hPhiTheta->Fill(phi, theta);
    }
    } // loop over tracks
} // loop over events

// Draw/save histograms.
...
}

```

The Kalman filter uses a default setting for its parameters. The default values may be modified if so desired. Information about the settings for the Kalman filter and how to retrieve the reconstruction results can be found in appendix A.



# Chapter 8

## Results

The Kalman filter has been tested with simulated Au+Au reactions at energies of  $1.25\text{A GeV}$ . Events were generated by UrQMD [UrQ04] which served as input for the HADES detector simulation package based on GEANT [CER93]. To increase statistics one additional electron and positron was generated for each sector in all events. These additional leptons are uniformly distributed in polar angles  $\Theta$  and momenta ranging from  $100\text{ MeV}/c$  to  $2000\text{ MeV}/c$ . The evaluation of the simulated data is carried out with the same procedure for track candidate search and segment reconstruction as with experimental data. The particle momenta and polar angles of the particles are not uniformly distributed (see figure 8.1). The proton sample contains a larger amount of tracks with medium momenta of  $600 - 1200\text{ MeV}/c$  and lower polar angles. The momentum of the leptons is more evenly distributed and more of them are found at high theta angles. Unlike the lepton samples the protons originate from the simulated Au+Au collisions.

An initial estimate for the track state vector  $\mathbf{x} = \{x, y, t_x, t_y, q/p\}^T$  is needed by the Kalman filter. The position coordinates  $x$  and  $y$  and direction parameters  $t_x$  and  $t_y$  were obtained from the segment fit. The initial momentum is provided by the spline task. The resolution of the segment reconstruction is  $0.2\text{ mm}$  in  $x$ - and  $0.1\text{ mm}$  in  $y$ -direction. The error in track direction due to the position uncertainty of the segment fit and the straight line approximation of the track between the drift chambers is around  $\tan(0.5^\circ)$ . The RMS of the spline task is 5.3% for protons (fig. 8.2). These values were used as the estimation error variances for the initial covariance matrix.

The evaluated particles were protons, positrons and electrons with momenta between  $0.1\text{ GeV}/c$  and  $2.0\text{ GeV}/c$ . Energy loss and multiple scattering is accounted for in the simulation. The momentum resolution is defined as

$$\frac{\Delta p}{p} \equiv \frac{p_{\text{Geant}} - p_{\text{Reconstructed}}}{p_{\text{Geant}}} \cdot 100.$$

The reconstructed momentum of the Kalman filter and DAF is the estimated momentum after smoothing at MDC-I. Because of the highly non-Gaussian

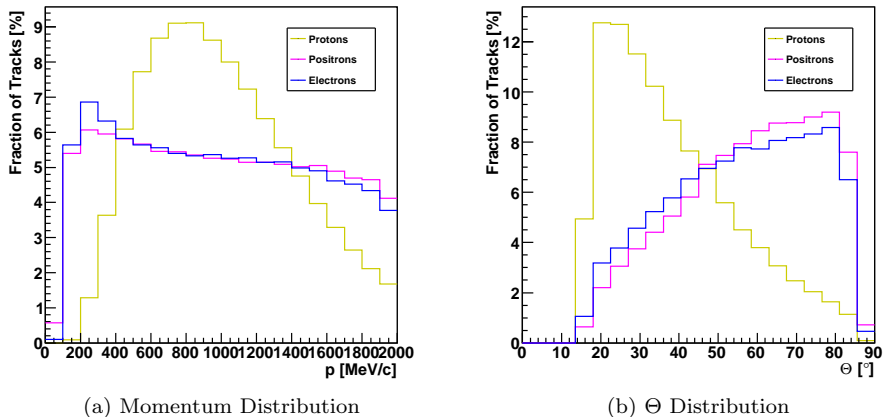


Figure 8.1: The momentum and polar angle distributions of the different particle species in the test sample. Included are only tracks that have been reconstructed by the segment fit.

distributions of the residuals, the momentum resolution will be given as the RMS of the residuals. As the RMS gives a large weight to outliers only values from  $-20\%$  to  $+20\%$  are taken into account.

In general, the efficiency is defined as the number of real reconstructed tracks divided by the number of all accepted real tracks. In this work, only the efficiencies of the Kalman filter and DAF themselves are considered. The previous steps of the reconstruction process are not included in the efficiency.

Track reconstruction is considered as failed if the  $\chi^2$  divided by the number degrees of freedom exceeded a certain threshold or an error was encountered during reconstruction. To allow for a comparison of different reconstruction methods the  $\chi^2$  threshold has been chosen so that the efficiency is around 99%. Since this number is an arbitrary choice the efficiency values are only meaningful relative to each other. It is possible to look for tracks where reconstruction is difficult.

In chapter 8.1 the results for the Kalman filter that fits reconstructed segment points are presented. Chapter 8.2 discusses dependencies on the input and chapter 8.3 the choice of parameter settings for the DAF. In 8.4 the different reconstruction methods are compared. Most of the tests have been done with proton tracks. Results refer to the proton sample unless mentioned otherwise.

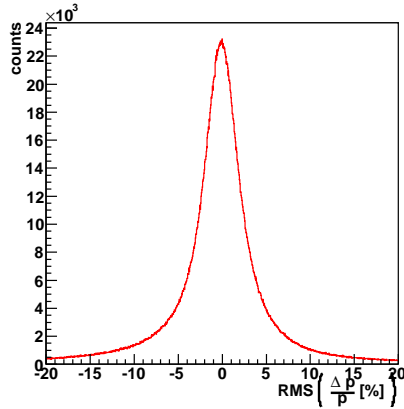


Figure 8.2: Plotted is the momentum resolution of the spline fit for the proton tracks.

## 8.1 Tests with the Kalman Filter

### 8.1.1 Parameters and Settings

The Kalman filter iterates through the measurement points in beam direction from the inner segment to the outer segment. Afterwards, track states are smoothed in the opposite direction. The smoothing is important to get a good estimation at the first MDC chamber and trace the track back to the vertex region. Table 8.1 shows the momentum resolution of the Kalman filter at the last measurement point and the smoothed momentum at the first measurement point. Smoothing could provide a small improvement in the momentum estimate for lepton tracks.

Particle	RMS $\left(\frac{\Delta p}{p} [\%]\right)$	
	Filtered	Smoothed
Positrons	4.52	4.36
Electrons	4.58	4.42
Protons	4.49	4.47

Table 8.1: Comparison of the momentum resolution without and with smoothing.

The initial momentum estimate for the Kalman filter is obtained from the spline fit (see chapter 3). The resolution of the spline fit is already 5.3% which the Kalman filter could improve to 4.5%. The residuals in momenta for both the spline fit and the Kalman filter show a non-gaussian distribution with pronounced tails (fig. 8.3). The tracks with already good starting estimates bene-

fitted the most while there were little improvements in the long tails.

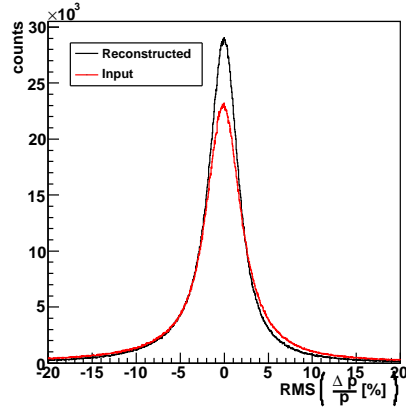


Figure 8.3: The momentum resolution of the Kalman filter (black line) compared to the initial estimate from the spline fit (red line).

The Kalman filter can be run in forward direction from MDC-I to MDC-IV or in backward direction from MDC-IV to MDC-I. The momentum resolution in backward direction decreased to 5.0% (fig. 8.4). When fitting backwards the initial state has to be obtained from the outer segment which has a worse resolution than the inner one resulting in a worse performance than in forward direction.

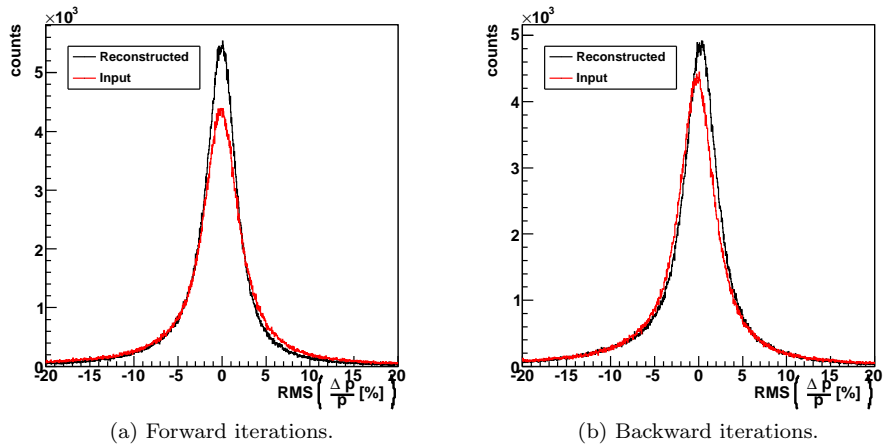


Figure 8.4: A comparison of the momentum resolution for forward and backward iteration.

No. of Iterations	RMS $\left(\frac{\Delta p}{p} [\%]\right)$	Mean $\left(\frac{\Delta p}{p} [\%]\right)$	Efficiency
1	4.47	-0.47	0.9898
2	4.58	-0.30	0.9939
3	4.49	-0.35	0.9928

Table 8.2: The Kalman filter with different amount of iterations.

Another test was to run several iterations of the Kalman filter. The estimate from the previous iteration serves as input for the next one and the measurements are then processed again by the Kalman filter in the opposite direction. Smoothing is only applied after the final iteration. As can be seen in table 8.2 multiple iterations did not improve the momentum estimate.

### 8.1.2 Covariance Matrix and Roundoff Errors

Filling the initial covariance matrix with realistic values that correspond to the estimation errors is beneficial for the Kalman filter performance. Assuming estimation errors that are too large can lead to a divergence of the filter and numerical instability. The variances of the initial estimation errors were set to realistic values and then increased by a factor of 10 and 100, respectively.

Additionally, several different formulations of the Kalman filter equations were compared for possible numerical instabilities. Roundoff errors are a consequence of the finite representation of numbers in computers. If they accumulate this may lead to a degradation of the Kalman filter performance. The covariance matrix was tested for positive definiteness and symmetry. An asymmetric covariance is an indication of numerical degradation. In order to avoid an overly high sensitivity, two off-diagonal elements of the matrix were not counted if their relative difference was smaller than  $10^{-9}$ .

Another possible factor that contributes to roundoff errors are ill-conditioned matrices that need to be inverted. In general, the condition of a problem describes the sensitivity of the output data to variations in the input data. A problem is called well-conditioned if a small change in the input produces only small changes in the output. Conversely, the problem is called ill-conditioned if small variations in the input results in large variations in the output [GA08, chap. 6.2]. The condition number  $\kappa$  of a matrix  $A$  [DB08, chap. 2]

$$\kappa(A) = \|A\| \cdot \|A^{-1}\| \quad (8.1)$$

is a quantity for the “easiness” of its inversion. Matrix norms can be defined with respect to vector norms  $\|\cdot\|_p$ :

$$\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p} = \max_{\|x\|=1} \|Ax\|_p. \quad (8.2)$$

The smallest  $p$ -norm for symmetric matrices is the Euclidian norm given by the

largest eigenvalue  $\lambda_{\max}$  of the matrix  $A^T \cdot A$ :

$$\|A\|_2 = \sqrt{\lambda_{\max}(A^T \cdot A)}. \quad (8.3)$$

The condition is equal to 1 with respect to the above norm for orthogonal matrices where the inversion can be achieved by a simple transposition. It approaches infinity for singular matrices. The problem of inverting the matrix  $A$  is called ill-conditioned if adding 1 to its condition produces no effects:

$$1 + \kappa(A) = \kappa(A) \text{ in machine precision.} \quad (8.4)$$

This is the case if  $\kappa(A) > 2/\epsilon$  where the machine epsilon or unit roundoff error  $\epsilon$  is defined as the largest number where

$$1 + \frac{\epsilon}{2} = 1 \text{ in machine precision and} \quad (8.5)$$

$$1 + \epsilon = 1 + \epsilon \text{ in machine precision.}$$

The conventional (equations 4.21b and 4.21c) and Joseph stabilized (equation 4.29) formulations of the Kalman filter equations need to invert a matrix with the dimension of the measurement vector while the Swerling formulation (equations 4.28) has to invert a matrix of the dimension of the state vector two times. The UD-filter (described in chapter 4.2.4) improves the condition of the covariance matrix by a decomposition of the matrix.

Table 8.3 shows that the momentum resolution deteriorated due to the over-estimation of the initial errors to a point where there was no more improvement compared to the input momentum. Track reconstruction is considered as failed if the  $\chi^2$  exceeds a threshold value. Since larger values in the covariance matrices tend to reduce the  $\chi^2$  of the fit the efficiency rises slightly. All calculations are performed in double floating point precision.

The amount of asymmetric elements in the covariance matrix also increased with larger initial covariance values. This problem could be greatly reduced by applying numerically more stable Joseph formulation of the Kalman filter equations. Only the Swerling formulation produced a very small amount of ill-conditioned covariance matrices. While these are warning signs of numerical degradation none of the methods had a significant impact on the final momentum estimate.

### 8.1.3 Energy Loss and Multiple Scattering

During track reconstruction the effects of multiple scattering and energy loss by radiation and ionization are taken into account as described in chapter 6.4.

#### Protons

Including the effects of multiple scattering in the Kalman filter has a significant impact on  $\chi^2$  and efficiency on low momenta proton tracks as can be seen in

Initial Values for Covariance	Kalman Formulation	RMS $\left(\frac{\Delta p}{p} [\%]\right)$	Efficiency
Normal	Conventional	4.47%	0.9898
10× higher	Conventional	5.09%	0.9948
100× higher	Conventional	5.62%	0.9956
Normal	Joseph	4.47%	0.9898
10× higher	Joseph	5.09%	0.9947
100× higher	Joseph	5.62%	0.9956
Normal	UD-Filter	4.47%	0.9898
10× higher	UD-Filter	5.09%	0.9947
100× higher	UD-Filter	5.62%	0.9956
Normal	Swerling	4.48%	0.9898
100× higher	Swerling	5.62%	0.9925
Initial Values for Covariance	Kalman Formulation	Symmetry Errors in Covariance	Ill-conditioned Matrices
Normal	Conventional	9026	0
10× higher	Conventional	553, 413	0
100× higher	Conventional	1, 728, 604	0
Normal	Joseph	20	0
10× higher	Joseph	26	0
100× higher	Joseph	25	0
Normal	UD-Filter	–	–
10× higher	UD-Filter	–	–
100× higher	UD-Filter	–	–
Normal	Swerling	317, 666	4
100× higher	Swerling	356, 540	16

Table 8.3: The effect of different initial covariance matrices and numerically more stable formulations of the Kalman filter equations.

