

Fachhochschule Aachen  
Campus Jülich

**Fachrichtung:** Scientific Programming  
**Fachbereich:** Medizintechnik und Technomathematik

Entwicklung eines  
Spurfindalgorithmus für das Forward  
Tracking System des  
 $\overline{\text{P}}\text{ANDA}$ -Detektors

Bachelorarbeit  
von  
**Felix Kibellus**  
Mat.Nr.: 4006197

Jülich, den 21. August 2016



# Eigenständigkeitserklärung

Diese Arbeit wurde von mir selbstständig angefertigt und verfasst. Es sind keine anderen Quellen als die angegebenen Quellen und Hilfsmittel verwendet worden.

---

Felix Kibellus  
Jülich, 21. August 2016

Diese Arbeit wurde betreut von:

- 1. Prüfer:** Prof. Dr. A. Terstegge, Fachhochschule Aachen, Campus Jülich
- 2. Prüfer:** G. Sterzenbach, Forschungszentrum Jülich GmbH, IKP-1

Diese Bachelorarbeit wurde in Kooperation mit der Forschungszentrum Jülich GmbH am Institut für Kernphysik (IKP) angefertigt.

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
<b>2. Die Beschleunigeranlage FAIR unter besonderer Betrachtung des <math>\bar{\text{PANDA}}</math>-Detektors</b>	<b>2</b>
2.1. Die Beschleunigeranlage FAIR . . . . .	2
2.1.1. An FAIR geplante Experimente . . . . .	2
2.1.2. Aufbau der Beschleunigeranlage . . . . .	3
2.2. Das $\bar{\text{PANDA}}$ -Experiment . . . . .	4
2.2.1. Physikalische Bedeutung von $\bar{\text{PANDA}}$ . . . . .	4
2.2.2. Aufbau des $\bar{\text{PANDA}}$ -Detektors . . . . .	5
2.2.3. Aufbau des Forward Tracking System . . . . .	7
2.2.4. Rekonstruktion von Teilchenflugbahnen mittels Straw-Tubes	10
<b>3. Das Framework PandaRoot</b>	<b>14</b>
3.1. ROOT . . . . .	14
3.1.1. Cint . . . . .	15
3.1.2. ROOT-Dateien . . . . .	15
3.2. FairRoot . . . . .	16
3.3. PandaRoot . . . . .	16
3.3.1. Der Simulationsprozess von PandaRoot . . . . .	16
3.3.2. Integration von eigenem Code in PandaRoot . . . . .	20
3.3.3. Visualisierung der Events . . . . .	21
<b>4. Beschreibung des Algorithmus</b>	<b>23</b>
4.1. Beschreibung des Verfahrens . . . . .	23
4.1.1. Gruppierung von Hits . . . . .	24
4.1.2. Approximierung der Teilchenflugbahnen außerhalb des Magnetfeldes . . . . .	26
4.1.3. Kombination der Tracks innerhalb einer Station . . . . .	30
4.1.4. Kombination der Teiltracks zwischen den Stationen . . . . .	36
4.1.5. Approximierung der Teilchenflugbahnen innerhalb des Magnetfeldes . . . . .	37
4.2. Hinzufügen von nicht gefundenen Hits . . . . .	40
4.3. Implementierung in C++ . . . . .	41
4.3.1. Übersicht der wichtigsten Klassen . . . . .	41

4.3.2.	Integration in PandaRoot . . . . .	44
<b>5.</b>	<b>Qualitätsanalyse</b>	<b>46</b>
5.1.	Realisierung der Qualitätskontrolle . . . . .	46
5.2.	Parameterstudien zur Wahl der Ähnlichkeitsparameter . . . . .	47
5.2.1.	Grenzwerte für das Generieren der Approximationen mit Isochronen (Verfahrensschritt 2) . . . . .	48
5.2.2.	Grenzwerte für das Zusammenfassen von Approximationen (Verfahrensschritt 3 und 4) . . . . .	50
5.2.3.	Grenzwerte für den Way-Follower (Verfahrensschritt 5) . . . . .	53
5.2.4.	Grenzwerte für das nachträgliche Hinzufügen von Hits (Verfahrensschritt 6) . . . . .	53
5.3.	Statistische Ergebnisse aus der Qualitätskontrolle . . . . .	54
5.4.	Gründe für fehlerhafte Rekonstruktionen . . . . .	55
5.4.1.	Tracks mit niedrigem Impuls . . . . .	55
5.4.2.	Events mit einer großen Anzahl Tracks . . . . .	58
5.4.3.	Tracks mit wenigen Hits . . . . .	60
5.4.4.	Statistische Ergebnisse unter Ausschluss der problematischen Tracks . . . . .	63
5.5.	Fehleruntersuchung in Y-Richtung . . . . .	65
<b>6.</b>	<b>Fazit und Ausblick</b>	<b>71</b>
<b>A.</b>	<b>Anhang</b>	<b>73</b>
A.0.1.	Beschreibung der implementierten Klassen . . . . .	73

# 1. Einleitung

Ziel der Hadronenphysik ist es, die Eigenschaft der Starken Kraft, welche Hadronen zusammenhält, besser zu verstehen. Neben den als Bestandteil von Atomkernen bekannten Nukleonen existieren zahlreiche weitere Hadronen. Diese kommen jedoch in der Natur nicht stabil vor, da sie nur unter erheblicher Energiezuführung entstehen und danach sehr schnell wieder zerfallen. Da diese Hadronen jedoch physikalisch interessant sind, werden sie in Teilchenbeschleunigern unter sehr hoher Energiezuführung erzeugt und untersucht. Beim  $\bar{\text{P}}\text{ANDA}$ -Experiment wird beispielsweise ein hochenergetischer Antiprotonenstrahl auf ein Targetmaterial geschossen. Dabei entstehen Sekundärteilchen, welche durch das umliegende Detektorsystem fliegen. Durch die Untersuchung dieser Sekundärteilchen können Rückschlüsse auf die Primärreaktion gezogen werden. Dabei sind die Kenngrößen Impuls, Energie, Art und Spur der resultierenden Teilchen von Interesse. Aus diesem Grund ist der Stoßpunkt von einem komplexen Detektorsystem umschlossen, in dem die einzelnen Sub-Detektoren auf die verschiedenen Rekonstruktionsaspekte spezialisiert sind. Das Forward Tracking System ist ein Sub-Detektor, welcher zur Rekonstruktion von Teilchenflugbahnen in Richtung des Antiprotonenstrahls eingesetzt wird. Im Rahmen dieser Arbeit wurde ein Verfahren zur Auswertung des Forward Tracking System implementiert. Das Forward Tracking System besteht aus mehreren Schichten unterschiedlich angeordneter Straw-Tubes, welche zur Erfassung der Teilchen genutzt werden. Um dem Aufbau des Detektors gerecht zu werden, rekonstruiert das implementierte Verfahren die Teilchenflugbahnen zunächst innerhalb dieser Schichten und kombiniert die so erhaltenen Spuren dann nach und nach zu einer kompletten Teilchenflugbahn. Diese rekonstruierten Teilchenflugbahnen werden Track genannt. Da das  $\bar{\text{P}}\text{ANDA}$ -Experiment Teil der Beschleunigeranlage FAIR ist, wird auf diese im folgenden Kapitel eingegangen. Anschließend wird der  $\bar{\text{P}}\text{ANDA}$ -Detektor thematisiert und dabei insbesondere das Forward-Tracking-System betrachtet. Danach wird die Softwareumgebung vorgestellt, welche zur Simulation und Auswertung des  $\bar{\text{P}}\text{ANDA}$ -Experiments benutzt wird. Das implementierte Verfahren wird im Anschluss daran vorgestellt. Abschließend wird erläutert, wie die Qualität des implementierten Verfahrens untersucht wurde.

## 2. Die Beschleunigeranlage FAIR unter besonderer Betrachtung des $\bar{\text{P}}\text{ANDA}$ -Detektors

Das implementierte Verfahren dient als Auswertungs-Algorithmus für das Forward Tracking System (FTS) des  $\bar{\text{P}}\text{ANDA}$ -Detektors. Da zum Verständnis des Verfahrens zunächst der Detektoraufbau eingeführt werden muss, wird in diesem Kapitel der  $\bar{\text{P}}\text{ANDA}$ -Detektor thematisiert. Der  $\bar{\text{P}}\text{ANDA}$ -Detektor ist ein Teil der Beschleunigeranlage FAIR. Im Folgenden wird deshalb zunächst die gesamte Beschleunigeranlage beschrieben um den  $\bar{\text{P}}\text{ANDA}$ -Detektor in den Kontext der gesamten Anlage einzuordnen. Danach wird genauer auf den  $\bar{\text{P}}\text{ANDA}$ -Detektor eingegangen um schließlich das FTS zu beschreiben.

### 2.1. Die Beschleunigeranlage FAIR

Die Bezeichnung FAIR steht für *Facility for Antiproton and Ion Research*. Die Beschleunigeranlage FAIR wird derzeit in Darmstadt gebaut und beinhaltet verschiedene Experimente. Mit Hilfe dieser Experimente sollen grundlegende Fragen zur Entstehung des Universums und der Struktur von Materie geklärt werden. FAIR fokussiert sich dabei auf die Forschungsgebiete Atom-, Plasma-, Antiproton-, Kernstruktur- und Kernmateriephysik. Neben reiner Grundlagenforschung verspricht FAIR neue Erkenntnisse bei der Erzeugung hochenergetischer Ionenstrahlung, welche bei der Erforschung der Kernfusion als alternative Energiequelle dienen können. Darüber hinaus werden durch die Experimente neue Erkenntnisse im Bereich der kosmischen Strahlung erwartet. Diese stellt in der Raumfahrt, beispielsweise für eine bemannte Mars-Mission, bisher noch eine große Herausforderung dar, da kosmischen Strahlung negative Auswirkungen auf Menschen und technische Komponenten haben kann. [5, S. 3-4]

#### 2.1.1. An FAIR geplante Experimente

Im Folgenden wird kurz auf die an FAIR geplanten Experimente eingegangen:

**APPA - Atomic, Plasma Physics and Applications** Mit diesem Experiment soll Plasma bei niedrigen Temperaturen und bei hohem Druck erforscht werden.

Darüber hinaus wird mit schweren Ionen die oben erwähnte kosmische Strahlung untersucht.[5, S. 3]

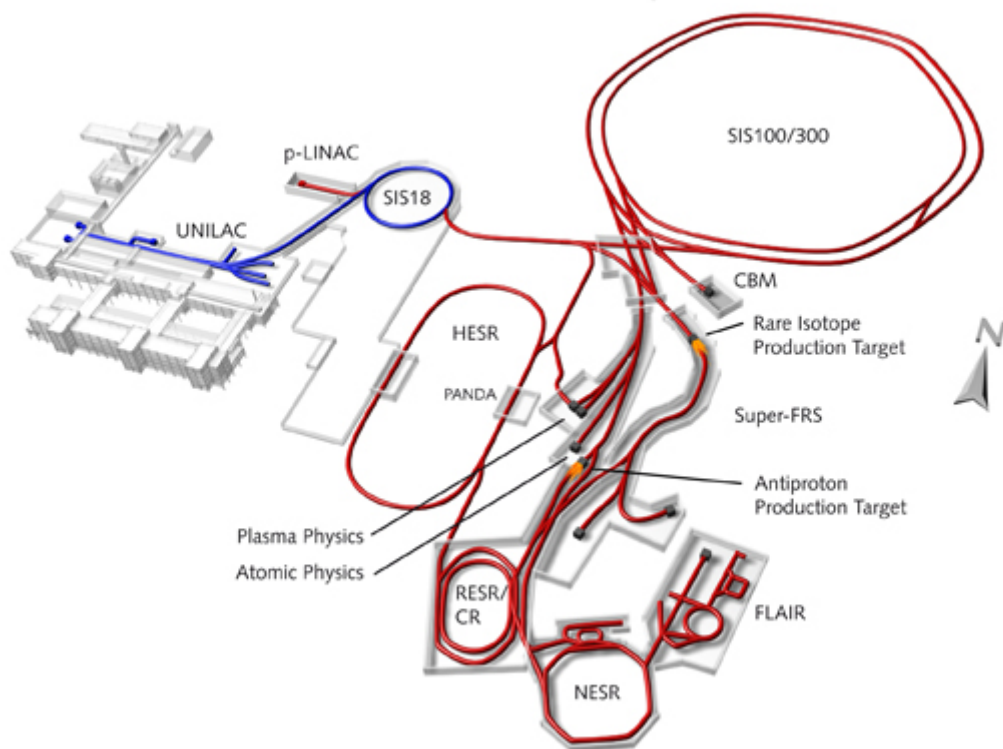
**CBM - Compressed Baryonic Matter** Um die Bedingungen in Frühstadien des Universums nachzustellen wird bei diesem Experiment mit komprimierter Kernmaterie, welche eine hohe energetische Dichte aufweist, experimentiert.[5, S. 3]

**NUSTAR - Nuclear Structure, Astrophysics and Reactions** Um astrophysikalische Phänomene zu erforschen, werden bei diesem Experiment seltene Elemente untersucht, welche schwerer sind als Eisen. Dazu wird der innere Aufbau seltener Isotope untersucht.[5, S. 4]

**PANDA - AntiProton Annihilation at Darmstadt** Die *Starke Kraft* hält Quarks in Hadronen zusammen. Um die Eigenschaften dieser Kraft genauer zu untersuchen, soll im Rahmen des PANDA-Experiments mit Antiprotonenstrahlen experimentiert werden. [5, S. 4]

### 2.1.2. Aufbau der Beschleunigeranlage

Eine schematische Abbildung der Beschleunigeranlage FAIR ist in Abbildung 2.1 zu sehen. Die in blau dargestellten Komponenten (SIS18) existieren bereits und sollen als Vorbeschleuniger eingesetzt werden. Die in rot dargestellten Komponenten befinden sich noch in der Planungs- bzw. Bauphase. SIS100 ist ein Beschleunigungsring und beschleunigt Protonen auf bis zu 29 GeV kinetische Energie. Abhängig davon, welches Experiment durchgeführt werden soll, wird der Protonenstrahl dann auf die unterschiedlichen Experimente verteilt. Dazu schließen sich mehrere Möglichkeiten der Strahlführung an. Für das PANDA-Experiment werden Antiprotonen benötigt. Da diese in der Natur nicht über längere Zeit vorkommen, müssen sie zunächst erzeugt werden. Dies ist darauf zurückzuführen, dass es beim Kontakt mit Materie sofort zu einer Annihilation kommt, bei der sich Materie und Antimaterie gegenseitig auslöschen. Zur Erzeugung der Antiprotonen werden deshalb die beschleunigten Protonen auf ein Wolfram-Target geschossen und die entstehenden Antiprotonen dann im *Collector Ring* (CR) gesammelt. Anschließend wird der so gewonnene Antiprotonenstrahl in den *High-Energy Storage Ring* (HESR) eingespeist und auf bis zu 14.5 GeV beschleunigt. Das PANDA-Experiment befindet sich direkt am HESR.[5, S. 4-5]



**Abbildung 2.1.:** Schematische Abbildung der Beschleunigeranlage FAIR [4]

## 2.2. Das $\bar{\text{P}}\text{ANDA}$ -Experiment

Das folgende Teilkapitel beschäftigt sich mit dem  $\bar{\text{P}}\text{ANDA}$ -Detektor. Zunächst wird auf die physikalische Bedeutung des Experiments eingegangen. Im Anschluss folgt dann eine Erläuterung des Detektoraufbaus. Das FTS, welches ein Subdetektor des  $\bar{\text{P}}\text{ANDA}$ -Detektors darstellt, wird abschließend thematisiert.

### 2.2.1. Physikalische Bedeutung von $\bar{\text{P}}\text{ANDA}$

Das  $\bar{\text{P}}\text{ANDA}$ -Experiment soll dazu dienen, Antiproton-Proton-Annihilationen und die Reaktion von Antiprotonen mit schweren Kernen zu untersuchen. Der Begriff der Annihilation beschreibt den Prozess der Paarvernichtung, bei dem ein Antiteilchen mit seinem korrespondierenden Fundamentarteilchen zusammenstößt und diese sich gegenseitig auslöschen. Dabei entstehen u.a. neuartige Teilchen, die wiederum in Sekundärteilchen zerfallen, welche vom Stoßpunkt ausgehend durch den Detektor fliegen. Da die Primärreaktion nicht direkt untersucht werden kann, müssen die entstehenden Sekundärteilchen durch ein komplexes Detektorsystem

erkannt und analysiert werden. Somit lassen sich dann Rückschlüsse auf die Primärreaktion ziehen.[5, S. 5]

Untersucht werden soll dabei die Natur der starken Wechselwirkung. Diese ist auch unter der Bezeichnung *Starke Kraft* bekannt und wird durch die Theorie der Quantenchromodynamik beschrieben. Diese beschreibt die Wechselwirkungen von Quarks und Gluonen. Quarks sind unter anderem die Bestandteile von Hadronen (Protonen und Neutronen). Gluonen dienen als Austauscheteilchen für die *Starke Kraft*. Die Quantenchromodynamik beschreibt diese Wechselwirkung jedoch nur für kleine Distanzen zwischen den Quarks. Wird die Distanz zu groß, wächst die *Starke Kraft* immer weiter an. Irgendwann wird sie so groß, dass sich aufgrund der Feldenergie zwischen den auseinandergezogenen Quarks neue Quark-Antiquark-Paare bilden. Diese formen dann neue Hadronen. Dieser Vorgang wird als *Confinement* bezeichnet und ist bisher noch nicht vollständig verstanden worden. Deshalb soll  $\bar{\text{PANDA}}$  über einen experimentellen Ansatz neue Erkenntnisse liefern. [5, S. 5-6]

## 2.2.2. Aufbau des $\bar{\text{PANDA}}$ -Detektors

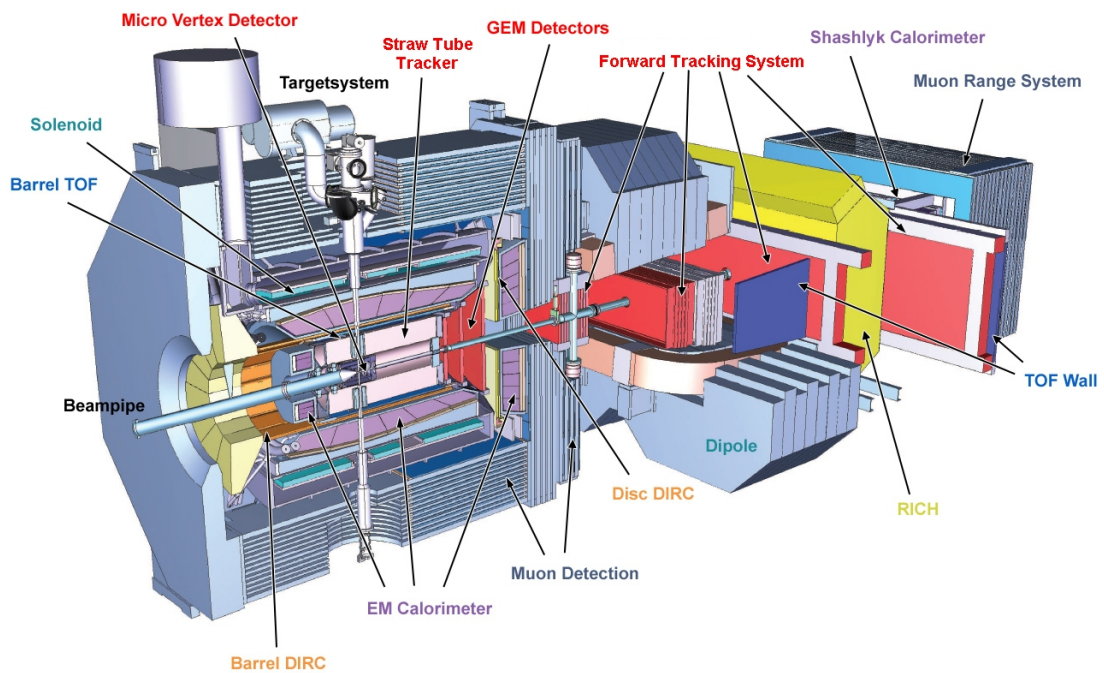


Abbildung 2.2.: Schematische Abbildung des  $\bar{\text{PANDA}}$ -Detektors [1]

Abbildung 2.2 zeigt eine schematische Abbildung des  $\bar{\text{PANDA}}$ -Detektors. Über das Strahlrohr wird der Antiprotonenstrahl vom HESR in den Detektor geführt.

Im vorderen Teil des Detektors befindet sich das Targetsystem, über welches das Target-Material zugeführt wird. Das Target-Material beinhaltet die Teilchen, mit denen der Antiprotonenstrahl kollidieren soll. Um Experimente mit verschiedenen Target-Materialien durchführen zu können, ist das Targetsystem austauschbar. Im Folgenden wird kurz auf die verschiedenen Target-Typen eingegangen [5, S. 7]:

**Pellet Target** Dieses Target führt mittels einer Düse gefrorenen Wasserstoff dem Antiprotonenstrahl zu.

**Cluster Jet Target** Durch wechselwirkende Kräfte zusammengehaltene Wasserstoff-Moleküle werden Cluster genannt. Das Cluster Jet Target wandelt über eine Düse Wasserstoffgas in einen Cluster-Strahl um, der mit dem Antiprotonenstrahl zur Kollision gebracht werden soll.

**Nuclear Target** Das Nuclear Target nutzt Draht und Folien um den Antiprotonenstrahl mit festen Materialien zur Kollision zu bringen. Damit soll die Reaktion von Antiprotonen mit schweren Kernen untersucht werden.

Der Detektor lässt sich in die beiden Teile Target- und Forward-Spektrometer aufteilen. Das Target-Spektrometer befindet sich im vorderen Teil um den Stoßpunkt herum. Das Forward-Spektrometer beinhaltet den hinteren Teil des Detektors und dient dazu Teilchen mit zu erfassen, welche mit einem kleinen Streuwinkel vom Stoßpunkt aus in Richtung der Strahlachse fliegen. Die in  $\bar{P}$ ANDA verwendeten Subdetektoren haben drei verschiedene Aufgaben. Es gibt spurgebende Detektoren, welche die Teilchenflugbahn rekonstruieren sollen. Darüber hinaus existieren Detektoren zur Energiebestimmung und Detektoren zur Bestimmung der Teilchenart. [5, S. 8-10]

Um den Impuls der Teilchen zu bestimmen wird die Lorenz-Kraft benutzt. Die Lorenzkraft beschreibt den Zusammenhang  $\vec{F}_L = q\vec{v} \times \vec{B}$ . Die Lorenzkraft  $\vec{F}_L$  wirkt auf geladene Teilchen, welche ein Magnetfeld mit der Feldstärke  $\vec{B}$  durchqueren. Bei Kenntnis der Ladung  $q$  lässt sich somit aufgrund der Ablenkung des Teilchens auf die Geschwindigkeit  $\vec{v}$  schließen. Bei Kenntnis der Masse kann dann auf den Impuls der Teilchen geschlossen werden. Um diesen Zusammenhang zu nutzen benötigt der Detektor die Magnetfelder, welche zur Ablenkung der Teilchen führen. Deshalb beinhaltet das Target-Spektrometer ein 2 T Solenoid-Magnetfeld. Im Forward-Teil des Detektors befindet sich ein 2 Tm Dipol-Magnet. [5, S. 8-10]

Zur Rekonstruktion der Teilchenflugbahn befindet sich im Target-Spektrometer der *Straw Tube Tracker* (STT), der *Micro Vertex Detector* (MVD) und der *Gas Electron Multiplier Detector* (GEM). Zur Teilchenidentifikation dient in erster Linie der *Detection-of-internally-reflected-cherenkov-light-Detektor* (DIRC) und der *Time-of-flight-Detektor* (TOF). Die verschiedenen Aufgaben der Detektoren lassen sich jedoch meist nicht immer vollständig voneinander abgrenzen. Der TOF kann beispielsweise auch zur Impulsbestimmung verwendet werden und liefert zusammen mit dem MVD auch eine Zeitinformation. Diese sogenannte  $t_0$ -Zeit wird

benötigt um den zeitlichen Beginn des Experiments zu bestimmen. Im Forward-Teil des Detektors dient das Forward Tracking System (FTS) als spurgebender Detektor. Da sich diese Arbeit mit der Rekonstruktion von Spuren im FTS beschäftigt wird darauf im späteren Verlauf des Kapitels noch weiter eingegangen. Die Teilchenidentifikation wird im Forward-Teil über den *Ring Imaging Cherenkov-Detektor* (RICH) durchgeführt. Wie im Target-Spectrometer kommen in Forward-Spectrometer auch ein TOF- und ein Myonen-Detektor zum Einsatz. Es sei noch erwähnt, dass im  $\bar{P}$ ANDA-Detektor die Z-Achse parallel zur Strahlachse verläuft und die Y-Achse nach oben zeigt. Diese Definition wird auch in der später folgenden Simulation und Auswertung in der vorliegenden Arbeit verwendet. [5, S. 8-10]

Da die jeweiligen Subdetektoren jeweils auf Spur-, Teilchenart- oder Impulsbestimmung spezialisiert sind, ist eine vollständige Rekonstruktion des Experiments nur durch eine Zusammenführung der gewonnenen Informationen möglich. Da die verschiedenen spurgebenden Detektoren jeweils nur einen Teil des Detektors abdecken, müssen die Spuren zunächst teilweise rekonstruiert und in einem späteren Schritt dann zusammengeführt werden. [5, S. 8-10]

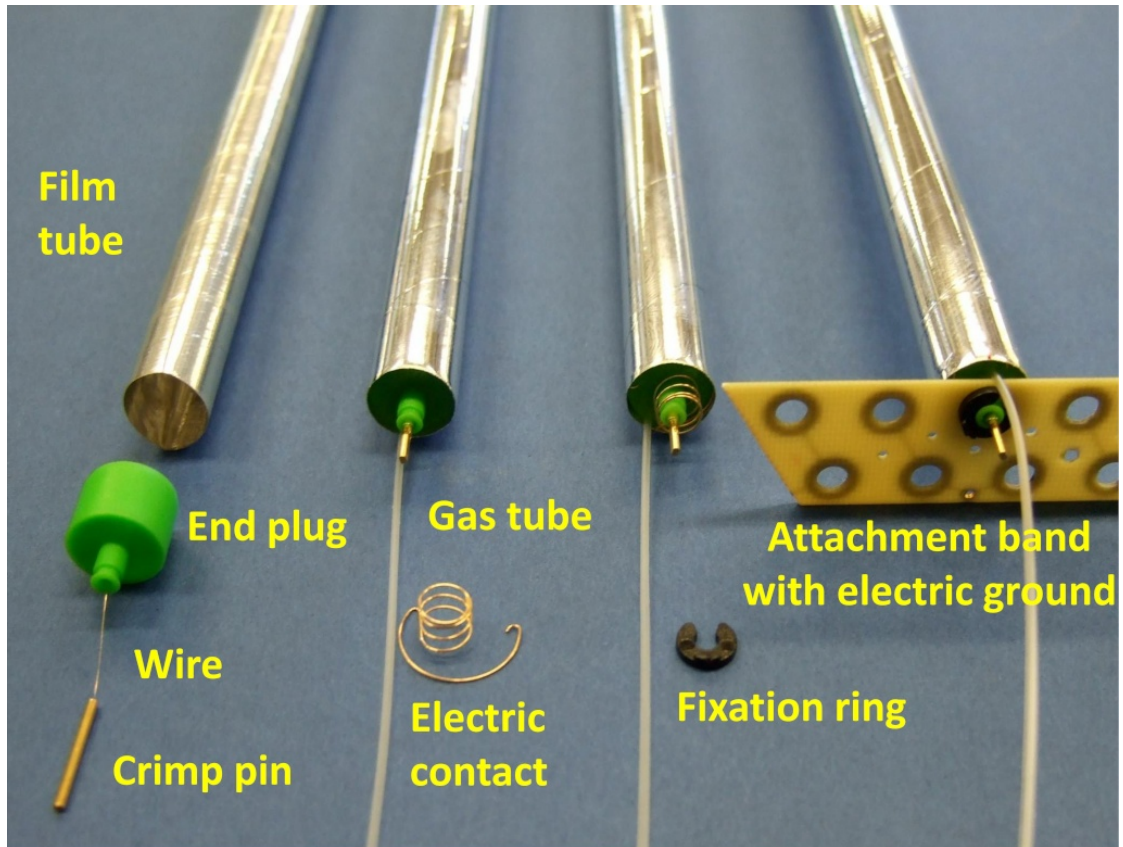
### 2.2.3. Aufbau des Forward Tracking System

Wie bereits erwähnt, dient das FTS der Rekonstruktion von Teilchenflugbahnen. Dazu nutzt es sogenannte Straw-Tubes. In diesem Teilkapitel wird zunächst die Funktionsweise von Straw-Tubes erklärt. Danach wird darauf eingegangen, wie es möglich ist mit Hilfe von Straw-Tubes Teilchenflugbahnen zu rekonstruieren. Abschließend wird dann erläutert, wie die einzelnen Straw-Tubes im FTS angeordnet sind.

#### Aufbau und Funktionsweise von Straw-Tubes

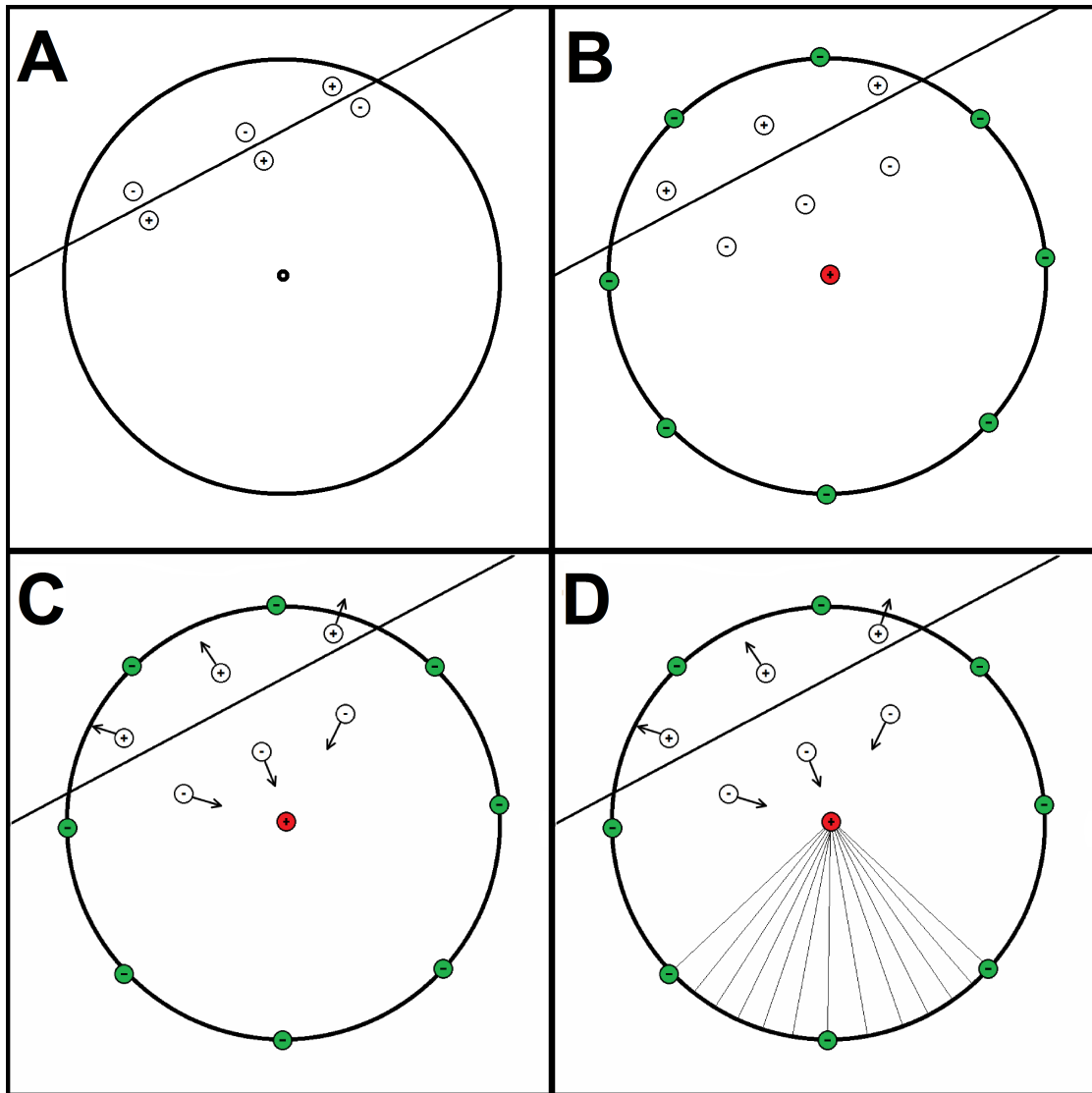
Eine in  $\bar{P}$ ANDA eingesetzte Straw-Tube besteht aus einem Röhrchen, welches 1500 mm lang ist und einen Durchmesser von 10 mm aufweist. Eine Abbildung der Straw-Tubes findet sich in Abbildung 2.3.

