Spurrekonstruktion im Silizium-Vertexdetektor des CDF II-Experiments

Zur Erlangung des akademischen Grades eines
DOKTORS DER NATURWISSENSCHAFTEN
von der Fakultät für Physik der
Universität Karlsruhe (TH)

genehmigte

DISSERTATION

von

Stephanie Menzemer
aus Karlsruhe

Tag der mündlichen Prüfung: 31.01.2003

Referent: Prof. Dr. M. Feindt, Institut für Experimentelle Kernphysik

Korreferent: Prof. Dr. G. Quast, Institut für Experimentelle Kernphysik

Deutsche Zusammenfassung

0.1 Einführung

Die Rekonstruktion der Spuren geladener Teilchen in den unterschiedlichen Detektorkomponenten eines modernen Collider-Experiments ist ein anspruchsvoller und wichtiger Teil der Ereignisrekonstruktion und die Grundlage jeder Physikanalyse.
Die vorliegende Dissertation behandelt die Spurrekonstruktion im Silizium-Vertexdetektor des CDF II-Experiments (Collider Detektor at Fermilab, RUN II), das sich am Protonen-Antiprotonen-Collider Tevatron am Fermilab befindet. Nach einer 5-jährigen Erneuerungs- und Aufbauphase werden dort seit Herbst 2001 erneut Daten bei einer erhöhten Schwerpunktsenergie von 2 TeV genommen.
Der Hauptbestandteil der CDF II-Spurrekonstruktionssoftware im Vertexdetektor ist das TrackingKal-Paket, das seit drei Jahren von der Karlsruher Arbeitsgruppe entwickelt wird. Es enthält zwei unterschiedliche Spurrekonstruktions-Algorithmen, einen Outside-In- (OI) und einen Silicon-Standalone-Algorithmus. Die OI-Strategie extrapoliert Spuren, die in der Driftkammer (COT, Central Outer Tracker) gefunden wurden, in den Vertexdetektor und fügt dort weitere Messungen (Hits) hinzu. Dieser Algorithmus ist wegen der limitierten Ausdehnung der COT auf den Zentralbereich des Vertexdetektors beschränkt. Im Vorwärtsbereich wird er durch die Silicon-Standalone-Strategie ergänzt, die ohne Informationen aus zusätzlichen Detektorkomponenten Spuren rekonstruiert. Beide Algorithmen benutzen einen hochoptimierten Kalman-Fitter, der ebenfalls innerhalb des TrackingKal-Pakets entwickelt wurde. Der Fitter dient zum einen dazu, während der Spursuche den Spuren die Siliziummessungen zuzuordnen, ist aber auch der genaueste und ebenso schnellste Fitter, der in der CDF II-Umgebung zur Verfügung steht. Er wird deshalb zum "Final Fit" aller Spuren im Detektor benutzt.

In der vorliegenden Dissertation wird zuerst der Aufbau des CDF II-Experiments und dessen Physik-Programm vorgestellt. Dabei wird näher auf die Geometrie des Silizium-Vertexdetektors eingegangen, insbesondere auf die Details, die während der Spurrekonstruktion oder beim Spurfitten berücksichtigt werden müssen. Das Konzept des Kalman-Fitters wird eingeführt und an einem eindimensionalen Beispiel erläutert. Der erarbeitete Formalismus wird dann auf den 5-dimensionalen Fall des Spurfittens übertragen und dessen mathematische Beschreibung detailliert hergeleitet. Die gute Leistungsfähigkeit des im TrackingKal-Paket implementierten Kalman-Fitters wird auf Monte-Carlo und auf Daten getestet. Auf dem Verständnis des Fitters aufbauend werden die beiden Spurrekonstruktions-Algorithmen vorgestellt und auf Effizienz, Reinheit, Schnelligkeit und Speicherbedarf untersucht. Dabei wird insbesondere auf die starke Korrelation dieser vier Kriterien eingegangen. Zum Schluss werden erste Physik-Resultate präsentiert, die die Funktionsfähigkeit des Vertexdetektors und dessen Spurrekonstruktion demonstrieren.
Die wichtigsten Erkenntnisse und Ergebnisse der Arbeit sind in dieser Zusammenfassung kurz dargestellt.

0.2 Der CDF-Detektor

Das Kernstück des CDF II-Detektors ist das Spurrekonstruktionssystem. Wie auf Abbildung 1.2 zu sehen ist, liegt es innerhalb einer supraleitenden Magnetspule, die ein gleichförmiges Feld von 1.4 T erzeugt. Das Spurrekonstruktionssystem besteht aus drei Komponenten. Das sind die Driftkammer, die auf grund ihrer großen radialen Ausdehnung die Krümmung und damit den Transversalimpuls mit hoher Auflösung misst, das TOF (Time Of Flight)-System, das entscheidend zur Teilchenidentifikation beiträgt, und der Silizium-Streifen-Vertexdetektor. Der Vertexdetektor besteht aus 6-7 doppelseitigen und einer einseitigen Lage, die den Radialbereich von 1.6 - 28 cm abdecken. Alle Lagen messen auf einer Seite die ϕ-Koordinate mit einer Auflösung von etwa 15 μm. Auf vier der doppelseitigen Lagen wird auf der Rückseite mit um 1.2° gedrehten Streifen die sogenannte shallow-angle-stereo (SAS)-Koordinate gemessen. In Kombination mit der ϕ-Messung kann eindeutig die Position des Durchgangs eines geladenen Teilchens durch die Lage in drei Dimensionen berechnet werden. Dafür ist die z-Auflösung dieser 3D-Hits um etwa einen Faktor 50 schlechter $(\frac{1}{\sin(1.2°)})$ als die z-Messungen der übrigen doppelseitigen Lagen (90° z-Lagen). Durch die in Vorwärtsrichtung verschobenen beiden Lagen des ISL (Intermediate Silicon Layers) hat der Vertexdetektor eine geometrische Akzeptanz von $|\eta| \leq 2$ (siehe Abbildung 2.3).
Das Spurrekonstruktionssystem wird außerhalb der Magnetspule durch Kalorimeter und Myonkammern ergänzt.
Eine detaillierte Beschreibung des CDF II-Detektors wird im Technical Design Report (TDR) [5] gegeben.