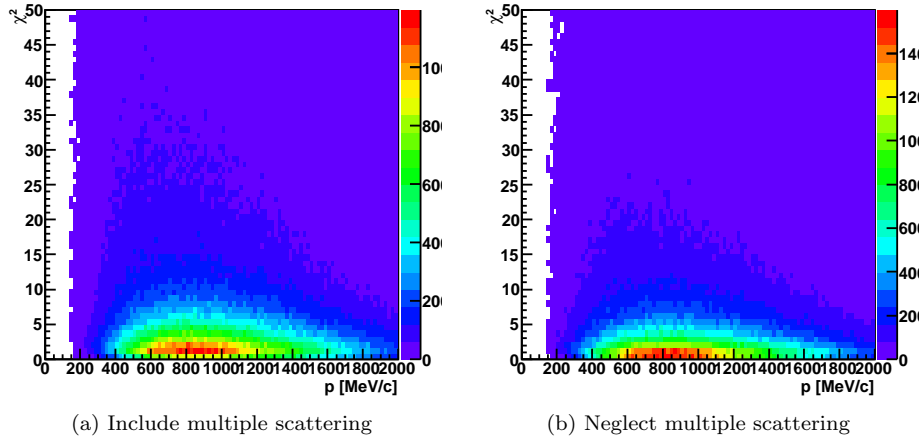


Figure 8.5: The  $\chi^2$  over the momentum when including (a) and excluding (b) the effects of multiple scattering.

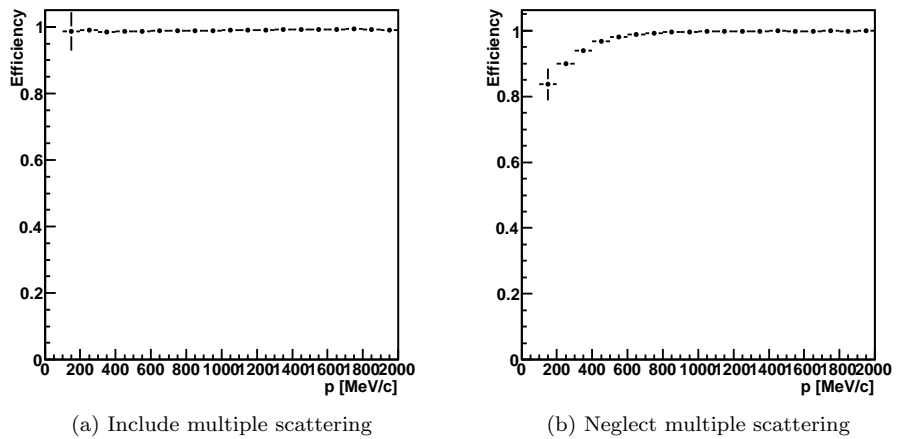


Figure 8.6: The efficiency plotted against the momentum with and without including the effects of multiple scattering.



figures 8.5 and 8.6. Multiple scattering increases the uncertainty of the track state and thus the covariance matrix which leads to lower  $\chi^2$  values in general.

The protons in the simulation data have momenta between  $100 \text{ MeV}/c$  and  $2 \text{ GeV}/c$  and are subject to energy loss by ionization that is calculated with the Bethe-Bloch formula (6.52) based on the momentum estimate. Their energy loss as calculated by the Kalman filter is plotted in figure 8.7a. The distribution looks similar to the energy loss done in the GEANT simulation (fig. 8.7b). In GEANT an additional scattering of the mean energy loss by a Landau distribution is applied while the Kalman filter only computes mean values of the energy loss. Aside from this there are considerable differences at momenta lower than  $400 \text{ MeV}/c$  (fig. 8.7c).

It is possible that the Kalman filter starts with a momentum estimate that is well below the real momentum. Because the energy loss increases with decreasing particle momentum this could lead to an overestimation of the energy loss and pull the Kalman filter further into the wrong direction. Additionally, the Bethe-Bloch formula 6.52 is no longer correct for low momenta. At  $\beta\gamma \lesssim 0.3$  which corresponds to a momentum of  $p \lesssim 280 \text{ MeV}/c$  respectively shell corrections contribute to the energy loss. If  $\beta\gamma \lesssim 0.1$  or momenta smaller than  $94 \text{ MeV}/c$  additional corrections to the Bethe-Bloch formula have to be made [N<sup>+</sup>10]. Including these corrections for low momenta protons could increase the accuracy of the Kalman filter as well.

## Leptons

In the case of positrons and electrons radiation loss dominates the loss by ionization (formulas 6.55 and 6.56). The energy loss of the positrons is shown in figure 8.8 and rises linearly with the particle energy as in the Bethe-Heitler formula (6.57).

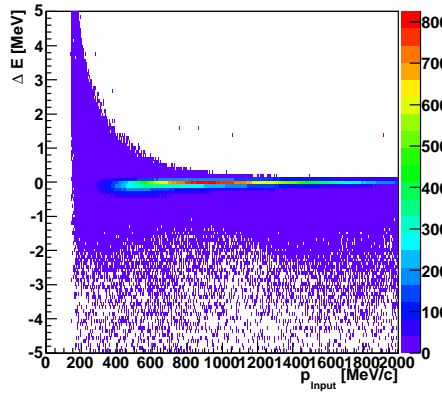
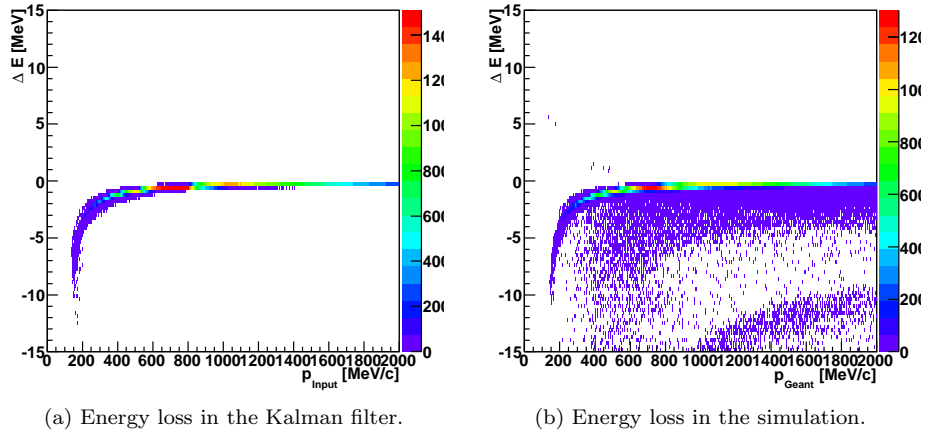


Figure 8.7: The energy loss calculated by the Kalman filter (a) and the GEANT simulation (b) for protons. In (c) the difference between  $\Delta E$  in the simulation and  $\Delta E$  computed by the Kalman filter can be seen. The energy loss  $\Delta E$  has been defined as  $\Delta E = E_{\text{final}} - E_{\text{start}}$ . Since the particle loses energy this value is always negative.

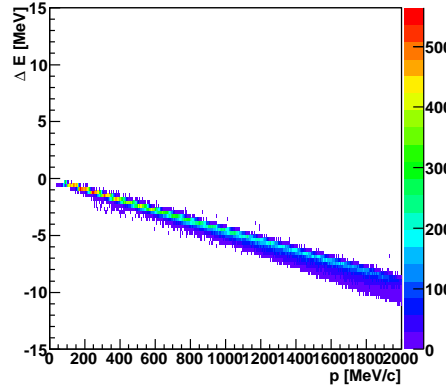


Figure 8.8: The energy loss of positrons as calculated by the Kalman filter.

#### 8.1.4 Dependencies of Track Reconstruction Performance on the Polar Angle

The HADES detector covers a wide angular range from  $18^\circ - 85^\circ$  in the polar angle  $\Theta$  and almost full azimuthal angle  $\phi$ . Figure 8.9 shows the momentum resolution plotted over the momentum for different  $\Theta$ -ranges. The resolution gets worse for larger polar angles because there is less magnetic field available than at smaller  $\Theta$ . The deflection by the magnetic field is lower making a momentum estimation more difficult. For smaller momenta tracks the resolution is dominated by multiple scattering. This is more relevant for protons than for leptons as could be expected from formula 6.50.

Figure 8.10 shows the polar angle  $\Theta$  as a function of the azimuthal angle  $\phi$  in the sector coordinate system for the rejected tracks that tend to cluster at the borders of the drift chambers where the magnetic coils are located and the inhomogeneity of the magnetic field is especially strong.

The state parameters  $t_x$  and  $t_y$  are the tangents of two angles. At high angles a small variation in the angle can lead to large changes in the tangent of that angle. The assumption done in the extended Kalman filter that the system stays approximately linear over small ranges is no longer valid. To avoid large angles the coordinate system is rotated in the initial track direction which has resulted in improvements of the momentum resolution for higher polar angles  $\Theta$  as can be seen in figure 8.11.

## 8.2 Effect of Non-Normal Error Distributions

The minimization with least-square methods is only optimal for normal error distributions. Both the Kalman filter and the global fit that is currently used for track reconstruction in HADES obtain the estimate for the initial track state

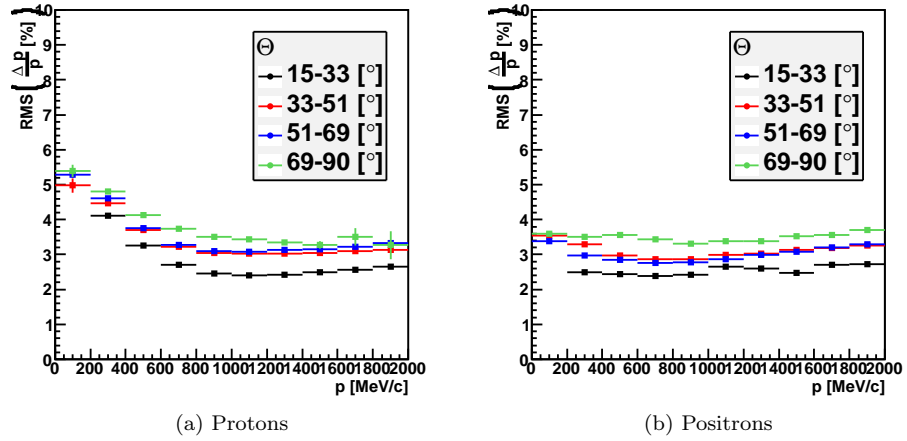


Figure 8.9: The momentum resolution as a function of the momentum for different polar angle  $\Theta$  ranges.

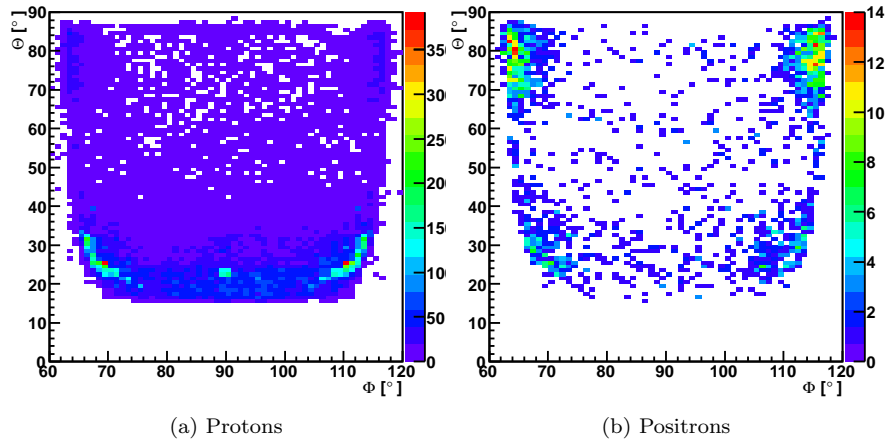


Figure 8.10: The polar angle  $\Theta$  as a function of the azimuthal angle  $\phi$  for tracks that could not be reconstructed or were cut due to high  $\chi^2$ .

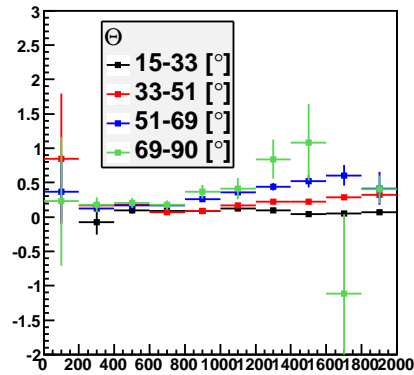
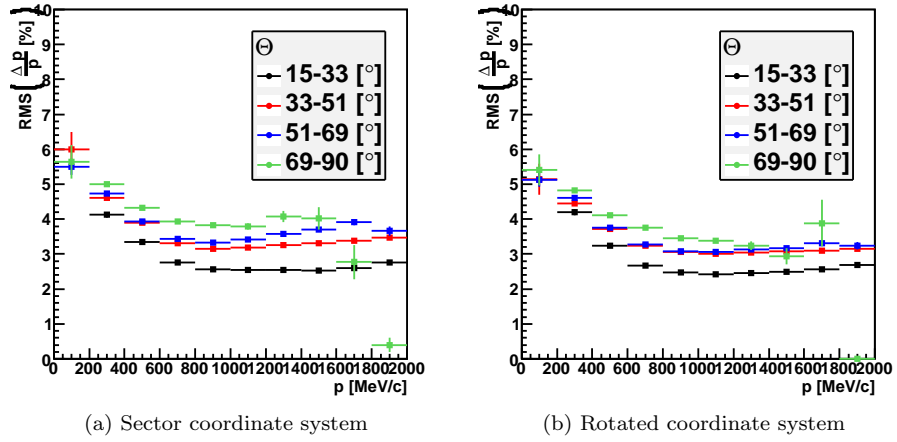


Figure 8.11: The momentum resolution plotted over the momentum for different polar angle ranges. In (a) the original sector coordinates were used while in (b) the coordinate system was rotated in direction of the initial estimate of the track direction. Figure (c) shows the difference between the two.

from the segment and spline fits that do not provide normal distributions. The spline fit also depends on the results of the segment fitter. In order to discriminate the effects of the segment and spline fit four cases have been compared:

- A Ideal tracking eliminates fake tracks and the errors from the segment fit are normally distributed. To obtain an initial momentum estimate the momentum from the simulation was smeared randomly by a Gaussian function with a width of 5%. In this case the uncertainty distributions of the initial track parameters state are normal.
- B The initial momentum estimate is now obtained from the spline fit. The distribution of the spline fit is non-normal under ideal tracking as figure 8.12a shows.
- C Ideal tracking is turned off. Thus, the errors of the segment fit are no longer normally distributed. The initial momentum estimate is obtained by smearing the simulation momentum as in case A.
- D The input is obtained from the segment and spline fit. The uncertainty distributions of all track state parameters are now non-normal.

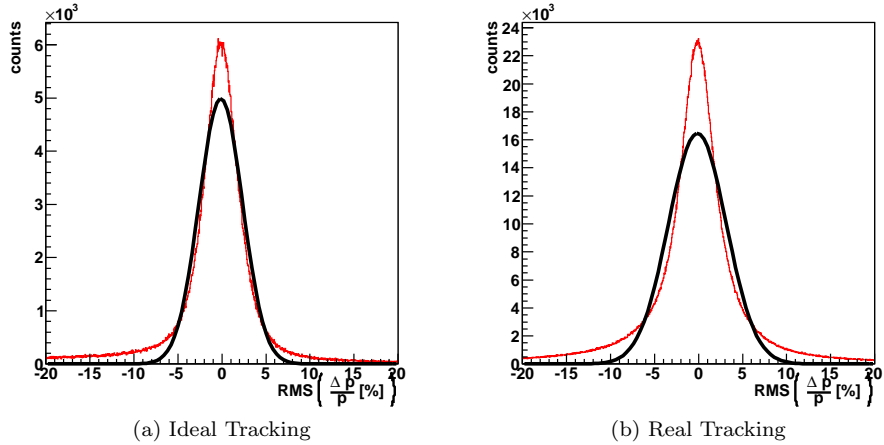


Figure 8.12: The momentum estimates from the spline fit of proton tracks with ideal and real tracking. The Gauss fit has a width of  $\sigma = 2.47 \pm 0.01$  in (a) and  $\sigma = 3.31 \pm 0.01$  in (b). The RMS of the distributions are 4.64% and 5.26%, respectively.