Die äußere Hülle der Straw-Tubes besteht aus einer Mylar-Folie der Dicke 27  $\mu\text{m}$ , welche von innen mit Aluminium beschichtet ist. Die Straw-Tubes sind mit einem Gas-Gemisch ( $\text{Ar}/\text{CO}_2$ ) gefüllt, welches einen Überdruck von 1 bar aufweist. Im Inneren der zylinderförmigen Straw-Tube verläuft ein Draht. Dieser Draht fungiert als Anode, während die leitende Innenbeschichtung als Kathode fungiert. Somit besteht zwischen Draht und Hülle ein elektrisches Feld. Durchquert ein geladenes Teilchen nun die Straw-Tube werden durch Wechselwirkungen mit dem Gas einige Elektronen der Gasteilchen abgespalten. Dadurch spalten sich die Gasteilchen zu positiv geladenen Ionen und negativ geladenen Elektronen auf, das Gas ionisiert. Die positiv geladenen Ionen werden dann durch das elektrische Feld zum Rand der Straw-Tubes gezogen. Die negativ geladenen Elektronen wandern zum Draht. Da



**Abbildung 2.3.:** Abbildung von zerlegten Straw-Tubes [3]

die Feldlinien des elektrischen Feldes in der Nähe des Drahtes sehr dicht verlaufen, werden die Elektronen dort so stark beschleunigt, dass sie weitere Elektronen von Gasteilchen abspalten können. Durch diesen Lawineneffekt kann eine Verstärkung des ursprünglichen Signals um den Faktor  $10^4$  bis  $10^5$  erreicht werden. Dadurch wird das Signal so stark, dass es messbar wird und ausgelesen werden kann. [5, S. 13-14]

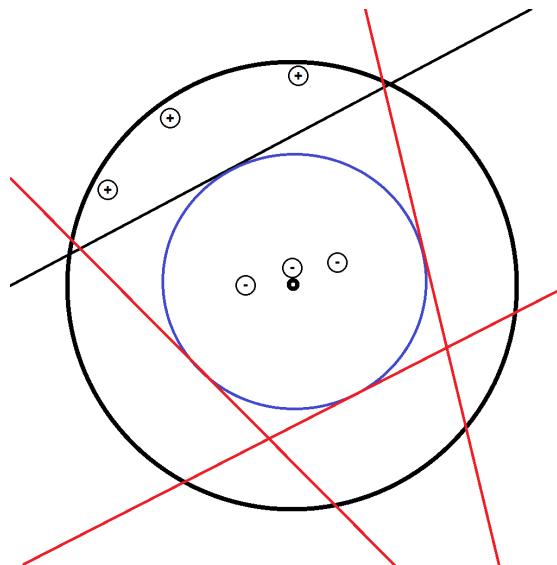


**Abbildung 2.4.:** Vorgänge in einer Straw-Tube beim Durchqueren eines geladenen Teilchens

Abbildung 2.4 verdeutlicht die Vorgänge innerhalb einer Straw-Tube beim durchqueren eines geladenen Teilchens. Dargestellt sind die Querschnitte durch die Straw-Tubes. Die schwarze Linie stellt die Flugbahn des geladenen Teilchens dar und die schwarzen Kreise die äußere Hülle der Straw-Tube. Der Draht ist durch den Punkt in der Mitte der Kreise markiert. In den Abbildungen B-D sind zusätzlich die Ladungen von Draht und Hülle markiert. Darüber hinaus zeigt Abbildung D zusätzlich den Verlauf der Feldlinien.

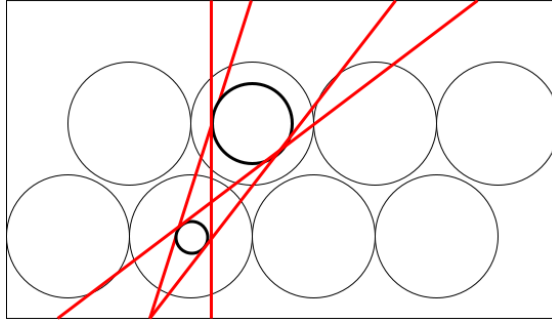
## 2.2.4. Rekonstruktion von Teilchenflugbahnen mittels Straw-Tubes

Über die bisher erläuterten Informationen lassen sich lediglich die Straw-Tubes bestimmen, welche von einem Teilchen durchquert worden sind. Um eine bessere Rekonstruktion der Teilchenflugbahnen zu ermöglichen ist es jedoch von Vorteil, den minimalen Abstand von Teilchenflugbahn und Draht zu kennen. Dieser kann über die sogenannte Driftzeit bestimmt werden. Die Driftzeit ist die Zeit, die ein freigewordenes Elektron benötigt um sich zum Draht zu bewegen um dort ein messbares Signal zu erzeugen. Mit Hilfe der Startzeit des Teilchens ( $t_0$ ) und der bekannten Driftgeschwindigkeit kann der minimale Abstand von Teilchenflugbahn und Draht bestimmt werden. Dieser Abstand wird als Isochronenradius bezeichnet. Es lässt sich nun ein Kreis konstruieren, dessen Mittelpunkt auf der Drahtposition liegt, und dessen Radius dem Isochronenradius entspricht. Dieser Kreis wird Isochrone genannt. [5, S. 14]



**Abbildung 2.5.:** Schematische Abbildung einer getroffenen Straw-Tube mit Isochronenradius und möglichen Teilchenflugbahnen

Abbildung 2.5 zeigt eine schematische Abbildung einer getroffenen Straw-Tube mit eingezeichnetem Isochron. Durch die Bestimmung des Isochronenradius lässt sich bei Betrachtung einer einzelnen Straw-Tube die Teilchenflugbahn nicht eindeutig bestimmen. Jede Tangente an den eingezeichneten Kreis stellt eine mögliche Teilchenflugbahn dar. Deshalb sind in den Straw-Tube-Detektoren mehrere Straw-Tubes in Ebenen angeordnet.



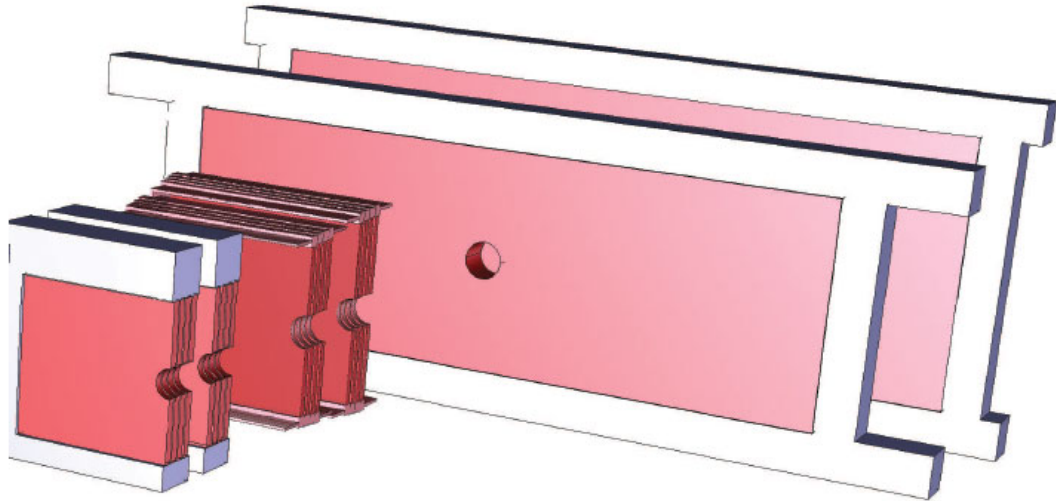
**Abbildung 2.6.:** Schematische Darstellung einer Doppelschicht Straw-Tubes mit den vier möglichen Teilchenflugbahnen, die tangential an den Isochronen liegen

Abbildung 2.6 zeigt zwei hintereinander angeordnete Ebenen von Straw-Tubes wie es beispielsweise im FTS verwendet wird. Durch die Betrachtung von zwei nebeneinander liegenden Straw-Tubes und der Einbeziehung der Isochronen ist eine genauere Bestimmung der Teilchenflugbahn möglich, falls sich die Flugbahn wie eine Gerade verhält. In diesem Fall gibt es nur noch vier mögliche Teilchenflugbahnen. Dabei handelt es sich immer um die beiden äußeren Tangenten an die Kreise und die inneren Tangenten, welche sich zwischen den Straw-Tubes kreuzen. Diese Herangehensweise wird auch im implementierten Verfahren verwendet und später erneut aufgegriffen. Bei der bisherigen Betrachtung wurde stets nur die zweidimensionale Projektion der Straw-Tubes betrachtet. Durch Betrachtung einer einzigen Straw-Tube ist es jedoch nicht möglich die Position entlang des Drahtes zu bestimmen, an der ein Teilchen die Straw-Tube durchquert hat. Dies wäre nur dann möglich, wenn sich an beiden Enden des Drahtes eine Ausleseelektronik befinden würde, dies ist jedoch bei  $\bar{\text{PANDA}}$  nicht der Fall. Es ist lediglich möglich, die Distanz von Teilchenflugbahn zum Draht zu bestimmen. Überträgt man die bisherige Betrachtung in den dreidimensionalen Raum, ergibt sich für die Menge der möglichen, minimalen Punkte, an denen das Teilchen am Draht vorbei geflogen sein könnte, ein Zylinder um den Draht herum mit einem Radius, der dem Isochronenradius entspricht. Im folgenden Teilkapitel wird unter anderem darauf eingegangen, wie durch die Betrachtung mehrerer Straw-Tubes trotz dieser Problematik eine Bestimmung der Position entlang des Drahtes durchgeführt werden kann.

### **Anordnung der Straw-Tubes im FTS**

Das FTS befindet sich im vorderen Teil des Detektors und besteht aus insgesamt sechs ähnlichen Stationen, welche hintereinander angeordnet sind. Diese Stationen

werden mit FTS1 - FTS6 bezeichnet und sind der Strahlachse entlang hintereinander angeordnet. Eine schematische Abbildung der sechs Stationen findet sich in Abbildung 2.7. Die Aussparung in der Mitte dient der Durchführung des Strahlrohrs und ist in allen sechs Stationen vorhanden. Die letzten beiden Stationen (FTS5 und FTS6) sind größer als die vorderen vier, um Teilchen mit einem größeren Streuwinkel im hinteren Teil des Detektors noch erfassen zu können. Das FTS ist in der Lage Teilchen mit einem vertikalen Streuwinkel von  $\pm 5^\circ$  und einem horizontalen Streuwinkel von  $\pm 10^\circ$  relativ zur Strahlachse zu erfassen. [6]



**Abbildung 2.7.:** Schematische Abbildung des FTS-Subdetektors

Jede Station besteht wiederum aus vier hintereinander liegenden Doppelschichten von Straw-Tubes. Innerhalb einer Doppelschicht sind alle Straw-Tubes parallel zueinander angeordnet. Die jeweils erste und die vierte Doppelschicht beinhaltet Straw-Tubes, welche parallel zur Y-Achse angeordnet sind. Die Straw-Tubes der zweiten Schicht sind um  $5^\circ$  nach rechts, die Straw-Tubes der dritten Schicht um  $5^\circ$  nach links geneigt. Somit lässt sich die bereits erwähnte Problematik beheben, dass eine einzelne Straw-Tube nur eine zweidimensionale Bestimmung der Teilchenflugbahn zulässt. Wird die Projektion in der XY-Ebene betrachtet, lässt sich über den Kreuzungspunkt zweier unterschiedlich geneigten Straw-Tubes die Position entlang des Drahtes bestimmen, in der ein Teilchen den Detektor durchquert hat. [6]

Tabelle 2.8 stellt für jede Station des FTS dar, wie viele Straw-Tubes in der jeweiligen Station verwendet werden. Die Einteilung in Module beschreibt, wie viele

Station	Module	Straw-Tubes	Z-Position [mm]	Größe $x \times y$ [mm]
FT1	$4 \times 8 = 32$	1024	2954	$1298 \times 640$
FT2	$4 \times 8 = 32$	1024	3274	$1298 \times 640$
FT3	$4 \times 12 = 48$	1536	3945	$1944 \times 690$
FT4	$4 \times 12 = 48$	1536	4385	$1944 \times 767$
FT5	$4 \times 25 = 100$	3200	6075	$4045 \times 1180$
FT6	$4 \times 37 = 148$	4736	7475	$5984 \times 1480$

**Abbildung 2.8.:** Tabelle der technischen Spezifikation des FTS [6]

Doppelschichten bzw. Pakete aus jeweils 32 Straw-Tubes in der entsprechenden Station verwendet werden. Darüber hinaus ist die Z-Position dargestellt, an der sich die jeweilige Station im Detektor befindet. Die Spalte "Größe" beinhaltet die Ausdehnung der jeweiligen Station in X- und Y-Richtung. Durch die Aussparung für das Strahlrohr in der Mitte der Stationen ist es nicht möglich eine durchgängige Straw-Tube zu verwenden. Aus diesem Grund sind dort kürzere Straw-Tubes oberhalb und unterhalb der Aussparung platziert.

## 3. Das Framework PandaRoot

Um die Vorgänge im  $\bar{P}$ ANDA-Detektor simulieren und auszuwerten zu können wird das Framework PandaRoot [7] verwendet. Dieses benutzt das übergeordnete Framework FairRoot [8], welches wiederum auf dem Framework ROOT [9] basiert. In diesem Kapitel wird zunächst ROOT eingeführt und anschließend FairRoot betrachtet. Daran anknüpfend wird dann darauf eingegangen, wie mit Hilfe von PandaRoot das  $\bar{P}$ ANDA-Experiment simuliert werden kann. Darüber hinaus wird erläutert, wie die Qualität des implementierten Verfahrens überprüft werden kann. Da das implementierte Verfahren in PandaRoot eingebettet werden soll, werden auch die dafür benötigten Schnittstellen vorgestellt.

### 3.1. ROOT

Das Framework ROOT wurde am CERN entwickelt, um Experimente aus dem Bereich der Hochenergiephysik simulieren und analysieren zu können. Inzwischen wird ROOT jedoch auch für zahlreiche, darüber hinausgehende Anwendungen verwendet, beispielsweise zum Data-Mining. ROOT bietet zahlreiche Möglichkeiten zur Erweiterung und Modifikation um genau auf das zu simulierende Experiment abgestimmt werden zu können. Zum wesentlichen Funktionsumfang von Root gehören unter anderem die folgenden Funktionalitäten [10]:

1. Erstellen von Histogrammen (2D und 3D)
2. Data management
3. Streudiagramme
4. Erstellen von Graphen
5. Fitten von Funktionen
6. Statistische Datenanalyse
7. Mathematische Standard-Funktionen
8. 3D-Visualisierung
9. Grafik-Export

Zur Simulation von komplexen Systemen wie  $\overline{\text{PANDA}}$  besitzt ROOT die Schnittstelle VirtualMC. Die Abkürzung **MC** steht dabei für Monte-Carlo. Die Monte-Carlo-Verfahren werden verwendet um den Transport der Teilchen durch den Detektor zu simulieren. Die Simulation des primären Stoßes wird über Event-Generatoren simuliert. Die einzelnen Simulationsschritte werden später noch genauer erläutert. VirtualMC kapselt den Zugriff auf den eigentlichen Simulationsalgorithmus, sodass dieser bei Bedarf einfach ausgetauscht werden kann. Für  $\overline{\text{PANDA}}$  wird dazu der Programmpaket Geant3 bzw. 4 [2] verwendet. Dabei kommen verschiedene Strategien zum Einsatz. Beispielsweise würde die im vorangegangenen Kapitel erwähnte Bahnkurve eines geladenen Teilchens im Magnetfeld mittels Lösen von Bewegungsgleichungen simuliert werden. Andererseits wird bei der Simulation von stark zufallsbasierten Ereignissen, wie der Zerfall eines Teilchens, ein stochastischer Ansatz, die sogenannte Monte-Carlo-Methode, zu Grunde gelegt. [5, S. 16]

### 3.1.1. Cint

Mit ROOT kann über den Interpreter Cint interagiert werden. Dabei handelt es sich um einen C++-Interpreter, der in der Lage ist C++-Code in einem interaktiven Modus ohne Kompilation auszuführen. Dies hat den Vorteil, dass ROOT schnell und einfach mit verschiedenen Konfigurationen gestartet werden kann. Wenn ROOT öfters mit der selben Konfiguration gestartet werden soll, kann der C++-Code auch in einer Datei gespeichert werden, welche ROOT dann zur Ausführung übergeben werden kann. Diese Dateien werden Macros genannt. Über Macros kann die zur Simulation benötigte Detektorgeometrie ausgewählt werden und Analysealgorithmen gestartet werden. Es ist üblich, verschiedene Aufgaben in unterschiedlichen Macros zu definieren. In Kapitel 3.3.1 wird genauer auf eine übliche Aufteilung der verschiedenen Aufgaben in Macros eingegangen. Diese Macros werden jedoch nur zur Steuerung der eigentlichen Verfahren verwendet. Die implementierten Verfahren werden als Bibliotheken kompiliert und dann über die Macros gestartet. Somit können die Vorteile aus schnellem kompilierten Code und dynamischem interpretiertem Code miteinander kombiniert werden.[5, S. 16]

### 3.1.2. ROOT-Dateien

Zur persistenten Speicherung von Daten verwendet ROOT ein spezielles Dateiformat. Sogenannte ROOT-Dateien verwenden eine baumartige Struktur, in der Daten abgelegt werden können. Dazu müssen Root-Branches erzeugt werden, welche dann mit Daten gefüllt werden können. Über Root-Dateien können Algorithmen unterschiedlicher Macros miteinander kommunizieren. Dazu schreibt ein Algorithmus Daten in einen bestimmten Root-Branch. Ein weiterer Algorithmus liest die Daten dann aus dem Root-Branch um diese weiter verarbeiten zu können. Ein Spurfindungs-Algorithmus liest ebenfalls die vom Detektor oder einer Simulation

erzeugten Messwerte aus einem Root-Branch. Die rekonstruierten Teilchenflugbahnen werden dann in einen weiteren Root-Branch geschrieben.

## 3.2. FairRoot

FairRoot basiert auf ROOT und stellt ein gemeinsames Programmiergerüst für alle FAIR-Experimente zur Verfügung. Beispielsweise existiert im FairRoot-Framework die Klasse FairHit, welche dazu verwendet wird den Treffer eines Teilchens in einem beliebigen Detektor zu repräsentieren. Ein FairHit enthält Informationen über die Position des Treffers, den Fehler und die DetektorID. In PandaRoot lassen sich die Klassen PndFtsHit und PndSttHit finden. Diese sind von FairHit abgeleitet und spezialisieren diese Klasse dahingehend, dass sie weitergehende Informationen enthalten, welche im STT bzw. dem FTS zur Verfügung stehen. Alle Klassen, welche zu FairRoot gehören beginnen mit dem Präfix "Fair". Die ROOT zugehörigen Klassen beginnen im Gegensatz dazu mit dem Präfix "T", die Klassen in PandaRoot beginnen mit "Pnd". Am Klassennamen kann also direkt das zugehörige Framework abgelesen werden.

## 3.3. PandaRoot

In den vorangegangenen Abschnitten wurde bereits die Aufteilung unterschiedlicher Aufgaben in Macros und die Kommunikation über Root-Dateien erwähnt. Im folgenden Teilkapitel wird nun die übliche Aufteilung dieser Aufgaben in PandaRoot thematisiert, um den Ablauf von Simulation und Auswertung zu skizzieren.

### 3.3.1. Der Simulationsprozess von PandaRoot

Der Simulationsprozess von PandaRoot wird üblicherweise wie folgt aufgeteilt:

1. Generieren von Events
2. Transport
3. Digitalisierung
4. Rekonstruktion
5. Physikalische Analyse

Dabei wird jeder Simulations- und Auswertungsschritt in einem eigenen Macro definiert, welches das jeweilige Verfahren in PandaRoot anstößt. Prinzipiell ist es natürlich auch möglich, mehrere Aufgaben in dem selben Macro zu definieren.

Dies ist jedoch meist nicht sinnvoll. Sollen beispielsweise mit den selben simulierten Teilchenflugbahnen mehrere, verschiedene Rekonstruktionsversuche gestartet werden bietet sich eine strikte Aufteilung in verschiedenen Macros an.

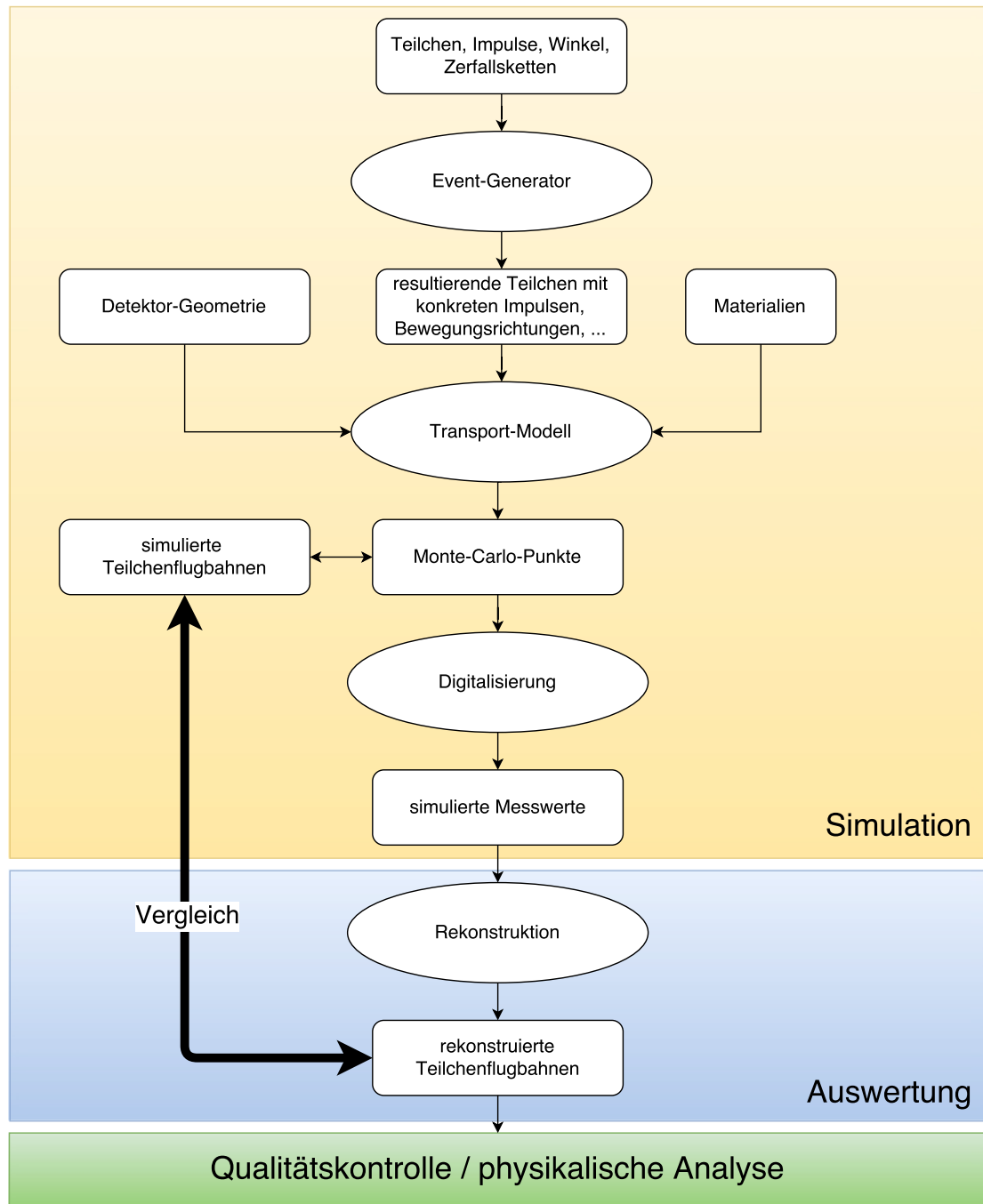


Abbildung 3.1.: Der Simulationsprozess in PandaRoot

Abbildung 3.1 zeigt die Aufteilung der verschiedenen Arbeitsschritte und welche Daten dabei erzeugt oder benötigt werden. Die ersten drei Schritte simulieren das physikalische Experiment. In der sich anschließenden Rekonstruktion wird versucht, das Experiment aufgrund der vom Detektor produzierten Messungen zu rekonstruieren und die entstandenen Teilchen zu charakterisieren. Abschließend werden die rekonstruierten Daten dann im Hinblick auf physikalisch interessante Erkenntnisse untersucht. Darüber hinaus ist es möglich über eine Qualitätskontrolle die Effektivität der in der Rekonstruktion implementierten Verfahren zu überprüfen. Diese Arbeitsschritte werden nun genauer erläutert.

## Generieren von Events

Zu Beginn wird die primäre physikalische Reaktion des Antiprotonenstrahls mit dem Targetmaterial simuliert. Diese Simulation kann auf verschiedene Arten parametrisiert werden. Damit eine große Vielfalt physikalisch verschiedener Experimente simuliert werden kann, gibt es unterschiedliche Event-Generatoren. Auf die wichtigsten Event-Generatoren wird nun kurz eingegangen:

**Box-Generator** Mit diesem Event-Generator können gezielt bestimmte Teilchen in eine definierbare Richtung geschossen werden. Der Winkel- und Impulsbereich der zu erzeugenden Teilchen lässt sich parametrisieren. Der Box-Generator erzeugt dann Teilchen, welche über diese Bereiche gleichverteilte Charakteristika aufweisen.

**EvtGen** Dieser Event-Generator ist in der Lage komplexe Zerfallsketten zu simulieren. Diese können auch im Vorfeld vom Benutzer definiert werden, sodass der Event-Generator darauf zurückgreifen kann.

**DPM** Antiproton-Proton-Kollisionen können mit dem Dual-Parton-Modell simuliert werden. Die Simulation des DPM-Generator stimmt zu großen Teilen mit den Vorgängen überein, welche sich bei einem realen Experiment auch im PANDA-Detektor abspielen würden. Aus diesem Grund sollte die Qualitätskontrolle eines implementierten Rekonstruktionsverfahren immer mit dem DPM-Generator durchgeführt werden.