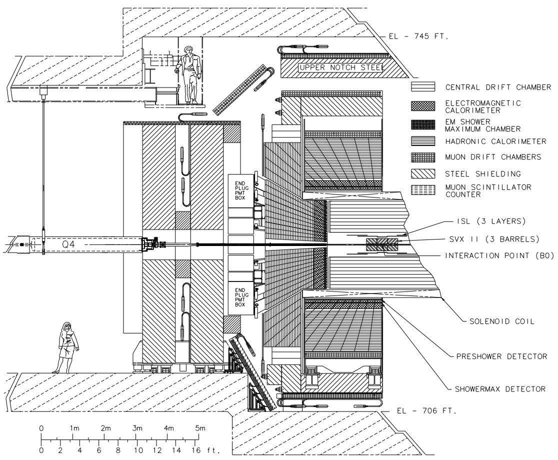


Abbildung 1: Querschnitt des CDF II-Detektors.

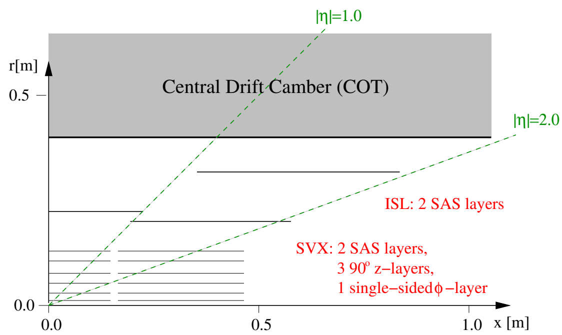


Abbildung 2: Querschnitt des Silizium-Detektors.

0.3 Der Kalman-Fitter

Der Kalman-Fitter ist zuerst in der elektronischen Signalverarbeitung entwickelt worden. Er ist ein progressiver, optimaler Fitteralgorithmus. Der Kalman-Fitter ist zum einem optimal, da er alle verfügbaren Informationen über die Messungen und die statistische Beschreibung des Systems benutzt, auch wenn die zugehörigen Fehler groß sind. Zum anderen ermittelt er - mathematisch bewiesen - die Wahrscheinlichkeitsverteilung für die zu fittenden Parameter, die im Mittel am besten der Wahrheit entspricht, unter folgenden Voraussetzungen:

- das System muss mit linearen Transportmodellen[1] beschrieben werden können;
- die Messfehler müssen Gauß-verteilt sein;
- der Untergrund[2] und die Messfehler müssen unkorreliert sein - im Fall der elektronischen Signalverarbeitung entspricht das "Weißem Rauschen";

Die meisten physikalischen Systeme erfüllen diese Voraussetzung zumindest näherungsweise.
Ein progressiver Fitter ist ein Fitter, der bei Hinzufügen einer neuen Messung nicht auf alle vorangegangen zurückgreifen muss, sondern direkt das letzte Fitergebnis aktualisieren kann. Diese Eigenschaft ist entscheidend für die effiziente Implementation des Fitters z.B. in den Spurrekonstruktions-Algorithmen.

Der im TrackingKal-Paket vorliegende Kalman-Fitter basiert auf dem Prinzip der χ^2-Minimierung. Die Lösung des Minimierungsproblems ist das gewichtete Mittel zwischen der extrapolierten Vorhersage aus vorangegangenen Messungen und jeder neuen Messung. Eine detaillierte Beschreibung des Kalman-Fitter-Formalismus übersteigt den Rahmen dieser Zusammenfassung, kann aber in der englischen Fassung im Kapitel 3 nachgelesen werden. Des weiteren wird auf [24] verwiesen.
Die erzielte Auflösung der Spurparameter in Perigee[3]-Parametrisierung für Spuren mit hohen Transversalimpulsen (p_T) ist (Abbildung 3.14):

- $\frac{\delta p_T}{p_T^2} \approx 3.5 \times 10^{-3} \frac{1}{GeV}$ (nur Silizium)
- $\approx 0.5 \times 10^{-3} \frac{1}{GeV}$ (Silizium+COT)
- $\delta D \approx 10\ \mu$m
- $\delta z_0 \approx 50\ \mu$m

[1] Ein Transportmodell beschreibt im Falle der Spurrekonstrukion die Helixparamter der Spur an einem Radius als Funktion der Helixparamter an einem anderen Radius.
[2] Im Falle der Spurrekonstruktion entspricht der Untergrund der Unsicherheit durch die Streuung bei Durchgang durch Materie (multiple scattering)
[3] Perigee ist der Punkt der Spur mit dem kürzesten Abstand zum Koordinatenursprung.