The results for proton and positron tracks can be found in table 8.5. Ideal tracking and Gaussian errors for all track states produced the best residuals in momenta. The momentum resolution degraded after obtaining the initial momentum estimate from the spline fit and further after adding realistic segments.

Tracking	Particle	RMS $\left(\frac{\Delta p}{p} [\%]\right)$	Skewness	Kurtosis
Ideal	Positrons	5.30	$-0.284 \pm 0.005$	$2.789 \pm 0.010$
Real	Positrons	5.32	$-0.096 \pm 0.005$	$2.572 \pm 0.010$
Ideal	Protons	4.64	$-0.591 \pm 0.003$	$4.623 \pm 0.005$
Real	Protons	5.22	$-0.145 \pm 0.001$	$2.781 \pm 0.002$

Table 8.4: Difference between ideal and real tracking for the spline fit.

In most cases the distributions of the residuals in momenta were skewed towards negative values that correspond to an overestimated reconstructed momentum. This is true for the spline fit (see table 8.4), the Kalman filter and the global fit (table 8.6). Even under the ideal conditions of case A the Kalman filter still provided an asymmetric, non-Gaussian momentum resolution. The momentum estimate of the Kalman filter was generally inferior compared to the global fit as can be seen in table 8.6.

Input	Particle	RMS $\left(\frac{\Delta p}{p} [\%]\right)$		Skewness	Kurtosis
		Filtered	Smoothed		
A	Positrons	2.48	2.36	$+1.167 \pm 0.005$	$9.394 \pm 0.010$
B	Positrons	3.60	3.77	$+0.295 \pm 0.005$	$6.077 \pm 0.009$
C	Positrons	4.17	3.57	$-0.332 \pm 0.005$	$5.873 \pm 0.010$
D	Positrons	4.52	4.37	$-0.203 \pm 0.005$	$3.978 \pm 0.010$
A	Protons	2.13	2.04	$-0.246 \pm 0.003$	$3.485 \pm 0.005$
B	Protons	3.00	2.90	$-0.437 \pm 0.003$	$7.422 \pm 0.005$
C	Protons	4.23	4.22	$-0.285 \pm 0.003$	$4.501 \pm 0.005$
D	Protons	4.49	4.47	$-0.249 \pm 0.001$	$3.945 \pm 0.002$

Table 8.5: Effects of different input methods on the Kalman filter. The filtered column shows the momentum estimates at MDC-IV and the smoothed column the estimate at MDC-I after smoothing. Skewness and kurtosis numbers refer to the smoothed momentum at MDC-I.

Input	Particle	RMS $\left(\frac{\Delta p}{p} [\%]\right)$	Skewness	Kurtosis
B	Positrons	2.35	$+0.298 \pm 0.005$	$8.229 \pm 0.009$
D	Positrons	4.18	$-0.584 \pm 0.005$	$4.466 \pm 0.010$
B	Protons	2.54	$-0.287 \pm 0.002$	$7.183 \pm 0.005$
D	Protons	4.42	$-0.547 \pm 0.001$	$3.801 \pm 0.003$

Table 8.6: Difference between ideal and real tracking for the global fit. The reconstruction efficiency for positrons with input method B is only 92.5% while it is around 99% for all the other test setups.

Table 8.7 lists the performance of the Kalman filter for different momentum uncertainties. Besides obtaining the initial momentum from the spline fit the GEANT momentum was smeared by a Gaussian function with widths of 5%, 10% and 15%, respectively, and used as input for the Kalman filter. With real segments a 15% momentum smearing yields similar results as momentum input from the spline fit. Better, more Gaussian shape input data yields better estimates by the Kalman filter.

Input	$\frac{\Delta p}{p} [\%]$		Efficiency
	RMS	Mean	
Spline Momentum	4.47	-0.47	0.9898
5% momentum smearing.	4.22	-0.39	0.9905
10% momentum smearing.	4.38	-0.42	0.9918
15% momentum smearing.	4.47	-0.47	0.9917

Table 8.7: Comparison of the momentum resolution for proton tracks with different input methods for the initial momentum.

Figure 8.13 shows the probability that the observed value for the  $\chi^2$  is equal or larger than  $\chi^2$  for a correct model (see chapter 4.2.3). The probability distribution should be flat for a correct  $\chi^2$  distribution. In case D the Kalman filter shows a large accumulation at low probabilities (fig. 8.13b). Possible reasons are that the system model may be wrong, the variances of the errors could be incorrect or the errors are not normally distributed. The latter can not be the sole reason as the probability curve is not flat either in the ideal case A as can be seen in figure 8.13a.

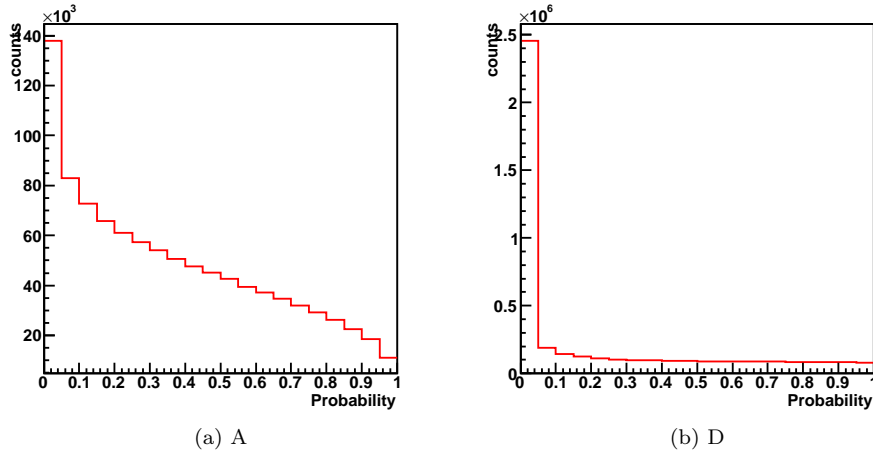


Figure 8.13: Probability distributions of the Kalman filter for proton tracks.



### 8.3 Parameters Settings for the Deterministic Annealing Filter

The parameters unique for the DAF are the cut-off parameter  $\chi_{\text{cut}}^2$  and the annealing factors  $T$ . Tests have been done with various settings for proton tracks. Since there are still open problems with the DAF (see 8.4) the results presented here should be viewed with caution. Table 8.8 lists the tested annealing schemes. While the amount of possible schemes are endless some observations can be made from this selection:

- Very high annealing factors and a lot of iterations (schemes A and B) are not necessary.
- The same is true when applying several medium temperatures (schemes C and D compared to F and G).
- “Deep-freezing” by a final temperature close to zero should be avoided. Figure 8.15 shows the weights of competing hits that both belong to the same track. The annealing schemes F and J are compared and differ only in the temperature of the last step. In 8.15a the final temperature is 1 while in 8.15b it is 0.1. Such a low temperature caused the DAF to reject a lot more real hits.
- Two final iterations at  $T = 1$  are preferable to one or three such iterations (schemes C–G).
- Schemes K and L that start at low temperatures are inferior to those that begin the annealing process at medium temperatures (schemes E–H).
- The scheme M only runs only one iteration at  $T = 1$ . This is similar to the standard Kalman filter except that an effective measurement is calculated in case of competing hits. The efficiency is much worse when annealing is omitted showing the effectiveness of annealing.
- The efficiency stayed almost constant for all annealing schemes except for scheme M.
- In most cases the effect of different annealing schemes were small.

Figure 8.16 shows the momentum resolution and efficiency for different cut-off parameters and the fixed annealing scheme  $T = \{81, 9, 4, 1, 1\}$ . The momentum resolution slowly deteriorates with increasing cut-off values.

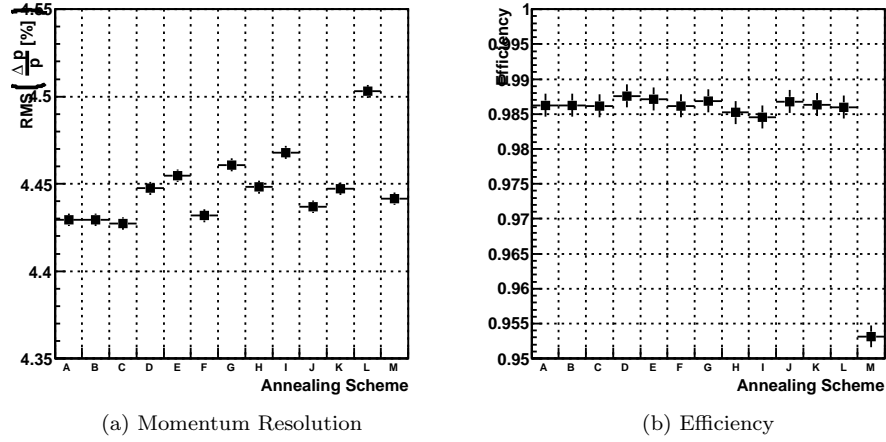


Figure 8.14: The momentum resolution and efficiency of the DAF for different annealing schemes with a cut-off parameter of  $\chi^2_{\text{cut}} = 4$ . The annealing schemes are listed in table 8.8.

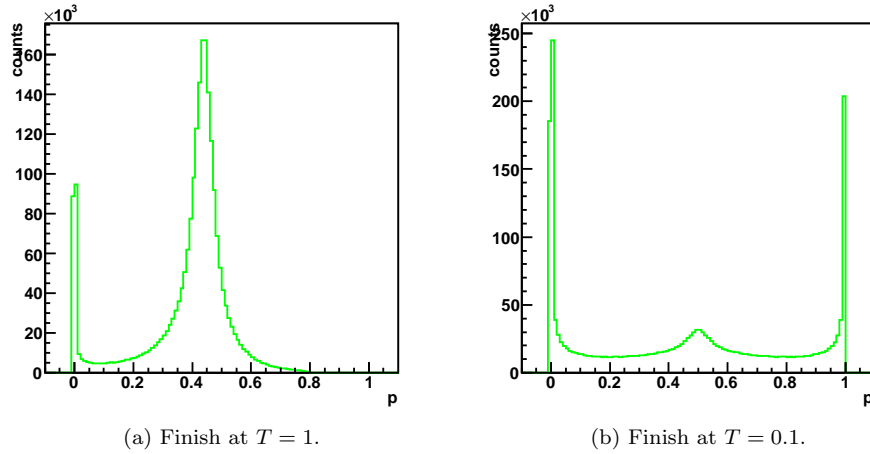


Figure 8.15: The assignment probabilities in the case that there are two competing measurements in an MDC layer that belong to the same track. The DAF was run with schemes F (plot (a)) and J (plot (b)) which only differ in the final temperature. Scheme F finishes with  $T = 1$  while scheme J finishes with the very low temperature of  $T = 0.1$ .

Scheme	Annealing Factors	Scheme	Annealing Factors
A	1000, 200, 81, 9, 4, 1, 1	H	81, 4, 1, 1
B	200, 81, 9, 4, 1, 1	I	81, 9, 4, 1, 0.1, 0.1
C	81, 21, 9, 4, 1, 1	J	81, 9, 4, 1, 0.1
D	81, 21, 9, 4, 1	K	9, 4, 1, 1
E	81, 9, 4, 1, 1, 1	L	1, 1, 1
F	81, 9, 4, 1, 1	M	1
G	81, 9, 4, 1		

Table 8.8: Tested annealing schemes.

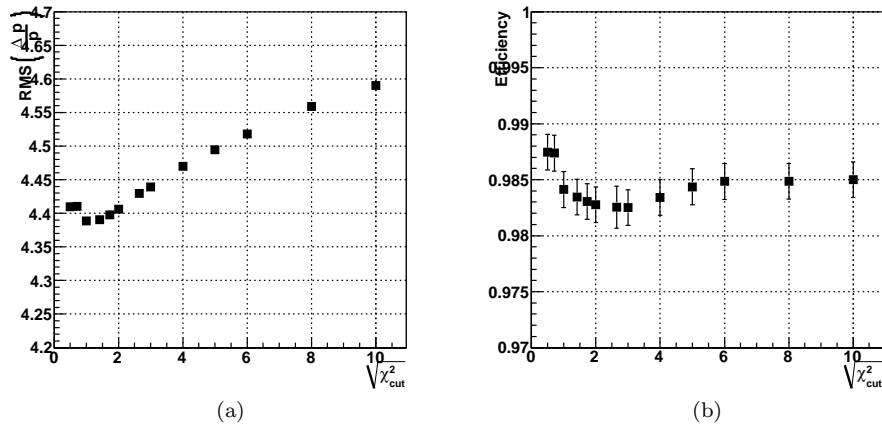


Figure 8.16: The momentum resolution (a) and efficiency (b) of the DAF for different cut-off parameters  $\chi_{\text{cut}}^2$  with an annealing scheme of  $T = \{81, 9, 4, 1, 1\}$ . The square-root of the cut-off parameter is plotted.

## 8.4 Comparison of Methods

Five different reconstruction procedures have been compared:

1. The current reconstruction method with a global fit as described in 3.
2. The Kalman filter worked with the reconstructed segment points and momentum from the spline fit as input. The projection function for this case can be found in chapter 6.3.1.
3. The Kalman filter was adapted to work directly with the drift chamber measurements. The projection is calculated as described in 6.3.2. If there are competing measurements in an MDC layer only the one with the smaller drift time is used.

4. The Deterministic Annealing Filter is used to fit the drift chamber measurements. Again, only one wire measurement per MDC layer is considered.
5. Competition between wire measurements in the same MDC layer is allowed. The method to calculate the projection function differs significantly from the previous cases and competing measurements have to be combined into an effective measurement as described in 6.3.2.

Approaches 3 and 4 are intermediate steps to get from the Kalman filter that works with space points to the DAF that works on competing drift time measurements. From the tests in the previous sections the following settings have shown to provide good results for the Kalman filter and the DAF:

- Use the conventional Kalman filter formulation (4.21b and 4.21c). The numerically more stable formulas have not shown a significant improvement.
- Rotate the coordinate system in initial track direction.
- Run one iteration in forward, i.e. beam, direction and smooth backwards.
- Calculate energy loss and multiple scattering.
- The DAF was run with a  $\chi_{cut}^2$  parameter of 4 and the annealing scheme  $T = \{81, 9, 4, 1, 1\}$ .

The residuals in momenta for the different reconstruction approaches are plotted in figures B.1, B.3 and B.8 for positron, electron and proton tracks, respectively. Values for the mean, RMS and skewness of the residuals in momenta are listed in tables 8.9, 8.10 and 8.11. Fitting the wire measurements without annealing produced the worst momentum estimation and is not an option for track fitting. The best results were consistently obtained with the global fit while the Kalman filter performed slightly worse at high  $\Theta$  angles and slightly better at momenta smaller than  $600 \text{ MeV}/c$  where multiple scattering becomes more relevant (figures 8.17, B.4 and 8.22).