Der benötigte Event-Generator kann im entsprechenden Macro ausgewählt und parametrisiert werden. Es ist zu beachten, dass bei der Event-Generation nur die resultierenden Teilchen der Primärreaktion mit entsprechenden Impulsen und Bewegungsrichtungen simuliert werden. Die simulierten Teilchen befinden sich zu diesem Zeitpunkt also noch am Stoßpunkt und fliegen noch nicht durch den Detektor. [5, S. 16-17]

## Transport-Modell

Die Flugbahn der Teilchen durch den Detektor wird im Transport-Modell simuliert. Als Eingabedaten dienen dabei neben den in der Event-Generation simulierten Teilchen auch die Detektor-Geometrie und die im Detektor verwendeten Materialien. Zur Simulation der Teilchenflugbahnen (MCTracks) kommt das Monte-Carlo-Programm Geant zum Einsatz, welches unter Berücksichtigung der Detektor-Geometrie sogenannte Monte-Carlo-Punkte erzeugt. Diese Punkte werden an den Stellen erzeugt, an denen ein Teilchen das Detektormaterial passiert hat. Dabei werden auch Wechselwirkungen mit dem Detektor in die Simulation mit einbezogen. Die bisher betrachteten Simulationsschritte simulieren das eigentliche physikalische Experiment. [5, S. 16]

## Digitalisierung

Die Simulation der Ausleseelektronik findet in der Digitalisierung statt. Dazu werden die zuvor simulierten Teilchenflugbahnen und Monte-Carlo-Punkte benutzt. In diesem Schritt werden genau die Daten erzeugt, welche der Detektor auch bei einem realen Experiment liefern würde. Dabei werden auch Messungenauigkeiten mit einbezogen und in die Simulation integriert. Beispielsweise kann das gemessene Signal nicht stark genug sein, um einen definierten Schwellwert zu überschreiten. In diesem Fall würde der betreffende Detektor das Teilchen nicht wahrnehmen. Die Digitalisierung im FTS erzeugt sogenannte Hits. Dies sind die Punkte, an denen ein Subdetektor ein Teilchen gemessen hat. Bei einem Straw-Tube-Tracker werden die Hits genau in der Mitte der getroffenen Straw-Tube, also in der Mitte des inneren Drahtes, erzeugt. Wichtig ist, dass diese Hits im Gegensatz zu den Monte-Carlo-Punkten nicht genau auf der simulierten Teilchenflugbahn liegen. Die bisher betrachteten Simulationsschritte simulieren den kompletten Detektor und lassen sich durch ein reales Experiment ersetzen. Für die sich anschließenden Verfahren sollte es also egal sein, ob die Eingabedaten von einem realen Experiment oder einer Simulation produziert worden sind. [5, S. 16]

## Rekonstruktion

An die Digitalisierung schließt sich die Rekonstruktion an. Hierbei kommen Spurfindungs-Algorithmen zum Einsatz, welche versuchen die gemessenen Hits zu Tracks zu gruppieren. Die sich anschließenden Spurfittings-Algorithmen versuchen dann über verschiedene mathematischen Methoden die Teilchenflugbahn zu approximieren. Oftmals lassen sich jedoch Spurfindung und Spurfitting nicht klar trennen und werden im selben Algorithmus implementiert. Bei dem im Rahmen dieser Arbeit implementierten Verfahren handelt es sich ebenfalls um ein Spurfindungs- und Spurfitting-Verfahren, welches im Bereich der Rekonstruktion eingesetzt werden soll. Zunächst werden die Teilchenflugbahnen nur in den einzelnen Sub-Detektoren gesucht. In einem späteren Verfahren werden diese Teilergebnisse dann zu globalen

Tracks zusammengeführt. Anschließend werden diese Rekonstruktionen zu ihren gemeinsamen Ursprungspunkt zurückverfolgt. Alle Tracks, welche aus der selben Rekonstruktion hervorgegangen sind werden dann zu einem Vertex zusammengefasst. Diese Vertices stellen zusammen mit den rekonstruierten Tracks die Ausgabe der Rekonstruktion dar. Darüber hinaus existieren auch Rekonstruktionsverfahren, welche Teilchenart, Impuls und Energie der rekonstruierten Teilchen bestimmen. [5, S. 17-18]

### Physikalische Analyse

Die rekonstruierten Teilchen und Events können in der sich anschließenden Analysephase dann hinsichtlich bestimmter Eigenschaften genauer untersucht werden. Dabei wird auch ermittelt, aus welchen primären Reaktionen die gemessenen Sekundärteilchen hervorgegangen sind um so Rückschlüsse auf das ursprüngliche Ereignis ziehen zu können. [5, S. 18]

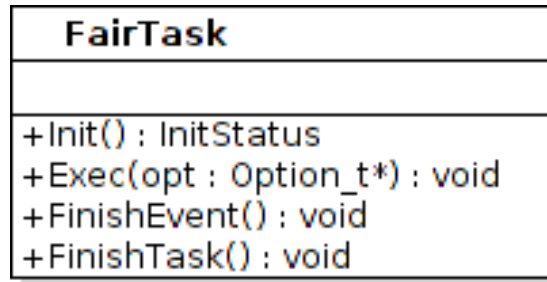
### Qualitätskontrolle

Um die Qualität der in Rahmen der Rekonstruktion implementierten Verfahren zu überprüfen, sollte eine Qualitätskontrolle durchgeführt werden. Dies ist relativ einfach möglich, wenn die simulierten Daten aus Event-Generation und Transport-Modell vorliegen. Die dort simulierten Teilchenflugbahnen sollen in der Rekonstruktionsphase rekonstruiert werden. Die Qualität dieser Rekonstruktion lässt sich somit einfach durch einen Vergleich der rekonstruierten Teilchenflugbahnen mit den simulierten Teilchenflugbahnen bestimmen (vgl. Abbildung 3.1). FairRoot bietet die Möglichkeit über die Klasse FairLink, ausgehend von den zur Rekonstruktion verwendeten Hits auf die zugrunde liegenden MCTracks zurückzugreifen. Dazu enthält ein FairLink den benötigten Root-Branch und die ID des MCTracks innerhalb des Branches. Eine Qualitätskontrolle bei einem realen Experiment ist nicht möglich, da hier die korrekten Teilchenflugbahnen natürlich nicht bekannt sind. [5, S. 18-20]

### 3.3.2. Integration von eigenem Code in PandaRoot

Das im Rahmen dieser Arbeit implementierte Verfahren soll in den Rekonstruktionsschritt von PandaRoot integriert werden. Dazu muss zunächst die benötigte Schnittstelle betrachtet werden. In FairRoot befindet sich die dafür benötigte Klasse FairTask. Um eigenen Code in PandaRoot integrieren zu können, muss eine eigene Unterklasse von FairTask implementiert werden. Ein UML-Diagramm, welches die wichtigsten Methoden der Klasse FairTask enthält findet sich in Abbildung 3.2.

Die Methoden der Klasse FairTask müssen überschrieben und mit entsprechendem eigenen Code gefüllt werden. Die Methode **Init** wird einmal zu Beginn des Verfah-



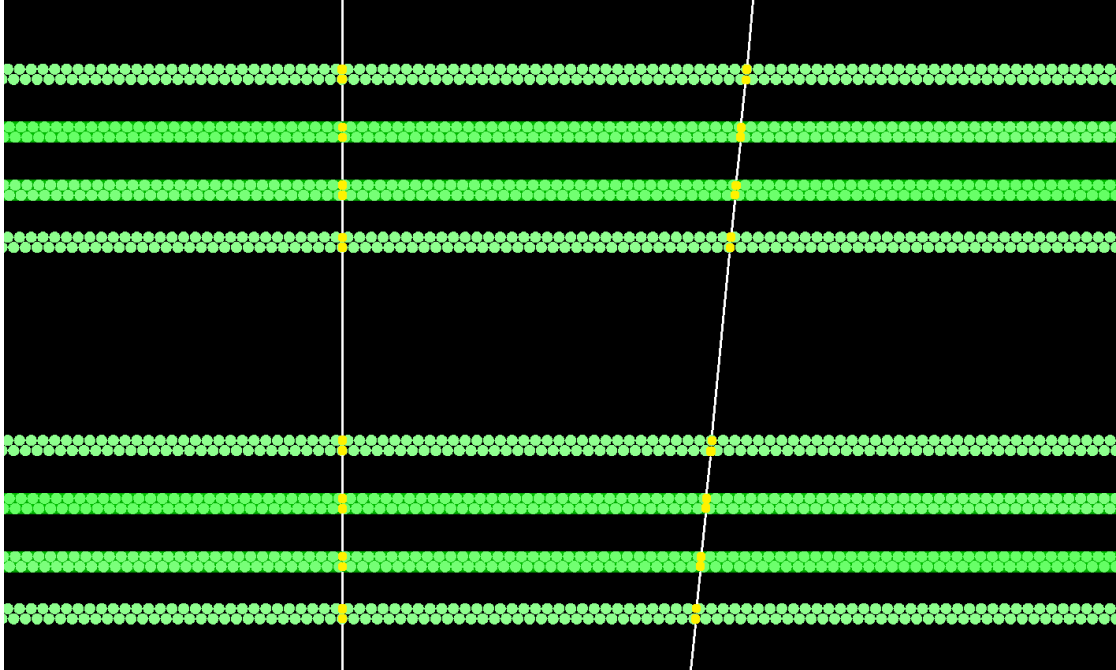
**Abbildung 3.2.:** UML-Diagramm der Klasse FairTask aus FairRoot

rens aufgerufen und dient beispielsweise der Initialisierung benötigter Datenstrukturen. Für jedes Event wird anschließend einmal die Methode **Exec** aufgerufen. Bei einem Event handelt es sich um eine simulierte Primärreaktion. Für jedes Event muss anschließend innerhalb der Exec-Methode das Rekonstruktionsverfahren gestartet werden. Die rekonstruierten Teilchenflugbahnen können dann in der Methode **FinishEvent** in einen Root-Branch geschrieben werden. Diese Methode wird jedes mal am Ende eines Events aufgerufen. Hier sollten auch alle verwendeten Datenstrukturen geleert werden. Sind alle Events bearbeitet worden, wird die Methode **FinishTask** aufgerufen. In dieser Methode sollten die verwendeten Datenstrukturen wieder freigegeben werden. Ein explizites Wechseln des Event ist nicht nötig. Wurde ein Event bearbeitet, wird automatisch die Exec-Methode für das nächste Event aufgerufen. Im entsprechenden Root-Macro für die Rekonstruktion muss ein Objekt des eigenen FairTasks erstellt werden. Dies kann über die Methode **AddTask** dem **FairRootManager** zur Ausführung übergeben werden, sodass beim Ausführen des Macros das eigene Verfahren integriert werden kann. Im Gegensatz zu den Macros kann der als FairTask implementierte Code jedoch kompiliert werden und lässt sich dann über ein interpretiertes Macro aufrufen. [5, S. 32]

### 3.3.3. Visualisierung der Events

Um die simulierten Events graphisch anzuzeigen, kann das in ROOT integrierte Programm Eve verwendet werden. Eve steht für **Event Visualization Environment** und ist in der Lage, den PANDA-Detektor in einer 3D-Ansicht zu visualisieren. Dabei werden automatisch 2D-Projektionen für den vom Benutzer gewünschten Blickwinkel erzeugt. Mit Eve können sowohl Detektor-Geometrie, als auch zu rekonstruierenden Tracks angezeigt werden. Darüber hinaus können Monte-Carlo-Punkte, Hits und rekonstruierte Tracks dargestellt werden. [5, S. 20] Abbildung 3.3 zeigt die Visualisierung eines Events mit Eve in der XZ-Projektion. Dargestellt sind die ersten zwei Stationen des FTS. Darüber hinaus befinden sich

in der Visualisierung zwei, in weiß dargestellte Tracks. Die Monte-Carlo-Punkte sind in gelb dargestellt.



**Abbildung 3.3.:** Visualisierung einer Simulation mit Eve. Zu sehen sind acht Doppelschichten von Straw-Tubes (FTS1 und FTS2), zwei Teilchenflugbahnen und die zu den Tracks gehörenden Monte-Carlo-Punkte im FTS.

# 4. Beschreibung des Algorithmus

In diesem Kapitel wird das implementierte Verfahren beschreiben. Dazu wird zunächst die abstrakte Idee des Verfahrens erklärt. Anschließend folgt eine Erläuterung der Realisierung in C++ und der Integration des Verfahrens in PandaRoot.

## 4.1. Beschreibung des Verfahrens

Um einen Gesamtüberblick zu erhalten werden zunächst die einzelnen Schritte des Verfahrens kurz vorgestellt:

- 1) **Gruppierung von Hits** Wie bereits im vorherigen Kapitel erwähnt, besteht das FTS aus insgesamt sechs Stationen. Jede Station besteht wiederum aus vier Schichten. Jede dieser Schichten besteht aus einer Doppelreihe Straw-Tubes. Im ersten Schritt werden alle Hits, welche innerhalb einer Doppelschicht nebeneinander liegen zu einem Track-Kandidaten gruppiert.
- 2) **Approximierung der Teilchenflugbahnen außerhalb des Magnetfeldes** In Abbildung 2.6 wurde verdeutlicht, wie aus der Kombination zweier getroffener Straw-Tubes und der Einbeziehung der Isochronenradien vier verschiedene, mögliche Teilchenflugbahnen bestimmt werden können. In diesem Schritt wird dies benutzt, um für jeden im vorherigen Schritt ermittelten Track-Kandidaten diese vier möglichen Approximationen zu bilden.
- 3) **Kombination der Tracks innerhalb einer Station** Die vorangegangenen Schritte beschränkten sich jeweils nur auf eine Doppelschicht der Stationen. In diesem Schritt werden die einzelnen Teiltracks, welche für jede Doppelschicht erzeugt wurden zu einem Track zusammengeführt. Die so erzeugten Tracks verlaufen dann durch die komplette betrachtete Station des FTS. Darüber hinaus wird in diesem Schritt auch mit verschiedenen geometrischen Überlegungen die Y-Koordinate der Tracks bestimmt.
- 4) **Kombination der Teiltracks zwischen den Stationen** Die im vorherigen Schritt erzeugten Tracks werden anschließend so zusammengeführt, dass der daraus resultierende Track mehrere Stationen umfasst.
- 5) **Approximierung der Teilchenflugbahnen innerhalb des Magnetfeldes** Da sich die Teilchenflugbahnen innerhalb des Magnetfeldes nicht wie Geraden verhalten muss dort ein alternatives Verfahren eingesetzt werden. Dazu wird ein

Way-Follower verwendet, der die Tracks, welche in der vorherigen Schicht rekonstruiert wurden durch das Magnetfeld verfolgt.

- 6) Hinzufügen von nicht gefundenen Hits** Es kann vorkommen, dass einige Hits nicht korrekt erkannt werden und deshalb nicht bei der Kombination zu einem Track verwendet werden. Da jedoch am Ende des Verfahrens die Flugbahn des Tracks hinreichend genau bekannt ist, können alle Hits welche aus der rekonstruierten Flugbahn hervorgegangen sein müssen nachträglich noch zum Track hinzugefügt werden.

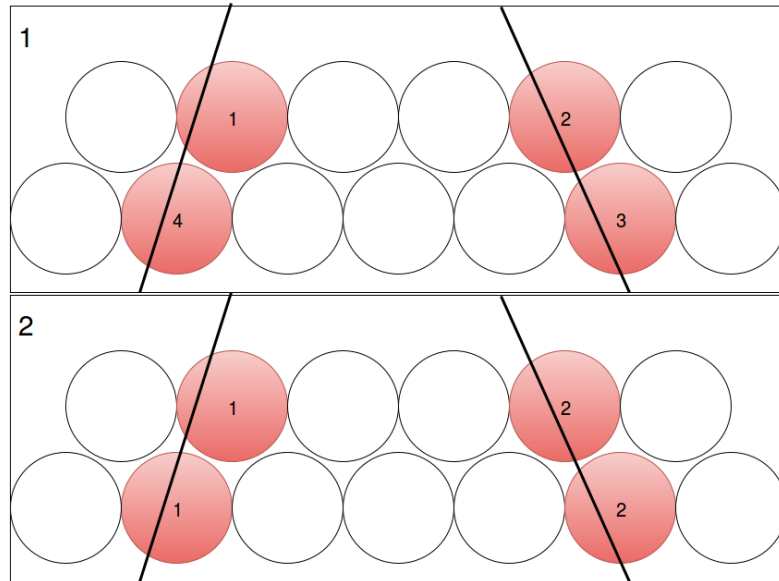
Es folgt eine ausführliche Erläuterung der einzelnen Verfahrensschritte.

#### 4.1.1. Gruppierung von Hits

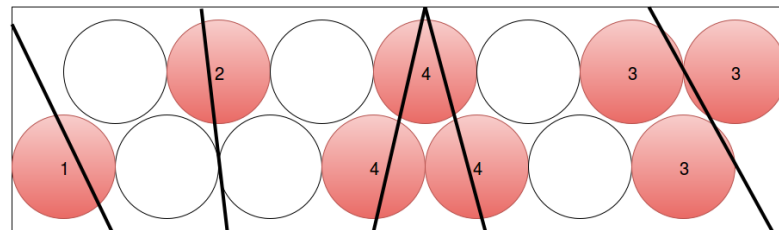
Die Gruppierung der Hits wurde mittels eines zellulären Automaten durchgeführt. Dieses Verfahren wurde bereits in einer Arbeit von Jette Schumann [5] für den STT implementiert. Dieses fertige Verfahren könnte jedoch nicht vollständig übernommen werden, da es an einigen Stellen sehr stark auf den STT spezialisiert war und zahlreiche weitere Funktionalitäten aufweist, welche im FTS nicht benötigt werden. Aus diesem Grund wurde das Verfahren in ähnlicher Form neu implementiert, um eine Spezialisierung auf die Detektorgeometrie des FTS möglich zu machen. Die vorliegende Arbeit orientiert sich jedoch stark an der Vorarbeit [5]. Zu Beginn des Verfahrens werden alle Hits mit einem eindeutigen Zustand versehen. Dazu wird die Hit-ID benutzt, welche der Anforderung der Eindeutigkeit genügt. Anschließend werden iterativ die Zustände geändert. Dazu betrachtet jeder Hit seine benachbarten Hits und setzt seinen Zustand auf das Minimum aus seinem Zustand und den Zuständen der benachbarten Hits. Dieses Verfahren wird so oft wiederholt, bis sich kein Zustand mehr ändert. Alle Hits, welche anschließend den selben Zustand haben gehören zu dem selben Track-Kandidaten. Das Verfahren beschränkt sich dabei jedoch nur auf die Nachbarschaften innerhalb der selben Doppelschicht. Dies hängt damit zusammen, dass der Abstand der Doppelschichten zu groß ist um von benachbarten Hits auf den selben Track schließen zu können.

Abbildung 4.1 zeigt eine schematische Abbildung von Doppelschichten aus Straw-Tubes, wie diese im FTS angeordnet sind. In diesem Fall wird die Doppelschicht von zwei Tracks durchquert. Da jeder Track dabei jeweils zwei Straw-Tubes trifft werden insgesamt vier Hits erzeugt. Wie vorhin beschrieben, erhält jeder Hit als Startzustand seine Hit-ID. Die Zustände sind im Bild in der Mitte der Hits vermerkt. Im zweiten Schritt werden die Zustände mit dem Minimum der Zustände der benachbarten Hits aktualisiert. Nach diesem Schritt ändert sich kein Zustand mehr und das Verfahren ist beendet. Aus der Abbildung geht hervor, dass alle Hits, welche zum selben Track gehören den selben Zustand aufweisen. Im Normalfall endet das Verfahren immer nach bereits einer Iteration, da bei der vor-

liegenden Detektorgeometrie die Wahrscheinlichkeit, dass ein Teilchen genau zwei Straw-Tubes trifft am höchsten ist.



**Abbildung 4.1.:** Schematische Darstellung der Schritte des Algorithmus



**Abbildung 4.2.:** Schematische Darstellung der Spezialfälle, bei denen nicht genau zwei Hits pro Track erzeugt werden

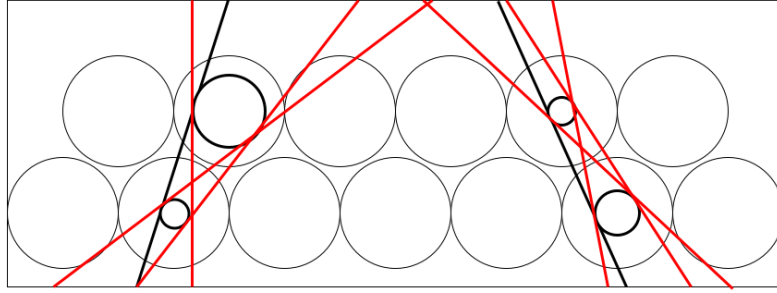
Möglich sind jedoch auch Tracks mit einem oder mehr als zwei Hits. In Abbildung 4.2 sind diese möglichen Spezialfälle dargestellt. Die Zustände in der Abbildung entsprechen der Ausgabe des zellulären Automaten. Der Fall, dass ein Teilchen nur eine einzige Straw-Tube trifft ist extrem unwahrscheinlich, kann theoretisch jedoch auch vorkommen (Track 1). Dieser Fall tritt dann auf, wenn das Teilchen das FTS nur am einem Rand streift und dann beispielsweise an der zweiten Schicht vorbei fliegt. Darüber hinaus ist es möglich, dass ein Teilchen in einer Schicht genau zwischen zwei Straw-Tubes hindurch fliegt (Track 2). Außerdem kann es sein, dass einer Straw-Tube defekt ist und deshalb nur eine Straw-Tube der Doppelschicht

einen Hit erzeugt. Da die Informationen, welche bei nur einem Treffer vom FTS über den Track gewonnen werden können, viel zu gering für eine sinnvolle Rekonstruktion sind, wird dieser spezielle Fall nicht weiter betrachtet. Diese einzelnen Hits können jedoch unter Umständen im sechsten Verfahrensschritt nachträglich hinzugefügt werden, wenn der übrige Track korrekt rekonstruiert werden kann. Teilchen, welche in einem sehr spitzen Winkel die Doppelschicht durchqueren können auch mehr Straw-Tubes treffen und erzeugen somit auch mehr als zwei Hits (Track 3). In diesem Fall dauert das Verfahren länger als eine Iteration. Darüber hinaus ist es möglich, dass sich zwei verschiedene Tracks in einer Doppelschicht Straw-Tubes überschneiden (Track 4). Dieser Fall ist nur sehr schwer sinnvoll zu behandeln, da der zelluläre Automat daraus nur einen Track-Kandidaten erstellt. Im Zuge der Analyse wurde untersucht, wie Wahrscheinlich die beschriebenen Spezialfälle sind. Da zum generieren der Events der DPM-Generator benutzt wurde entsprechen die Ergebnisse dieser Untersuchung auch näherungsweise den Wahrscheinlichkeiten bei einem realen Experiment. Die Untersuchung ergab, dass 15% aller Tracks in mindestens einer Doppelschicht nur einen Hit erzeugt haben. 52% aller Tracks weisen in mindestens einer Doppelschicht mehr als zwei Hits auf. Der Fall, dass mehrere Tracks die selbe Straw-Tube treffen betrifft 30% aller Tracks für mindestens einen Hit. Diese große Anzahl wird im wesentlichen durch Events mit sehr vielen Tracks verursacht, welche aufgrund der hohen Trackanzahl viel stärker in diese Statistik einfließen. Diese statistische Auswertung bezieht sich jeweils auf Tracks, welche in mindestens einer Doppelschicht problematische Fälle aufweisen. In den meisten Fällen beschränkt sich diese Problematik auch auf lediglich ein bis zwei problematische Doppelschichten. Im übrigen Detektor verhalten sich die Tracks dann unproblematisch.

#### 4.1.2. Approximierung der Teilchenflugbahnen außerhalb des Magnetfeldes

Im folgenden Schritt werden durch die Betrachtung der Isochronenradien der im vorherigen Schritt gruppierten Hits Approximationen für die Teilchenflugbahn generiert. Dazu werden alle Geraden konstruiert, welche tangential zu den Isochronenradien verlaufen. Bei zwei Hits entstehen dabei jeweils vier verschiedene Approximationen. Eine dieser Approximationen stellt die korrekte Approximation des Tracks dar. Die anderen drei sind zwar falsch, dies ist jedoch bei dem bisher betrachteten Informationsgehalt noch nicht feststellbar. Diese falschen Approximationen werden im späteren Verlauf des Verfahrens entfernt.

Abbildung 4.3 greift das Beispiel aus Abbildung 4.1 auf, und zeigt, wie im weiteren Verlauf die vorher erstellten Track-Kandidaten verarbeitet werden. Die schwarze Linie zeigt den korrekten Track, welcher tangential zu den Isochronenradien verläuft. Die roten Linien zeigen die Geraden, welche fälschlicherweise als mögliche Approximation des Tracks konstruiert werden. Falls im vorherigen Schritt mehr als



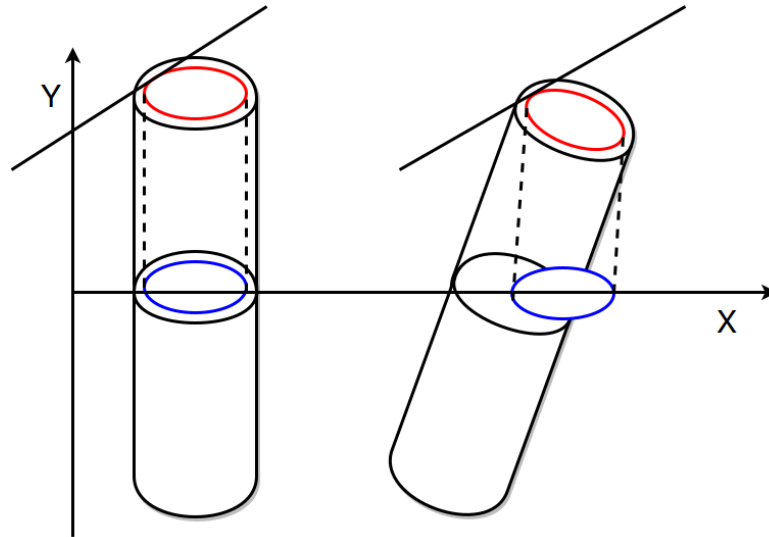
**Abbildung 4.3.:** Schematische Darstellung der möglichen Teilchenflugbahnen

zwei Hits zu einem Track gruppiert worden sind, ist meist eine eindeutige Vorhersage des Tracks möglich. In diesem Fall wird das gerade vorgestellte Verfahren für alle benachbarten Hits durchgeführt, und anschließend die Gerade als Approximation verwendet, welche am häufigsten generiert worden ist. Besonders problematisch ist hier der Fall, dass ein Track-Kandidat aus mehr als zwei Hits besteht, welche von verschiedenen Tracks stammen. In diesem Fall kann keine eindeutige Approximation gefunden werden. Der Algorithmus betrachtet eine Approximation als nicht eindeutig, wenn sich die Approximation aus mindestens einer Kombination der benachbarten Straw-Tubes nicht erzeugen lässt. In diesem Fall werden wie bei Track-Kandidaten aus je zwei Hits alle Approximationen generiert. Problematisch ist dabei jedoch, dass der Isochronenradius von Hits, welche von zwei verschiedenen Tracks getroffen wurden immer der Entfernung des Tracks entspricht, welcher mit dem kleinsten Abstand am Draht vorbei geflogen ist. Dies hängt damit zusammen, dass der Isochronenradius dem minimalen Abstand der Teilchenflugbahn zum Track entspricht. Treffen mehrere Teilchen die selbe Straw-Tube erreichen die Elektronen den Draht zuerst, welche von dem Teilchen freigesetzt wurden, das mit dem kleinsten Abstand am Draht vorbei geflogen ist. Der entferntere Track ist dann für die Straw-Tube nicht zu erkennen. Dies kann dazu führen, dass Tracks für eine Doppelschicht Straw-Tubes nicht zu rekonstruieren sind. Das sich anschließende Verfahren muss also so konstruiert werden, dass bei der Rekonstruktion auf die Teillösungen aus einzelnen Schichten verzichtet werden kann.

### **Behandlung gedrehter Straw-Tubes**

In den vorangegangenen Betrachtungen wurde stets die Projektion in der XZ-Ebene betrachtet. Da die Höhe, an der eine Straw-Tube getroffen wurde nicht zu bestimmen ist, beschreibt die approximierte Gerade die mögliche Teilchenflugbahn nur in der entsprechenden Projektion. Wird diese Betrachtung in den 3-dimensionalen Raum erweitert, so ergibt sich eine Ebene, welche parallel zu den entsprechenden Straw-Tubes verläuft. Diese Herangehensweise ist bei nicht gedrehten Straw-Tubes problemlos möglich, da bei der Projektion die X- und die

Z-Koordinate nicht verändert werden. Die Y-Höhe, in der die Straw-Tube getroffen wurde, hat keinen Einfluss auf die betrachtete Projektion. Bei gedrehten Straw-Tubes ist diese Annahme jedoch nicht gerechtfertigt. Dort weicht die erhaltene Projektion von der korrekten Lage des Isochrones ab, wenn die Straw-Tube nicht in der Höhe  $Y = 0$  getroffen wurde.

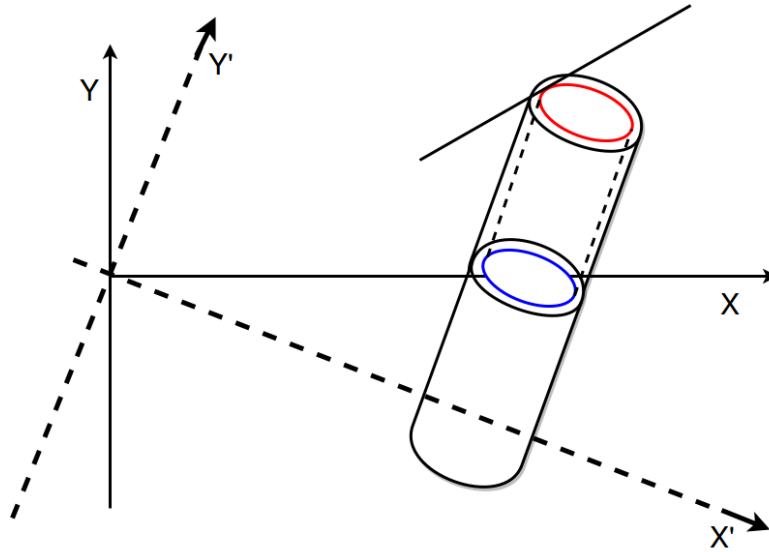


**Abbildung 4.4.:** Schematische Darstellung der Projektionen auf die XZ-Ebene

Abbildung 4.4 verdeutlicht die Problematik. Links ist eine nicht gedrehte Straw-Tube dargestellt, rechts eine gedrehte Straw-Tube. In rot ist der Kreis mit Isochronenradius auf der Höhe dargestellt, in der die Straw-Tube von einem Teilchen getroffen wurde. In blau ist die im Verfahren betrachtete Projektion dargestellt. Es wird deutlich, dass die Projektion bei der linken Straw-Tube problemlos möglich ist, bei der rechten jedoch von dem gewünschten Ergebnis abweicht. Dieses Problem kann mittels einer Koordinatentransformation umgangen werden. Wird das Koordinatensystem um den Neigungswinkel der Straw-Tubes um die Z-Achse gedreht, ist eine problemlose Projektion auf die X'Z'-Ebene möglich und das Verfahren kann wie zuvor beschrieben durchgeführt werden.

Abbildung 4.5 zeigt die Projektion nach einer Koordinatentransformation. Die in Blau dargestellte Projektion liegt nun unabhängig von der Höhe, in der die Straw-Tube von einem Teilchen getroffen wurde immer an der selben Position. Zur Durchführung der Koordinatentransformation wird eine Rotationsmatrix benötigt, welche das Koordinatensystem um einen Winkel  $\alpha$  um die Z-Achse dreht. Diese Matrix ist folgendermaßen definiert:

$$R_z(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



**Abbildung 4.5.:** Schematische Darstellung der Projektion in die  $X'Z'$ -Ebene nach einer Koordinatentransformation

Die Hit-Positionen  $P'$  werden durch Multiplikation mit der Matrix in das neue Koordinatensystem überführt:

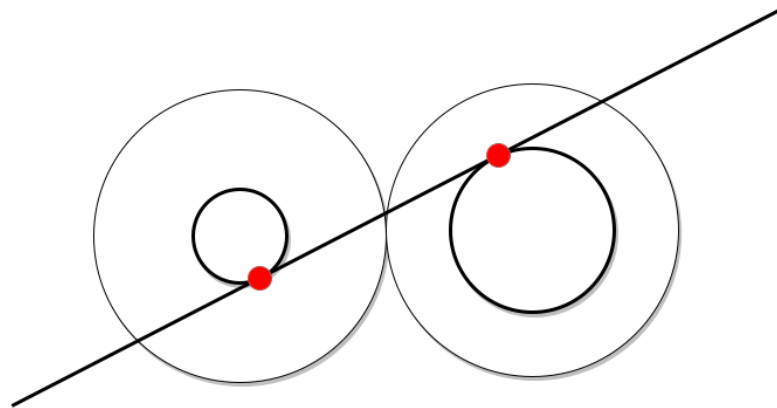
$$\begin{pmatrix} P'_x \\ P'_y \\ P'_z \end{pmatrix} = \vec{P}' = R_z(\alpha) \cdot \vec{P} = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix}$$

Der Winkel  $\alpha$  entspricht dabei der Neigung der Straw-Tubes, also  $5^\circ$  für die jeweils zweite und  $-5^\circ$  für die jeweils dritte Schicht einer jeden Station. Nach der Berechnung der Approximationen müssen die beiden Punkte, welche die approximierte Gerade definieren wieder in das ursprüngliche Koordinatensystem überführt werden. Dazu kann die gleiche Matrix mit dem negativen Drehwinkel der ursprünglichen Drehung verwendet werden.