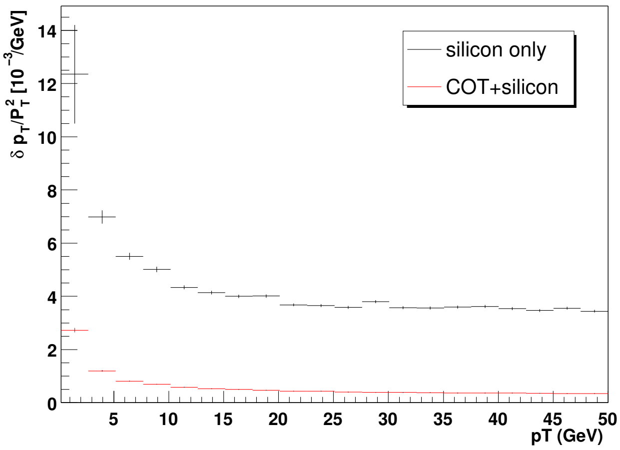


Abbildung 3: Auflösung des Transversalimpulses des Kalman-Fitters.

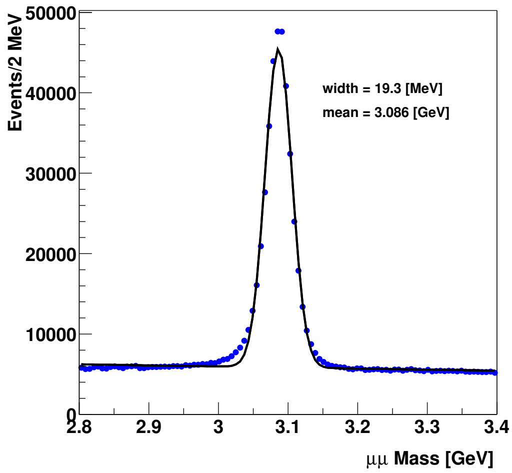


Abbildung 4: Die J/Ψ-Masse, rekonstruiert mit dem Kalman-Fitter des TrackingKal-Pakets (ohne Energiekorrektur).

wobei z_0 die z-Position der Spur am Perigee und D der Impaktparameter ist.
Die Funktionalität des Fitters auf Daten wurde anhand der rekonsturierten J/Ψ-Masse (Abbildung 4) getestet und daraus auch die Energie-Kalibration bestimmt [28]. Die erreichten Auflösungen sind besser als die im TDR erwarteten Ergebnisse.
Der Kalman-Fitter des TrackingKal-Pakets ist der von der Tracking-Gruppe für die Physikanalysen empfohlene Fitter.

0.4 Die Spurrekonstruktions-Algorithmen

Die Spurrekonstruktionssoftware im CDF II-Tracking-System besteht aus einer Kombination verschiedener Algorithmen. Da die Driftkammer sich bei größeren Radien befindet, ist dort die Spurdichte niedriger als im Vertexdetektor. Die Spuren sind isolierter und leichter zu rekonstruieren. Deshalb wird dieser Algorithmus zuerst angewandt. Eine Beschreibung der COT-Algorithmen kann z.B. in [29] und [30] gefunden werden. Der nächste Schritt ist der OI-Algorithmus. Er extrapoliert die in der COT gefunden Spuren in den Vertexdetektor und fügt dort Siliziummessungen hinzu. Jede Spur, die in der COT rekonstruiert wird, kann als gute Spur betrachtet werden. Damit hat der OI-Algorithmus nicht die Schwierigkeit, zwischen zufälligen Kombinationen aus Hits, die eine Helix bilden (Fakes) und wahren Spuren zu unterscheiden. Das ist ein großer Vorteil gegenüber der Silicon-Standalone-Spurrekonstruktion. Diese wird nach dem OI-Algorithmus auf den unbenutzten Hits prozessiert und verwendet neben den Messungen im Vertexdetektor keine zusätzliche Information. Deshalb muß sie alle möglichen Kombination aus Silizium-Hits testen und dabei die wahren Spuren aus einer großen Anzahl von Fake-Spuren heraus filtern. Wegen der größeren geometrischen Akzeptanz des Vertexdetektors ($|\eta| \leq 2$ verglichen mit $|\eta| \leq 1$ für die Driftkammer) ist die Hauptaufgabe der Standalone-Spurrekonstruktion, Spuren im Vorwärtsbereich zu finden. Aber auch Ineffizienzen der COT/OI-Spurrekonstruktion im Zentralbereich können ausgeglichen werden. Der letzte Schritt sollte eine Inside-Out-Strategie sein, die die Standalone-Spuren in die Driftkammer extrapoliert, um dort zusätzliche COT-Signale aufzusammeln. Eine solche Strategie ist derzeit noch nicht in der CDF II-Software eingebaut, ist aber dabei entwickelt zu werden.

0.4.1 Outside-In-Spurrekonstruktion

Der OI-Algorithmus benutzt die COT-Spuren als Ausgangspunkt. Sie dienen gleichzeitig als Referenz des Fitters und als erste Messung der Spur. Die Spur wird zum Schnittpunkt mit der äußersten Siliziumlage extrapoliert. Dort wird in einem mehrere Standardabweichungen großen Fenster um den Schnittpunkt nach weiteren Hits gesucht. Pro Signal wird die Spur einmal kopiert (geklont) und mit jeweils einem der Hits neu gefittet. Um eventuelle Ineffizienzen der Lagen zu berücksichtigen, wird auch ein Klon ohne Hit angefertigt. Alle Klone werden in die nächste Lage extrapoliert. Dort wird um den jeweiligen Schnittpunkt erneut nach weiteren Hits gesucht und die Liste der Spurkandidaten entsprechend weiter aufgespalten (siehe Abbildung 4.1). Dieses Verfahren wird Lage für Lage angewandt. Am Ende wird dann mit Hilfe einer Kombination aus der Anzahl der aufgesammelten Hits und dem χ^2 des Spurfits der beste Klon ausgewählt.