The Kalman filter and DAF versions that fit wire measurements without competition between hits show a substantial degradation in momentum resolution at large polar angles  $\Theta > 50^\circ$  which becomes worse with higher momenta. Additionally, the amount of rejected tracks starts to increase with higher azimuthal angles  $\Phi$  (figures 8.19 and 8.23). As these systematics are present in both approaches the annealing can be excluded as the reason. Adding annealing significantly improved the overall momentum resolution when fitting wire measurements. The bad performance of the DAF version that works with competing hits should be investigated.

The residuals are mostly skewed towards negative values that correspond to an overestimation of the reconstructed momentum. The asymmetry could be alleviated by fitting the wire measurements instead of the segment points, especially in case of the electrons.

The global fit and Kalman filter show clusters of rejected tracks at the borders of the chambers where the magnetic coils are located.

Method	$\frac{\Delta p}{p} [\%]$			Efficiency
	Mean	RMS	Skewness	
Input from spline	-0.11	5.32	$-0.096 \pm 0.005$	-
Global fit	-0.51	4.18	$-0.584 \pm 0.005$	0.9893
Kalman filter	-0.56	4.37	$-0.203 \pm 0.005$	0.9898
Kalman filter for wire hits	-0.62	5.39	$-0.132 \pm 0.005$	0.9894
DAF for single wire hits	+0.72	5.28	$-0.267 \pm 0.005$	0.9846
DAF	-0.03	6.22	$+0.025 \pm 0.005$	0.8772

Table 8.9: Comparison of the momentum resolution for positron tracks.

Method	$\frac{\Delta p}{p} [\%]$			Efficiency
	Mean	RMS	Skewness	
Input from spline	-0.56	5.03	$-0.276 \pm 0.005$	-
Global fit	-0.89	4.25	$-0.933 \pm 0.005$	0.9901
Kalman filter	-0.56	4.42	$-0.564 \pm 0.005$	0.9809
Kalman filter for wire hits	+0.21	5.28	$-0.065 \pm 0.005$	0.9814
DAF for single wire hits	+0.20	4.99	$+0.003 \pm 0.005$	0.9810
DAF	-0.49	6.03	$-0.089 \pm 0.006$	0.8578

Table 8.10: Comparison of the momentum resolution for electron tracks.

Method	$\frac{\Delta p}{p} [\%]$			Efficiency
	Mean	RMS	Skewness	
Input from spline	-0.33	5.22	$-0.143 \pm 0.001$	-
Global fit	-0.58	4.42	$-0.547 \pm 0.001$	0.9903
Kalman filter	-0.46	4.47	$-0.249 \pm 0.001$	0.9898
Kalman filter for wire hits	-0.20	4.70	$-0.105 \pm 0.001$	0.9906
DAF for single wire hits	-0.32	4.33	$-0.366 \pm 0.001$	0.9874
DAF	-0.29	6.18	$-0.069 \pm 0.001$	0.8502

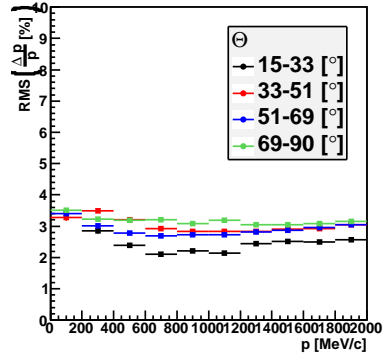
Table 8.11: Comparison of the momentum resolution for proton tracks.

Figures 8.18 and B.10 show the probabilities that the observed values for the  $\chi^2$  is equal or higher than  $\chi^2$  for a correct model (also see chapter 4.2.3). The Kalman filter yields a large accumulation at low probabilities, i.e. there are too many high  $\chi^2$  values. Possible reasons are that the system model may be wrong, the variances of the errors could be incorrect or the errors are not normally distributed. The non-normal error distributions can not be the only

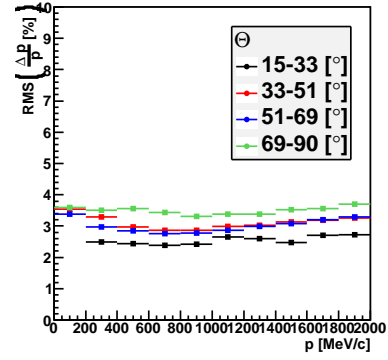
reason as has been discussed in chapter 8.2. The versions that work with wire measurements and the global fit additionally yield too many low  $\chi^2$  values that indicates an overestimation of errors. In general, the  $\chi^2$  distributions show some improvement compared to the global fit (figures B.2 and B.9).

The reconstruction efficiency of the global fit falls with decreasing particle momentum while it remains constant for the Kalman filter. The methods that work with wire measurements experience a drop in efficiency at lower momenta as well, though to a lesser degree than the global fit. The decline in efficiency is much less pronounced for leptons (figures 8.19 and B.5) than for protons (figure 8.23). The Kalman filter and DAF could benefit from the inclusion of multiple scattering in the momentum estimation [PTVF07, chap. 15].

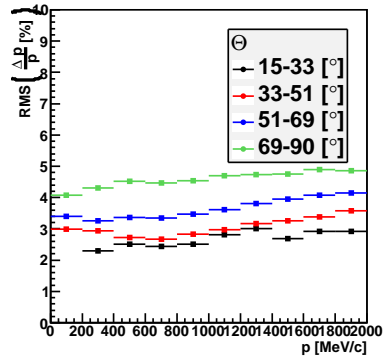
The efficiency falls slightly at very low polar angles of around  $\Theta \sim 20^\circ$  and sharply at high polar angles  $\Theta > 65^\circ$  with all the tested methods (figures 8.21, B.7 and 8.25). The magnetic field strength is lower at higher polar angles making the momentum reconstruction more difficult. A second dip in efficiency can be observed for the Kalman filter and DAF that fit wire measurements without competing hits at  $\Theta \sim 50^\circ$  which corresponds to a perpendicular impact on the drift chamber. This structure can also be observed in plots 8.19, B.5 and 8.23 that show  $\Theta$  over  $\Phi$  for the rejected tracks.



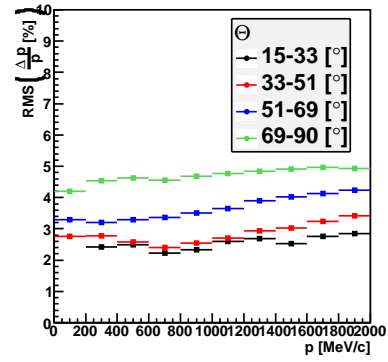
(a) Global fit



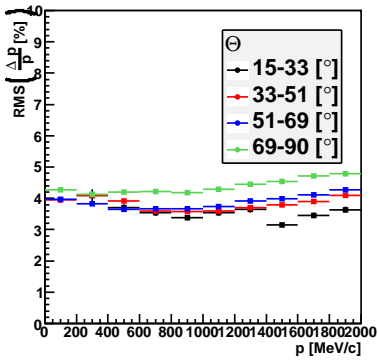
(b) Kalman filter



(c) Kalman filter for single wire hits

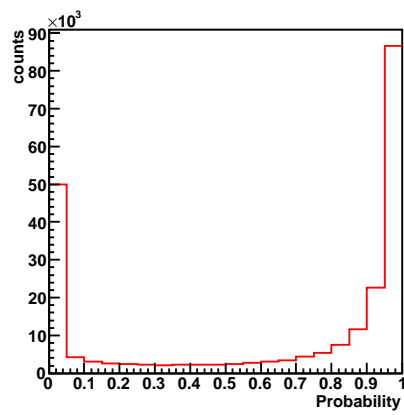


(d) DAF for single wire hits

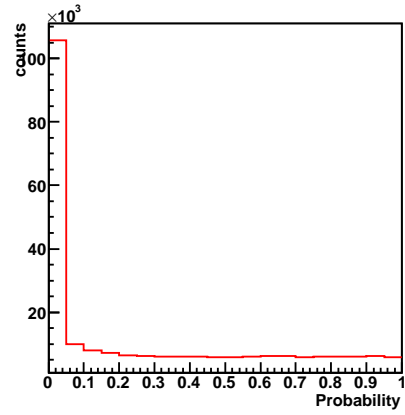


(e) DAF

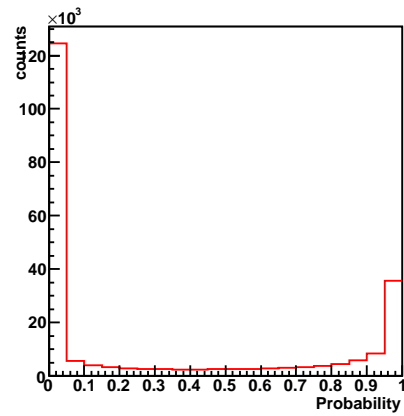
Figure 8.17: Comparison of the momentum resolution for different  $\Theta$  ranges for positron tracks.



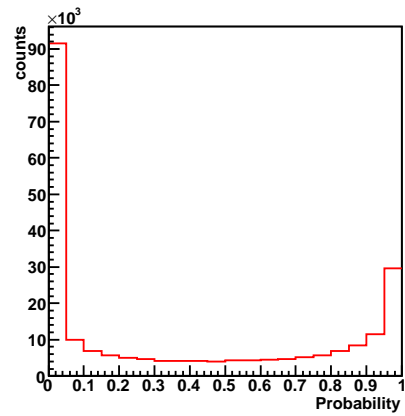
(a) Global fit



(b) Kalman filter



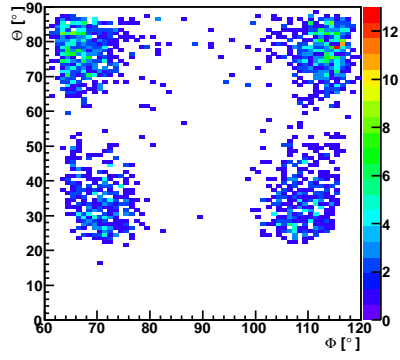
(c) Kalman filter for single wire hits



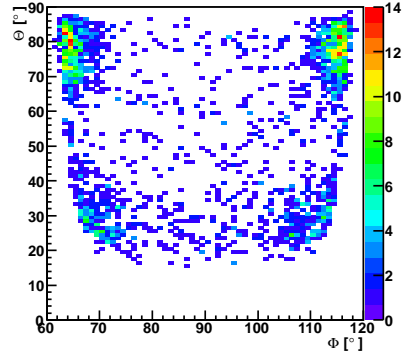
(d) DAF for single wire hits

Figure 8.18: Probability distributions for positron tracks.

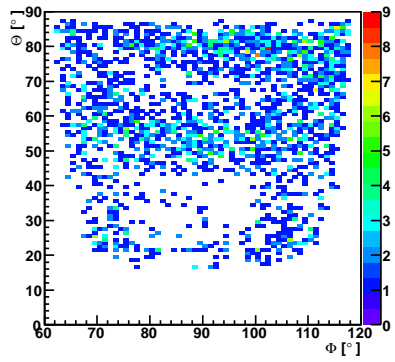




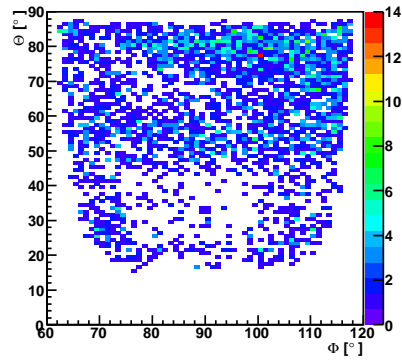
(a) Global fit



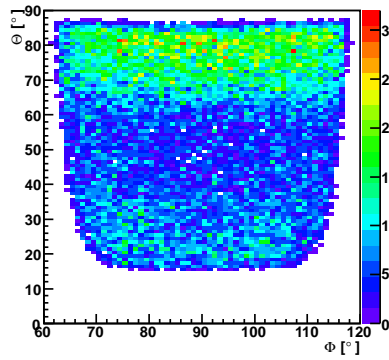
(b) Kalman filter



(c) Kalman filter for single wire hits

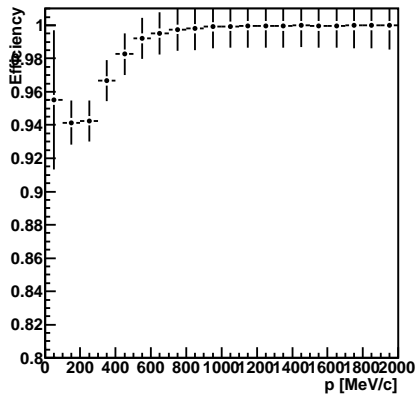


(d) DAF for single wire hits

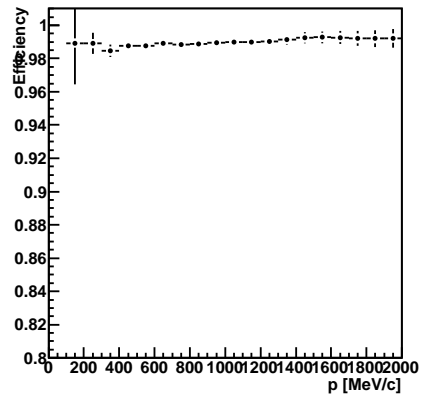


(e) DAF

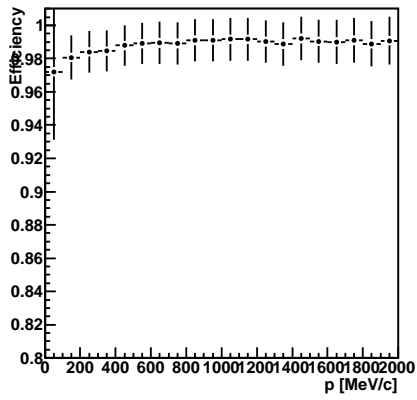
Figure 8.19: The polar angle  $\Theta$  over the azimuthal angle  $\phi$  for positron tracks that could not be reconstructed or were cut due to high  $\chi^2$ .



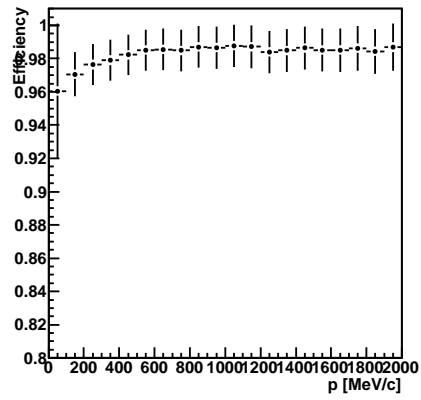
(a) Global fit



(b) Kalman filter

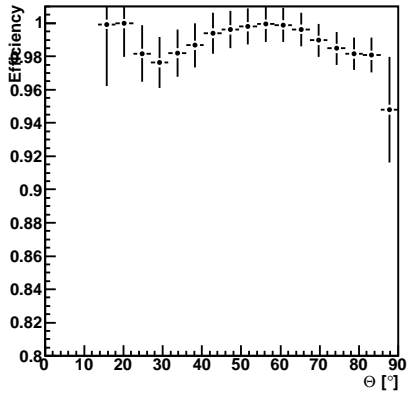


(c) Kalman filter for single wire hits

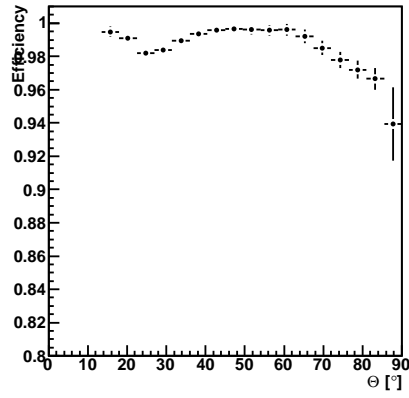


(d) DAF for single wire hits

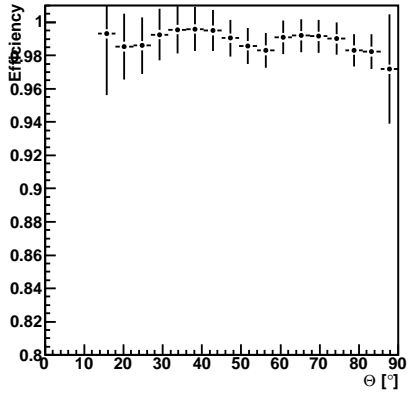
Figure 8.20: Efficiency over momentum for positron tracks.