### Korrigieren der Hit-Positionen

In Kapitel 3.3.1 wurde darauf eingegangen, dass die Position eines Hits immer genau in der Mitte der Straw-Tube erzeugt wird. Durch die durch die Einbeziehung der Isochronenradien ermittelte Approximation kann diese Position nun genauer angegeben werden. Dies ist deshalb vorteilhaft, weil im späteren Verlauf des Verfahrens ein Fitting der Tracks vorgenommen wird. Durch eine genauere Position der zum Fitting verwendeten Hits verbessert sich auch die Qualität des Fitts. Die korrigierten Hit-Positionen liegen genau an den Punkten, an denen der appro-

ximierte Track tangential zum Isochron verläuft. Diese Korrektur muss für jede mögliche Approximation separat vorgenommen werden, da diese Punkte abhängig von der Lage der approximierten Gerade an verschiedenen Stellen liegen.



**Abbildung 4.6.:** Schematische Darstellung von Hits mit korrigierter Hit-Position

Abbildung 4.6 zeigt zwei Straw-Tubes mit Isochronen und einer der vier möglichen Approximationen. In rot sind die korrigierten Hit-Positionen dargestellt. Hier erfolgt nur eine Korrektur der X- und der Z-Position der Hits. Eine Korrektur der Y-Position ist nicht möglich, da im bisherigen Verfahren nur die Projektion in die XY-Ebene betrachtet wurde. Diese Problematik wird jedoch im folgenden Verfahrensschritt beseitigt.

### 4.1.3. Kombination der Tracks innerhalb einer Station

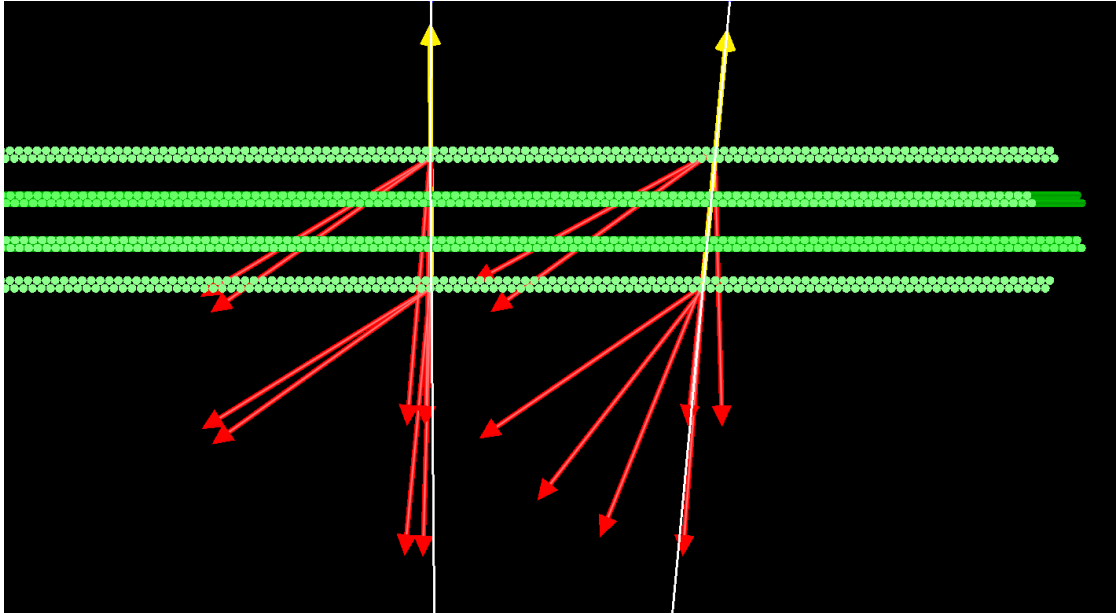
In diesem Verfahrensschritt werden alle Teilapproximationen innerhalb einer Station zusammengeführt. Dazu wird zunächst die jeweils erste Schicht mit der jeweils vierten Schicht einer jeden Station kombiniert, da diese beide Schichten nicht geneigt sind und somit eine zweidimensionale Betrachtung ausreicht. Dabei wird für jede Approximation der ersten Schicht überprüft, ob eine ähnliche Approximation in der vierten Schicht existiert. Die Ähnlichkeit der Approximationen ist wie folgt definiert:

**Winkel** Zunächst wird der Winkel zwischen den beiden approximierten Geraden berechnet. Überschreitet dieser Winkel einen definierbaren Grenzwert, sind die Geraden nicht ähnlich.

**Abstand** Wenn der Winkel zwischen beiden Geraden sehr klein ist, wird anschließend der Durchstoßpunkt der Geraden und der Ebene  $z = z_0$  berechnet. Beträgt der Abstand beider Durchstoßpunkte weniger als ein definierbarer Grenzwert, so werden die Geraden als ähnlich beurteilt.

Die Definition des Abstands als Abstand der beiden Durchstoßpunkte hat den Grund, dass lediglich die Ähnlichkeit beider Geraden in einem bestimmten Bereich definiert werden soll. Wird direkt der Abstand beider Geraden für die Definition verwendet, kann der Fall auftreten, dass beide Geraden in einem uninteressanten Bereich zwar einen geringen Abstand besitzen, im interessanten Bereich jedoch sehr weit auseinander liegen. Dies ist im Hinblick darauf, dass die meisten Tracks sowieso aus dem selben Vertex (Stoßpunkt) stammen sogar ein sehr wahrscheinlicher Fall. Diese Tracks sollen jedoch in diesem Verfahren zunächst als verschiedene Tracks aufgefasst werden und erst in einem späteren Schritt zu einem Vertex zusammengeführt werden. Um die Grenzwerte für Winkel und Abstand, ab denen Approximationen als ähnlich akzeptiert werden zu bestimmen wurde eine Parameterstudie durchgeführt, welche in Kapitel 5 thematisiert wird. Wenn zwei Geraden als ähnlich akzeptiert werden, wird eine lineare Regression mit den korrigierten Hits, welche den beiden verwendeten Approximationen zugeordnet wurden durchgeführt. Die daraus resultierenden Approximationen werden im weiteren Verlauf der Verfahrens verwendet. Alle nicht kombinierten Approximationen werden nicht weiter betrachtet, da es sich dabei nicht um die korrekte Teilchenflugbahn handeln kann. Dies hat den Vorteil, den kombinatorischen Aufwand in weiteren Verfahrensschritten deutlich zu reduzieren. Da pro Doppelschicht vier mögliche Approximationen erzeugt werden, können durch die Zusammenführung der beiden äußeren Schichten 16 mögliche Kombinationen zu einer einzigen reduziert werden. Nachteilig ist hingegen, dass Tracks, welche von einer der beiden äußeren Schichten nicht erkannt werden konnten, nicht weiter kombiniert werden. Da zu diesen Tracks keine ähnliche Approximation aus einer anderen Doppelschicht gefunden werden kann geht der Algorithmus davon aus, dass es sich um eine fehlerhafte Rekonstruktion handelt. Aus diesem Grund kann der Algorithmus die fehlerhaften Approximationen nur dann entfernen, wenn mindestens eine Approximation aus einer der äußeren Schichten mit einer Approximation der anderen äußeren Schicht kombiniert wurde. Findet der Algorithmus keine entsprechende Approximation, kann davon ausgegangen werden, dass der Track in der anderen Schicht nicht erkannt wurde. Da jedoch in der untersuchten Schicht Hits erzeugt wurden kann auch davon ausgegangen werden, dass sich dort mit Sicherheit ein Track befindet. Aus diesem Grund werden, wenn keine ähnlichen Approximationen in der anderen Schicht gefunden wurden, alle möglichen Approximationen für die weiteren Verfahrensschritte benutzt. In diesem Fall ist kein Aussortieren fehlerhafter Approximationen möglich.

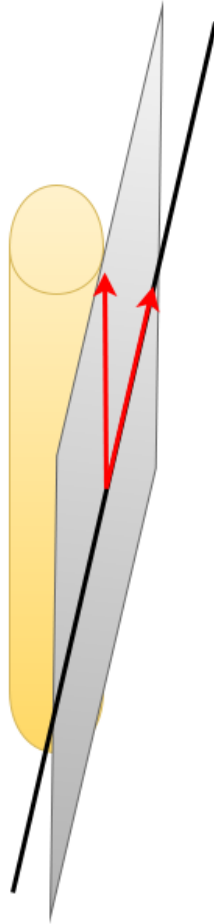
Abbildung 4.7 zeigt die in diesem Verfahrensschritt betrachteten Approximationen in der ersten FTS-Station. In Rot sind die möglichen Approximationen aus dem vorangegangenen Verfahrensschritt dargestellt. In Gelb ist die in diesem Verfahrensschritt berechnete Approximation dargestellt, die aus den beiden roten Approximationen generiert wurde, welche sich am ähnlichsten sind. Im folgenden Verfahren werden die erste und die vierte Schicht stets zusammengefasst betrachtet, und die neue Approximation für weitere Berechnungen verwendet.



**Abbildung 4.7.:** Darstellung der kombinierten Approximationen in der Simulation

Im Folgenden werden die Approximationen verwendet, welche in den gedrehten Schichten generiert wurden. Dazu wird zunächst aus jeder Approximation des vorherigen Schrittes eine Ebenen erzeugt. Als zweiter Richtungsvektor dient dazu ein Vektor, welcher parallel zu den Straw-Tubes verläuft, die für die Erstellung der Approximation verwendet wurden. Bei Betrachtung einer Approximation aus der ersten Schicht wird also ein Vektor verwendet, der parallel zu den Straw-Tubes der ersten Schicht verläuft. Die im vorherigen Schritt zusammengefassten Approximationen aus der ersten und vierten Schicht erhalten also den Richtungsvektor  $(0, 1, 0)^T$ , da die Straw-Tubes dieser Schichten parallel zur Y-Achse verlaufen. Für die zweite bzw. dritte Schicht wird der Richtungsvektor  $(\sin(-5), \cos(-5), 0)^T$  bzw.  $(\sin(5), \cos(5), 0)^T$  verwendet.

Abbildung 4.8 zeigt eine schematische Darstellung der generierten Ebene. In schwarz ist die Teilchenflugbahn eingezeichnet, in rot die beiden Richtungsvektoren der Ebene. Die Teilchenflugbahn verläuft immer in der generierten Ebene. Die Ebene beschreibt mathematisch alle möglichen Punkte, welche bei gegebenen Isochronenradien zur Teilchenflugbahn gehören könnten. Die Schnittgerade zwischen einer Ebenen der ersten Schicht und einer Ebene der zweiten Schicht liefert dann die eindeutige Teilchenflugbahn. Dabei wird auch die Höhe des Tracks entlang des Drahtes, also die bisher nicht rekonstruierbare Y-Koordinate, berechnet. Abbildung 4.9 zeigt die Ebenen, welche von für unterschiedlich geneigte Straw-Tubes generiert wurden. In rot ist die berechnete Schnittgerade zu sehen, welcher der approximierten Teilchenflugbahn entspricht.

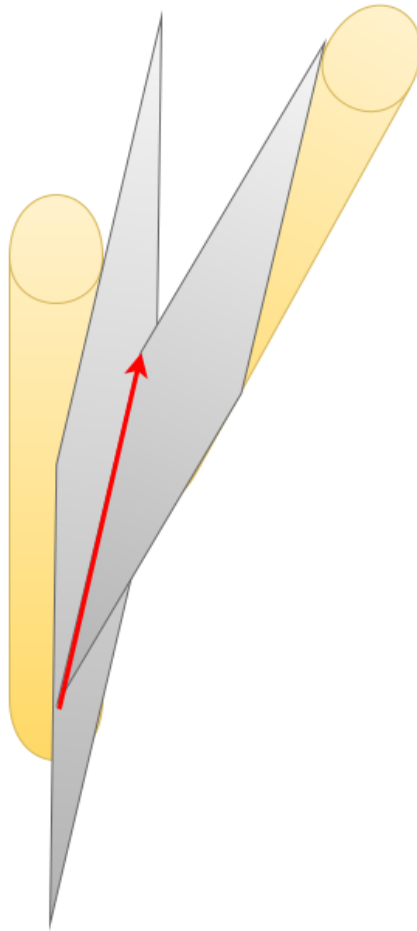


**Abbildung 4.8.:** Schematische Darstellung der generierten Ebenen

Die Schnittgeraden werden durch Kombination der folgenden Schichten berechnet:

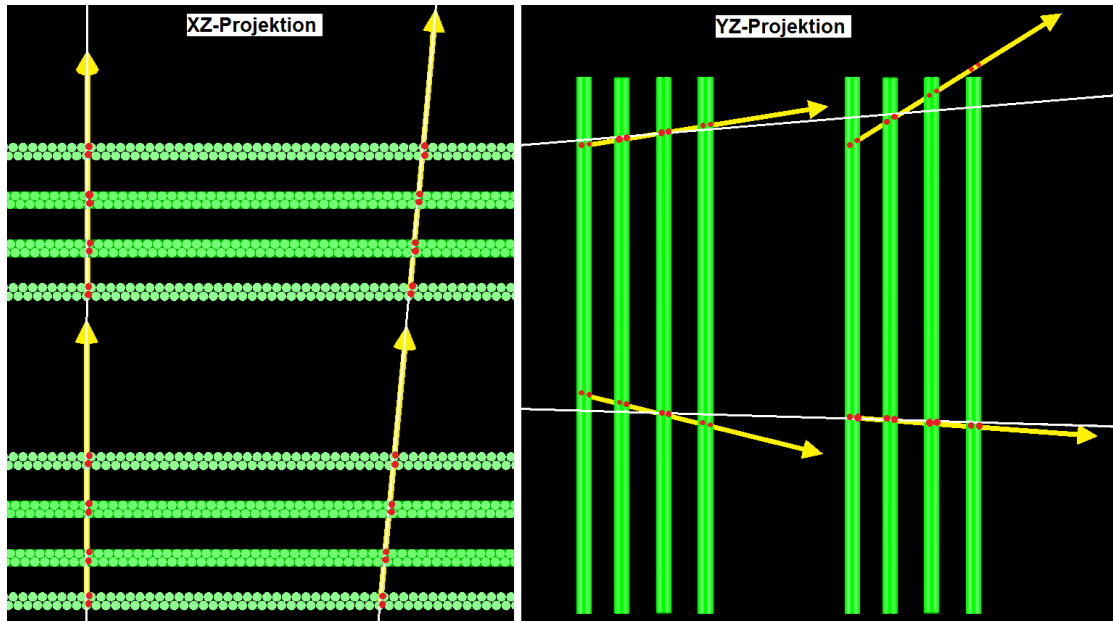
1. Zusammengefasste Schicht 1 und 4 mit Schicht 2
2. Zusammengefasste Schicht 1 und 4 mit Schicht 3
3. Schicht 2 mit Schicht 3

Da insgesamt drei unterschiedliche Doppelschichten miteinander kombiniert werden, entstehen insgesamt drei Schnittgeraden, welche den selben Track rekonstruieren. Darüber hinaus entstehen jedoch auch andere, fehlerhafte Approximationen. Diese können beispielsweise dadurch entstehen, dass Ebenen, welche zu unterschiedlichen Tracks gehören miteinander kombiniert werden. Außerdem konnten im bisherigen Verfahren für die zweite und dritte Doppelschicht noch nicht die jeweils drei falschen Approximationen entfernt werden, welche im vorherigen



**Abbildung 4.9.:** Schematische Darstellung der Schnittgeraden zweier Ebenen von unterschiedlich geneigten Straw-Tubes

Schritt generiert wurden. Deshalb wird anschließend in den generierten Schnittgeraden nach ähnlichen Geraden gesucht. Alle ähnlichen Geraden werden dann zu einer neuen Approximation zusammengefasst, indem erneut eine lineare Regression durchgeführt wird. Dazu wird zuerst eine lineare Regression in der ZX-Ebene und anschließend eine lineare Regression in der ZY-Ebene durchgeführt, und diese dann zu einer dreidimensionalen Geraden zusammengefasst. In diesem Verfahrensschritt wird anschließend auch die Y-Koordinate der Hit-Position abhängig von der neuen Approximation berechnet. Dabei wird die Y-Koordinate so bestimmt, dass der Abstand des Hits von dem approximierten Track minimal wird. Die Qualität der Korrektur der Y-Koordinate ist folglich stark von der Qualität der generierten Approximation abhängig.



**Abbildung 4.10.:** Darstellung der rekonstruierten Tracks in den Stationen FTS1 und FTS2

Abbildung 4.10 zeigt die Stationen FTS1 und FTS2 in verschiedenen Projektionen. Neben den in gelb dargestellten rekonstruierten Tracks sind nun auch die korrigierten Hits in rot dargestellt. Es fällt auf, dass die Approximationen in der XZ-Ebene die weißen Tracks sehr gut Approximieren. Die Approximation in der YZ-Ebene ist vergleichsweise eher schlecht. Die Gründe dafür wurden genauer untersucht und in Kapitel 5 genauer erläutert.

### **Probleme bei nicht rekonstruierten Teilapproximationen in gedrehten Schichten**

Im Verlaufe dieses Kapitels wurde bereits erwähnt, dass der Algorithmus auch dann in der Lage sein muss einen Track zu rekonstruieren, wenn für einzelne Schichten keine Teillösungen vorliegen. Der Fall, dass bei einer der beiden äußeren Schichten eine Rekonstruktion nicht möglich ist wurde bereits thematisiert. Nun wird der Spezialfall betrachtet, dass eine der beiden gedrehten Schichten keine Teilapproximationen für den gesuchten Track generiert hat. Es wurde bereits beschrieben, dass nach der Zusammenführung der beiden äußeren Schichten je drei Schichten miteinander kombiniert werden. Dabei entstehen insgesamt drei Gruppen kombinierter Teiltracks, welche anschließend auf Ähnlichkeit untersucht werden. Innerhalb dieser Gruppen finden sich auch zahlreiche fehlerhafte Tracks, welche durch die Kombination von unterschiedlichen Track-Kandidaten oder falschen Approximationen entstehen. Diese werden dadurch eliminiert, dass ein Track nur dann

akzeptiert wird, wenn sich in jeder anderen Gruppe der kombinierten Teilapproximationen auch ein Track befindet, der zu der Flugbahn des Tracks passt. Existieren nun in einer der mittleren Schichten keine korrekten Approximationen zum gesuchten Track, kann nur die andere mittlere Schicht mit der zusammengefassten äußeren Schichten kombiniert werden. Dabei entsteht nur eine Gruppe möglicher Tracks. Dies führt zu der Problematik, dass die dabei entstehenden fehlerhaften Tracks nicht eliminiert werden können. Bei der Zusammenfassung der beiden äußeren Schichten wurden jedoch zwei Approximationen gefunden, welche zueinander ähnlich sind. Folglich kann davon ausgegangen werden, dass sich dort ein Track befindet. Eine Auflösung der Flugbahn ist in der XZ-Projektion ebenfalls möglich. Eine Bestimmung der Y-Koordinaten des Tracks führt hingegen zu Problemen, da eine einzige Schicht gedrehter Straw-Tubes dazu nicht ausreicht. Durch die Kombination mit der gedrehten Schicht kann zwar eine Approximation mit den korrekten Y-Koordinaten generiert werden, sie kann jedoch nicht von den fehlerhaft generierten Approximationen unterschieden werden. Deshalb wird im soeben beschriebenen Fall der rekonstruierte Track trotzdem generiert, die Y-Koordinaten der Hits werden jedoch nicht berechnet. Darüber hinaus wird für den Teiltrack gespeichert, dass die Y-Koordinaten noch nicht berechnet wurden, damit diese nicht in das Fitting der sich anschließenden Schritte einbezogen werden. Werden diese Teiltracks dann mit Teiltracks anderer Stationen kombiniert, wird die Y-Koordinate an den generierten Gesamttrack angepasst.

#### 4.1.4. Kombination der Teiltracks zwischen den Stationen

Die bisher berechneten Approximationen wurden für jede FTS-Station einzeln berechnet. In diesem Verfahrensschritt müssen diese Approximationen noch zu einem Stationsübergreifenden Track zusammengefasst werden. Dazu wird erneut die Ähnlichkeit der Teiltracks bezüglich der oben definierten Kriterien betrachtet. Wie im vorherigen Schritt werden ähnliche Geraden zu einer neuen Approximation zusammengeführt und anschließend eine neue Gerade mit den Hits aller Teiltracks approximiert. Mit diesem Verfahren werden die Teilapproximationen aus FTS1 mit denen aus FTS2 verbunden und die Teilapproximationen aus FTS5 mit FTS6 verbunden.

Abbildung 4.11 zeigt die Ergebnisse des Verfahrens an einer Simulation. Die gelben Approximationen gelten nun für beide FTS-Stationen. Darüber hinaus fällt auf, dass die starken Ungenauigkeiten in der ZY-Projektion nach diesem Schritt deutlich abgenommen haben. Insbesondere beim oberen Track liegen die in rot dargestellten, korrigierten Hit-Positionen nun deutlich näher an der korrekten Teilchenflugbahn.

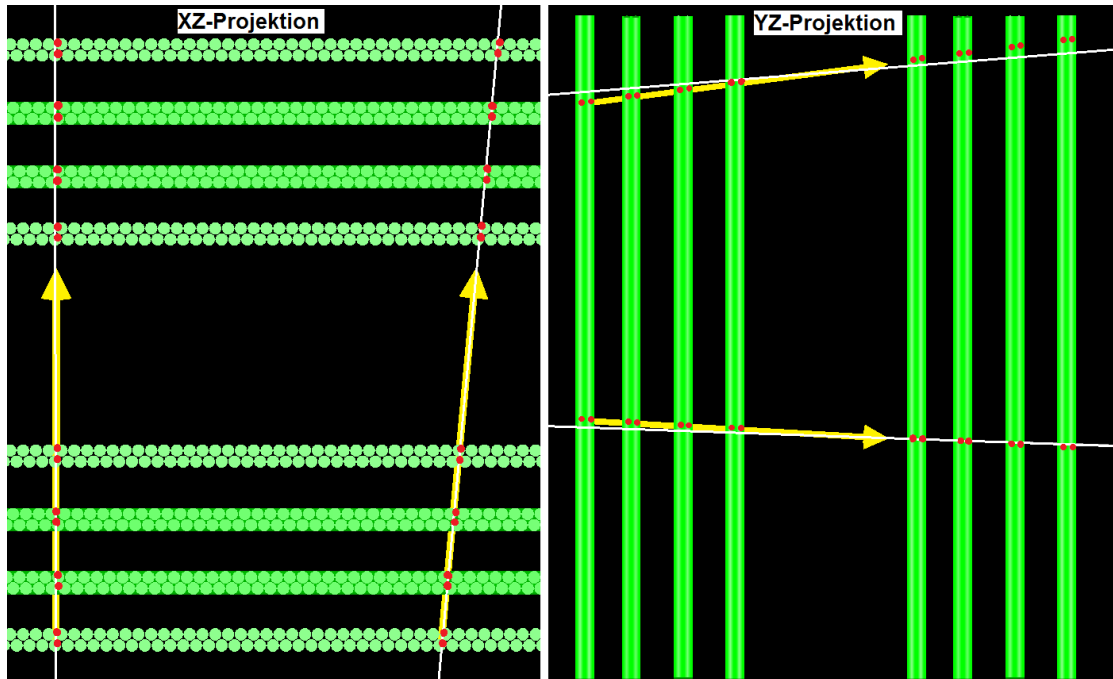
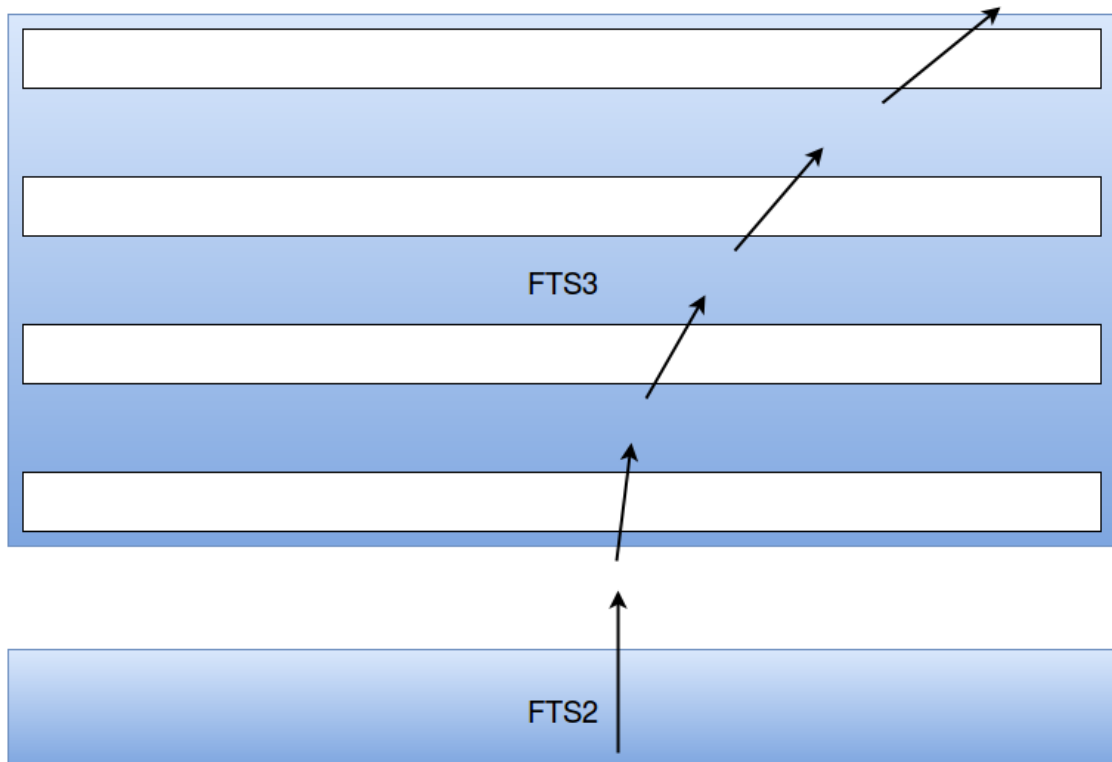


Abbildung 4.11.: Darstellung der zusammengeführten Tracks der Stationen FTS1 und FTS2

#### 4.1.5. Approximierung der Teilchenflugbahnen innerhalb des Magnetfeldes

Die FTS-Stationen FTS3 und FTS4 befinden sich innerhalb eines Magnetfeldes. Dadurch werden die Tracks durch die Lorenz-Kraft in X-Richtung abgelenkt. Die Tracks lassen sich also in der XZ-Ebene nicht durch eine Gerade approximieren, da dort eine gebogene Flugbahn vorliegt. In der YZ-Projektion erscheinen die Tracks jedoch weiterhin als Gerade, da lediglich eine Ablenkung in X-Richtung erfolgt. Um der veränderten Teilchenflugbahn gerecht zu werden, muss innerhalb des Magnetfeldes ein alternatives Verfahren verwendet werden. Dazu wird ein Way-Follower verwendet. Dieser betrachtet zunächst die Tracks, welche in den FTS-Stationen FTS1 und FTS2 rekonstruiert wurden und versucht die Flugbahn dieser Tracks durch das Magnetfeld weiter zu verfolgen. Der Way-Follower sucht zu jedem Track der vorherigen Stationen die Approximation aus der ersten Doppelschicht von FTS3, welche den Track geeignet fortsetzt. Dieses Verfahren folgt der Annahme, dass sich die Tracks innerhalb des gesamten Magnetfeldes nicht wie eine Gerade verhalten, in einem kleinen Bereich jedoch trotzdem durch eine Gerade beschreiben lassen. Zum Finden einer geeigneten Fortsetzung werden die Durchstoßpunkte der Tracks und der Ebene  $z = z_0$  gebildet. Dabei entspricht  $z_0$  der Z-Position der ersten Doppelschicht von FTS3. Danach wird die Approximation aus FTS3 ge-

sucht, welche sich am nächsten am Durchstoßpunkt befindet und deren Winkel zum Track minimal ist. Dazu werden die in diesem Kapitel bereits vorgestellten Ähnlichkeitskriterien verwendet. Die gefundene Approximation wird dann verwendet, um die passende Approximation der nächsten Doppelschicht zu finden. Bei Schichten mit gedrehten Straw-Tubes wird zusätzlich das in diesem Kapitel bereits beschriebene Verfahren der Schnittgeraden zweier Ebenen verwendet, um die Y-Position des approximierten Tracks zu verbessern. Der Way-Follower verfolgt mit diesem Verfahren den Track bis zum Beginn der Station FTS5 und verbindet somit auch die Teilapproximationen der Stationen FTS1 und FTS2 mit den Teilapproximationen der Stationen FTS5 und FTS6. Der Way-Follower führt darüber hinaus auch eine lineare Regression in der YZ-Ebene durch, da in dieser Projektion die Tracks durch eine Gerade beschrieben werden können.



**Abbildung 4.12.:** Schematische Darstellung der Teilapproximationen

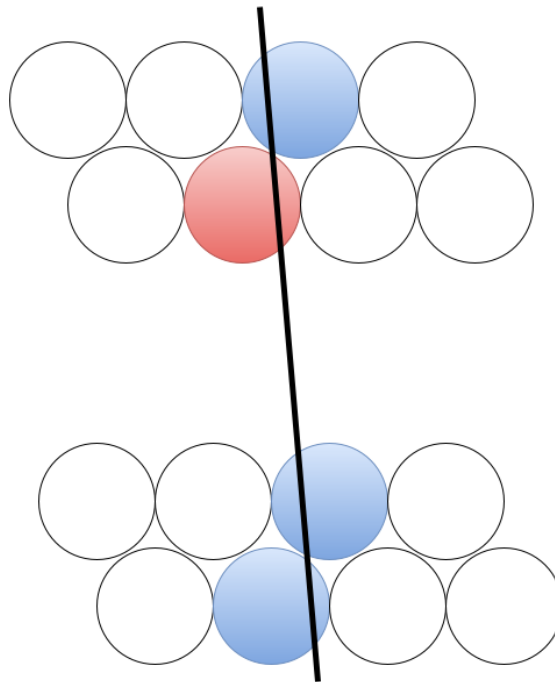
Abbildung 4.12 zeigt die vier Doppelschichten der Station FTS3, welche sich im Magnetfeld befindet. Der unterste schwarze Pfeil stellt die approximierte Teilchenflugbahn dar, welche in FTS1 und FTS2 approximiert wurde. Die schwarzen Pfeile innerhalb von FTS3 stellen die Teilapproximationen dar, welche durch Betrachtung der Isochronenradien für die einzelnen Doppelschichten erstellt wurde. Es wird deutlich, dass der Winkel zwischen zwei aufeinanderfolgenden Teilappropxi-



chen erzeugt wurden. Das erste Teilchen bewegt sich durch die gesamte FTS-Station und verursacht keine besonderen Probleme. Das zweite Teilchen verlässt bereits nach der zweiten Doppelschicht das FTS. Bei den in rot markierten Teiltracks handelt es sich um eine vom Way-Follower zusammengesetzte Flugbahn für den Track des problematischen Teilchens. Bis zur zweiten Doppelschicht gelingt die Wegverfolgung problemlos. Da jedoch in der dritten Doppelschicht keine passende Teilapproximation vorliegt würde das Verfahren für die Fortsetzung die Teilapproximationen des anderen Tracks benutzen. Durch die Einführung eines Grenzwertes für die Mindestqualität der Fortsetzung lässt sich dieses Problem beheben.