0.4.2 Silicon-Standalone-Spurrekonstruktion

Die Grundidee der Hitsuche im Silizium ist für die Standalone-Strategie dieselbe wie für die OI-Strategie. Der entscheidende Unterschied liegt in der Berechnung der Spurkandidaten. Der OI-Algorithmus verwendet die COT-Spuren, das sind je nach Ereignisgröße bis zu 200. Für den Standalone-Algorithmus gibt es bis zu 50.000 Kombinationsmöglichkeiten, die als Ausgangspunkt für die Spursuche verwendet werden.

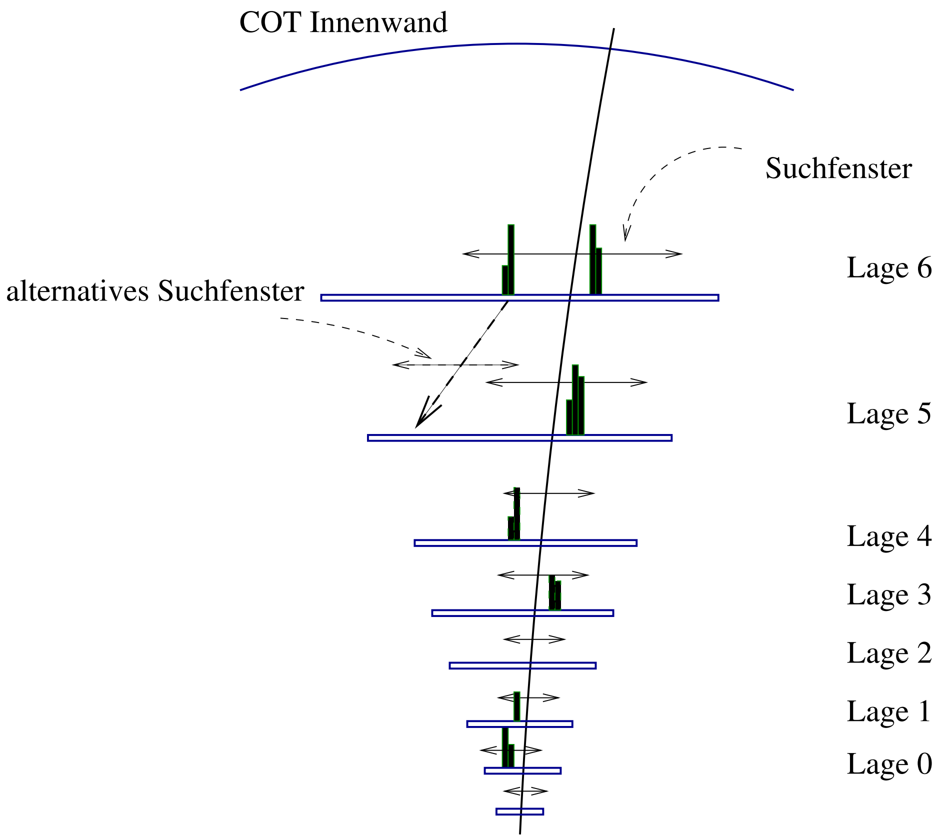


Abbildung 5: Extrapolation einer Spur aus der COT in den Vertexdetektor.

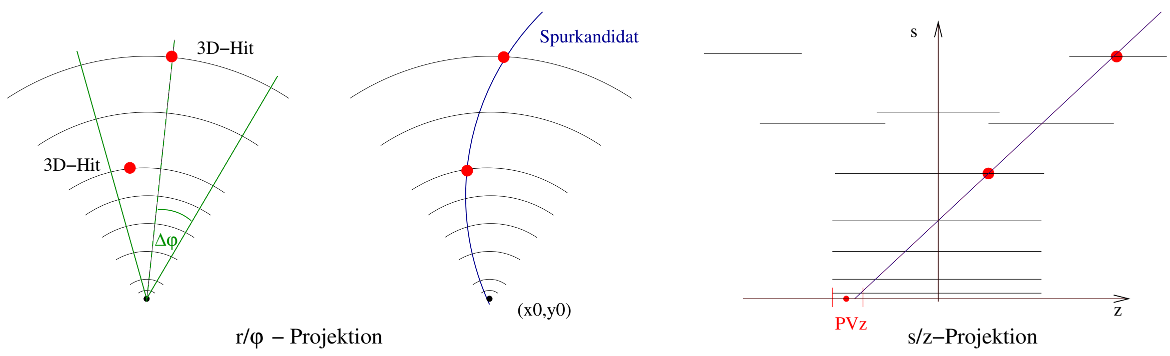


Abbildung 6: Nominierung und Konstruktion der Spurkandiaten des Standalone-Rekonstruktionsalgorithmus.

Nominieren der Spurkandidaten

Um eine dreidimensionale Spur zu bestimmen, benötigt man drei Punkte in der r/ϕ Ebene und zwei in r/z. Die Spurkandidaten des Standalone-Algorithmus werden aus zwei 3D-Punkten der SAS-Lagen und der Strahlposition (x_0, y_0) konstruiert. Obwohl die Strahlposition nicht in den Spurfit eingeht, ist durch diese Wahl der Spurkandidaten die Strategie auf Spuren aus dem Strahlrohr beschränkt. Um die Anzahl der Spurkandidaten zu reduzieren, wird verlangt, dass die beiden 3D-Hits in ihrer ϕ-Position übereinstimmen und die z-Koordinate der Helix am Strahl mit der z-Position eines der zuvor rekonstruierten Primärvertices (PV) konsistent ist. Dadurch ist die Effizienz der Standalone-Strategie direkt mit der Effizienz des PV-Finders gekoppelt. Der PV-Finder-Algorithmus ist auch Teil des TrackingKal-Pakets und wird in [32] ausführlich dokumentiert.