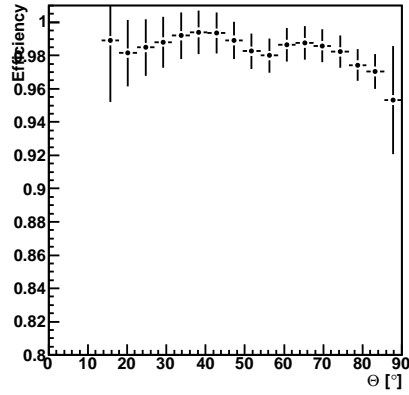
(a) Global fit



(b) Kalman filter

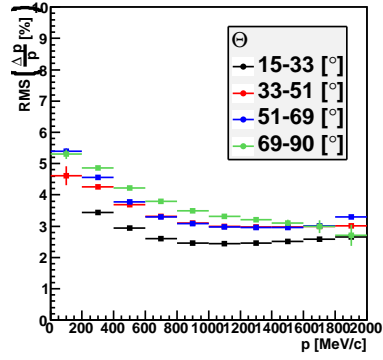


(c) Kalman filter for single wire hits

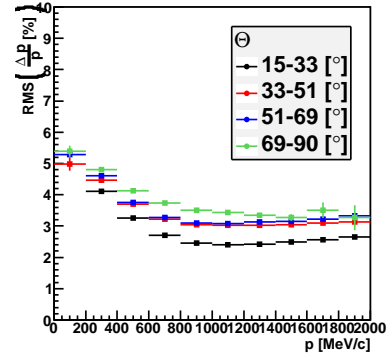


(d) DAF for single wire hits

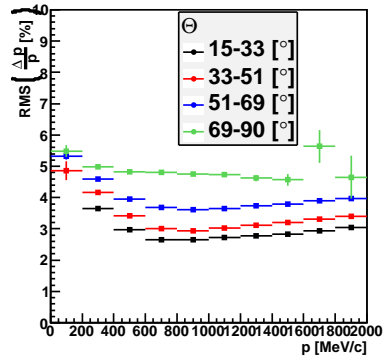
Figure 8.21: Efficiency over the polar angle  $\Theta$  for positron tracks.



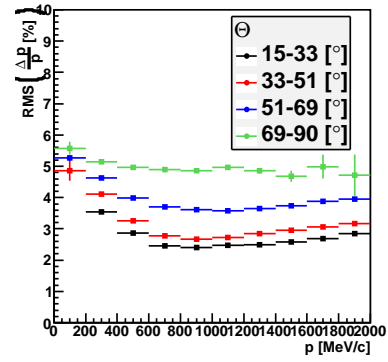
(a) Global fit



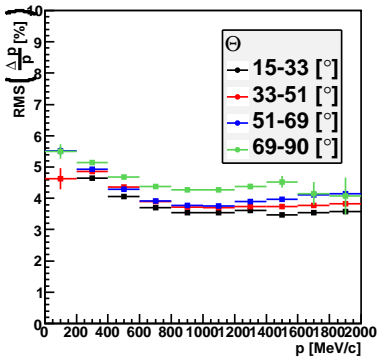
(b) Kalman filter



(c) Kalman filter with wire hits

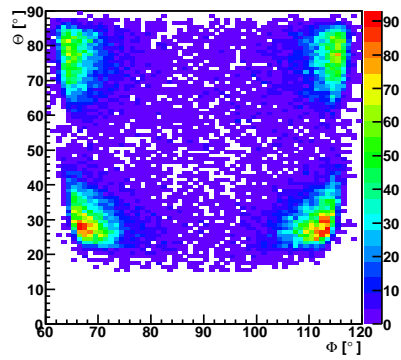


(d) DAF for single wire hits

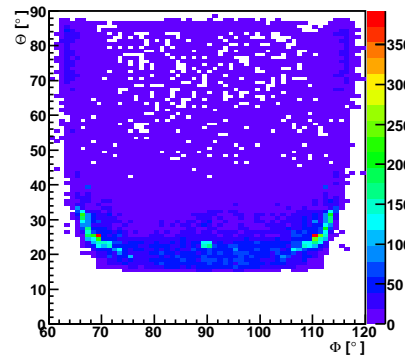


(e) DAF

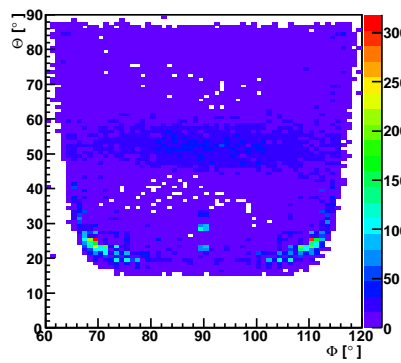
Figure 8.22: Comparison of the momentum resolution for different  $\Theta$  ranges for proton tracks.



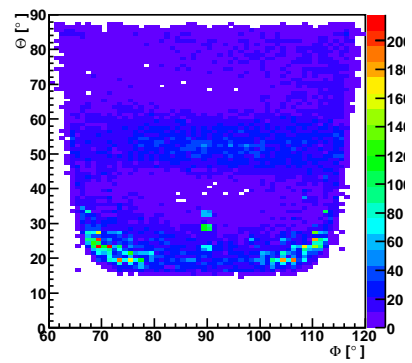
(a) Global fit



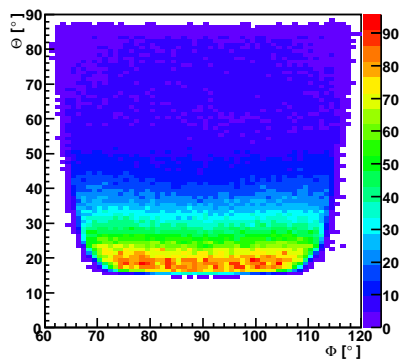
(b) Kalman filter



(c) Kalman filter with wire hits

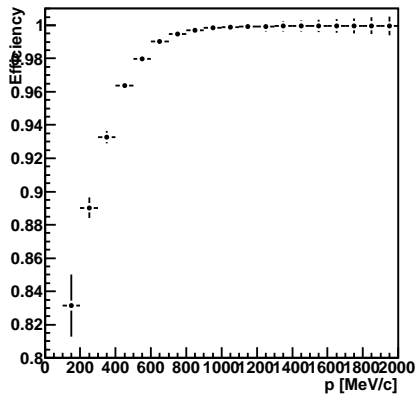


(d) DAF for single wire hits

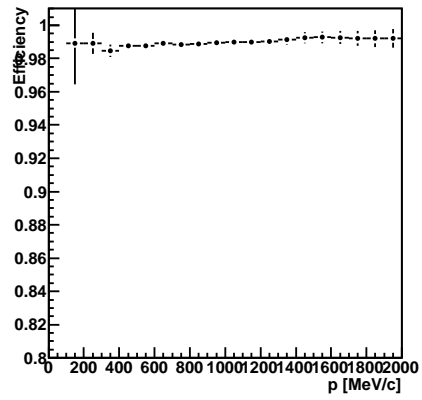


(e) DAF

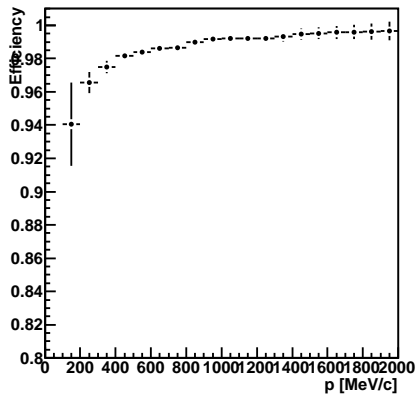
Figure 8.23: The polar angle  $\Theta$  over the azimuthal angle  $\phi$  for proton tracks that could not be reconstructed or were cut due to high  $\chi^2$ .



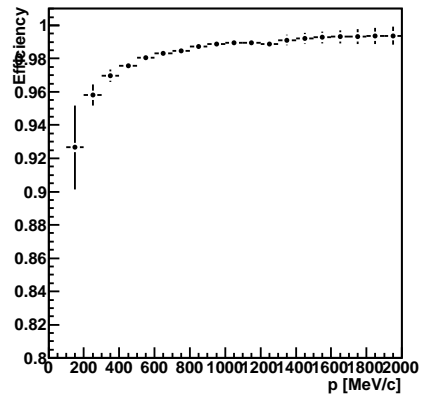
(a) Global fit



(b) Kalman filter

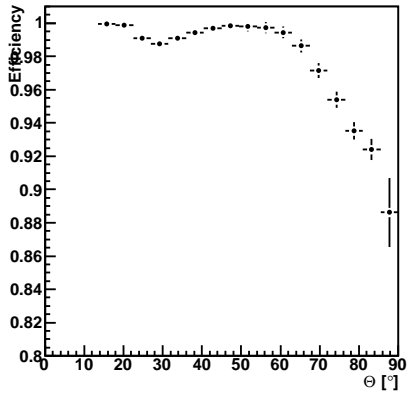


(c) Kalman filter for single wire hits

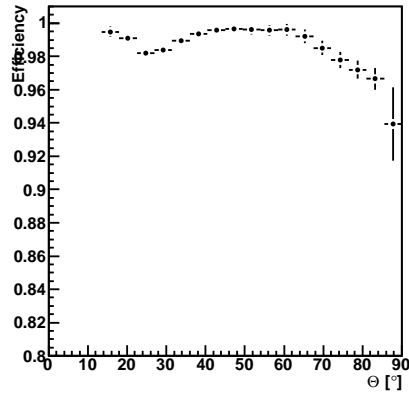


(d) DAF for single wire hits

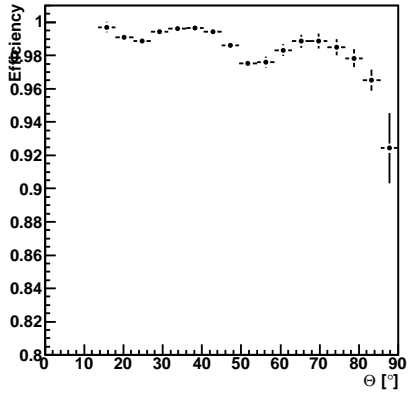
Figure 8.24: Efficiency over momentum for proton tracks.



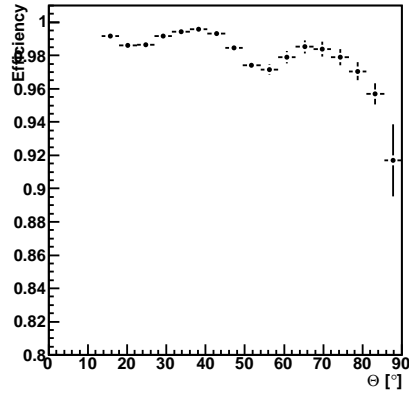
(a) Global fit



(b) Kalman filter



(c) Kalman filter for single wire hits



(d) DAF for single wire hits

Figure 8.25: Efficiency over the polar angle  $\Theta$  for proton tracks.

## Chapter 9

# Summary and Outlook

An extended Kalman filter and a Deterministic Annealing Filter have been implemented and tested. The Kalman filter is able to work both with reconstructed segment space points and the drift time measurements from the drift chambers. The Kalman filter that works with reconstructed segment points yields an overall momentum resolution comparable to the currently used global fit. The inclusion of multiple scattering in the estimation process has resulted in improved performance at lower momenta. The observed values for the  $\chi^2$  are closer to a true  $\chi^2$  distribution and a better measurement for the fit quality. At high momenta and polar angles the momentum resolution of the Kalman filter is still inferior to the currently used global fit. The numerically more reliable Joseph stabilization and UD-filter are available should roundoff errors become an issue with the conventional formulation of the Kalman filter.

The handling of protons with low momenta needs to be improved as the Kalman filter systematically overestimates the ionization loss that starts to play a significant role at low momenta and rises with decreasing momentum. This may pull the Kalman filter further into the wrong direction if the initial estimate of the momentum is already too low. A method to obtain an initial particle hypothesis and the possibility to trace the track to the RICH- and META-detectors have to be added before the Kalman filter can be tested with real data. Some of the matrices like the projection and process noise matrix have many elements that are always zero and the propagator matrix is a band-matrix. Exploiting such structures would enhance the speed of the necessary matrix operations.

In the hope to reduce systematics in the segment fit it is planned to implement a Deterministic Annealing Filter that works directly with the drift time information. As an intermediate step the Kalman filter has been adapted to fit the drift time measurements taking at most one measurement per MDC layer into account. In a further step annealing has been added which has proven to be important for a good momentum reconstruction with wire measurements. While the asymmetry in the momentum residuals has been reduced and momentum resolution has improved for lower momenta and medium polar angles around



$30^\circ - 50^\circ$ , the dependencies on the azimuthal angle and the vast deterioration of performance at higher polar angles need to be resolved before this may be a viable option. The drift distance carries an ambiguity on which side the particle passed the sense wire. If the expected and true track state lie on opposite sides of the wire then this may pull the estimate into the wrong direction. Incorporating this ambiguity into the DAF could contribute to stabilize the fit. The probability distribution also indicates that some components of the drift time errors are still overestimated.

Finally, the program has been modified to handle competing wire measurements in an MDC layer. Competing measurements are combined into an effective measurements and the projection function vastly differs from the previous approaches. The results show that the implementation or procedure is still faulty.

The version of the Kalman filter that works with reconstructed segment hits shows some promising improvements and would be testable with real data if the remaining issues are resolved. The versions that work with wire measurements are not usable in their present state.

# Appendix A

## Settings for the Kalman Filter and Retrieving Output Data

### A.1 Settings for the Kalman Filter

The Kalman filter uses a default setting for its parameters. The default values may be modified if so desired. It is important that the Kalman filter task has been initialized before which happens automatically for all tasks that have been added to the task set of the global `Hades` object when calling `gHades->init()`. Otherwise the Kalman filter objects are not created yet. The `getTask()` function allows to retrieve the Kalman filter task:

```
HKalFilterTask *kalFilter = (HKalFilterTask*) gHades->getTask("KalmanFilterTask");
```

Now the set functions for the various options may be called:

- `setConstField(Bool_t constField)`  
Tells the Kalman filter to use a constant magnetic field instead of the HADES field map. Default is off. The value for the field strength is set separately with `setFieldVector(const TVector3 &B)`.
- `setCounterStep(Int_t c)`  
Print a notification every time  $c$  events have been filtered. Default is to print no message ( $-1$ ).
- `setDafPars(Double_t chi2cut, const Double_t *T, Int_t n)`  
Change Parameters for the deterministic annealing filter.