## 4.2. Hinzufügen von nicht gefundenen Hits

Die Ausgabe der vorherigen Verfahrensschritts beinhaltet prinzipiell fertige FTS-Tracks. Es wurden jedoch einige Gründe dafür erläutert, weshalb einzelne Hits nicht korrekt zugeordnet werden können. Dies führt dazu, dass einige Tracks zwar korrekt rekonstruiert werden, jedoch nicht vollständig sind. Die rekonstruierten Tracks beschreiben jedoch die gesamte Flugbahn des Tracks. Folglich können alle Hits erkannt werden, welche vom rekonstruierten Track erzeugt wurden. Dies ist für alle Hits der Fall, wo der Draht der zugehörigen Straw-Tube weniger als der Isochronenradius vom rekonstruierten Track entfernt ist.



**Abbildung 4.14.:** Schematische Darstellung von einem nicht korrekt hinzugefügten Hit

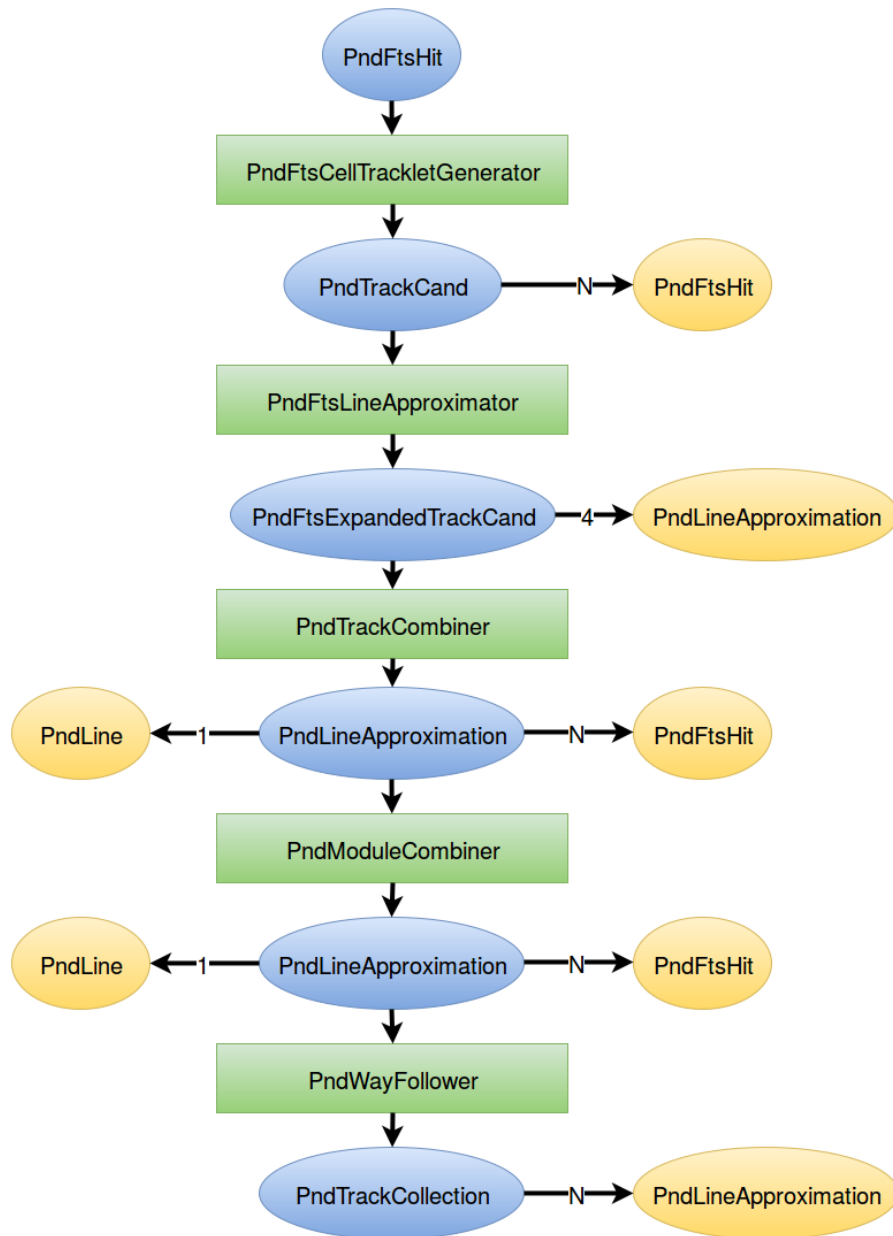
Abbildung 4.14 zeigt eine schematische Darstellung der beschriebenen Situation. Die Abbildung zeigt zwei FTS-Schichten, welche von einem Track getroffen wurden. Alle entstandenen Hits sind farbig markiert. In Blau sind die Hits dargestellt, welche dem Track zugeordnet wurden. Die roten Hits wurden dem Track nicht zugeordnet. Aus der Abbildung geht hervor, dass der rote Hit beim vorliegenden Track-Verlauf zwangsläufig zum Track gehören muss. Folglich kann dieser Hit dem Track nachträglich zugeordnet werden. Dieses Verfahren liefert nur dann positive Resultate, wenn die rekonstruierten Flugbahnen sich nicht wesentlich von den korrekten Flugbahnen unterscheiden. Bei einer starken Abweichung könnten Hits hinzugefügt werden, welche nicht vom korrekten Track getroffen werden. Da die Ausgabe des vorherigen Verfahrensschrittes jedoch globale Tracks enthält, welche durch lineare Regression verbessert wurden ist davon auszugehen, dass dieser Fall nur sehr selten auftritt.

## 4.3. Implementierung in C++

### 4.3.1. Übersicht der wichtigsten Klassen

Im Folgenden wird beschrieben, wie das erarbeitete Verfahren in eine objektorientierte C++-Implementierung überführt und in PandaRoot integriert wurde. Dazu soll zunächst ein Gesamtüberblick über die benötigten Klassen gegeben werden und später eine genauere Erläuterung der Implementierung folgen. Die implementierten Klassen lassen sich in zwei Kategorien gruppieren. Bei der ersten Kategorie handelt es sich um Klassen, welche als Datenstruktur dienen. Diese Klassen dienen der Repräsentation einer Teilrekonstruktion oder eines kompletten Tracks. Bei der zweiten Kategorie handelt es sich um Klassen, welche Teile des Verfahrens beinhalten. Beim Erstellen der Programmarchitektur wurde darauf geachtet, verschiedene Teilverfahren in unterschiedlichen Klassen zu modularisieren um diese später einfach austauschen zu können.

Abbildung 4.15 zeigt die wichtigsten Klassen des implementierten Verfahrens. Alle Datenstrukturen sind blau und alle Verfahrensklassen grün dargestellt. Setzt sich eine Datenstruktur aus anderen Datenstrukturen zusammen, sind die Bestandteile in gelb dargestellt. Durch die schwarzen Pfeile wird deutlich, von welchen Verfahrensklassen die Datenstrukturen erzeugt werden und welche Verfahrensklassen diese als Eingabe erwarten. Es folgt nun eine kurze Beschreibung der vorgestellten Klassen. Eine detaillierte Dokumentation der Methoden und Attribute befindet sich im Anhang.



**Abbildung 4.15.:** Schematische Darstellung für die Verfahrensschritte benötigten Klassen

**PndFtsHit** Die Klasse PndFtsHit wurde nicht im Rahmen dieser Arbeit implementiert sondern befindet sich schon in PandaRoot. Ein PndFtsHit repräsentiert ein Hit im FTS und beinhaltet unter anderem die Hit-Position, den Isochronenradius und die TubeID.

**PndFtsCellTrackletGenerator** Der TrackletGenerator verwendet alle FTS-Hits und generiert daraus mittels eines zellulären Automaten Track-Kandidaten.

**PndTrackCand** Die Klasse PndTrackCand befindet sich ebenfalls bereits im PandaRoot und fasst mehrere FairHits zu einem Track-Kandidaten zusammen.

**PndFtsLineApproximator** Aus den erstellten Track-Kandidaten werden im PndFtsLineApproximator mit Hilfe der Isochronenradien Geraden approximiert. Dabei bei der Betrachtung von zwei Hits insgesamt vier mögliche Approximationen generiert werden können, erstellt der LineApproximator im Standardfall mehrere Approximationen zu einem Track-Kandidaten.

**PndFtsExpandedTrackCand** Um diese möglichen Approximationen zu repräsentieren wurde die Klasse PndFtsExpandedTrackCand entwickelt. Diese enthält neben den originalen Hits des Track-Kandidaten auch mehrere mögliche Approximationen zu diesen Hits. Im Standardfall sind dies vier, in den beschriebenen Spezialfällen ist jedoch auch eine andere Anzahl an Approximationen denkbar.

**PndLineApproximation** Die Klasse PndLineApproximation repräsentiert eine mögliche, generierte Approximation. Dazu enthält sie ein Objekt der Klasse PndLine, welches die Teilchenflugbahn beschreibt. Darüber hinaus speichert diese Klasse alle zugeordneten Hits, welche anhand der gespeicherten Geraden korrigiert worden sind.

**PndTrackCombiner** Im TrackCombiner werden zunächst die Approximationen der beiden äußeren Schichten zusammengeführt und anschließend mit den beiden mittleren Schichten kombiniert. Dabei werden alle fehlerhaften Objekte der Klasse PndLineApproximation aus dem Track-Kandidaten herausgefiltert. Als Ausgabe wird deshalb nur ein Objekt der Klasse PndLineApproximation pro Teiltrack generiert. Der TrackCombiner ist nur in der Lage, Teiltracks innerhalb einer Station zu generieren.

**PndModuleCombiner** Die Teiltracks der verschiedenen Stationen werden anschließend von der Klasse PndModuleCombiner zu einem kompletten Track zusammengesetzt. Der ModuleCombiner kombiniert dabei die Tracks der Stationen FTS1 mit FTS2 und die Tracks der Stationen FTS5 mit FTS6.

**PndWayFollower** Die vom ModuleCombiner zusammengesetzten Tracks für FTS1 und FTS2 werden vom WayFollower durch das Magnetfeld in FTS3 und FTS4 verfolgt. Anschließend wird mit dem ModuleCombiner der zusammengesetzte Track aus FTS5 und FTS6 gesucht, welcher den Track am besten fortsetzt. Dabei werden Objekte der Klasse PndTrackCollection erzeugt.

**PndTrackCollection** Da sich die Tracks im Magnetfeld nicht wie eine Gerade verhalten, kann der gesamte Track nicht als eine einzige Gerade dargestellt

werden. Die Klasse `PndTrackCollection` enthält deshalb mehrere Objekte der Klasse `PndLineApproximation`, welche zusammen den kompletten Track ergeben.

### 4.3.2. Integration in PandaRoot

Das implementierte Verfahren soll in PandaRoot integriert werden. In Kapitel 2 wurde bereits beschrieben, wie eigener Programmcode in PandaRoot ausgeführt werden kann. Es wurde dazu die Klasse `PndForwardTrackFinderTask` implementiert, welche von `FairTask` erbt und die dafür benötigten Methoden überschreibt. Abbildung 4.16 zeigt das UML-Diagramm dieser Klasse. Die beiden Attribute **fHits** und **fFinalSolution** sind vom Typ `TClonesArray*` und werden für den Zugriff auf Root-Branches einer Root-Datei benötigt. Über **fHits** liest das Verfahren alle FTS-Hits aus der Datei ein, über **fFinalSolution** werden alle gefundenen Tracks in die Datei geschrieben. Die Map **fOriginalHits** bildet von der Tube-ID eines Hits auf den Hit ab, der ursprünglich aus **fHits** gelesen wurde. Dies ist nötig, da bei der Korrektur der Hits jeweils Kopien erzeugt werden. Für die anschließende Qualitätskontrolle ist es jedoch wichtig, dass die originalen Hits gespeichert werden um diese den simulierten Tracks zuordnen zu können. In der überschriebenen Methode **Init** werden alle Datenstrukturen initialisiert. Darüber hinaus wird jeweils ein Objekt der implementierten Verfahrensklassen initialisiert und ein Root-Branch mit dem Namen **FtsTracks** angelegt. Dort werden später die rekonstruierten Tracks abgespeichert. In der Methode **Exec** werden dann die Hits aus dem Root-Branch gelesen und die Verfahren nacheinander aufgerufen. Abschließend wird die Methode **saveTrackCollection** aufgerufen um die rekonstruierten Tracks in dem Root-Branch zu speichern. Über den Parameter **orgHits** kann angegeben werden, ob beim Speichern die korrigierten Hits oder die originalen Hits verwendet werden sollen. Die Methoden **FinishEvent** und **Finish** werden benutzt um Ressourcen freizugeben und die entsprechenden Verfahren oder Datenstrukturen zurückzusetzen.

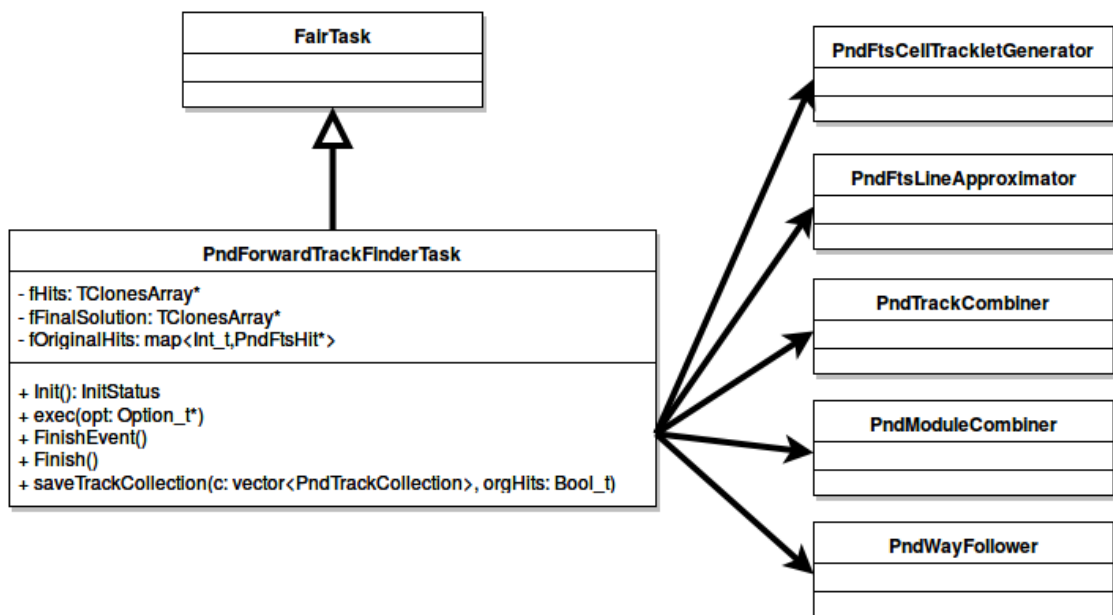


Abbildung 4.16.: UML-Diagramm der Klasse `PndForwardTrackFinderTask`

# 5. Qualitätsanalyse

In diesem Kapitel wird die Qualität des implementierten Verfahrens untersucht. Dazu wird zunächst erläutert, wie die Qualitätskontrolle konkret realisiert wurde und welche Qualitätskriterien verwendet wurden. Darüber hinaus wird auch vorgestellt, wie mittels Parameterstudien die optimalen Grenzwerte für die Ähnlichkeitskriterien gefunden wurden. Anschließend wird anhand der Qualitätskriterien die Auswertungsstatistik vorgestellt. Im Folgenden werden problematische Tracks näher betrachtet und die Gründe für fehlerhafte Rekonstruktionen erläutert. Abschließend erfolgt eine Analyse der bereits erwähnten Problematik, dass die Rekonstruktionsqualität des implementierten Verfahrens in der YZ-Projektion deutlich schlechter als in der XY-Projektion ist.

## 5.1. Realisierung der Qualitätskontrolle

In Kapitel 3.3.1 wurde bereits der Simulations- und Rekonstruktionsprozess in PandaRoot erläutert. Dort wurde unter anderem erwähnt, dass bei simulierten Daten eine Qualitätskontrolle der Rekonstruktion möglich ist. Um diese zu realisieren wurde eine weitere Task implementiert, welche in der Lage ist die Qualität eines rekonstruierten Tracks zu überprüfen. Dazu wird unter anderem ein Ideal-Track-Finder benötigt. Dieser benutzt die Informationen aus der Simulation um alle Tracks im FTS optimal zu rekonstruieren. Der Vergleich der Tracks des implementierten Verfahren mit den Tracks des Ideal-Track-Finders liefert dann eine Soll/Ist Statistik. In der implementierten Task werden zunächst alle Tracks, die der Ideal-Track-Finder gefunden hat betrachtet. Von jedem Track des Ideal-Track-Finders werden dann die Hits abgefragt und nach der Track-ID gruppiert. Die so erstellen Gruppen von Hits sollten vom implementierten Verfahren ebenfalls gefunden und zu einem Track zusammengefasst werden. Für jeden MC-Track wird dann überprüft, ob dieser vom implementierten Verfahren gefunden wurde. Ist dies der Fall, erfolgt eine Analyse der Rekonstruktionsqualität. Dabei wurden die folgende Gruppierung benutzt, welche dazu üblicherweise in PandaRoot verwendet wird:

**Reiner, vollständiger Track** Der rekonstruierten Track enthält alle Hits des MC-Tracks und keine Hits, die zu anderen MC-Tracks gehören.

**Unreiner, vollständiger Track** Der rekonstruierte Track enthält alle Hits des MC-Tracks, beinhaltet jedoch auch Hits von anderen MC-Tracks.

**Reiner, unvollständiger Track** Der rekonstruierte Track enthält mindestens 70%, jedoch nicht alle Hits des MC-Tracks und keine Hits, die zu anderen MC-Tracks gehören.

**Unreiner, unvollständiger Track** Der rekonstruierte Track enthält mindestens 70%, jedoch nicht alle Hits des MC-Tracks und beinhaltet Hits anderer MC-Tracks.

**Rekonstruktion nicht ausreichend** Der MC-Track wurde gefunden, der rekonstruierte Track beinhaltet aber weniger als 70% der Hits des MC-Tracks.

**Track nicht gefunden** Der MC-Track wurde nicht gefunden, kein rekonstruierter Track beinhaltet Hits des MC-Tracks.

Die implementierte Task ordnet jeden Track in eine solche Kategorie ein und erstellt daraus eine Statistik, welche im Folgenden genauer untersucht wird.

## 5.2. Parameterstudien zur Wahl der Ähnlichkeitsparameter

Im implementierten Verfahren befinden sich noch einige freie Parameter, welche noch so optimal wie möglich zu wählen sind. Diese werden im Folgenden kurz beschrieben.

**Generieren der Approximationen mit Isochronen** Beim Generieren der ersten Approximationen werden die Isochronen der Hits verwendet, welche zu einem Track-Kandidaten zusammengefasst wurden. Besteht ein Track-Kandidat aus mehr als zwei Hits wird überprüft, ob eine eindeutige Approximation vorliegt. Dazu müssen die Approximationen aller benachbarten Straw-Tubes auf Ähnlichkeit untersucht werden. Hier liegen freie Parameter für die Grenzwerte zu Winkel und Abstand vor.

**Zusammenfassen von Approximationen** Bei der Kombination von Teilapproximationen werden ebenfalls ähnliche Tracks zusammengefasst. Die freien Parameter der Ähnlichkeitskriterien müssen auch hier noch sinnvoll bestimmt werden. Die Ähnlichkeitskriterien müssen für jeden Verfahrensschritt einzeln bestimmt werden, da sie unter Umständen aufgrund der Beschaffenheit des Verfahrens voneinander abweichen könnten.

**Grenzwerte für den Way-Follower** Bei der Beschreibung des Way-Followers wurde erwähnt, dass Einschränkungen vorgenommen werden müssen, um fehlerhafte Fortsetzungen bei Tracks zu vermeiden, die das FTS zu früh verlassen. Auch hier müssen die Grenzwerte bestimmt werden, ab denen eine Fortsetzung noch akzeptiert wird. Diese Grenzwerte dürfen nicht so streng wie im

übrigen Verfahren sein. Dies hängt damit zusammen, dass die Tracks im Magnetfeld abgelenkt werden und eine Approximation als Gerade nur bedingt zulässig ist.

**Nachträgliches Hinzufügen von Hits** Nach der abgeschlossenen Rekonstruktion werden die Hits hinzugefügt, welche so nah am Track befinden, dass sie höchstwahrscheinlich zum Track gehören. Ab welcher Nähe die Hits hinzugefügt werden ist ebenfalls noch nicht bestimmt worden.

Im Folgenden werden zur Bestimmung dieser freien Parameter Parameterstudien durchgeführt. Dazu werden die freien Parameter variiert und dann das implementierte Verfahren für 1000 Events ausgeführt. Anschließend wird die Task zur Qualitätskontrolle ausgeführt, sodass die Reonstruktionsqualität mit dem Parameter analysiert werden kann. Die Bestimmung der Parameter erfolgt nicht gleichzeitig. Dies hängt damit zusammen, dass die Laufzeiten für 1000 Events viel zu hoch sind um in angemessener Zeit eine Optimierung über alle Parameter durchführen zu können. Eine Verringerung der Eventanzahl würde hingegen zu keiner statistisch signifikanten Aussage mehr führen. Eine nicht gleichzeitige Bestimmung der Parameter ist jedoch bedingt vertretbar, da die meisten Parameter zu unterschiedlichen Teilverfahren gehören, welche vollständig unabhängig voneinander funktionieren. Bei der Bestimmung eines Parameters müssen folglich alle übrigen Parameter schon irgendwie initialisiert werden. Dazu wird meist  $4^\circ$  als Grenzwert für die Winkel, 0.5 cm als Grenzwert für den Abstand der Tracks und 0.1 cm als Grenzwert für das nachträgliche Hinzufügen von Hits verwendet. Diese Werte ergaben sich bei der Entwicklung des Verfahrens durch Analyse einer kleinen Menge Events. Die Werte sind deshalb zwar bedingt sinnvoll gewählt, jedoch nicht statistisch repräsentativ.

### 5.2.1. Grenzwerte für das Generieren der Approximationen mit Isochronen (Verfahrensschritt 2)

Es wurde damit begonnen, den Grenzwert für den Abstand zu variieren, den zwei Geraden haben dürfen um noch als ähnlich akzeptiert zu werden. Dabei ergaben sich die in Tabelle 5.1 dargestellten Resultate. Insgesamt wurden 1637 Tracks im FTS vom Ideal-Track-Finder gefunden.

In der ersten Spalte findet sich der freie Parameter, der variiert wurde. Die zweite Zeile beinhaltet den Anteil der gefundenen Tracks, bei denen mehr als 70% der Hits korrekt zugeordnet wurden. Die dritte Zeile stellt dar, wie viele der gefundenen Tracks anteilmäßig vollständig und rein rekonstruiert wurden. Für die nicht gefundenen Tracks ist jeweils dargestellt, wie viele nur unzureichend (weniger als 70%) rekonstruiert wurden und wie viele gar nicht gefunden wurden. Die Spalte der Tracks mit weniger als 70% der Hits gibt an, wie viele nicht gefundene Tracks nicht akzeptiert wurden, weil die Rekonstruktionsqualität zu schlecht ist. Diese Spalte ist also besonders wichtig bei der folgenden Betrachtung, da diese Tracks

Abstand	Gefunden	davon rein u. vollständig	nicht gefunden	<70%	0%
0 cm	56,02%	51,91%	43,98%	<b>55,14%</b>	44,86%
0.2 cm	<b>56,51%</b>	55,68%	<b>43,49%</b>	58,01%	41,99%
0.4 cm	56,26%	56,03%	43,74%	58,66%	41,34%
0.6 cm	56,32%	56,18%	43,68%	58,18%	41,82%
0.8 cm	56,38%	56,23%	43,62%	58,26%	41,74%
1.0 cm	55,96%	<b>56,55%</b>	44,04%	58,53%	41,47%

**Tabelle 5.1.:** Ergebnisse der Task zur Qualitätskontrolle mit unterschiedlichen Grenzwerten für den Abstand der Geraden

durch eine geeignete Parameterwahl wesentlich verbessert werden kann. Hingegen ist es unwahrscheinlicher, dass Tracks mit 0 gefundenen Hits durch geeignete Parameterwahl gefunden werden können. Das Optimum ist für jede Spalte jeweils Fett markiert. Diese Darstellung wird bei allen folgenden Parameterstudien beibehalten. Zunächst fällt auf, dass sich die Wahl des Parameters nicht stark auf die Qualität des Verfahrens auswirkt. Dies hängt damit zusammen, dass die Wahrscheinlichkeit für Track-Kandidaten mit mehr als zwei Hits sehr gering ist. Aus diesem Grund wurde auch auf eine feinere Schrittweite des Abstands im interessanten Bereich verzichtet. Wird als Qualitätskriterium die Anzahl der gefundenen Tracks benutzt liegt das Optimum bei 0.2 cm. Die Qualität der rekonstruierten Tracks wird jedoch für einen größeren Abstand besser. Dies ist dadurch zu erklären, dass bei kleinen Abständen nur sehr wenig Tracks als ähnlich akzeptiert werden und keine eindeutige Approximation gefunden wird. In diesem Fall werden alle denkbaren Approximationen für das weiterführende Verfahren benutzt. Dies hat den Vorteil, dass tendenziell mehr Tracks gefunden werden weil unter der großen Menge an Approximationen einige korrekte sein können. Die große Menge der falschen Approximationen verfälscht jedoch die korrekt rekonstruierten Tracks. Es wurde entschieden hier die Anzahl der rekonstruierten Tracks zu bevorzugen und ein Optimum einen Abstand von 0.2 zu wählen. Anschließend wurde die gleiche Betrachtung für einen sich ändernden Grenzwert für den Winkel durchgeführt. Dabei ergaben sich die in Tabelle 5.2 dargestellten Ergebnisse. Da hier die Auswirkungen auf die Rekonstruktionsqualität größer sind wurde für den interessanten Bereich zwischen  $0^\circ$  und  $1^\circ$  eine kleinere Schrittweite gewählt. Auch hier liegt aufgrund der oben genannten Gründen ein Konflikt zwischen guter Track-Qualität und der Anzahl der gefundenen Tracks vor. Um hier eine ausgeglichene Balance zu schaffen wurde  $0.2^\circ$  als Grenzwert für den Winkel benutzt.

Winkel	Gefunden	davon rein u. vollsfiguretändig	nicht gefunden	<70%	0%
0°	<b>58,64%</b>	50,52%	<b>41,36%</b>	<b>56,28%</b>	43,72%
0.2°	57,42%	53,09%	42,58%	57,53%	42,47%
0.4°	57,12%	53,90%	42,88%	57,41%	42,59%
0.6°	57,06%	54,82%	42,94%	57,75%	42,25%
0.8°	56,93%	55,26%	43,07%	58,16%	41,84%
1°	56,93%	55,26%	43,07%	58,30%	41,70%
2°	57,12%	55,19%	42,88%	58,12%	41,88%
3°	56,69%	55,60%	43,31%	58,11%	41,89%
4°	56,51%	56,32%	43,49%	58,01%	41,99%
5°	56,38%	56,23%	43,62%	58,26%	41,74%
10°	56,08%	55,77%	43,92%	58,83%	41,17%
15°	55,53%	56,66%	44,47%	58,10%	41,90%
20°	54,73%	<b>56,92%</b>	45,27%	57,49%	42,51%

**Tabelle 5.2.:** Ergebnisse der Task zur Qualitätskontrolle mit unterschiedlichen Grenzwerten für den Winkel der Geraden

### 5.2.2. Grenzwerte für das Zusammenfassen von Approximationen (Verfahrensschritt 3 und 4)

In den Klasse *PndTrackCombiner* und *PndModuleCombiner* werden Teilapproximationen abhängig von Winkel und Abstand zusammengeführt. Auch hier wurde zunächst eine Variation des Grenzwertes für den Abstand durchgeführt. Die Resultate finden sich in Tabelle 5.3.

Die Änderung der Distanz wirkt sich nicht wesentlich auf die Qualität des Verfahrens aus. Dies hängt damit zusammen, dass bei der eigentlichen Kombination der Tracks in der Klasse *PndTrackCombiner* nicht auf Abstand und Winkel zurückgegriffen wird, sondern ein gemeinsames Qualitätskriterium verwendet wird. Darauf wird bei der späteren Parameterstudie noch genauer eingegangen. Lediglich beim Kombinieren zwischen Stationen und beim Zusammenfassen der beiden äußeren Schichten werden Winkel und Distanz einzeln abgefragt und verglichen. Im vorliegenden Fall liegt das Optimum der Rekonstruktionsqualität bei einem geringem Abstand. Dies hängt damit zusammen, ein kleinerer Grenzwert zu weniger generierten Tracks führt. Im oben behandelten Fall führt ein kleiner Grenzwert hingegen zu weniger eindeutigen Approximationen und somit zu mehr Tracks. Es wurde entschieden, einen Grenzwert von 0.8 cm für den Abstand zu benutzen.