Sortieren der Spurkandidaten

Wird nach der Hitsuche analog zum OI-Algorithmus ein Spurkandiat als echte Spur akzeptiert, werden die dafür benutzten Hits markiert und im folgenden nicht wieder verwendet. Dadurch wird verhindert, Spuren doppelt zu finden. Das bedeutet aber, dass alle Spurkandidaten, die aus Hits der akzeptierten Spur bestehen, automatisch aus der Kandidatenliste gestrichen werden. Da es aus Zeitgründen nicht möglich ist, alle Kandidaten zu untersuchen und hinterher die guten Spuren auszuwählen, ist die Reihenfolge in der die Kandidaten bearbeitet werden, entscheidend. Ein Sortierungskriterium ist der Transveralimpuls, ein zweites die Kombination der SAS Lagen, aus denen der Kandidat konstruiert wurde.
Während bei der OI-Strategie falsch zugeordnete Hits nur die jeweilige Spur betreffen, können Fehlentscheidungen in der Spurrekonstruktion im Standalone-Algorithmus den Verlust mehrerer Spuren nach sich ziehen.

0.5 Validierung der Spurrekonstruktions-Algorithmen

Es gibt verschiedene Kriterien, nach denen ein Algorithmus beurteilt wird. Das sind seine Effizienz, seine Reinheit, seine Auflösung, sein Zeitverhalten und sein Speicherbedarf. Diese Größen sind streng korrelliert, und es gilt, die Strategien in Bezug auf alle Kriterien zu optimieren.
Die Effizienz ist das Verhältnis zwischen rekonstruierten zu rekonstruierbaren Spuren. Deshalb hängt ihr Wert stark von der Definition von rekonstruiert und rekonstruierbar ab. Unabhängig davon ist die Effizienz jedoch ein Maß, das das Verhältnis zwischen

physikalisch interessanten Ereignissen im Detektor und den zugehörigen rekonstruierten Signalen bestimmt.
Die Reinheit beschreibt die Zuverlässigkeit der Spuren, die ein Algorithmus findet. Die Auflösung ist zum einen durch die intrinsische Auflösung der Signale im Vertexdetektor und der Güte des Fitters, zum anderen aber auch von der Anzahl der gefundenen Hits bestimmt. Reinheit und Auflösung sind beide entscheidend, um Signal von Untergrund zu trennen z.B. in der Rekonstruktion von verschobenen Sekundärvertices.
Ein vernünftiges Zeitverhalten und Speicherbedarf sind essentiell, um die Algorithmen auf Trigger-Level oder auf den Produktions-Farmen laufen zu lassen. Der beste Algorithmus ist nutzlos, wenn er nicht in der Lage ist, die vorgefilterten Daten in nahezu Echtzeit zu prozessieren.

0.5.1 Validierung auf Monte-Carlo

Für die folgenden Ergebnisse wurden zwei verschiedene Monte-Carlo (MC)-Samples untersucht. Das erste besteht aus 500 $t\bar{t}$-Ereignissen mit im Mittel drei Poisson-verteilten Untergrundereignissen (minimum bias events (mbr)) entsprechend den erwarteten Ereignis-Multiplizitäten auf Daten. Das zweite besteht aus 5.000 $b\bar{b}$-Ereignissen mit ebenfalls drei mbr-Ereignissen.
Physikalische Ereignisse mit einer ähnlichen Struktur wie die $t\bar{t}$-Ereignisse sind auf Daten selten, ungefähr 10.000 werden in RUN IIA erwartet. Dieses Sample dient als "worst case" Studie. Die meisten Ereignisse auf Daten haben ein Struktur vergleichbar den $b\bar{b}$-Ereignissen.

Effizienz des Outside-In-Algorithmus

Der Nenner der Effizienz der OI-Strategie besteht aus COT-Spuren mit

- mindestens 70 von 96 COT-Hits,
- mindestens 6 verfügbaren ϕ-Messungen im Silizium
- und einem Transversalimpuls größer als 0.5 GeV.

Der Effizienz-Zähler besteht aus allen rekonstruierten Spuren, deren Spurparameter konsistent mit den Spurparametern der MC-Wahrheit innerhalb folgender Grenzen sind:

- $\Delta C \leq 0.00025$ cm^{-1}
- $\Delta D \leq 0.015$ cm
- $\Delta z_0 \leq 0.05$ cm

Dabei ist C die halbe Krümmung, D der Impaktparamter und z_0 die z-Position am Perigee. Da die OI-Spuren mit Hilfe ihrer COT-Spuren den MC-Teilchen zugeordnet werden, gibt es per Definition keine Fake-Spuren im OI-Tracking.

Folgende Effizienzen werden gemessen:

Ereignistyp	keine Schnitte	Schnitte auf C und D	Schnitte auf C, D und z_0
$t\bar{t} + 3$ mbr	98.2 %	92.1 %	86.2 %
$b\bar{b} + 3$ mbr	97.0 %	92.5 %	88.8 %

Für über 97 % der Spuren im Nenner werden zugehörige OI-Spuren gefunden. Die zusätzlichen Schnitte auf Krümmung, Impaktparameter und z-Position verwerfen Spuren, die entweder zu wenig Hits oder falsche Hits im Silizium zugeordnent bekommen haben. Im ersten Fall wird nicht die notwendige Auflösung erzielt, um die Parameterschnitte zu erfüllen, im zweiten Fall wird das Fitergebnis falsch beeinflusst, und deshalb werden die Paramterschnitte ebenfalls nicht erfüllt.