**chi2cut:** The  $\chi^2$ -cut-off parameter. Default: 4.

**T:** Array with annealing factors. Default:  $T = \{81, 9, 4, 1, 1\}$ .

**n**: Number of annealing iterations. Default: 5.

- **setDirection(Bool\_t dir)**  
Set the propagation direction to forward (`kIterForward = kTRUE`) or backward (`kIterBackward = kFALSE`). The direction be reversed after each iteration. Default is forward direction.
- **setDoDaf(Bool\_t daf)**  
Enable or disable the Annealing Filter. Default is on when fitting wire measurements and off when fitting reconstructed segments.
- **setDoEnerLoss(Bool\_t dedx)**  
Turn energy loss calculation on or off. Default is on.
- **setDoMultScat(Bool\_t ms)**  
Include multiple scattering in the Kalman filter. Default is on.
- **setDoPid(Int\_t pid[], Int\_t n)**  
Only particles with certain GEANT particle IDs will be filtered. This has no effect for real data. Default setting is to filter all particles.
- **setErrors(Double\_t dx, Double\_t dy, Double\_t dtx, Double\_t dty, Double\_t dmom)**

Changes the error estimates that will be used to fill the initial covariance matrix.

**dx, dy**: Position errors in sector coordinates. Default is  $0.2\text{ mm}$  in  $x$ - and  $0.1\text{ mm}$  in  $y$ -direction. This corresponds to the resolution of the inner reconstructed segment.

**dtx, dty**: Uncertainty in the direction tangents  $tx = \tan(p_x/p_z)$  and  $ty = \tan(p_y/p_z)$ . Default is  $\tan(0.5^\circ)$ .

**dmom**: Momentum uncertainty as a fraction. For example 0.1 means initial estimate is within 10% of the true momentum. Default is 5% which is roughly the uncertainty of the spline fit.

- **setFillPointsArrays(Bool\_t fill)**  
Stores the position and magnetic field strength of every Runge-Kutta step for debug purposes. Default is off.
- **setFieldMap(HMdcTrackGField \*fMap, Double\_t scale)**  
Change the field map and scaling factor.
- **setFieldVector(const TVector3 &B)**  
Set vector of magnetic field in Tesla if a constant field has been selected with `setConstField(Bool_t constField)`.

- **setFilterMethod(Kalman::filtMethod type)**  
Determines which formulation of the Kalman equations will be used. The formulations differ in speed and numerical stability.
  - Kalman::kKalConv:** Use equation 4.21b to compute the Kalman gain and 4.21c for the covariance update. This is the default value.
  - Kalman::kKalJoseph:** Joseph stabilized form of the Kalman filter that guarantees positive semi-definiteness of the covariance matrix (see chapter 4.2.4).
  - Kalman::kKalUD:** Use the numerically more stable UD-Filter [GA08, chapter 6]. The UD filter does currently not work together with the DAF.
  - Kalman::kKalSeq:** The sequential filter avoids matrix inversions (see chapter 4.2.4).
  - Kalman::kKalSwer:** The Swerling formulation of the Kalman filter (see chapter 4.2.4).
- **setNumIterations(Int\_t kalRuns)**  
Number of filter iterations. After each iteration the Kalman filter will use the result of the previous iteration as starting value and filter the measurement sites again. The direction will be reversed after each iteration. This value is not used by the DAF. Default is 1.
- **setRotation(Kalman::kalRotateOptions rotate)**  
May rotate the coordinate system before the filter process:
  - Kalman::kVarRot:** (default): Tilt the coordinate system along the initial track direction to avoid large values for the track state parameters  $t_x$  and  $t_y$ .
  - Kalman::kNoRot:** Do no coordinate transformation. This setting is not suitable for the large polar angles at HADES.

The results will be transformed back into the sector coordinate system after the Kalman filter is finished.
- **setSmoothing(Bool\_t smooth)**  
Turn smoothing on or off. Default is on.
- **setVerbose(Int\_t verb)**  
Set which kind of messages will be printed: error and warning messages (2), error messages only (1) or no messages at all (0). Default is to print all messages (2).

The Kalman library offers a rudimentary debug mode. A higher level will enable more detailed debug messages and print intermediate data. The available settings are:

- 0: Debug mode is disabled.
- 1: Will make some additional consistency checks like dimension checks when performing matrix multiplications or looking for bad elements in the covariance matrices.
- 2: Print the stage where the program currently is, for example the prediction step of the Kalman filter, a Runge-Kutta step or smoothing a measurement site.
- 3: Print calculated intermediate data like state vectors, for example.
- 4: More detailed as 3.

A debug level will perform all the tasks of the lower levels as well. By default the debug mode is turned off. To increase performance the debug mode is implemented using preprocessor directives. Therefore changing the debug mode requires recompiling the library. Debug levels can be set separately for the Kalman filter and the track propagation with a Runge-Kutta method by modifying the variables called “`kalDebug`” and “`rkDebug`”, respectively, in the file called “`hkaldef.h`”.

## A.2 Retrieving Output Data

The `Hades` object can write data relevant to the event processing to an output file. This has to be enabled by the user [sg11, chap. 2.3] and the categories `catKalTrack`, `catKalSite`, `catKalHitWire` and `catKalHit2d` must be persistent. The values for these categories are stored in `hmdctrackg.def`.

The category `catKalTrack` holds objects of class `HKalTrack`. Information about the track can be obtained with the following methods:

- `Bool_t getAccepted()`  
Returns if track reconstruction has been successful.
- `Float_t getChi2()`  
Return the  $\chi^2$  of the reconstructed track.
- `Float_t getDafChi2Cut()`  
The cut-off parameter used by the DAF.
- `Float_t getDafT(Int_t i)`  
The annealing factor used in the  $i$ -th iteration of the DAF. Only values of the last `getMaxNdafs()` iterations are stored.
- `Bool_t getDirection()`  
The propagation direction of the track. Returns `kIterForward = kTRUE` for downstream and `kIterBackward = kFALSE` for upstream.

- `Bool_t getDoDaf()`  
Returns `kTRUE` if the DAF has been used.
- `Bool_t getDoMultScat()`  
Track reconstruction included multiple scattering.
- `Bool_t getDoEnerLoss()`  
Track reconstruction included energy loss.
- `Int_t getIdxFirst()`  
Index of the measurement site that is closest to the target.
- `Int_t getIdxLast()`  
Index of the measurement site that is furthest away from the target.
- `Bool_t getIsSegHit()`  
The Kalman filter worked with reconstructed segment hits.
- `Bool_t getIsWireHit()`  
The Kalman filter/DAF worked with MDC drift times.
- `Int_t getMaxNdafs()`  
The maximum amount of DAF annealing factors for which results will be stored. The DAF itself can run with more annealing factors, but then the results of the first iterations will not be saved.
- `Float_t getMomInput()`  
Initial momentum estimate.
- `Int_t getNdafs()`  
The number of annealing factors for the DAF. This can be either lower than or equal to `getMaxNdafs()`.
- `Float_t getNdf()`  
Number degrees of freedom.
- `Int_t getNiter()`  
Number of Kalman filter iterations.
- `Float_t getTrackLength()`  
Track length in mm.
- `Int_t getPid()`  
The GEANT particle ID for simulation data or the initial particle hypothesis in case of real data.

The class `HKalTrack` inherits from `HBaseTrack`. Currently the following members are filled with data:

- `Float_t getBeta()`  
Relativistic velocity evaluated at the measurement site most upstream.

- `Float_t getMass2()`  
Squared mass derived from the initial particle hypothesis.
- `Float_t getP()`  
`Float_t getErrP()`  
Smoothed momentum in  $MeV/c$  of the particle and variance at the first measurement site. This momentum will be  $-1$  if the standard settings have been modified so that the Kalman filter did not execute the smoothing.
- `Float_t getPhi()`  
`Float_t getErrPhi()`  
Azimuthal angle  $\phi$  and its error in sector coordinates from the inner track segment. The angle is given in radians.
- `Char_t getPolarity()`  
Particle charge derived from initial particle hypothesis.
- `Float_t getR()`  
`Float_t getErrR()`  
The  $r$ -coordinate and its error in the sector coordinate system in mm.
- `Char_t getSector()`  
The HADES sector number (0...5).
- `Float_t getTheta()`  
`Float_t getErrTheta()`  
Polar angle  $\Theta$  and its error in sector coordinates from the inner track segment. The angle is given in radians.
- `Float_t getZ()`  
`Float_t getErrZ()`  
The  $z$ -coordinate and its error in the sector coordinate system in mm.

The functions `getIdxFirst()` and `getIdxLast()` return indices to `HKalSite` objects which may be retrieved from the category `catKalSite`. A site is an observation point like an MDC layer. The information a site contains are:

- `Int_t getSec()`  
Index of sector (0...5).
- `Int_t getMod()`  
Index of MDC module (0...3).
- `Int_t getLay()`  
Index of MDC layer (0...5).
- `Float_t getEnerLoss()`  
Calculated energy loss up to this site.

- `Int_t getIdxFirst()`  
Index of site's first competing hit.
- `Int_t getIdxLast()`  
Index of site's last competing hit.
- `Float_t getMomFilt()`  
Momentum from filter step.
- `Float_t getMomReal()`  
Momentum from simulation data.
- `Float_t getMomSmoo()`  
Smoothed momentum.
- `Int_t getNcomp()`  
Number of competing hits.
- `Int_t getTrackNum()`  
Geant track number.

A site may contain several competing hits. There are two different categories depending on the hit type the Kalman filter worked with: `fCatKalHit2d` for reconstructed segment hits and `fCatKalHitWire` for drift time measurements. It is therefore important to use the correct category. The correct hit type can be obtained from the functions `HKalTrack::getIsSegHit()` or `HKalTrack::getIsWireHit()`.

The category `fCatKalHit2d` returns `HKalHit2d` objects that contain position information. Since the points are located on the virtual segment planes only two coordinates are stored:

- `Float_t getXmeas()`  
Measured position in sector coordinates in mm.
- `Float_t getYmeas()`  
Measured position in sector coordinates in mm.
- `Float_t getXreco()`  
Reconstructed position in sector coordinates in mm.
- `Float_t getYreco()`  
Reconstructed position in sector coordinates in mm.

If the Kalman filter worked with drift time measurements the hit information is written to an `HKalHitWire` object.

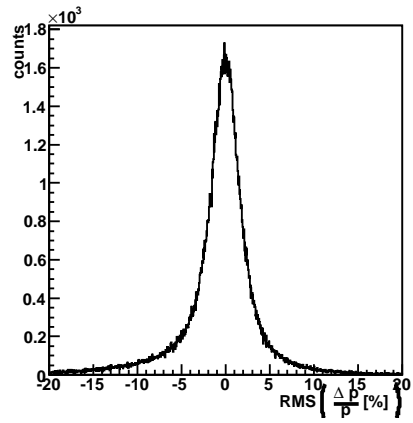
- `Float_t getAlpha()`  
Reconstructed angle  $\alpha$  of track and wire plane in degrees.  $0 < \alpha < 90^\circ$ .
- `Int_t getCell()`  
Drift cell number.



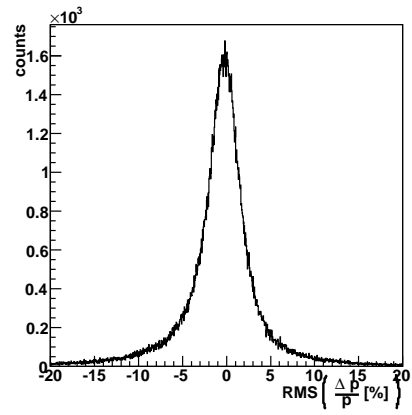
- `Float_t getChi2Daf()`  
Hit's  $\chi^2$ , i.e. the weighted, squared distance of the measurement to the smoothed track state.
- `Float_t getMinDist()`  
Reconstructed drift radius in mm.
- `Int_t getNdafs()`  
The maximum amount of DAF iterations for which results will be stored.
- `Float_t getTime1()`  
Drift time of first hit in ns. This is equivalent to `time1` from `HMdcCal1Sim`.
- `Float_t getTime2()`  
Drift time of second hit in ns. This is equivalent to `time2` from `HMdcCal1Sim`.
- `Float_t getTime1Err()`  
Error of time 1 from `HMdcCal1Sim` in ns.
- `Float_t getTimeMeas()`  
Measured time. Input for the Kalman filter. Includes `tWireOffset` and `tTof`.
- `Float_t getTimeReco()`  
Reconstructed drift `time1` in ns. Calculated from `alpha` and `mindist`.
- `Float_t getTimeTof()`  
Time of flight correction of `time1` used by the Kalman filter.
- `Float_t getTimeTofCal1()`  
Time of flight correction of `time1` from `HMdcCal1Sim`.
- `Float_t getTimeWireOffset()`  
Travel time on wire correction of `time1`.
- `Float_t getWeight(Int_t i)`  
Assignment probabilities in the  $i$ -th DAF iteration.

## Appendix B

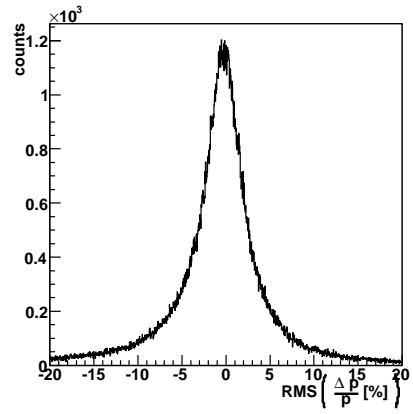
# Additional Plots



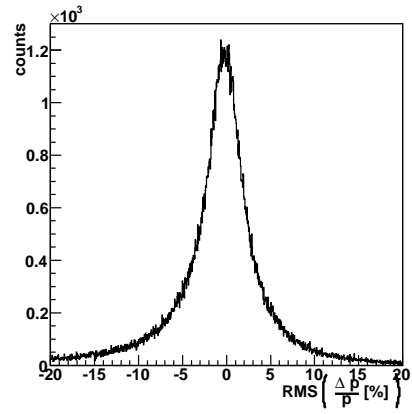
(a) Global fit



(b) Kalman Filter for segment hits

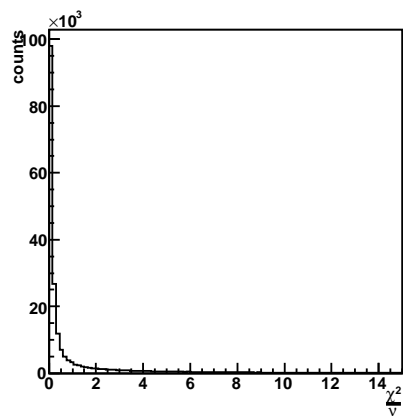


(c) Kalman filter for single wire hits

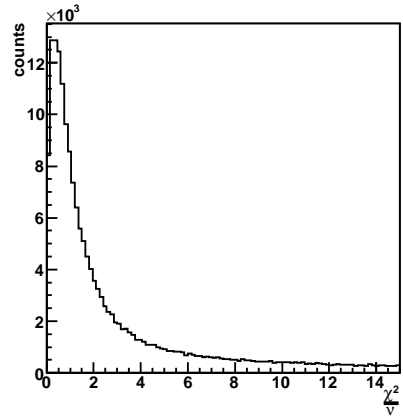


(d) DAF for single wire hits

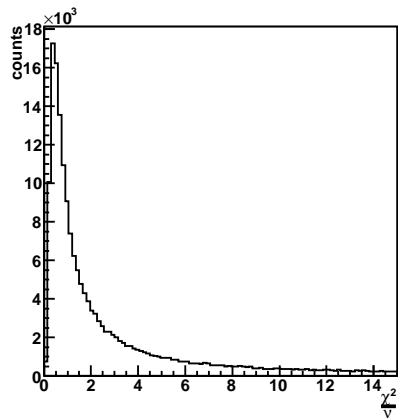
Figure B.1: Comparison of the momentum resolution for positron tracks.