In Tabelle 5.4 zeigt die Resultate für verschiedene Grenzwerte des Winkels, welcher zum Vergleich der Ähnlichkeit benutzt wird. Auch hier lassen sich kaum Auswirkungen auf die Qualität des Algorithmus feststellen. Es wurde ein Grenzwert von 7° verwendet, da bei diesem Grenzwert am meisten Tracks rekonstruiert wurden

Abstand	Gefunden	davon rein u. vollständig	nicht gefunden	<70%	0%
0.2 cm	56,93%	52,15%	43,07%	58,87%	41,13%
0.4 cm	57,24%	<b>52,83%</b>	42,76%	58,00%	42,00%
0.6 cm	57,54%	52,76%	42,46%	57,70%	42,30%
0.8 cm	<b>57,85%</b>	52,59%	<b>42,15%</b>	57,25%	42,75%
1 cm	57,73%	52,49%	42,27%	56,94%	43,06%
2 cm	57,42%	52,66%	42,58%	56,53%	43,47%
3 cm	57,42%	52,66%	42,58%	<b>56,38%</b>	43,62%
4 cm	57,42%	52,55%	42,58%	56,81%	43,19%
5 cm	57,30%	52,35%	42,70%	57,37%	42,63%

**Tabelle 5.3.:** Ergebnisse der Task zur Qualitätskontrolle mit unterschiedlichen Grenzwerten für den Abstand der Geraden

Winkel	Gefunden	davon rein u. vollständig	nicht gefunden	<70%	0%
1°	57,73%	51,85%	42,27%	57,66%	42,34%
3°	57,73%	<b>52,91%</b>	42,27%	56,94%	43,06%
5°	57,73%	<b>52,91%</b>	42,27%	57,37%	42,63%
7°	<b>57,85%</b>	52,90%	<b>42,15%</b>	<b>56,81%</b>	43,19%
9°	<b>57,85%</b>	52,90%	<b>42,15%</b>	57,10%	42,90%

**Tabelle 5.4.:** Ergebnisse der Task zur Qualitätskontrolle mit unterschiedlichen Grenzwerten für den Winkel der Geraden

$q_0$	Gefunden	davon rein u. vollständig	nicht gefunden	<70%	0%
300	55,04%	51,50%	44,96%	<b>55,30%</b>	44,70%
500	57,12%	<b>53,69%</b>	42,88%	56,70%	43,30%
700	58,28%	53,35%	41,72%	56,95%	43,05%
900	<b>58,83%</b>	52,54%	<b>41,17%</b>	57,12%	42,88%

**Tabelle 5.5.:** Ergebnisse der Task zur Qualitätskontrolle mit unterschiedlichen Grenzwerten für das Qualitätskriterium

und die Rekonstruktionsqualität nur minimal schlechter ist. Für die eigentliche Kombination der Tracks wird ein gemischtes Qualitätskriterium verwendet. Die Begründung dafür findet sich in Kapitel 5.5. Dieses Kriterium ist wie folgt definiert:

$$q_{2D} = 300 \cdot d_{2D} + 100 \cdot a_{2D}$$

$$q_{3D} = 300 \cdot d_{3D} + 100 \cdot a_{3D}$$

$$q = 0.9 \cdot q_{2D} + 0.1 \cdot q_{3D}$$

Wenn  $q$  kleiner ist als  $q_0$  werden die Approximationen als ähnlich betrachtet. Dabei steht  $d$  für die Distanz zwischen den Tracks und  $a$  für den Winkel. Der Index gibt an, ob die Bewertung im dreidimensionalen Raum oder in der XZ-Projektion durchgeführt wurde. Ein kleines  $q$  steht für eine starke Ähnlichkeit, ein großes  $q$  für stark verschiedene Approximationen. Diese Berechnung liefert den Vorteil, dass ein einzelner numerischer Wert die Ähnlichkeit der Tracks angibt und nicht Winkel und Abstand einzeln überprüft werden muss. Es wurde überprüft, wie sich eine Variation des Grenzwertes  $q_0$  auf die Qualität der Rekonstruktion auswirkt. Tabelle 5.5 zeigt die verschiedenen Ergebnisse bei einer Variation des Grenzwertes für das neue Qualitätskriterium. Für höhere Grenzwerte werden mehr Tracks gefunden. Dies hängt damit zusammen, dass in diesem Fall auch die Kombination von schlechteren Approximationen in die weiterführenden Berechnungen aufgenommen werden. Wie bereits zu erwarten war, leidet darunter jedoch die Qualität der Approximationen. Die Auswirkung des Grenzwertes auf die analysierten Daten ist auch hier geringer als vermutet. Dies weist darauf hin, dass Probleme bei der Rekonstruktion nicht auf die Kombination innerhalb von Modulen zurückzuführen ist. Dies wird im späteren Verlauf des Kapitels noch genauer analysiert. Es wurde ein Grenzwert von 700 verwendet um eine Balance zwischen der Anzahl gefundener Tracks und der Rekonstruktionsqualität zu schaffen.

### 5.2.3. Grenzwerte für den Way-Follower (Verfahrensschritt 5)

In Kapitel 4 wurde beschrieben, dass der Way-Follower einen Grenzwert benutzt, um eine fehlerhafte Verfolgung von Tracks zu verhindern, welche das FTS frühzeitig verlassen. Dieser Grenzwert wird im Folgenden bestimmt. Verwendet wurde dazu ebenfalls das definierte Qualitätskriterium, sodass nur ein Parameter bestimmt werden muss.

$q_0$	Gefunden	davon rein u. vollständig	nicht gefunden	<70%	0%
100	43,37%	15,49%	56,63%	58,36%	41,64%
500	46,61%	24,90%	53,39%	57,21%	42,79%
800	49,66%	39,85%	50,34%	56,43%	43,57%
1100	52,78%	50,23%	47,22%	56,02%	43,98%
2100	56,63%	<b>59,87%</b>	43,37%	53,10%	46,90%
3100	57,97%	58,59%	42,03%	52,18%	47,82%
4000	58,52%	58,04%	41,48%	<b>52,14%</b>	47,86%
6000	<b>58,58%</b>	57,46%	<b>41,42%</b>	52,51%	47,49%

**Tabelle 5.6.:** Ergebnisse der Task zur Qualitätskontrolle mit unterschiedlichen Grenzwerten für das Qualitätskriterium

Tabelle 5.6 enthält die Ergebnisse der Parameterstudie. Aus der Tabelle geht hervor, dass die Rekonstruktionsqualität für niedrige oder hohe Grenzwerten schlechter wird. Dies ist darauf zurückzuführen, dass bei niedrigen Grenzwerten weniger Tracks vollständig rekonstruiert werden, da die eigentlich korrekten Fortsetzungen nicht dem Qualitätskriterium genügen. Bei großen Grenzwerten werden die Tracks zwar vollständig rekonstruiert, werden jedoch durch falsche Teiltracks fortgesetzt und genügen deshalb nicht mehr dem Kriterium der Reinheit. Die meisten Tracks werden für große Grenzwerte gefunden, da dann auch Tracks verfolgt werden, welche sehr stark vom Magnetfeld abgelenkt werden. Es wurde entschieden, einen Grenzwert von 4000 zu verwenden.

### 5.2.4. Grenzwerte für das nachträgliche Hinzufügen von Hits (Verfahrensschritt 6)

Der Mindestabstand, den ein Hit vom rekonstruierten Track haben muss, um nachträglich hinzugefügt zu werden muss abschließend ebenfalls bestimmt werden. Der zu bestimmende Parameter beschreibt jedoch nicht direkt den Abstand zum Track, sondern den Sicherheits-Summands, der zum Isochronenradius addiert wird.

$\delta$	Gefunden	davon rein u. vollständig	nicht gefunden	<70%	0%
0 cm	55,35%	34,77%	44,65%	51,85%	48,15%
1 cm	58,70%	<b>58,38%</b>	41,30%	51,63%	48,37%
2 cm	60,48%	55,86%	39,52%	51,00%	49,00%
3 cm	61,58%	54,66%	38,42%	50,56%	49,44%
4 cm	<b>63,04%</b>	52,52%	<b>36,96%</b>	<b>49,09%</b>	50,91%

**Tabelle 5.7.:** Ergebnisse der Task zur Qualitätskontrolle mit unterschiedlichen  $\delta$

Ein Hit wird hinzugefügt wenn

$$d < r + \delta$$

gilt. Dabei steht  $d$  für den Abstand zum Track,  $r$  für den Isochronenradius und  $\delta$  für den Sicherheits-Summand. Dieser wird in der folgenden Parameterstudie bestimmt.

Tabelle 5.7 zeigt die Ergebnisse für unterschiedliches  $\delta$ . Es fällt auf, dass sich die Qualität des Verfahrens stark durch die Wahl von  $\delta$  zu beeinflussen ist. Für große  $\delta$  werden die meisten Tracks rekonstruiert, die Qualität der Rekonstruktionen nimmt jedoch ab. Dies hängt damit zusammen, dass für sehr große  $\delta$  alle FTS-Hits zum Track hinzugefügt werden. Folglich werden zwar alle Tracks gefunden, die Tracks enthalten jedoch auch die fehlerhaften Hits der anderen Tracks. Es wurde entschieden einen Grenzwert von 3 cm zu verwenden. Durch Betrachtung der Detektorgeometrie ergibt sich jedoch eigentlich ein hartes Limit von  $\delta = r_s = 0.5$ , da dies dem Radius einer Straw-Tube entspricht. Eigentlich ist es nicht möglich, dass der Track weiter als  $r_s$  von einer Straw-Tube entfernt ist und trotzdem einen Hit erzeugt. Jedoch liegen die rekonstruierten Tracks nicht exakt auf den korrekten Tracks. Insbesondere die schlechte Auflösung der Y-Koordinaten ist problematisch. Aus diesem Grund ist es vertretbar, hier einen größeren  $\delta$  zu wählen. Jedoch kann dies dazu führen, dass Hits anderer Tracks fehlerhaft hinzugefügt werden. Durch ein größeres  $\delta$  verbessert sich also die Vollständigkeit der Tracks auf Kosten von Reinheit.

### 5.3. Statistische Ergebnisse aus der Qualitätskontrolle

Es folgt nun eine statistische Auswertung des optimierten Algorithmus. Dazu werden alle Tracks in die vorher definierten Qualitätsklassen einsortiert.

Tabellen 5.8 zeigt die Ergebnisse der Qualitätskontrolle. Es ist zu beachten, dass die relative Angabe sich diesmal nicht auf den Anteil an den gefundenen oder nicht gefundenen Tracks bezieht, sondern auf den Anteil an allen Tracks. Insgesamt werden 61,58% aller Tracks gefunden. Insgesamt werden 33,66% aller Tracks rein

	absolut	relativ
<b>Gefundene Tracks</b>	<b>1008</b>	<b>61,58%</b>
reine u. vollständige Tracks	551	33,66%
unreine u. vollständige Tracks	252	15,4%
reine u. unvollständige Tracks	113	6,9%
unreine u. unvollständige Tracks	92	5,62%
<b>nicht gefundene Tracks</b>	<b>629</b>	<b>38,43 %</b>
unter 70% gefunden	318	19,43%
nicht gefunden	311	19%

**Tabelle 5.8.:** Ausführliche Ergebnisse der Qualitätsanalyse des optimierten Algorithmus ohne Einschränkungen an die betrachteten Tracks

und vollständig rekonstruiert. Dies entspricht einem Anteil von etwa 50% an den gefundenen Tracks. Die meisten Tracks werden vollständig rekonstruiert. Der Anteil der unreinen Tracks an allen Tracks beträgt 21,02%. Dies hängt mit den hohen Grenzwerten für das nachträgliche Hinzufügen von Hits und für den Way-Follower zusammen.

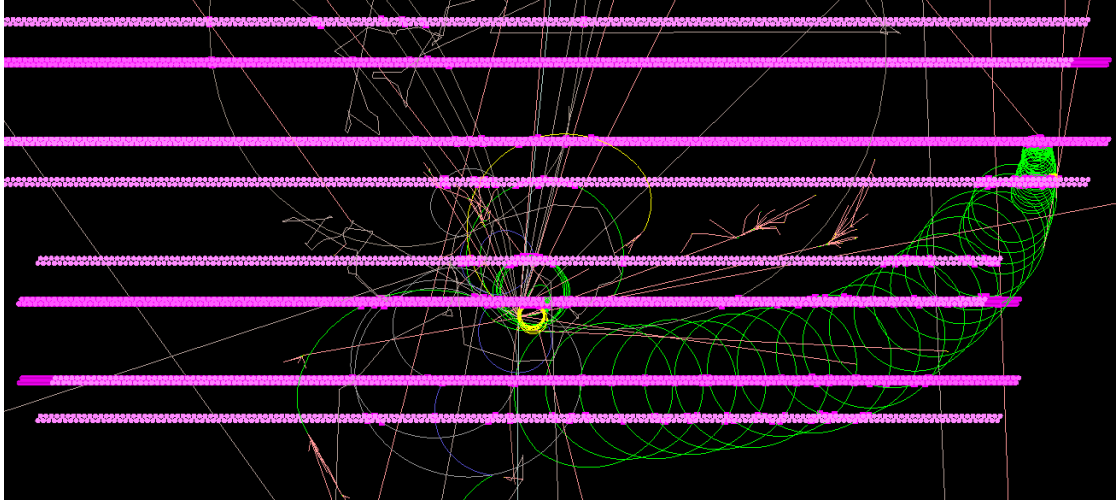
## 5.4. Gründe für fehlerhafte Rekonstruktionen

In der analysierten Statistik existieren noch zahlreiche Tracks, welche vom implementierten Verfahren nicht gefunden werden. Deshalb werden die wichtigsten Klassen von problematischen Tracks vorgestellt und erläutert, warum diese für das Verfahren problematisch sind.

### 5.4.1. Tracks mit niedrigem Impuls

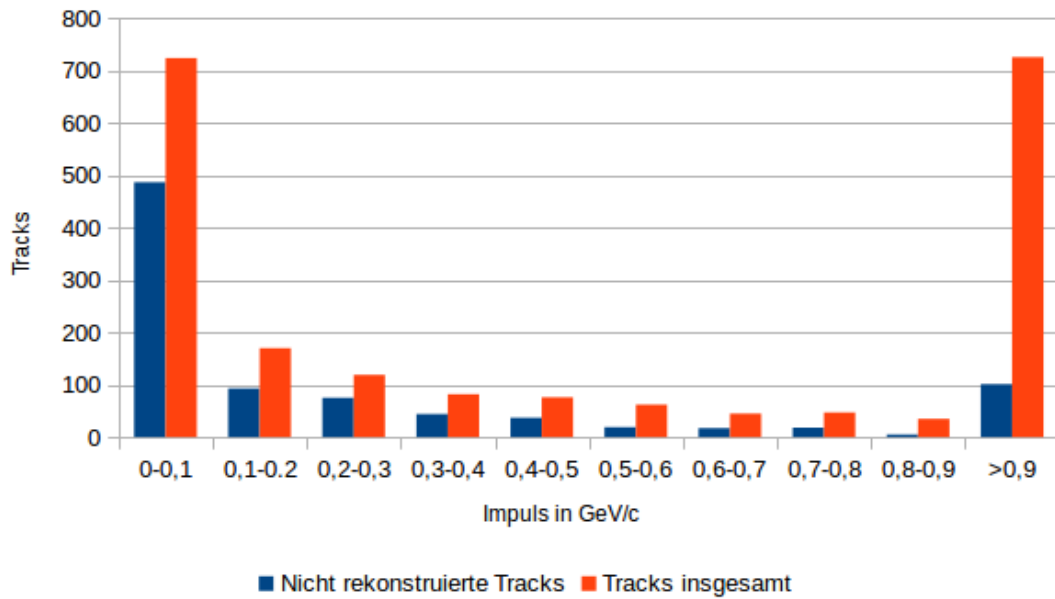
Ein großes Problem stellen Tracks dar, welche sich stark kreisend durch den Detektor bewegen. Ein Event mit stark kreisenden Tracks ist in Abbildung 5.1 dargestellt. Ein solcher Fall führt zu mehreren Problemen. Zum einen ist das Verfahren darauf ausgelegt, dass sich alle Tracks vom Stoßpunkt aus in Richtung der Z-Achse bewegen. Tracks, welche sich stark seitwärts oder in die entgegengesetzte Richtung bewegen können nicht erkannt werden. Zum anderen erzeugen solche Tracks eine sehr große Anzahl an Hits. Dies führt dazu, dass zahlreiche Teilapproximationen auf engem Raum erzeugt werden. Diese können mit den Ähnlichkeitskriterien nicht ausreichend gut unterschieden werden. Problematisch ist auch, dass sich die stark kreisenden Tracks meist nicht durch alle FTS-Stationen bewegen und somit nicht die benötigte Anzahl an Teilapproximationen erreichen, welche zur weiteren Be-

trachtung nötig ist. Das größte Problem liegt jedoch darin, dass diese Tracks nicht ansatzweise als Gerade aufgefasst werden können. Da dies jedoch in fast allen Schritten des implementierten Verfahrens vorausgesetzt wird ist eine Rekonstruktion dieser Tracks nicht möglich. Der Way-Follower geht beispielsweise davon aus, dass sich die Tracks auf einem kleinen Bereich wie eine Gerade verhalten. Die starke Kreisbewegung ist auf eine starke Ablenkung durch das Magnetfeld zurückzuführen. Für niedrige Impulse ist diese Ablenkung besonders groß. Deshalb wurde eine Korrelationsuntersuchung durchgeführt, in welcher der Zusammenhang zwischen Impuls und Rekonstruktionsqualität untersucht wurde.

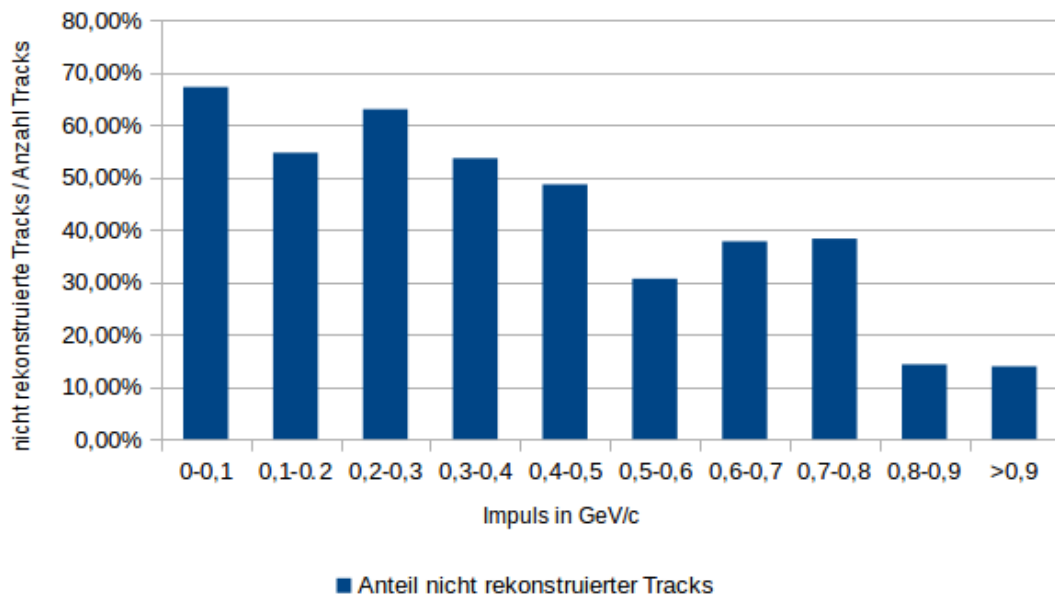


**Abbildung 5.1.:** Darstellung eines stark kreisenden Tracks

Die Ergebnisse dieser Korrelationsuntersuchung sind in Abbildung 5.2 dargestellt. In dieser und allen folgenden Korrelationsuntersuchungen wurden mit anderen Simulationsdaten durchgeführt, in der insgesamt 896 von 2086 Tracks nicht rekonstruiert wurden. Aus der Abbildung geht hervor, dass sehr viele Tracks einen Impuls von 0-0.1 GeV aufweisen. Das Auftreten größerer Impulse ist zunehmend unwahrscheinlicher. Die hohe Anzahl Tracks mit einem Impuls von mehr als 0.9 GeV/c lässt sich mit der akkumulierten Darstellung eines wesentlich größeren Intervalls begründen. Abbildung 5.3 zeigt den Anteil der nicht rekonstruierten Tracks an allen existenten Tracks. Aus der Abbildung geht hervor, dass die Rekonstruktionsqualität für höhere Impulse zunimmt. Für Impulse ab 0.8 GeV/c liegt der Anteil fehlerhafter Rekonstruktionen mit 15% deutlich unter dem Durchschnitt für alle Tracks. Folglich kann davon ausgegangen werden, dass Rekonstruktionsqualität und Impuls korreliert sind.



**Abbildung 5.2.:** Verteilung von nicht rekonstruierten Tracks und aller Tracks auf verschiedene Impulsbereiche



**Abbildung 5.3.:** Verteilung des Anteils nicht rekonstruierter Tracks auf verschiedene Impulsbereiche

### 5.4.2. Events mit einer großen Anzahl Tracks

Es kann der Fall eintreten, dass ein Teilchen kurz vor dem FTS oder beim durchqueren des FTS mit dem Detektormaterial reagiert, beispielsweise wegen einem Zusammenstoß mit einem anderen Atomkern. Dabei kann eine große Menge weiterer Teilchen freiwerden. Dieser Fall ist für das implementierte Verfahren ebenfalls stark problematisch. Die Anzahl der Hits ist dann so hoch, dass die Flugbahnen nicht korrekt unterschieden werden können. Dieses Problem wird dadurch verstärkt, dass die Sekundärteilchen meist Flugbahnen aufweisen, welche zueinander fast parallel ist. Darüber hinaus ist der zelluläre Automat nicht in der Lage, aus den verschiedenen Sekundärteilchen auch verschiedene Track-Kandidaten zu bilden. Dieses liegt daran, dass die Hits der verschiedenen Tracks benachbart sind, wenn die Tracks sehr nah aneinander liegen. Abbildung 5.4 zeigt den beschriebenen Fall. Der beschriebene Fall hat für die statistische Untersuchung besonders fatale Auswirkungen, da darin die Anzahl der rekonstruierten Tracks untersucht wird. Events mit einer hohen Anzahl Tracks gehen also besonders stark in die Statistik ein. Beispielsweise könnte das implementierte Verfahren zehn Events mit jeweils einem Track vollkommen korrekt rekonstruieren. Wenn das elfte Event dann ein problematisches Teilchen mit zehn Sekundärtracks enthält, welche nicht zu rekonstruieren sind, ergibt sich in der Statistik nur ein Anteil von 50% für die gefundenen Tracks. Auch hier wurde eine Korrelationsuntersuchung durchgeführt, um den Zusammenhang zwischen problematischen Tracks und Events mit vielen Tracks zu ermitteln.

Abbildung 5.5 zeigt die Anzahl nicht rekonstruierter Tracks und die Anzahl aller Tracks. Die Anzahl aller Tracks ist bezüglich der Anzahl Tracks im gleichen Event annähernd gleichverteilt. Dies bedeutet nicht, dass Events mit vielen Tracks genauso wahrscheinlich sind wie Events mit wenigen Tracks, da Events mit vielen Tracks auch stärker in die Statistik eingehen. Events mit vielen Tracks werden also zunehmend unwahrscheinlicher. Der Anteil der nicht rekonstruierter Tracks ist in Abbildung 5.6 dargestellt. Die Abbildung zeigt eine deutliche Problematik für Events mit mehr als sechs Tracks. Events mit nur einem Track werden hingegen zu 85% rekonstruiert. Auch hier wird eine Korrelation von Trackanzahl und fehlerhafter Rekonstruktion sehr deutlich.

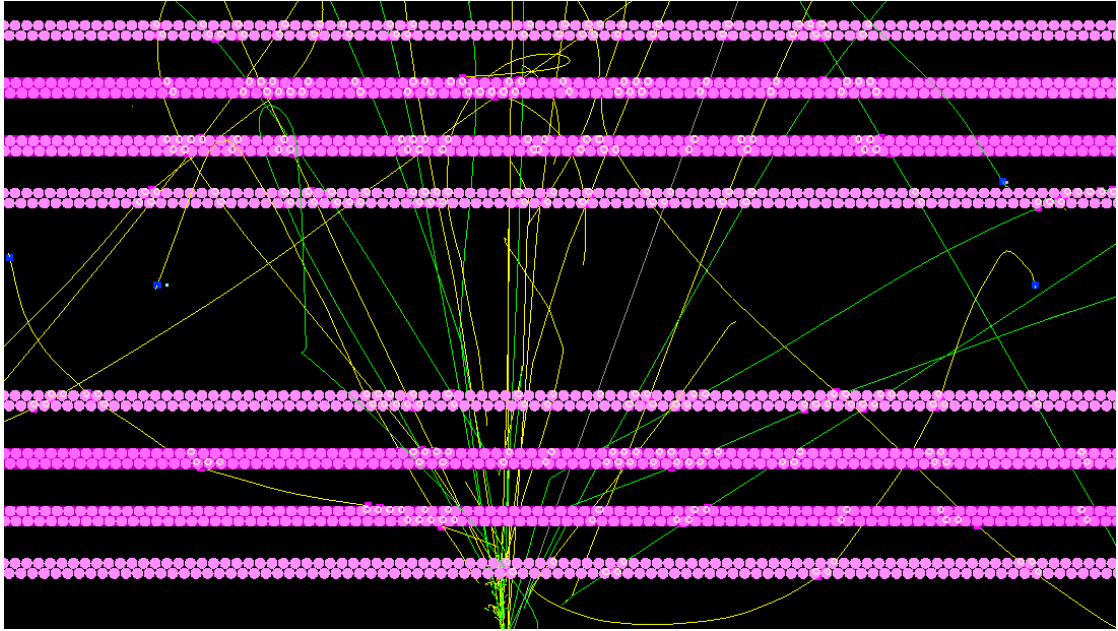


Abbildung 5.4.: Darstellung eines Teilchens, welches vor dem FTS stark zerfällt

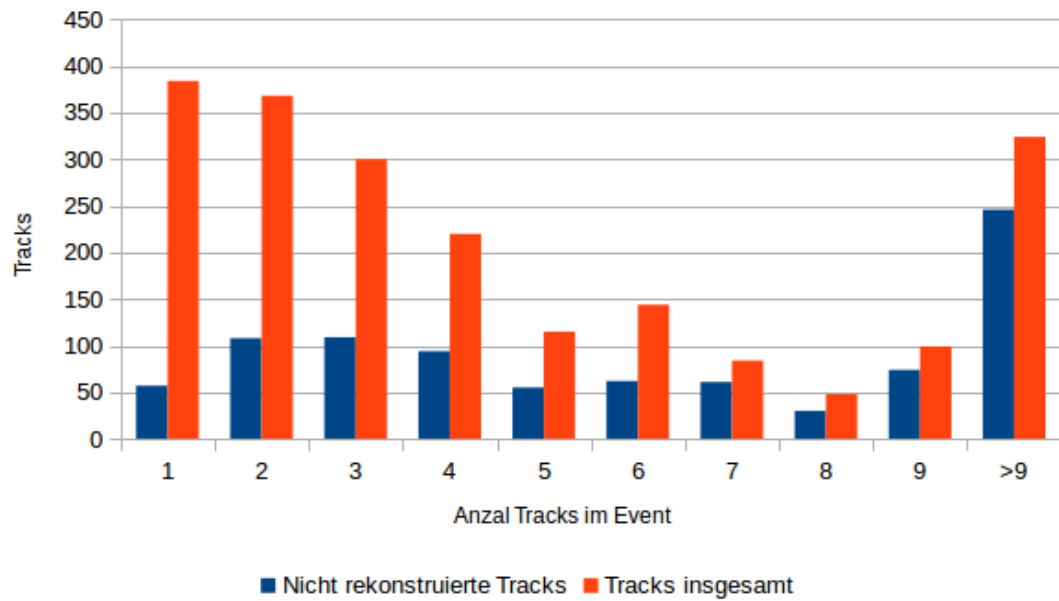
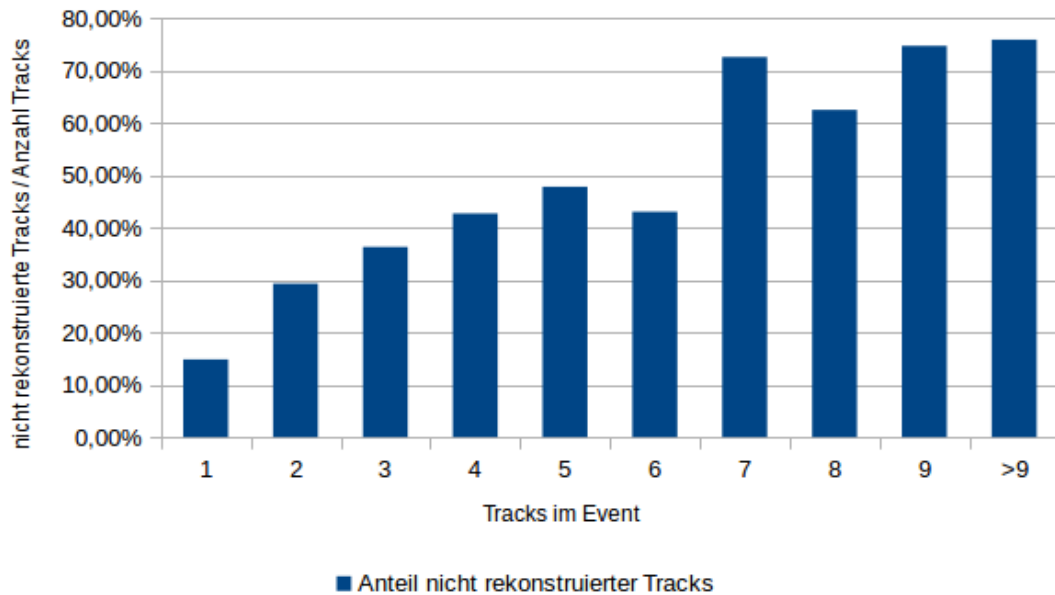


Abbildung 5.5.: Verteilung von nicht rekonstruierten Tracks und aller Tracks auf die Anzahl an Tracks im selben Event

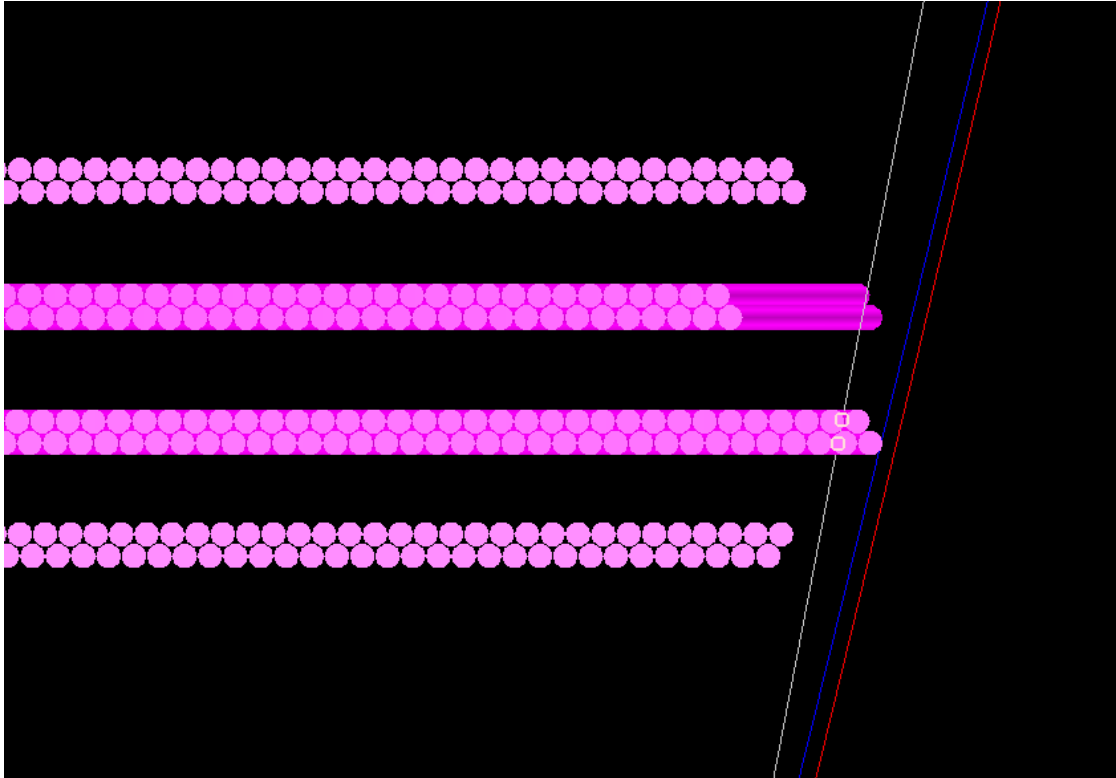


**Abbildung 5.6.:** Verteilung des Anteils nicht rekonstruierter Tracks auf die Anzahl an Tracks im selben Event

### 5.4.3. Tracks mit wenigen Hits

Abbildung 5.7 zeigt einen Track, der seitlich am FTS vorfliegt. Dabei werden nur zwei Hits erzeugt. Dieser Fall ist für eine Rekonstruktion ebenfalls sehr problematisch. Die erste Teilapproximation wird zwar korrekt erzeugt, kann jedoch anschließend nicht mit anderen Teilapproximationen kombiniert werden. Eine Bestimmung der Y-Koordinaten ist im vorliegenden Fall ebenfalls nicht möglich. Aus diesem Grund kann die Approximation nicht in weiterführende Verfahrensschritte integriert werden. Das Verfahren wurde so entworfen, dass auf eine Schicht verzichtet werden kann. Existieren jedoch nur Hits von weniger als drei Schichten ist eine Rekonstruktion nicht möglich. Darüber hinaus können andere FTS-Stationen ebenfalls nicht zur Kombination oder Verbesserung des Tracks herangezogen werden, da der Track an folgenden Stationen vorbei fliegt.