Effizienz des Standalone-Algorithmus

Der Nenner der Effizienz der Standalone-Spuren schließt alle Spuren im Vertexdetektor ein, die

- mindestens zwei 3D-Hits in verschiedene SAS-Lagen
- und mindestens 6 ϕ-Messungen haben,
- innerhalb von ± 1 cm mit der z-Position eines der rekonstruierten PVs kompatibel
- und innerhalb des Strahlrohrs entstanden sind
- und die einen Transversalimpuls größer als 0.5 GeV haben.

Es gibt zwei verschiedene Möglichkeiten die Standalone-Strategie zu validieren. Die erste misst die Effizienz auf der gesamten Kombinatorik, ohne zuvor die OI-Strategie prozessiert zu haben. Die zweite misst die Effizienz, nachdem die OI-Stategie schon den Großteil der Spuren im Zentralbereich rekonstruiert hat. Für die zweite Version gibt es eine zusätzliche Bedingung an den Nenner:

- noch nicht vom OI-Algorithmus rekonstruiert.

Die gefundenen Spuren werden einem simulierten Teilchen zugeordnet, wenn mindestens 70 % der Hits auf der gefundenen Spur zu diesem Teilchen gehören. Alle anderen Spuren werden als Fake-Spuren gewertet. Die Spuren im Zähler müssen ebenfalls zusätzliche Bedingungen erfüllen, die allerdings wegen der schlechteren Auflösung schwächer sind als die des OI-Zählers:

- $\Delta D \leq 0.03$ cm,
- $\Delta z_0 \leq 0.1$ cm.

Die mit dieser Definition gemessenen Effizienzen sind in den Tabellen 5.1 und 5.3 für die beiden unterschiedlichen Ansätze dargestellt. Wie für den OI-Algorithmus hat das $t\bar{t}$-Ereignis eine niedrigere Effizienz als die schwächer besetzten $b\bar{b}$-Ereignisse. Zudem weisen die $t\bar{t}$-Ereignisse eine höhere Fake-Rate auf. Die Effizienz des Standalone-Algorithmus wird stärker durch die Parameterschnitte als die OI-Strategie verringert, besonders durch den Schnitt auf die z-Position. In die dichte Umgebung im Siliziumdetektor wird ohne die zusätzliche COT-Information hineinextrapoliert, was zu einem erhöhten Anteil falsch zugeordneter Hits führt.
Die Ergebnisse der Strategie auf der gesamten Kombinatorik sind besser als die auf den verbleibenden Hits nach dem OI-Algorithmus. Das hängt damit zusammen, dass die Spuren in den Vorwärtsregionen, besonders in z wegen der höheren Pseudorapidität $|\eta|$, eine schlechtere Auflösung haben.

Ereignistyp	keine Schnitte	Schnitt auf D	Schnitte auf D und z_0	Fakes
$t\bar{t} + 3$ mbr	83.8 %	78.6 %	65.8 %	14.6 %
$b\bar{b} + 3$ mbr	92.2 %	88.8 %	80.2 %	3.0 %

Tabelle 1: Effizienz des Standalone-Algorithmus nach dem OI-Algorithmus.

Ereignistyp	keine Schnitte	Schnitt auf D	Schnitte auf D und z_0	Fakes
$t\bar{t} + 3$ mbr	87.1 %	82.8 %	70.7 %	8.2 %
$b\bar{b} + 3$ mbr	95.3 %	93.5 %	87.5 %	1.4 %

Tabelle 2: Effizienz des Standalone-Algorithmus auf der vollen Kombinatorik.

Reinheit

Im Gegensatz zur Effizienz gibt es für den Nenner der Reinheit nur eine Definition. Das sind alle Spuren, die die Spurrekonstruktions-Algorithmen finden, ohne zusätzlichen Schnitte.
Ein Maß der Unreinheit ist die Anzahl der Spuren eines Algorithmus, die keinem simulierten Teilchen zugeordnet werden können. Das sind die oben schon erwähnten Fake-Spuren. Ein weiteres Maß ist der Anteil der falschen Hits auf den verbleibenden Spuren.
Falsche Hits werden den Spuren vor allem in dichten Umgebungen zugeordnet. Dicht ist hier relativ zu der Auflösung der bis dahin gefitteten Spur (σ_{fit}) und zu der Auflösung des Hits (σ_{hit}). Deshalb sind zum Beispiel Ereignisse, die für OI-Spuren nicht dicht erscheinen für Standalone-Spuren dicht, da ihnen die zusätzliche Messung in der COT fehlt. Es wurde ein Maß für die relative Hitdichte eingeführt (Abb.: 5.14):

$$\Delta_{norm} = \frac{||d1| - |d2||}{\sqrt{\sigma_{fit}^2 + \sigma_{hit}^2}} \tag{1}$$

um kritische Situationen in der Spurrekonstruktion zu identifizieren. Gegebenenfalls ist es besser, dort der Spur keinen Hit zuzuordnen, anstatt zu riskieren, sich für einen falschen zu entscheiden.