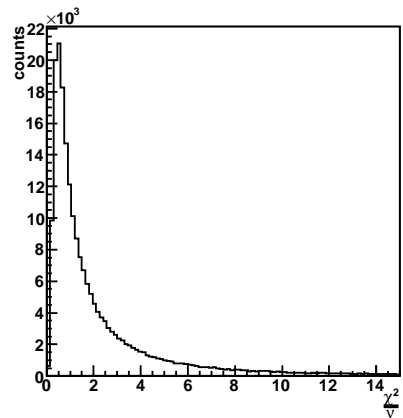
(a) Global fit



(b) Kalman filter

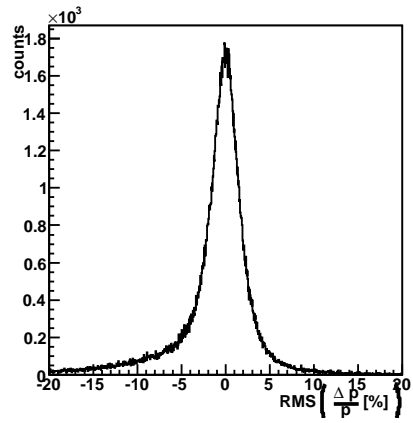


(c) Kalman filter for single wire hits

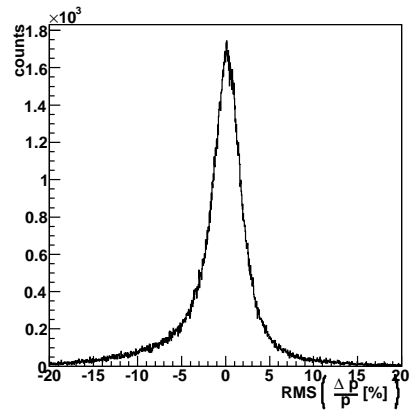


(d) DAF for single wire hits

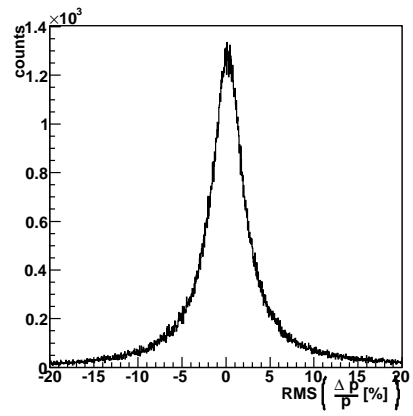
Figure B.2: Comparison of  $\chi^2$  distributions for positron tracks.



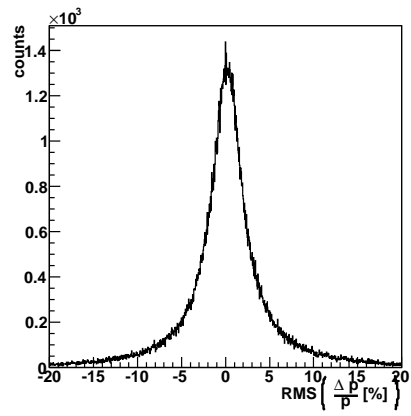
(a) Global fit



(b) Kalman filter

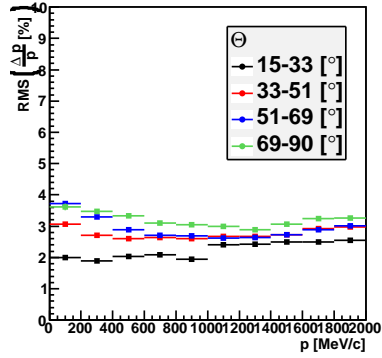


(c) Kalman filter for single wire hits

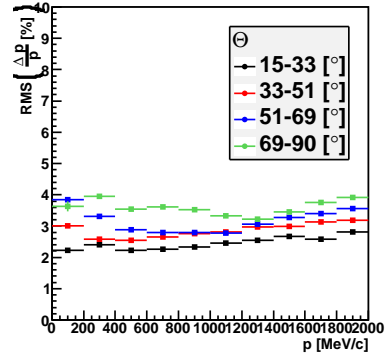


(d) DAF for single wire hits

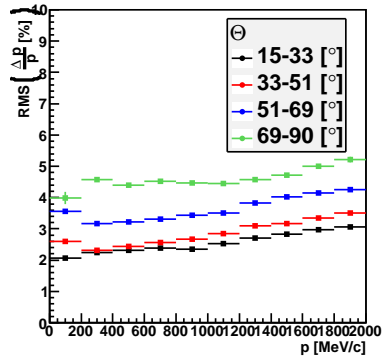
Figure B.3: Comparison of the momentum resolution for electron tracks.



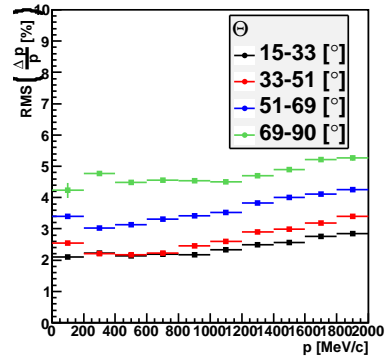
(a) Global fit



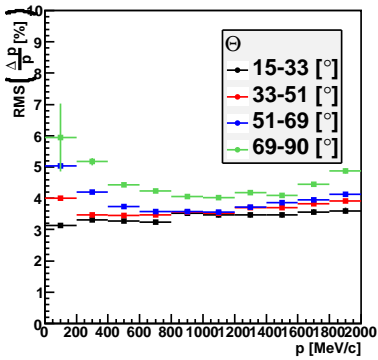
(b) Kalman filter



(c) Kalman filter for single wire hits

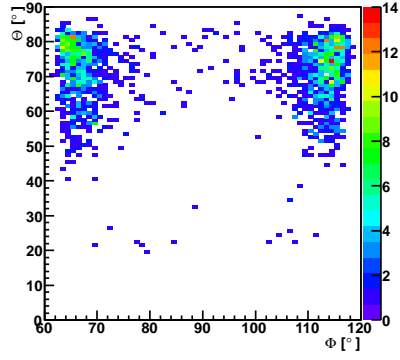


(d) DAF for single wire hits

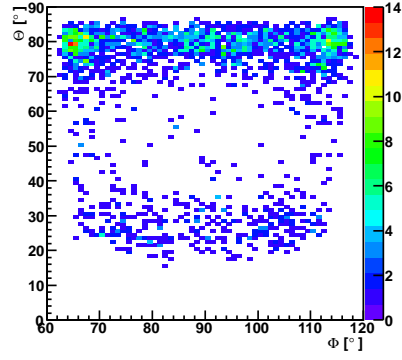


(e) DAF

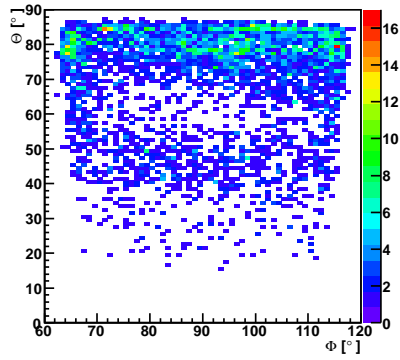
Figure B.4: Comparison of the momentum resolution for different  $\Theta$  ranges for electrons.



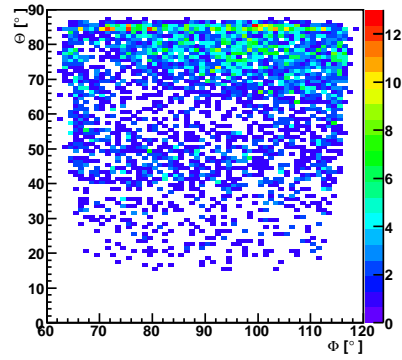
(a) Global fit



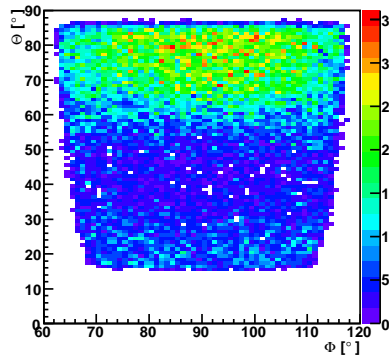
(b) Kalman filter



(c) Kalman filter for single wire hits

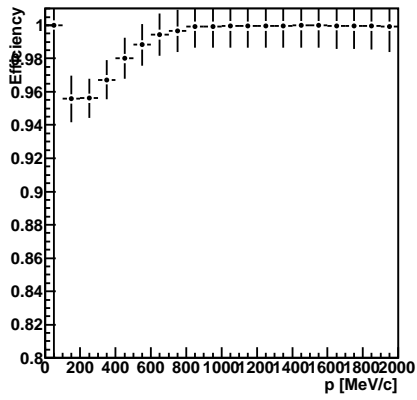


(d) DAF for single wire hits

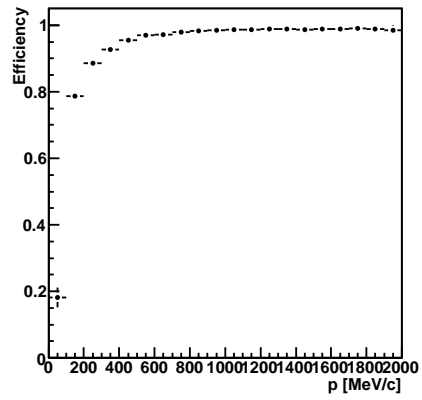


(e) DAF

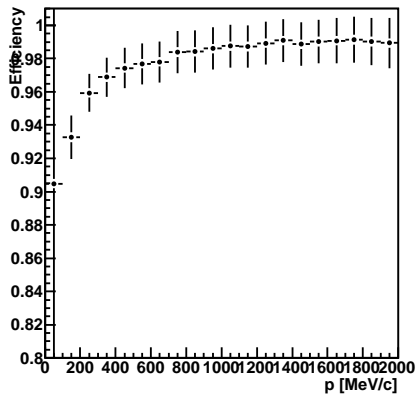
Figure B.5: The polar angle  $\Theta$  over the azimuthal angle  $\phi$  for electron tracks that could not be reconstructed or were cut due to high  $\chi^2$ .



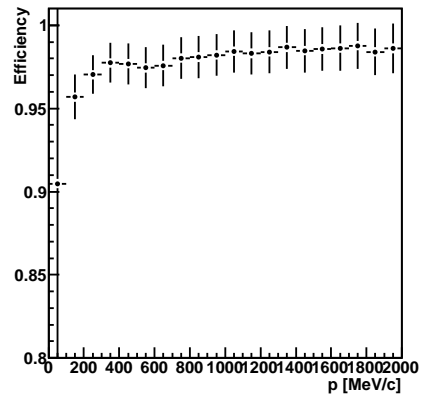
(a) Global fit



(b) Kalman filter



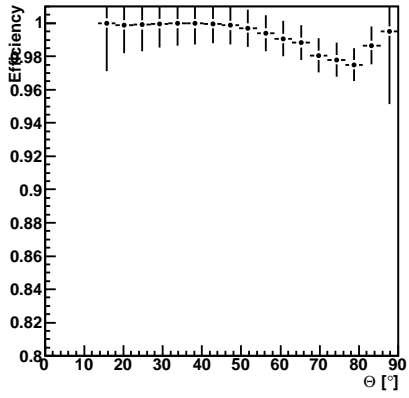
(c) Kalman filter for single wire hits



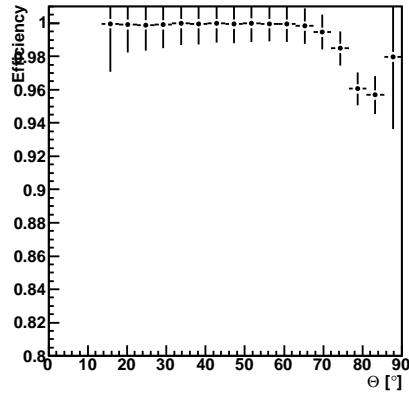
(d) DAF for single wire hits

Figure B.6: Efficiency over momentum for electron tracks.

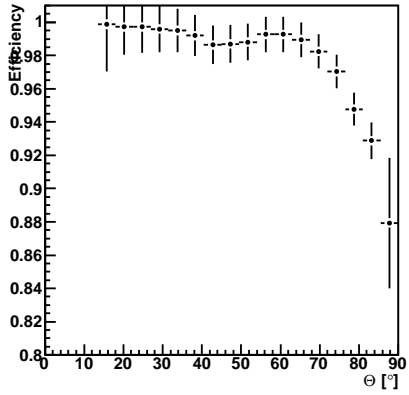




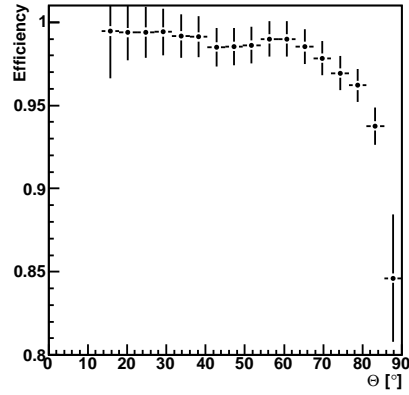
(a) Global fit



(b) Kalman filter

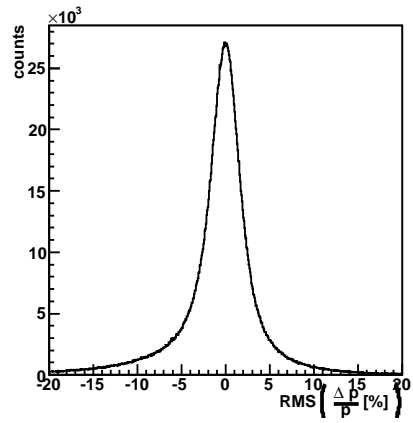


(c) Kalman filter for single wire hits

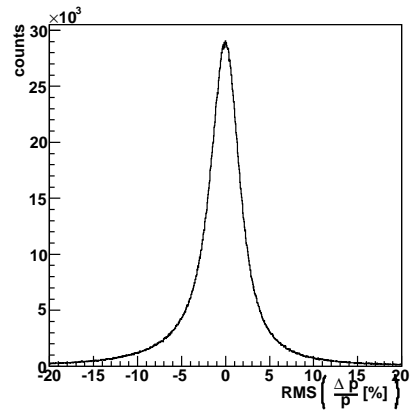


(d) DAF for single wire hits

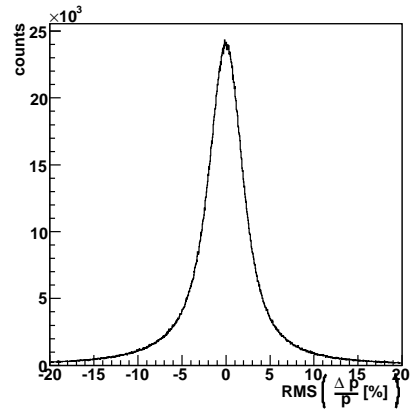
Figure B.7: Efficiency over the polar angle  $\Theta$  for electron tracks.