Die Ergebnisse der für diesen Fall durchgeführten Korrelationsuntersuchung finden sich in Abbildung 5.8. Zunächst fällt die große Anzahl Tracks mit 40 Hits auf. Obwohl hier die Intervallbreite im Gegensatz zu den übrigen Intervallen nicht acht sondern eins beträgt, existieren außergewöhnlich viele Tracks mit 40 Hits. Ein Track, welcher alle Schichten des FTS durchquert produziert immer 48 Hits. Dieser Wert ergibt sich durch folgende Berechnung.

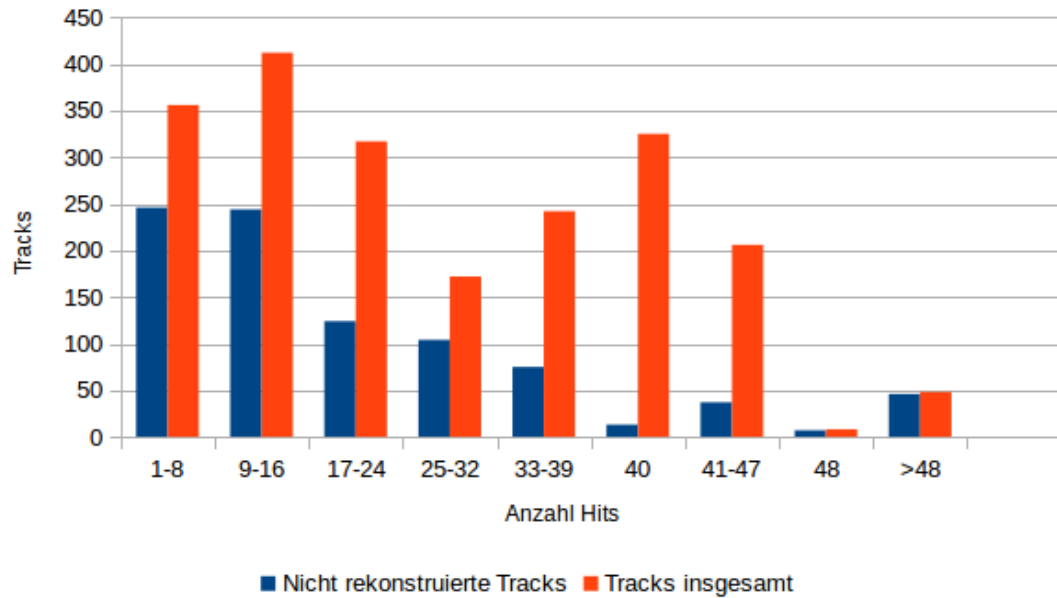


**Abbildung 5.7.:** Darstellung eines Tracks, der nur zwei Hits verursacht

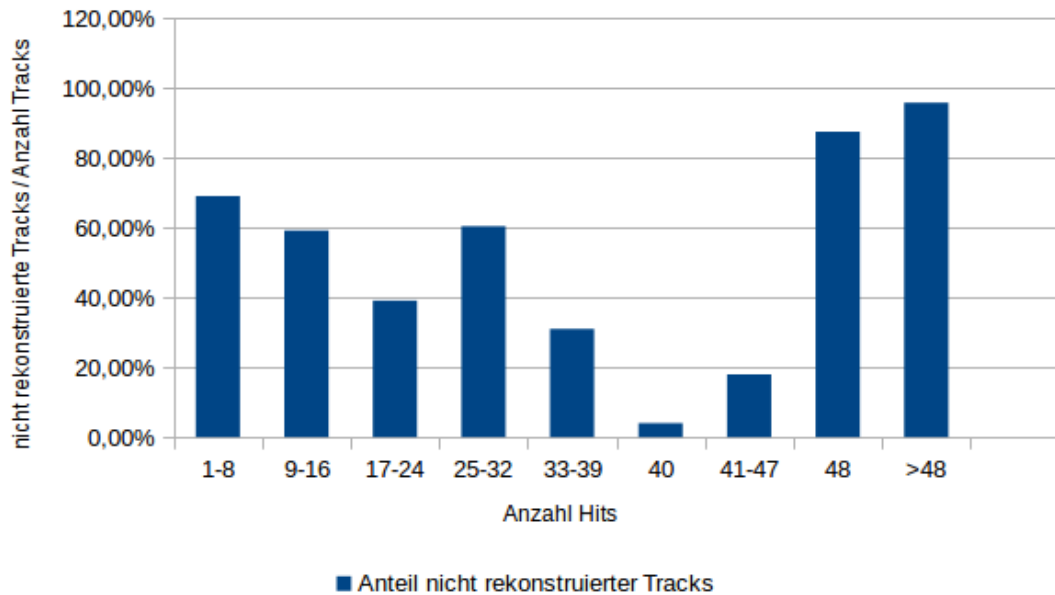
$$48 = \underbrace{2}_{\text{Hits pro Doppelschicht}} \cdot \underbrace{4}_{\text{Schichten pro Station}} \cdot \underbrace{6}_{\text{FTS-Stationen}}$$

Tracks mit 48 Hits beinhalten folglich die maximale Information und sollten für das Verfahren am besten zu rekonstruieren sein. Leider existiert derzeit noch ein Fehler in PandaRoot. Aufgrund dieses Fehlers werden bei einer Simulation in der Station FTS6 in den allermeisten Fällen keine MC-Points und somit auch keine Hits erzeugt. Abzüglich der acht Hits für die letzte FTS-Station ergibt sich also ein Wert von 40 Hits für den wahrscheinlichsten Fall. Tracks mit mehr als 48 Hits können folglich nur dann zustande kommen, wenn die Tracks stark kreisen. Diese Tracks sollten für das Verfahren kaum zu rekonstruieren sein. Abbildung 5.9 bestätigt diese Vermutung. Für Tracks mit mehr als 48 Hits beträgt der Anteil nicht rekonstruierter Tracks nahezu 100%. Wie bereits erwartet werden auch die Tracks mit 40 Hits eindeutig am besten rekonstruiert. Allerdings lässt sich keine signifikante Abweichung zwischen Tracks mit sehr wenigen Hits und Tracks mit 25-39 Hits feststellen. Dies weist darauf hin, dass das Verfahren für Tracks mit wenigen Hits prinzipiell problematisch ist. Tracks mit 48 Hits werden ebenfalls sehr schlecht rekonstruiert, obwohl diese eigentlich optimal für das Verfahren sein

sollten. Dies liegt zum einen daran, dass diese Tracks zum Großteil ebenfalls von stark kreisende Teilchen stammen, welche zufällig exakt 48 Hits generiert haben. Zum anderen ist die Anzahl an Tracks mit 48 Hits viel zu gering, um hier statistisch signifikante Aussagen treffen zu können.



**Abbildung 5.8.:** Verteilung von nicht rekonstruierten Tracks und aller Tracks auf die Anzahl der Hits dieser Tracks



**Abbildung 5.9.:** Verteilung des Anteils nicht rekonstruierter Tracks auf die Anzahl der Hits dieser Tracks

#### 5.4.4. Statistische Ergebnisse unter Ausschluss der problematischen Tracks

Es wurden einige Track-Klassen vorgestellt, bei deren Bearbeitung das implementierte Verfahren keine Chance hat, eine erfolgreiche Rekonstruktion durchzuführen. Es macht also Sinn, bei der Qualitätskontrolle nur die Tracks zu betrachten, welche zumindest theoretisch gefunden werden können. Deshalb wurde eine erneute Qualitätskontrolle durchgeführt, in welcher die folgenden Einschränkungen getroffen wurden:

**Keine Tracks mit einem Impuls von weniger als 0.3 GeV/c** Bei der durchgeführten Korrelationsuntersuchung wurde festgestellt, dass stark kreisende Tracks auf einen niedrigen Impuls zurückzuführen sind. Deshalb werden in der folgenden Qualitätsanalyse nur Tracks mit einem Impuls von mehr als 0.3 GeV/c betrachtet.

**Keine Events mit mehr als sechs Tracks** Es wurde festgestellt, dass bei Events mit vielen Tracks eine Rekonstruktion problematisch werden kann. Aufgrund dieser Tatsache werden im Folgenden nur Events mit weniger als sechs Tracks betrachtet.

**Keine Tracks mit weniger als sechs Hits** Um vom Verfahren erkannt zu werden muss ein Track von mindestens einer der beiden ersten oder einer der

	absolut	relativ
<b>Gefundene Tracks</b>	<b>832</b>	<b>77,54%</b>
reine u. vollständige Tracks	468	43,62%
unreine u. vollständige Tracks	187	17,43%
reine u. unvollständige Tracks	93	8,67%
unreine u. unvollständige Tracks	84	7,83%
<b>nicht gefundene Tracks</b>	<b>241</b>	<b>22,47 %</b>
unter 70% gefunden	86	8,02%
nicht gefunden	155	14,45%

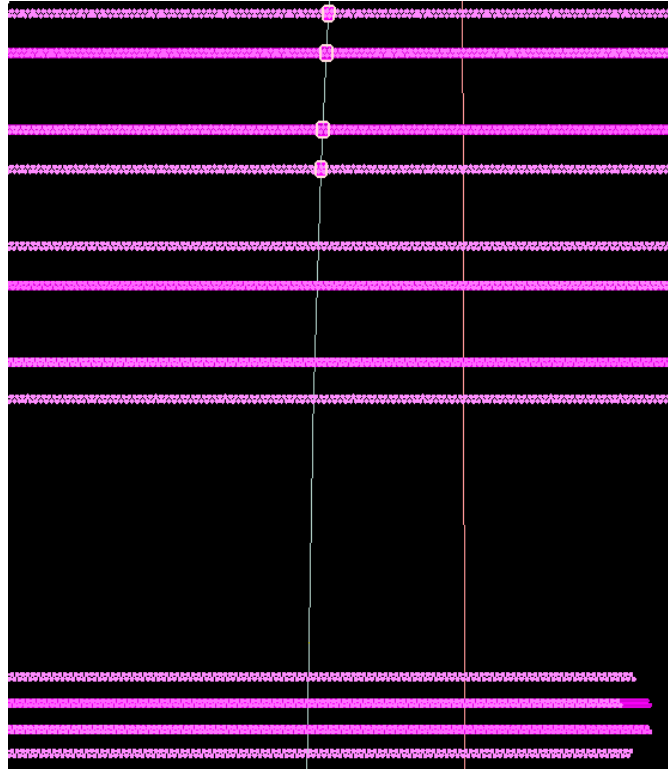
**Tabelle 5.9.:** Ausführliche Ergebnisse der Qualitätsanalyse des optimierten Algorithmus mit Einschränkungen an die betrachteten Tracks

beiden letzten FTS-Stationen erkannt werden. Es wurde beschrieben, dass dazu mindestens sechs Hits notwendig sind. Deshalb werden anschließend nur Tracks mit mehr als fünf Hits analysiert.

Bei der Qualitätskontrolle unter den definierten Einschränkungen ergaben sich die in Tabelle 5.9 dargestellten Ergebnisse.

Unter den vorgestellten Einschränkungen werden insgesamt 1073 der ursprünglich 1637 Tracks analysiert. Der Anteil der rekonstruierten Tracks steigt dabei auf 77.52%. Aufgrund der hohen Grenzwerte für Way-Follower und nachträgliches Hinzufügen von Hits existiert auch weiterhin noch ein hoher Anteil unreiner Tracks. Dies könnte geändert werden, indem die Grenzwerte mehr zugunsten der Reinheit und weniger zugunsten der Anzahl gefundener Tracks bestimmt werden. Der Anteil Tracks mit weniger als 70% gefundener Hits ist auf 8% gefallen. Dies weist darauf hin, dass durch die Einschränkungen vornehmlich Tracks entfernt wurden, welche prinzipiell schon rekonstruiert werden können. Die Anzahl der korrekt zugeordneten Hits dieser Tracks ist jedoch zu klein, damit diese den gefunden Tracks zugeordnet werden können. Der Anteil der Tracks, von denen keine Hits gefunden wurden konnte jedoch durch die Einschränkungen nicht so stark verringert werden. Dies ist meist auf Probleme mit der Wegverfolgung zurückzuführen. Abbildung 5.10 zeigt einen problematischen Fall. Der Track verläuft zu Beginn durch das Loch in der Mitte des FTS. Erst ab der dritten Schicht von FTS4 erzeugt der Track Hits. Das Verfahren wurde so konstruiert, dass die in FTS1 und FTS2 gefundenen Tracks durch das Magnetfeld verfolgt werden. Wenn die Tracks also in den vorderen Schichten nicht gefunden wurden ist eine Verfolgung dieser Tracks nicht möglich. Eine generelle Erkennung von neuen Tracks im Magnetfeld ist im Verfahren nicht vorgesehen. Dies hängt damit zusammen, dass Tracks mit einer hinreichend guten Genauigkeit bekannt sein müssen um effektiv durch das Ma-

gnetfeld verfolgbar zu sein. Diese Genauigkeit wird in den Stationen FTS1 und FTS2 durch die lineare Regression erzielt. Sind jedoch nur gefundene Teilapproximationen aus den Schichten FTS3 oder FTS4 verfügbar sind die Ungenauigkeiten, insbesondere in der Y-Koordinate, zu groß um die nachfolgenden Teilapproximationen zu finden.



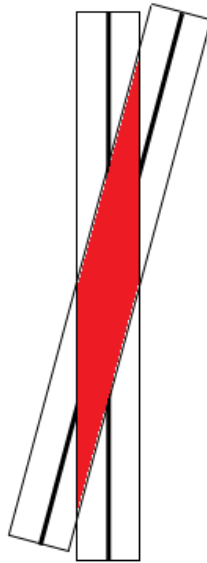
**Abbildung 5.10.:** Abbildung eines Tracks, von dem keine Hits gefunden wurden

## 5.5. Fehleruntersuchung in Y-Richtung

Es wurde festgestellt, dass die rekonstruierten Tracks in der XZ-Projektion die korrekten Tracks sehr gut approximieren. In der YZ-Projektion wurden jedoch erhebliche Abweichungen zum korrekten Track festgestellt. In Abbildung 4.10 wird diese unterschiedliche Genauigkeit besonders deutlich. Deshalb wurde untersucht, worauf diese unterschiedlich hohe Auflösung zurückzuführen ist.

Zu Beginn wird eine intuitive Erklärung eingeführt die Idee zu verdeutlichen. Später folgt dann eine mathematische Erklärung, in der auch ein fester Wert für die Fehlerverstärkung berechnet wird. Die berechneten Ebenen liegen nicht vollständig parallel zur exakten Teilchenflugbahn. Die Gründe dafür werden ebenfalls noch

im späteren Verlauf dieses Abschnitts erläutert. Dies führt dazu, dass sich für jede Ebene ein Band ergibt, welches die mögliche Abweichung nach links und rechts beschreibt. Die korrekte Ebene liegt irgendwo innerhalb dieses Bandes.



**Abbildung 5.11.:** Schematische Abbildung der Fehlerverstärkung bei ungenauen Berechnungen der Ebenen

Abbildung 5.11 zeigt die Bänder zweier Ebenen, welche miteinander geschnitten werden sollen. Die Schnittgerade verläuft also irgendwo durch die rot markierte Fläche. Aus der Abbildung wird deutlich, dass die Ungenauigkeiten in Y-Richtung für einen kleinen Winkel zwischen den Ebenen wesentlich größer sind als die Ungenauigkeiten in X-Richtung. Bei den dargestellten Ebenen wurde ein Winkel von  $20^\circ$  verwendet. Der Neigungswinkel der Straw-Tubes beträgt jedoch nur  $5^\circ$ . Dies bedeutet, dass bei einer solchen Konstruktion des Detektors die Fehlerverstärkung in Y-Richtung noch um ein vielfaches größer als in der Abbildung ist. Es folgt nun eine genauere Herleitung der Fehlerverstärkung.

Ungenauigkeiten im implementierten Verfahren sind auf Ungenauigkeiten in den Isochronenradien zurückzuführen. Diese Ungenauigkeiten führen dazu, dass die approximierte Gerade von dem korrekten Track abweicht.

Abbildung 5.12 zeigt zwei Straw-Tubes mit Isochronen. Die schwarze Gerade entspricht dem korrekten Track. Die fehlerbehafteten Isochronen und die daraus fehlerhaft rekonstruierte Gerade sind in rot dargestellt. In dem kleinen Bereich um die Straw-Tubes weichen beide Geraden nicht stark voneinander ab. Im implementierten Verfahren wird die generierte Gerade jedoch bis zur nächsten Doppelschicht propagiert. Die fehlerbehaftete Gerade trifft also an einem Punkt auf die nächste

Ebene Straw-Tubes, der  $d_1$  von der korrekten Gerade entfernt ist. Es wird nun untersucht, wie stark sich dieser Fehler  $d_1$  auf die Y-Koordinate der rekonstruierten Gerade auswirkt. Zur Bestimmung der Y-Koordinate wird aus der approximierten Gerade eine Ebene erstellt, welche mit der generierten Ebene der nächsten Schicht geschnitten wird. Da die approximierte Gerade in der zweiten Schicht um  $d_1$  von der korrekten Gerade abweicht, liegt in der XY-Projektion die generierte Ebene auch um die Distanz  $d_1$  verschoben.

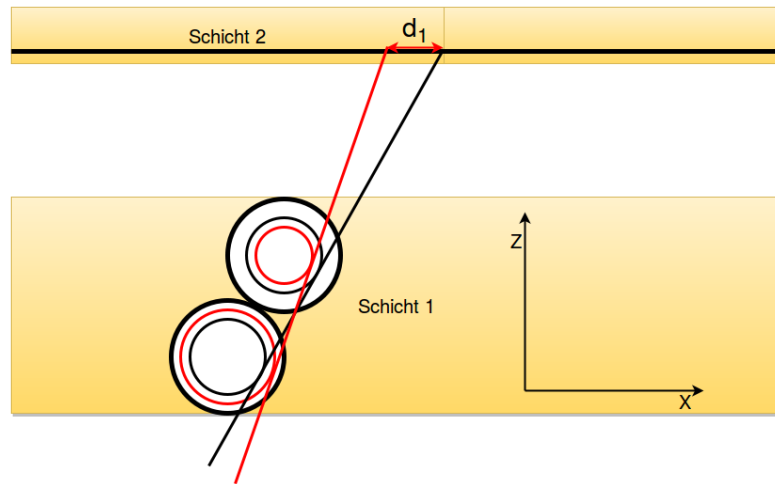
Abbildung 5.13 stellt die generierten Ebenen in der XY-Projektion an der Z-Position der nächsten Schicht Straw-Tubes dar. Die Darstellung entspricht also einer Ansicht, bei der in Richtung der Z-Achse auf die in Abbildung 5.12 als Schicht 2 markierte schwarze Fläche geschaut wird. Die generierten Ebenen erscheinen in dieser Projektion also als Schnittgeraden mit dieser Fläche. Bei der senkrechten schwarzen Ebene handelt es sich um die korrekte Ebene aus der ersten Schicht. In rot ist die um  $d_1$  verschobene, fehlerbehaftete Ebene der ersten Schicht zu sehen. Die um den Winkel  $\alpha$  geneigte Ebene, wurde in der zweiten Schicht generiert. Der Winkel  $\alpha$  entspricht dem Neigungswinkel der Straw-Tubes. Bei der blau markierten Fläche handelt es sich um ein rechtwinkliges Dreieck. Mit Hilfe des Tangens kann also die Differenz  $d_2$  angegeben werden.

$$\tan(\alpha) = \frac{d_1}{d_2} \Leftrightarrow d_2 = \frac{d_1}{\tan(\alpha)}$$

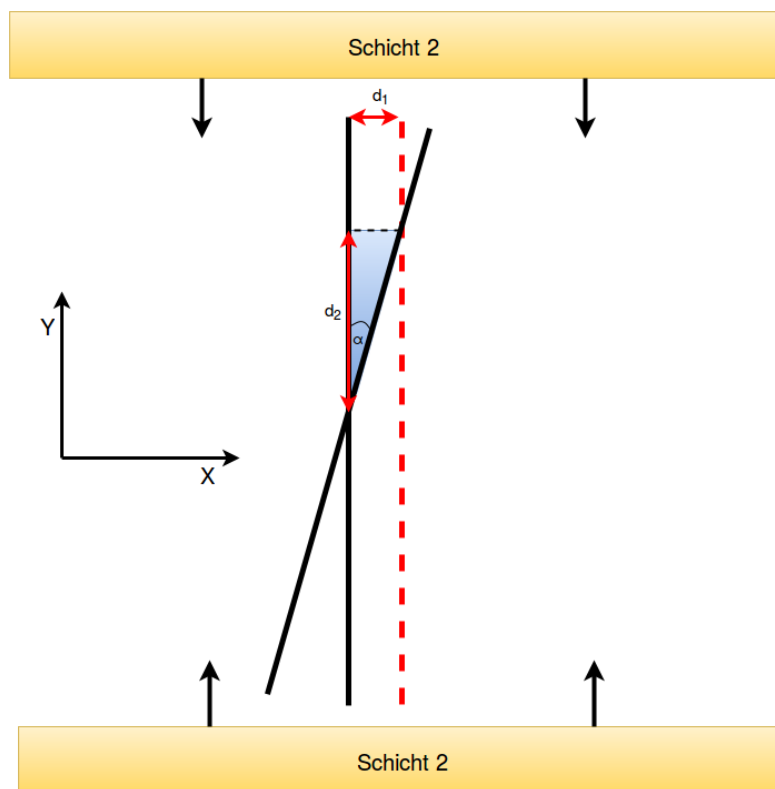
Da die Straw-Tubes der mittleren beiden Schichten um jeweils  $\pm 5^\circ$  gedreht sind, gilt  $\alpha = 5$ . Daraus ergibt sich die folgende Fehlerverstärkung:

$$d_2 = \frac{d_1}{\tan(5)} = \frac{d_1}{0.087} = 11.43 \cdot d_1$$

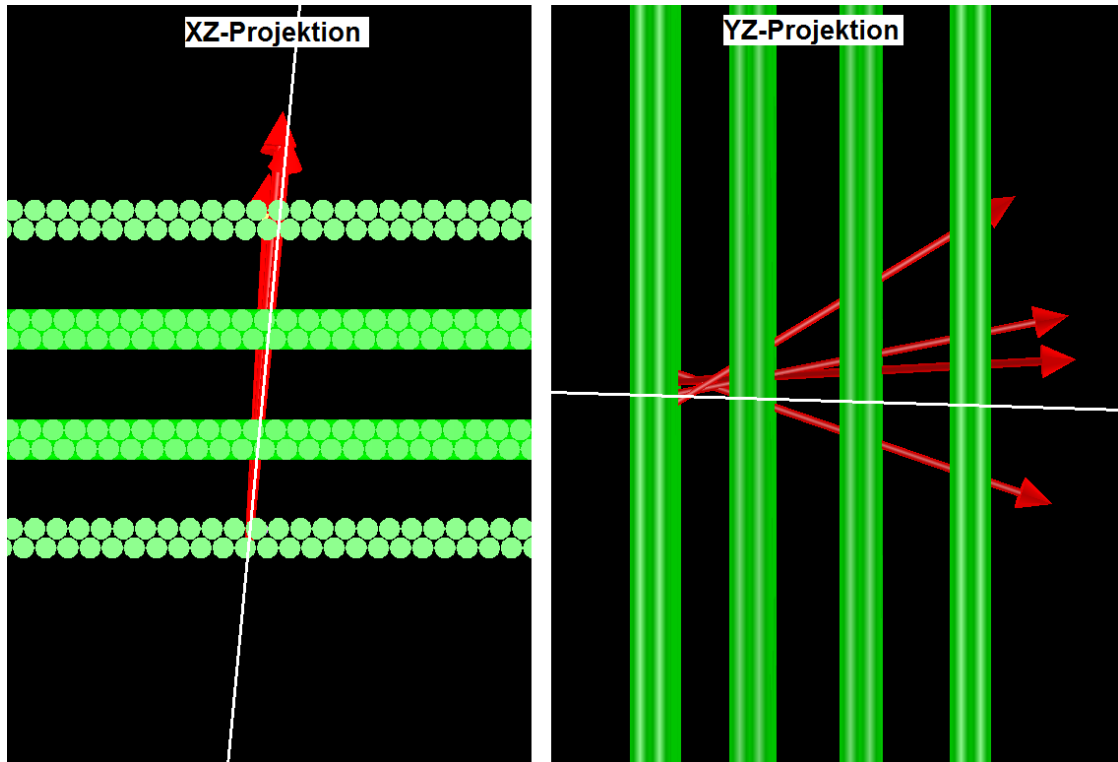
Somit wird der Fehler aus der XZ-Projektion bei einem Drehwinkel der Straw-Tubes von  $5^\circ$  um das 11.43-fache verstärkt. Da der Tangens für kleine  $\alpha$  gegen 0 geht, wird die Auflösung in Y-Richtung schlechter, je kleiner der Winkel  $\alpha$  gewählt wird. Dieser Problematik könnte entgegengewirkt werden, indem bei der Konstruktion des Detektor ein größerer Neigungswinkel verwendet wird. Damit könnte die Y-Auflösung des Detektors erheblich verbessert werden. Bei der vorgenommenen Betrachtung wurde zur Vereinfachung der Berechnung nur der Fehler in der senkrechten Ebene berücksichtigt, der Fehler in der gedrehten Ebene jedoch vernachlässigt. Diese Vereinfachung wirkt sich jedoch nicht signifikant auf die Fehlerverstärkung aus, da dieser Fehler bei kleiner Z-Abweichung von der geneigten Ebene sehr klein ist. Der Fehler  $d_1$  ist jedoch nicht zu vernachlässigen und wird größer, je weiter die vorgenommene Approximation propagiert wird.



**Abbildung 5.12.:** Schematische Abbildung einer fehlerbehafteten Approximation



**Abbildung 5.13.:** Schematische Abbildung der Fehlerverstärkung



**Abbildung 5.14.:** Darstellung der aus fehlerhaften Isochronen generierten Tracks

FairRoot bietet die Möglichkeit, den maximalen Fehler des Isochronenradius abzufragen. Um die soeben durchgeführte, mathematische Betrachtung auf Plausibilität zu überprüfen, wurden die Auswirkungen der Fehlerverstärkung anhand von Simulationsdaten überprüft. Dazu wurde für jeden Hit jeweils der positive und der negative Fehler auf den Isochronenradius addiert. Bei zwei Hits in einer Schicht ergeben sich dabei insgesamt vier verschiedene Approximationen:

1. beide Hits mit negativem Fehler
2. beide Hits mit positivem Fehler
3. erster Hit mit positivem, zweiter Hit mit negativem Fehler
4. erster Hit mit negativem, zweiter Hit mit positivem Fehler

Somit entstehen insgesamt vier Approximationen, wobei die beiden Äußeren ein Band aufspannen, in dem die mögliche Teilchenflugbahn liegt. Mit diesen vier Approximationen wurde das implementierte Verfahren zur Bestimmung der Y-Koordinaten gestartet. Dabei werden diese in die zweite Schicht propagiert, wobei

sich der verstärkte Fehler auf die Berechnung auswirkt. Abbildung 5.14 zeigt das Ergebnis des Verfahrens. Es wird deutlich, dass der Fehler in der XZ-Projektion nur sehr gering ist und alle Approximationen nur minimal vom Track abweichen. In der YZ-Projektion hingegen weichen die Approximationen erheblich vom Track ab. Das Ergebnis des implementierten Verfahrens liegt hier irgendwo innerhalb des Bandes, welches von den äußersten Approximationen aufgespannt wird. Folglich ist hier eine erhebliche Abweichung möglich. Dieser Problematik wurde durch eine lineare Regression in der YZ-Ebene entgegengewirkt. Liegen Hits aus allen sechs FTS-Stationen vor, kann somit die Ungenauigkeit in der Y-Position stark reduziert werden. Darüber hinaus wurden die soeben vorgestellten Erkenntnisse auch bei der Untersuchung zweier Geraden auf Ähnlichkeit berücksichtigt. Die Ähnlichkeit zweier Geraden in der XZ-Projektion wird mit 0.9 gewichtet, wohingegen die Ähnlichkeit im dreidimensionalen Raum mit nur 0.1 gewichtet wird. So können die meisten Geraden unterschieden werden, welche auf einer unterschiedlichen Y-Höhe liegen. Ungenauigkeiten in der Y-Position der rekonstruierten Teilapproximationen führen jedoch nicht dazu, dass zwei zusammengehörige Teiltracks nicht verschieden eingestuft werden.

## 6. Fazit und Ausblick

Im Rahmen dieser Arbeit wurde ein Verfahren entwickelt, welches in der Lage ist Teilchenflugbahnen im FTS zu rekonstruieren. Dazu wurden zunächst Teilapproximationen innerhalb der einzelnen Schichten generiert und diese dann Schrittweise zu größeren Tracks zusammengeführt. Darüber hinaus wurde mit Hilfe von geometrischen Überlegungen Ebenen erzeugt und Schnittgeraden berechnet um die Y-Positionen der Hits anzunähern. Hier konnten teils massive Probleme festgestellt werden, welche zu einer sehr ungenauen Rekonstruktionsqualität in der YZ-Projektion führen. Durch eine mathematische Analyse wurden diese Ungenauigkeiten der Y-Positionen mit dem geringen Neigungswinkel der gedrehten Straw-Tubes erklärt. Die abschließend durchgeführte Qualitätskontrolle der rekonstruierten Tracks ergab, dass insgesamt 61% aller Tracks rekonstruiert werden können. Als problematisch wurden im Besonderen stark kreisende Tracks, Tracks mit wenigen Hits und Events mit sehr vielen Tracks betrachtet. Werden diese Tracks bei der Analyse nicht betrachtet werden 77% aller Tracks rekonstruiert. Es besteht folglich noch Verbesserungsbedarf beim implementierten Verfahren. Hier könnte beispielsweise der Way-Follower derart verändert werden, dass das Verfahren den Grad der Ablenkung des Tracks über den Winkel zwischen den Teilapproximationen bestimmt. Somit kann der Track genauer als mit der Approximation einer Geraden verfolgt werden. Darüber hinaus könnten dann Tracks, welche in zwei verschiedene Richtungen abgelenkt werden als ungültig identifiziert werden. Dies ist möglich, da die Ablenkung abhängig von der Ladung entweder nach rechts oder nach links erfolgt, sich bei gleichem Verlauf der Feldlinien jedoch niemals ändern kann. Die Optimierung der Verfahrensparameter wurde über eine Parameterstudie durchgeführt. Dabei wurden nicht alle Parameter gemeinsam optimiert. Hier ergibt sich ebenfalls noch Optimierungspotential, indem alle Parameter gleichzeitig über ein Gradientenabstigsverfahren optimiert werden. Dazu wird jedoch zusätzliche Rechenleistung benötigt, sodass dies im Rahmen dieser Arbeit nicht möglich war.