In den inneren ϕ- und $90°z$-Lagen wird in beiden Algorithmen auf die relative Hitdichte Δ_{norm} geschnitten. Die kritischen Hits werden während der Spurrekonstruktion in den Fit einbezogen, aber markiert. Erst nachdem ein Klon ausgewählt wurde, werden diese Hits von der Spur entfernt. Diese Vorgehensweise ist einleuchtend, wenn man bedenkt, dass einige Prozent der fälschlich als kritisch identifizierten Hits in absoluter Zahl wesentlich mehr sind als die gesamte Anzahl der falsche zugeordneten Hits. Ignoriert man die kritischen Hits gleich bei der Spurrekonstruktion, sind die Suchfenster in der nächsten Lage entsprechend größer, und noch mehr falsche Hits werden aufgesammelt.

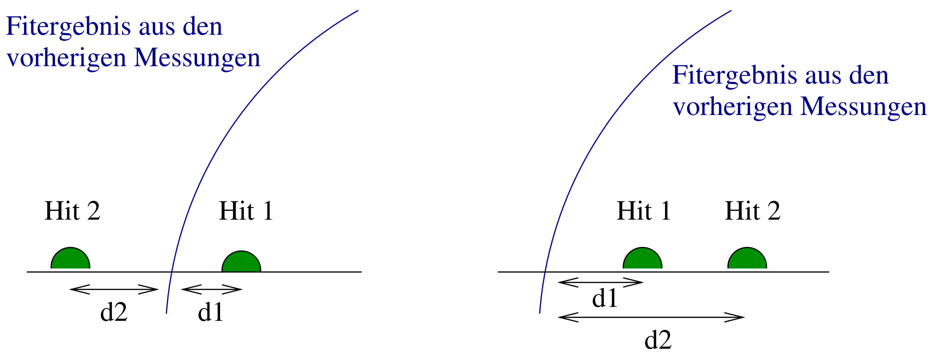
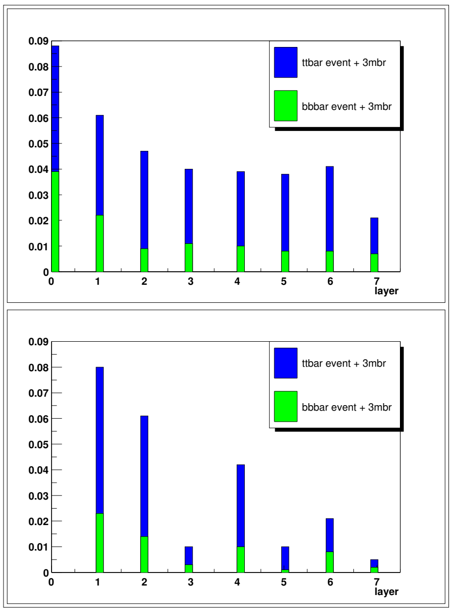


Abbildung 7: Definition der relativen Hitdichte Δ_{norm}.



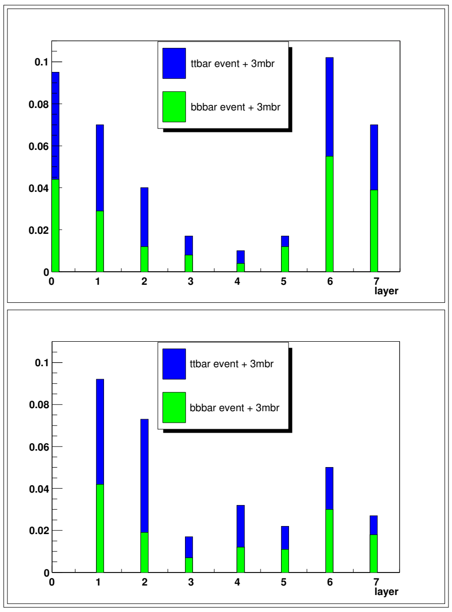


Abbildung 8: Anteil der falsch zugeordneten Hits auf OI-Spuren (ϕ Lagen oben, z und SAS Lagen unten).

Abbildung 9: Anteil der falsch zugeordneten Hits auf Standalone-Spuren (ϕ Lagen oben, z und SAS Lagen unten).

Die Anteile der falschen zugeordneten Hits in den verschiedenen Lagen ist für OI- und Standalone-Spuren in den Abbildungen 5.24 und 5.25 zu sehen.
Der Anteil der falsch zugeordneten Hits in den ISL-Lagen ist für die Standalone-Strategie deutlich höher als für die OI-Strategie. Das liegt daran, dass in den äußeren Lagen der Fit der Standalone-Spuren noch nicht vollständig bestimmt ist. Deswegen ist es nicht möglich, auf χ^2 oder eine andere Größe des Fitters zu schneiden, um zwischen den verschiedenen Hits zu unterscheiden. In den inneren Lagen ist der Anteil der falsch zugeordneten Hits für beide Strategien etwa gleich. Dabei muss man allerdings berücksichtigen, dass die schlechtesten Spuren des Standalone-Algorithmus nicht in diesen Plots auftauchen, da sie keinem MC-Teilchen zugeordnet werden konnten. Insgesamt sieht man, dass die Reinheit auf $b\bar{b}$-Ereignissen wesentlich höher ist als auf den $t\bar{t}$-Ereignissen, da diese wesentlich weniger dicht besetzt sind.

0.5.2 Zeitverhalten

Die folgenden Zeiten sind auf einem 1.4 GHz Athlon mit KCC Version 4_0f ohne Optimierung gemessen. Um das Zeitverhalten der Spurrekonstruktions-Algorithmen im

Silizium besser einzuordnen, sind auch die Zeiten der anderen Beiträge zur Spurrekonstruktion aufgelistet. Mit der derzeitigen Ereignisrate entspricht ein typisches Ereignis auf Daten in seinem Zeitbedarf etwa einem $b\bar{b}$ ohne mbr-Ereignisse.
Das Zeitverhalten ist hinreichend schnell, um die gesamte Spurrekonstruktion auf Trigger Level laufen zu lassen. Es ist das erste Mal an einem Hadronencollider, dass dies möglich ist.