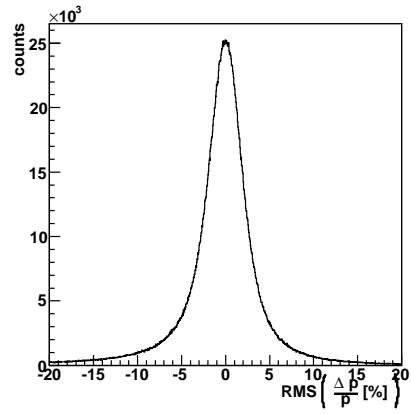
(a) Global fit



(b) Kalman filter

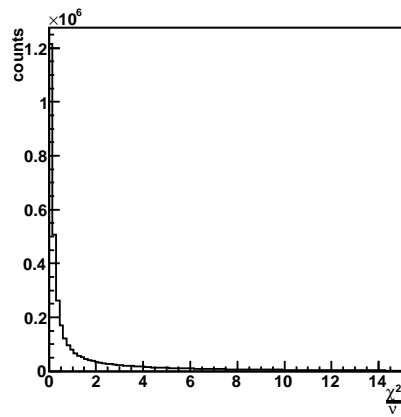


(c) Kalman filter for single wire hits

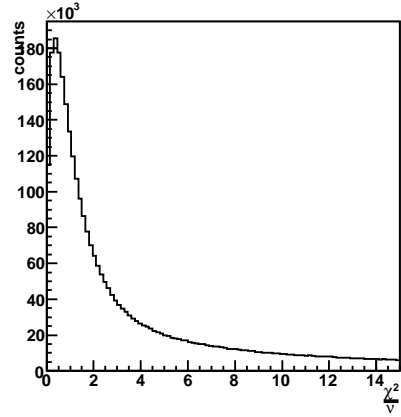


(d) DAF for single wire hits

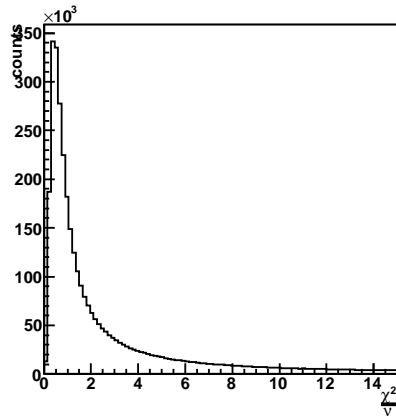
Figure B.8: Comparison of the momentum resolution for proton tracks.



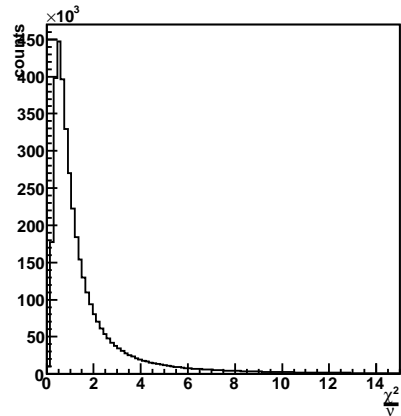
(a) Global fit



(b) Kalman filter

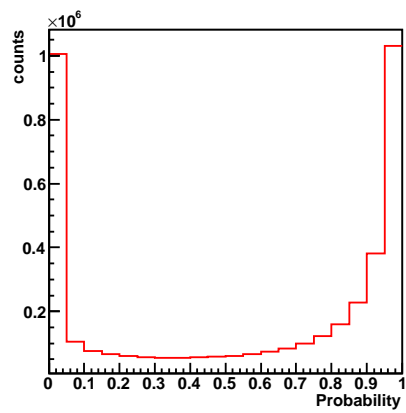


(c) Kalman filter for single wire hits

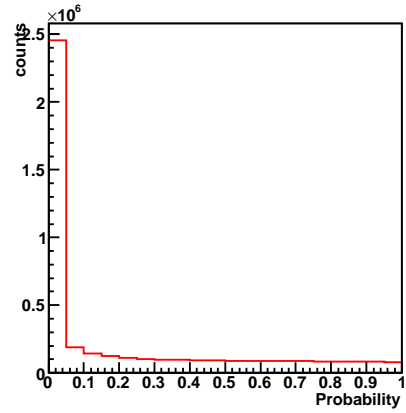


(d) DAF

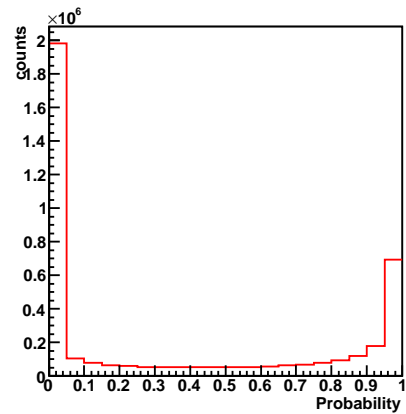
Figure B.9: Comparison of  $\chi^2$  distributions for proton tracks.



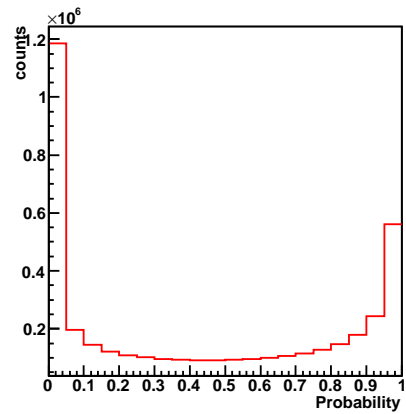
(a) Global fit



(b) Kalman filter



(c) Kalman filter for single wire hits



(d) DAF for single wire hits

Figure B.10: Probability distributions for proton tracks.

# Bibliography

- [A<sup>+</sup>02] C. Agodi et al. The HADES time-of-flight. *Nucl. Instrum. and Meth. in Physics Research A*, 492:14–25, 2002. 14
- [B<sup>+</sup>04] A. Balanda et al. The HADES Pre-Shower detector. *Nucl. Instrum. and Meth. in Physics Research A*, 531:445–458, 2004. 15
- [B<sup>+</sup>09] D. Belver et al. The HADES RPC inner TOF wall. *Nucl. Instrum. and Meth. in Physics Research A*, 602:687–690, 2009. 11, 15
- [BJ68] R.S. Bucy and P.D. Joseph. *Filtering for stochastic processes with applications to guidance*. John Wiley & Sons, New York, 1968. 32
- [BSGM06] I.N. Bronstein, K.A. Semendjajew, G.Musoil, and H. Mühlig. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, 6 edition, 2005, 2006. ISBN-13 978-3-8171-2006-2. 31
- [CER93] CERN Program Library Long Writeup W5013. *GEANT3 Detector description and simulation tool*, 1993. 48, 57, 71
- [CER00] CERN. *GARFIELD. Simulation of gaseous detectors.*, 7.02 edition, 2000. Online User Guide. 47
- [C.L00] C.Lippmann. Aufbau und Inbetriebnahme eines Gasqualitätsmonitors für die HADES-Driftkammern. Master’s thesis, Goethe-Universität, Frankfurt am Main, 2000. 18
- [DB08] Peter Deuffhard and Folkmar Bornemann. *Gewöhnliche Differentialgleichungen*, volume 2 of *Numierische Mathematik*. Walter de Gruyter, 3 edition, 2008. ISBN 978-3-11-020356-1. 37, 75
- [Ebe01] David Eberly. *3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics*. Morgan Kaufmann, 2001. ISBN 1558605932. 52
- [EPIM97] Heinz Erzberger, Russell A. Paielli, Douglas R. Isaacson, and Michelle M.Eshowl. Conflict detection and resolution in the presence of prediction error. In *Air Traffic Management Research and Development Seminar*, number 1, Saclay, France, June 1997. 55

- [Fle06] Sebastian Fleischmann. Track reconstruction in the ATLAS experiment - the Deterministic Annealing Filter. Master's thesis, Bergische Universität Wuppertal, 10 2006. 34
- [Frü87] Rudolf Frühwirth. Application of Kalman filtering to track and vertex fitting. *Nucl. Instr. and Meth. in Physics Research A*, 262:444–450, 1987. 29
- [FS99] Rudolf Frühwirth and Are Strandlie. Track fitting with ambiguities and noise: a study of elastic tracking and nonlinear filters. *Computer Physics Communications*, (120):197–214, 1999. 33
- [FS06] Rudolf Frühwirth and Are Strandlie. Application of adaptive filters to track finding. *Nucl. Instr. and Meth. in Physics Research A*, 559:162–166, 2006. 33
- [FSW00] P. Fonte, A. Smirnitski, and M.C.S. Williams. A new high-resolution TOF technology. *Nucl. Instr. and Meth. in Physics Research A*, 443:201–204, 2000. 15
- [Fuj] Keisuke Fujii. *Extended Kalman Filter*. The ACFA Sim-J Group. 28, 29, 31
- [GA08] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering Theory and Practice Using MATLAB*. John Wiley & Sons, 3 edition, 2008. ISBN 978-0-470-17366-4. 32, 33, 75, 106
- [Gal09] Tetyana Galatyuk. *Di-electron spectroscopy in HADES and CBM: from  $p + p$  and  $n + p$  collisions at GSI to  $Au + Au$  collisions at FAIR*. PhD thesis, Goethe-Universität, Frankfurt am Main, 2009. 9
- [Gal12] Tetyana Galatyuk, 03 2012. Personal note. 3, 40
- [GK06] S. Gorbunov and I. Kisel. An analytic formula for track extrapolation in an inhomogenous magnetic field. *Nucl. Instr. and Meth, A(559):148–152*, 2006. 40, 45
- [HAD09] The HADES Collaboration. The high-acceptance dielectron spectrometer HADES. *Eur. Phys. J.*, A(41):243–277, 2009. 3, 11, 17, 18, 21, 22, 23, 46
- [Kal60] Rudolph E. Kalman. A new approach to linear filtering and prediction problems. *ASME Journal of Basic Engineering*, 82:35–45, 1960. 26, 32
- [Klo11] Peter Kloeden. Numerical methods for differential equations, 2011. Lecture script. 37
- [L<sup>+</sup>09] K. Lapidus et al. Dielectron production in pp and dp collisions at 1.25 GeV/u with HADES. *arXiv:0904.1128*, 2009. 20

- [M<sup>+</sup>04] Christian Müntz et al. The HADES tracking system. *Nuclear Instruments and Methods in Physics Research A*, 535:242–246, 2004. 16
- [Man95] Rainer Mankel. Application of the kalman filter technique in the HERA-B track reconstruction. *Public HERA-B Notes*, 1995. 51
- [Man99] Rainer Mankel. ranger - a pattern recognition algorithm for the HERA-B main tracking system. *Public HERA-B Notes*, (98-079), 1999. 40, 61
- [Man04] Rainer Mankel. Pattern recognition and event reconstruction in particle physics experiments. *Rep. Prog. Phys.*, 67(4), 2004. 59, 60, 61, 62
- [Mar05] Jochen Markert. *Untersuchung zum Ansprechverhalten der Vieldraht-Driftkammern niedriger Massenbelegung des HADES Experimentes*. PhD thesis, Goethe-Universität, Frankfurt am Main, 2005. 3, 4, 9, 17, 18, 46, 47, 48, 58
- [Mar11a] Jochen Markert, 11 2011. Personal note. 22
- [Mar11b] Jochen Markert, 12 2011. Personal note. 46
- [MS85] Leonard A. McGee and Stanley F. Schmidt. Discovery of the Kalman filter as a practical tool for aerospace and industry. *NASA Technical Memorandum 86847*, 1985. 26
- [Mün10] Christian Müntz, 07 2010. Personal note. 6, 57, 59
- [N<sup>+</sup>10] K. Nakamura et al. Passage of particles through matter. *JPG*, 37(075021), 2010. 59, 60, 61, 79
- [Ore82] Jay Orear. Notes on statistics for physicists, revised, 1982. 31
- [PTVF07] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *NUMERICAL RECIPES The Art of Scientific Computing*. Cambridge University Press, 2007. ISBN-13 978-0-521-88068-8. 7, 31, 37, 92
- [RTS65] H. Rauch, F. Tung, and C. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, 1965. 29
- [Run95] Carl Runge. Über die numerische Auflösung von Differentialgleichungen. *Math. Ann.*, 46:167–178, 1895. 37
- [Rus06] Anar Rustamov. *Exclusive  $\eta$  Meson Reconstruction in Proton-Proton Collisions at 2.2 GeV with the HADES Spectrometer and High Resolution Tracking*. PhD thesis, Technische Universität Darmstadt, 2006. 3, 41

- [SB82] Stephen J. Seltzer and Martin J. Berger. Evaluation of the collision stopping power of elements and compounds for electrons and positrons. *Int. J. Appl. Radiat. Isot.*, 33:1189–1218, 1982. 58
- [SC81] R. Santonico and R. Cardarelli. Development of resistive plate counters. *Nuclear Instruments and Methods*, (187):377–380, 1981. 15
- [sg11] HADES software group. *HYDRA2 manual*, 2011. 63, 67, 107
- [Sim06] Dan Simon. *Optimal State Estimation Kalman,  $H_\infty$ , and Nonlinear Approaches*. John Wiley & Sons, 2006. ISBN-13 978-0-471-70858-2. 26, 28, 29, 30, 32, 33, 47
- [SRF94] D. Stampfer, M. Regler, and R. Frühwirth. Track fitting with energy loss. *Computer Physics Communications*, 79:157–164, 1994. 60, 62
- [Swe59] Peter Swerling. First order error propagation in a stagewise differential smoothing procedure for satellite observations. *Journal of Astronomical Sciences*, 6:46–52, 1959. 32
- [Tar10] Attilio Tarantola. *Dielectron analysis in p-p collisions at 3.5 GeV with the HADES spectrometer:  $\omega$ -meson line shape and a new electronics readout for the Multi-wire Drift Chambers*. PhD thesis, Goethe-Universität, Frankfurt am Main, 2010. 3, 13, 16
- [UrQ04] The UrQMD Collaboration. *The UrQMD Model*, 2004. <http://urqmd.org/>. 71
- [Win02] Matthias Winkler. *A comparative study of track reconstruction methods in the context of CMS physics*. PhD thesis, Technische Universität Wien, 2002. 35
- [Z<sup>+</sup>96] E. Cerron Zeballos et al. A new type of resistive plate chamber: the multigap RPC. *Nuclear Instruments and Methods in Physics Research A*, 374:132–135, 1996. 15
- [Z<sup>+</sup>99] K. Zeitelhack et al. The HADES RICH detector. *Nuclear Instruments and Methods in Physics Research A*, 433:201–206, 1999. 13



## Danksagung

Ich danke Prof. Dr. Joachim Stroth für die Aufnahme in seine Arbeitsgruppe und die Möglichkeit eine Arbeit über dieses interessante Thema schreiben zu können.

Mein ganz besonderer Dank gilt Dr. Jochen Markert, der mir immer eine unersetzliche Hilfe war.

Außerdem danke ich der Arbeitsgruppe und meinen Arbeitskollegen im Büro für die angenehme Atmosphäre.

## Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst, keine anderen als die angegebenen Hilfsmittel verwendet und sämtliche Stellen, die benutzten Werken im Wortlaut oder dem Sinne nach entnommen sind, mit Quellen- bzw. Herkunftsangaben kenntlich gemacht habe.

Frankfurt am Main, den