Darüber hinaus müssen für ein reales Experiment die gemessenen Daten noch in Events zerlegt werden. Diese Problematik liegt bei einer Simulation mit PandaRoot nicht vor, da die Simulationsdaten bereits in Events zerlegt generiert werden. Bei einem realen Experiment ist dies jedoch nicht der Fall. Hier kann es auch zu der Problematik kommen, dass sich Tracks verschiedener Events überschneiden. Dabei könnte das FTS Teile von Tracks messen, welche eigentlich zu einem anderen Event gehören. Daraus ergeben sich zusätzliche Anforderungen an das Verfahren. Zunächst muss eine Analyse darüber erfolgen, wie robust das Verfahren auf sol-

che Tracks reagiert. Dabei können eventuelle Problematiken ersichtlich werden, sodass das Verfahren angepasst werden muss. Darüber hinaus müssen diese eventübergreifenden Teiltracks erkannt und den Tracks aus anderen Events zugeordnet werden.

Hinsichtlich des Laufzeitverhaltens bestehen an die Rekonstruktionsverfahren besondere Anforderungen, da die Rekonstruktion während des Experiments durchgeführt werden muss. Dies hat den Hintergrund, dass die vom Detektor erzeugte Datenmenge erheblich zu groß für eine vollständige Speicherung ist. Durch eine Parallelisierung kann die Laufzeit des Verfahrens verbessert werden. Das implementierte Verfahren eignet sich sehr gut für eine Parallelisierung auf Algorithmusebene, da die Rekonstruktion der Teiltracks in den einzelnen Schichten vollständig unabhängig voneinander durchgeführt wird. Lediglich die Zusammenführung der einzelnen Teiltracks muss durch seriellen Code durchgeführt werden. In der vorangegangenen Arbeit von Jette Schumann [5] wurde jedoch bereits festgestellt, dass eine Parallelisierung auf Algorithmusebene mit Grafikkarten sehr ineffektiv ist. Dies ist darauf zurückzuführen, dass die Übertragung der zu verarbeiteten Daten in den Grafikspeicher im Verhältnis zur eigentlichen Spurrekonstruktion zu viel Zeit in Anspruch nimmt. Deshalb eignet sich eine Parallelisierung auf Eventebene deutlich besser.

# A. Anhang

## A.0.1. Beschreibung der implementierten Klassen

Es folgt nun eine detaillierte Beschreibung der wichtigsten implementierten Klassen und der relevanten Methoden.

### PndFtsTrackletGenerator (Abbildung A.1)

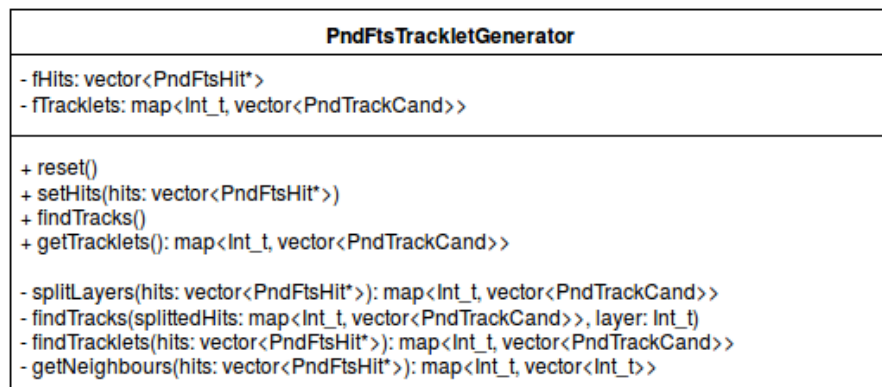


Abbildung A.1.: UML-Diagramm der Klasse PndFtsTrackletGenerator

**fHits** Dieser Vektor enthält alle FTS-Hits, welche in einem Event aufgetreten sind.

**fTracklets** Das Ergebnis des TrackletGenerators wird in einer Map gespeichert, welche von der Layer-ID auf einen Vektor aus Track-Kandidaten der entsprechenden Schicht abbildet.

**reset()** Diese Methode sollte nach jedem Event aufgerufen werden, um alle im TrackletGenerator verwendeten Datenstrukturen zurückzusetzen. Beispielsweise wird in der reset-Methode die Map mit den Track-Kandidaten geleert.

**setHits(vector<PndFtsHit\*> hits)** Vor der Bearbeitung eines Events müssen über diese Schnittstelle die Hits initialisiert werden, welche im entsprechenden Event aufgetreten sind.

**findTracks()** Um den Prozess der Tracklet-Generierung zu starten, werden in dieser Methode zunächst die Hits in Schichten aufgeteilt und der zelluläre Automat dann für die verschiedenen Schichten aufgerufen.

**splitLayers(vector<PndFtsHit\*> hits)** Diese Methode benutzt die Layer-ID der PndFtsHits, um eine Map zu erstellen, welche von der Layer-ID auf einen Vektor aller Hits der jeweiligen Schicht abbildet.

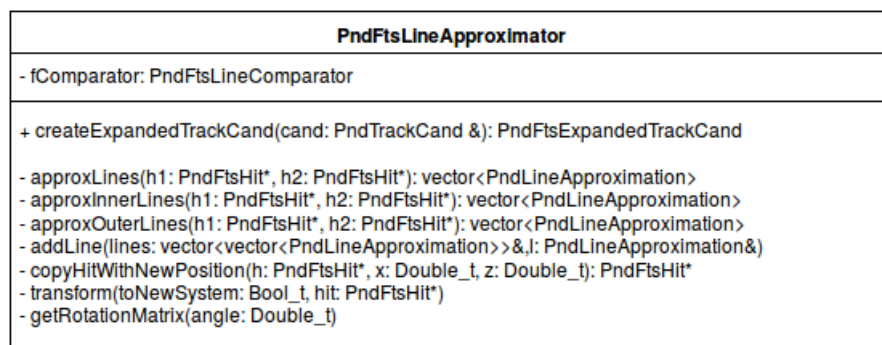
**findTracks(map<Int\_t, vector<PndTrackCand>> splittedHits)**

Zunächst werden alle benötigten Hits der gewünschten Schicht von der Map abgefragt. Anschließend wird der Tracklet-Generator mit diesen Hits aufgerufen und das Ergebnis in fTracklets für die entsprechende Schicht abgespeichert. Dabei werden aus den vom zellulären Automaten zurückgegebenen Hits Objekte der Klasse PndTrackCand erzeugt.

**findTracklets(vector<PndFtsHit\*> hits)** Diese Methode enthält den eigentlichen zellulären Automaten. Zunächst wird die Methode getNeighbours genutzt um die Nachbarschaftsbeziehungen der übergebenen Hits zu generieren. Anschließend werden die Zustände der Hits initialisiert und mittels des beschriebenen Verfahrens so lange iteriert, bis sich kein Zustand mehr ändert. Abschließend wird eine Map erzeugt, welche von der Track-ID auf alle Hits abbildet, welche zu einem Track-Kandidaten gruppiert wurden.

**getNeighbours(vector<PndFtsHit\*> hits)** Erzeugt eine Map, welche von der Tube-ID einer Straw-Tube auf die Tube-IDs aller benachbarter Straw-Tubes abbildet.

## PndFtsLineApproximator (Abbildung A.2)



**Abbildung A.2.:** UML-Diagramm der Klasse  
PndFtsLineApproximator

**fComparator** Dieses Objekt der Klasse PndFtsLineComparator wird verwendet um zwei Geraden auf Ähnlichkeit zu untersuchen.

**createExpandedTrackCand(PndTrackCand& cand)** Beim Aufruf dieser Methode wird aus dem übergebenen Track-Kandidaten ein Objekt der Klasse PndFtsExpandedTrackCand erstellt. Dazu werden die Methoden approxLine und copyHitWithNewPosition benutzt, um abhängig von den Isochronenradien Geraden zu erzeugen und die korrigierten Hit-Positionen zu generieren. Besitzt der Track-Kandidat mehr als zwei Hits wird in dieser Methode geprüft, ob eine eindeutige Approximation vorliegt.

**approxLines(h1: PndFtsHit\*, h2: PndFtsHit\*)** In dieser Methode werden die Methoden approxInnerLines und approxOuterLines aufgerufen und die Resultate in einem Vektor zusammengeführt.

**approxInnerLines(h1: PndFtsHit\*, h2: PndFtsHit\*)** Generiert die beiden inneren Geraden abhängig von den Isochronenradien der übergebenen Hits.

**approxOuterLines(h1: PndFtsHit\*, h2: PndFtsHit\*)** Generiert die beiden äußeren Geraden abhängig von den Isochronenradien der übergebenen Hits.

**addLine(lines: vector<vector<PndLineApproximation>>&, l: PndLineApproximation)**

Diese Methode wird nur bei Track-Kandidaten benötigt, welche mehr als zwei Hits beinhalten. Der Parameter lines repräsentiert einen Vektor, welcher Vektoren aus ähnlichen Geraden enthält. Wird eine neue Gerade hinzugefügt, wird überprüft ob diese Gerade einer dieser Gruppen zugeordnet werden kann. Ist dies nicht der Fall, wird diese in eine neue Gruppe eingeordnet. So kann später die Gruppe mit den meisten Elementen bestimmt und ein Refit aller Elemente durchgeführt werden um die eindeutige Approximation für den gesamten Track-Kandidaten zu finden.

**copyHitWithNewPosition(h: PndFtsHit\*, x: Double\_t, y: Double\_t: z)** Kopiert den übergebenen Hit und korrigiert die Hit-Position.

**transform(toNewSystem: Bool\_t, hit: PndFtsHit\*)** Führt eine Koordinatentransformation abhängig von dem Neigungswinkel der Straw-Tube durch. Der Parameter toNewSystem gibt an, ob in das gedrehte System oder zurück in das PandaRoot-Referenzsystem System zurück transferiert werden soll.

**getRotationMatrix(angle: Double\_t)** Generiert eine Rotationsmatrix, welche das Koordinatensystem um den Winkel angle um die Z-Achse dreht.

### PndFtsExpandedTrackCand (Abbildung A.3)

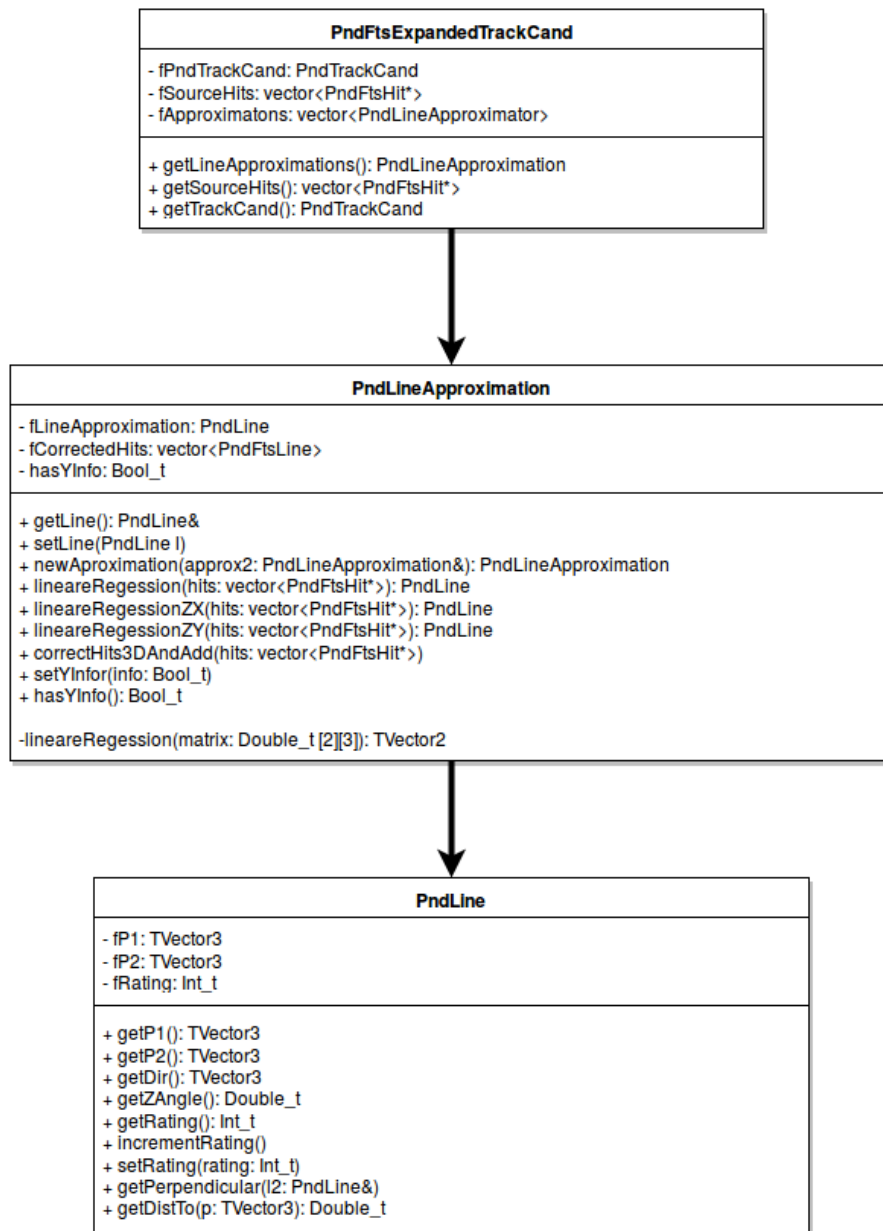


Abbildung A.3.: UML-Diagramm der Klassen PndFtsExpandedTrackCand, PndLineApproximation und PndLine

**fPndTrackCand** Speichert den Track-Kandidaten, welcher durch diese Klasse mit möglichen Approximationen erweitert werden soll

**fSourceHits** Vektor mit den originalen Hits, welche nicht die korrigierte Hit-Position enthalten

**fApproximations** Vektor mit Objekten der Klasse PndLineApproximation. Diese repräsentieren mögliche Approximationen des erweiterten Track-Kandidaten.

### **PndLineApproximation (Abbildung A.3)**

**fLineApproximation** Speichert ein Objekt der Klasse PndLine, welches die Gerade der Approximation repräsentiert.

**fCorrectedHits** Vektor mit den korrigierten Hits, welche abhängig von der approximierten Gerade korrigiert worden sind.

**hasYInfo** Es wurde beschrieben, dass beim Ausfall einer gedrehten Schicht die Y-Koordinate einer Teilapproximation nicht berechnet werden kann. Über dieses Attribut wird für jede Approximation gespeichert, ob korrigierte Y-Werte vorliegen.

**newApproximation(PndLineApproximation& approx2)** Im implementierten Verfahren müssen mehrmals ähnliche Approximationen zu einem Gesamt-Track zusammengeführt werden. In dieser Methode wird eine lineare Regression mit den Hits beider Approximationen durchgeführt und anschließend die Hit-Position abhängig von der neuen Approximation korrigiert. Beim Aufruf dieser Methode wird also eine gemeinsame Approximation aus dem übergebenen Parameter und dem Objekt, auf dem diese Methode aufgerufen wurde erstellt. Der Rückgabewert dieser Methode entspricht der zusammengeführten Approximation.

**lineareRegression(vector<PndFtsHit\*> hits)** Beim Aufruf dieser Methode wird eine lineare Regression mit den übergebenen Hits in der ZX-Projektion und in der ZY-Projektion durchgeführt und beide dann zu einer neuen, dreidimensionalen Approximation zusammengeführt.

**correctHits3DAndAdd(vector<PndFtsHit\*> hits)** Alle Hits des übergebenen Vektors werden in ihrer Y-Koordinate abhängig von der aktuellen Approximation korrigiert und dann zu der Approximation hinzugefügt.

**lineareRegression(Double\_t matrix[2][3])** Diese Methode kann benutzt werden um eine lineare Regression für beliebige Punkte durchzuführen. Die übergebene Matrix enthält das Gleichungssystem, welches für die lineare Regression benötigt wird. Der Rückgabewert dieser Methode ist ein zweidimensionaler Vektor, welcher die Koeffizienten der approximierten Gerade enthält.

### PndLine (Abbildung A.3)

**fP1 und fP2** Bei diesen Attributen handelt es sich um Objekte der Klasse TVector3. Diese beiden dreidimensionalen Punkte definieren die Gerade im Raum.

**fRating** Eine initiale Gerade besitzt eine Bewertung von eins. Werden zwei Geraden zu einer neuen zusammengeführt ergibt sich die neue Bewertung aus der Summe der beiden Einzelbewertungen. So kann für jede Gerade entschieden werden, wie viele Teilapproximationen für den rekonstruierten Verlauf der Teilchenflugbahn sprechen. Geraden mit einer geringen Bewertung können dann aussortiert werden, da sie höchstwahrscheinlich nicht die Teilchenflugbahn approximieren.

**getPerpendicular(PndLine& l2)** Diese Methode berechnet den Lotfußpunkt der Geraden auf der diese Methode aufgerufen wurde zur übergebenen Gerade. Diese Methode wird benötigt um die Hit-Positionen in der Y-Komponente zu korrigieren. Dazu wird eine Gerade erstellt, welche durch den bereits in X und Z korrigierten Hit und parallel zur entsprechenden Straw-Tube verläuft. Anschließend wird über die Bestimmung des Lotfußpunktes zum approximierten Track die neue Hit-Position berechnet.

### PndModuleCombiner (Abbildung A.4)

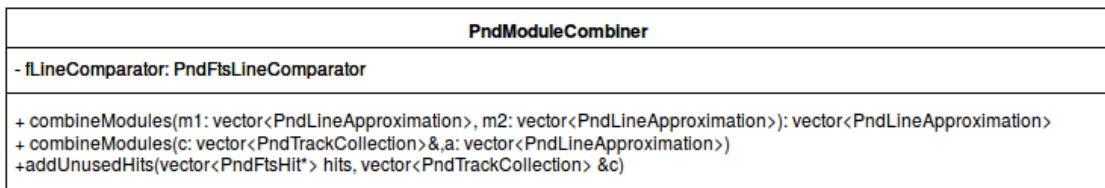


Abbildung A.4.: UML-Diagramm der Klasse PndModuleCombiner

**combineModules(vector<PndLineApproximation> m1, vector<PndLineApproximation> m2)**

Mit dieser Methode werden zwei Teiltracks von unterschiedlichen FTS-Stationen außerhalb des Magnetfeldes zusammengeführt. Dazu wird die Ähnlichkeit der Approximationen verwendet.

**combineModules(vector<PndTrackCollection> & c, vector<PndLineApproximation> a)**

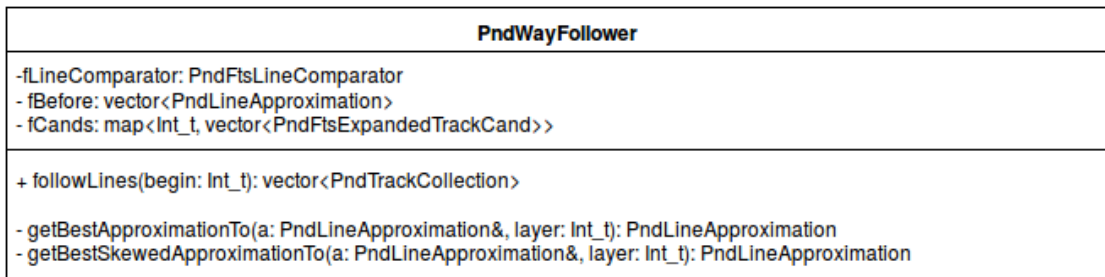
In dieser Methode sucht der Module-Combiner die am besten passende Approximation aus dem Parameter a, welche die PndTrackCollection am geeignetsten ergänzt. Diese Methode wird verwendet, um die zusammengeführten Tracks aus FTS1-FTS4 mit den zusammengeführten Tracks aus FTS5 und

FTS6 zu verbinden. Eine Berücksichtigung des Magnetfeldes ist hier nicht mehr notwendig, die Tracks nach dem Verlassen von FTS4 nicht mehr durch das Magnetfeld beeinflusst werden.

**addUnusedHits(vector<PndFtsHit\*> hits, vector<PndTrackCollection> &c)**

Diese Methode fügt Hits aus dem übergebenen Vektor den TrackCollections hinzu, wenn diese Hits eigentlich zum rekonstruierten Track gehören sollten. Dazu wird der Abstand des Drahtes der zugehörigen Straw-Tube zum rekonstruierten Track benutzt. Wenn dieser Abstand abzüglich einer parametrisierbaren Toleranz kleiner als der Isochronenradius ist, wird der Hit zur Track-Collection hinzugefügt. Dieser freie Parameter wurde ebenfalls im Rahmen einer in Kapitel 5 thematisierten Parameterstudie bestimmt.

**PndWayFollower (Abbildung A.5)**



**Abbildung A.5.:** UML-Diagramm der Klasse PndWayFollower

**fBefore** Dieser Vektor enthält die zusammengeführten Approximationen von FTS1 und FTS2, welche vor dem Magnetfeld rekonstruiert werden. Diese Tracks bieten den Ausgangspunkt für die Wegverfolgung.

**fCands** Die Track-Kandidaten, welche von der Klasse PndFtsLineApproximator erzeugt wurden, werden im Verlauf der Wegverfolgung benutzt, um den Track zu ergänzen.

**followLines(Int\_t begin)** Mit dieser Methode wird die Wegverfolgung der Tracks durch das Magnetfeld ausgeführt. Der Parameter begin steht dabei für die Schicht, in der mit der Verfolgung der Tracks begonnen werden soll.

**getBestApproxTo(PndLineApproximation &a, Int\_t layer)** Diese Methode sucht in der übergebenen Schicht nach der bestmöglichen Approximation, welche die übergebene Approximation geeignet fortsetzt.

**getBestSkewedApproxTo(PndLineApproximation &a, Int\_t layer)** Wenn es sich bei der gewünschten Schicht um eine gedrehte Schicht handelt, kann

diese benutzt werden um die Y-Position der Hits zu bestimmen. Diese Methode funktioniert ähnlich wie die oben beschriebene, bildet jedoch zusätzlich Hilfsebenen entlang der Straw-Tubes und entlang des Tracks um diese miteinander zu schneiden. Dabei entstehen Geraden mit Y-Information, welche zur Fortsetzung des Tracks benutzt werden.

### PndTrackCollection (Abbildung A.6)

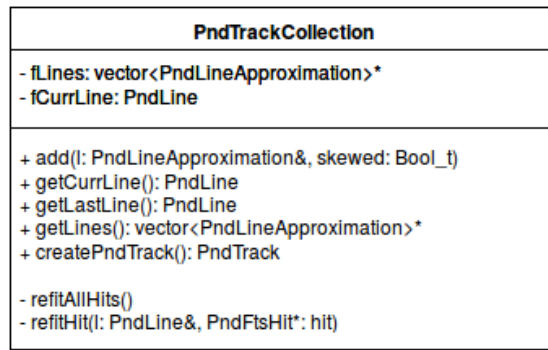


Abbildung A.6.: UML-Diagramm der Klasse PndTrackCollection

**fLines** Eine PndTrackCollection setzt sich aus mehreren Objekten der Klasse PndLineApproximation zusammen, welche die gesamte Teilchenflugbahn auf einem jeweils kleinen Bereich approximieren. Werden diese Teilapproximationen hintereinander gesetzt, ergibt sich der gesamte Track.

**fCurrLine** Der gesamte Track lässt sich in der YZ-Projektion durch eine Gerade beschreiben. Die aktuell berechnete Approximation dieser Geraden wird in diesem Attribut gespeichert.

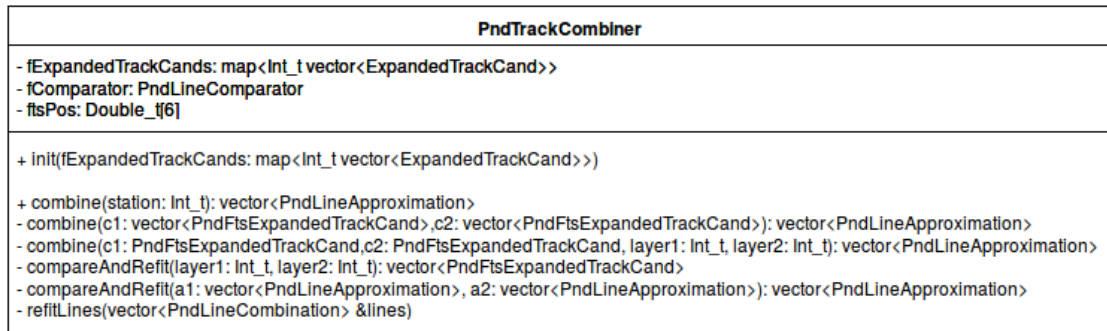
**add(PndLineApproximation& l, Bool\_t skewed)** Über diese Methode können neue Approximationen zur PndTrackCollection hinzugefügt werden. Dabei wird jedes mal die Methode refitAllHits aufgerufen um alle zugehörigen Hit-Positionen mit den zusätzlichen Informationen bezüglich ihrer Y-Position neu zu berechnen. Der Parameter skewed gibt an, ob die hinzugefügte Approximation aus gedrehten Straw-Tubes berechnet wurde.

**getLastLine()** Um die PndTrackCollection mit weiteren Teilapproximationen zu erweitern kann über diese Methode die zuletzt eingefügte Approximation abgefragt werden. Diese wird dann benutzt um eine geeignete Fortsetzung des Tracks zu finden.

**createPndTrack()** Erzeugt ein Objekt der Klasse PndTrack, welches alle Hits des Tracks enthält. Darüber hinaus wird dem PndTrack die Richtung des

Anfangs- und End-Impulses des rekonstruierten Teilchens zugewiesen. Der PndTrack stellt das finale Ergebnis des Verfahrens dar.

### PndTrackCombiner (Abbildung A.7)



**Abbildung A.7.:** UML-Diagramm der Klasse  
PndTrackCombiner

**fExpandedTrackCands** Dieses Attribut speichert die Track-Kandidaten auf denen der Track-Combiner operiert. Diese Daten werden von der Klasse PndLineApproximator erzeugt und über die Methode init an den Track-Combiner übergeben.

**fComparator** Wie bereits die Klasse PndFtsLineApproximator, enthält der Track-Combiner auch ein Objekt der Klasse PndFtsLineComparator. Es werden mehrere Objekte dieser Klasse benötigt, um den Komparatoren unterschiedliche Grenzwerte zuweisen zu können, ab denen zwei Approximationen als ähnlich gelten.

**ftsPos** In diesem Feld mit sechs Elementen befinden sich die Y-Positionen der sechs FTS-Stationen. Diese werden benötigt, um die Durchstoßpunkte der Geraden in den verschiedenen FTS-Stationen zu berechnen und somit den Abstand der Geraden zueinander bestimmen zu können.

**combine(Int\_t ftsStation)** Bei einem Aufruf dieser Methode werden sämtliche Teiltracks innerhalb der übergebenen FTS-Station zusammengeführt und mittels Berechnung der Schnittgeraden auch die Y-Positionen bestimmt. Dazu wird die komplexe Struktur der als Map aus PndFtsExpandedTrackCands gebündelten Teilapproximationen in den überladenen combine-Methoden immer weiter aufgelöst. Diese Methoden werden durch eine unterschiedliche Wahl der Parameterliste überladen. Dies ist notwendig, um die Extraktion der PndLineApproximations aus der Map übersichtlich und leserlich zu implementieren. Dazu werden zunächst die PndFtsExpandedTrackCands von

der Map abgefragt und miteinander kombiniert. In der überladenen Methode werden anschließend aus den jeweiligen `PndFtsExpandedTrackCands` die Approximationen abgefragt um diese miteinander zu kombinieren. In der dritten `combine`-Methode werden dann sämtliche Kombinationen der übergebenen Track-Kandidaten aus beiden Schichten gebildet. Darüber hinaus werden dort auch die Ebenen abhängig vom Neigungswinkel der Straw-Tubes erzeugt und miteinander geschnitten. Alle so entstehenden Teilapproximationen werden dann transitiv bis zur obersten `combine`-Methode zurückgegeben, dort gesammelt und zusammengeführt. Anschließend erfolgt eine Filterung der Approximationen nach der in der Klasse `PndLine` gespeicherten Bewertung. Alle Approximationen mit einer Bewertung von mindestens drei werden anschließend unter Einbeziehung aller Hits erneut gefittet und zurückgegeben.

**`compareAndRefit(Int_t layer1, Int_t layer2)`** Zur Zusammenführung der beiden äußeren Schichten wird diese Methode verwendet. Auch hier wird zunächst die komplexe Struktur aufgelöst und in der überladenen Methode auf Ähnlichkeit untersucht. Im Gegensatz zur `combine`-Methode erfolgt hier keine Bestimmung der Y-Koordinate durch das Bilden von Ebenen, da dies bei gleich geneigten Straw-Tubes zu keinen sinnvollen Resultaten führt. Werden zwei ähnliche Geraden gefunden, werden diese durch eine lineare Regression gemittelt und zurückgegeben. Alle übrigen Geraden werden in diesem Schritt aussortiert.

# Literaturverzeichnis

- [1] PANDA Collaboration. The panda detector - overview. <https://panda.gsi.de/oldwww/framework/detector.php>. Aufgerufen am 26.07.2016.
- [2] Application Software Group Computing and Networks Division. Geant - detector description and simulation tool. <http://hep.fi.infn.it/geant.pdf>. Aufgerufen am 06.08.2016.
- [3] FZJ. Straw-spurdetektoren. <http://www.fz-juelich.de/ikp/DE/Forschung/ExperimentelleEntwicklungen/DriftrohrKammern/DriftrohrKammern.html>. Aufgerufen am 26.07.2016.
- [4] FAIR GmbH. Fair - eine internationale beschleuniger-anlage zur forschung mit antiprotonen und ionen. <http://www.fair-center.de/>. Aufgerufen am 26.07.2016.
- [5] Jette Schumann. Beschleunigung eines Spurfindealgorithmus für den Straw Tube Tracker des PANDA-Detektors durch Parallelisierung mit CUDA C. Master's thesis, FH Aachen, Forschungszentrum Jülich, IKP-1, 2015.
- [6] Jerzy Smyrski. The panda detector - tracking. [https://panda.gsi.de/oldwww/framework/det\\_iframe.php?section=Tracking](https://panda.gsi.de/oldwww/framework/det_iframe.php?section=Tracking). Aufgerufen am 26.07.2016.
- [7] Stefano Spataro. The pandaroot framework for simulation, reconstruction and analysis. *Journal of Physics: Conference Series* 331.3 (2011), S. 032031. doi: 10.1088/17426596/331/3/032031.
- [8] M. Al-Turany u. a. The fairroot framework. *Journal of Physics: Conference Series* 396.2 (2012), S. 022001. doi: doi:10.1088/17426596/396/2/022001.
- [9] Rene Brun und Fons Rademakers. Root — an object oriented data analysis framework. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 389.1 (1997), S. 81–86.
- [10] Fons Rademakers und Rene Brun. Root: An object-oriented data analysis framework. *Linux Journal*, 1998.