	$t\bar{t}$ + 3 mbr	$b\bar{b}$ + 3 mbr	$b\bar{b}$
COT-Spurrekonstruktion	0.84 s/ev.	0.29 s/ev.	0.10 s/ev.
Clustering im Silizium	0.05 s/ev.	0.04 s/ev.	0.02 s/ev.
PVz-Finder	0.11 s/ev.	0.04 s/ev.	0.02 s/ev.
OI-Algorithmus	0.50 s/ev.	0.12 s/ev.	0.04 s/ev.
Standalone-Algorithmus	0.70 s/ev.	0.22 s/ev.	0.04 s/ev.
Σ	2.2 s/ev.	1.26 s/ev.	0.22 s/ev.

0.5.3 Erste Ergebnisse auf Daten

Der Silizium-Vertexdetektor arbeitet nach einer längeren Testphase jetzt stabil und zuverlässig. Er nimmt in 80 % seiner Module gute Daten. Diese Zahl kann noch erhöht werden, wenn einige Reparaturen durchgeführt werden können. Dafür muss allerdings die Datennahme kurzzeitig ausgesetzt und der Detektor geöffnet werden. So eine Pause ist im Januar 2003 geplant.

Die Validierung der Spurrekonstruktions-Algorithmen auf Daten ist um einiges komplizierter als auf Monte-Carlo, da es keine bekannte Wahrheit als Referenz gibt. Die einzige gute Möglichkeit, die Algorithmen zu validieren, ist die Qualität der Physikanalysen zu studieren, die Siliziumspuren benutzen. Einige erste Physikanalysen sind im Kapitel 6 der Dissertation vorgestellt.
Um trotzdem einen Eindruck von dem Verhalten der Algorithmen auf Daten zu bekommen, wurden die rekonstruierten COT-Spuren als Nenner für die Effizienz der OI-Strategie benutzt. Zum Zähler gehören alle gefundenen OI-Spuren. Damit wurde eine Effizienz von 85 % gemessen. Dieses Ergebnis liegt schon über dem in RUN I erreichten Wert, kann aber mit dem vollständigen Alignment (Ausrichtung des Detektors) und entsprechend angepassten Strategien noch verbessert werden.
Um die Effizienz der Standalone-Strategie zu messen, wurden die OI-Spuren, die mindestens 4 ϕ- und 2 SAS-Hits haben und deren z-Position mit einem der rekonstru-

ierten PVs übereinstimmt, in den Zähler aufgenommen. Alle Standalone-Spure die mindestens 70 % ihrer Hits mit einer OI-Spur gemeinsam haben, wurden als gefunden gewertet. Mit dieser Definition wurde eine Effizienz von 72.5 % erzielt. Lässt man die zusätzlichen Bedingungen an den Zähler bezüglich der minimalen Anzahl der Hits weg, reduziert sich allerdings die Effizienz auf 55.5 %. Das ist verständlich, da der Detektor noch nicht in z ausgerichtet ist. Sind die 3D-Hits in z soweit verschoben, dass der OI-Algorithmus sie nicht den Spuren zuordnen kann, kann der Standalone-Algorithmus auch keine brauchbaren Spurkandidaten aus ihnen bilden.
Die mittlere Anzahl der aufgesammelten Hits der Spuren auf Daten ist im Moment noch um 3.2 niedriger im Vergleich zu den simulierten Ereignisse. Das liegt zum einem an den schon erwähnten, noch nicht korrigierten Verschiebungen der einzelnen Module, zum anderen daran, dass nur 80 % des Vertexdetektors derzeit funktionsfähig sind. Das Verhältnis der Gesamtanzahl der gefundenen Spuren des Standalone-Algorithmus im Vorwärtsbereich und des OI-Algorithmus im Zentralbereich ist um etwa einen Faktor zwei niedriger als auf Monte-Carlos. Dies ist ein eindeutiger Hinweis auf Probleme im Vorwärtsbereich. Diese Beobachtung ist konsistent damit, dass ein Großteil der nichtfunktionsfähigen Module im Vertexdetektor im Vorwärtsbereich liegt, und dieser noch nicht so gut ausgerichtet ist wie der Zentralbereich.

0.6 Schlußbemerkung

Die vorliegende Zusammenfassung stellte kurz die Spurrekonstruktions-Algorithmen und den Kalman-Fitter des TrackingKal-Software-Pakets vor. Die Studien auf Monte-Carlo und auf Daten zeigten, dass der Silizium-Vertexdetekor und dessen Funktionsweise gut verstanden sind.
Obwohl der größte Teil der Software fertiggestellt ist, geht die Arbeit am Verständnis des Detektors und der Algorithmen auf Daten weiter. Insbesondere mit der Umstellung auf den RUN IIB-Vertexdetektor wird erneuter Wartungsaufwand entstehen. Deshalb ist es wichtig, dass die Spurrekonstruktionssoftware gut dokumentiert ist, insbesondere, wenn die Aufgabe der Wartung an andere Mitglieder des Experimentes übergeben wird. Die vorliegende Arbeit ist deshalb auch als eine ausführliche Dokumentation des Fitters und der Strategien von Bedeutung.

Literaturverzeichnis

[1] CDF Collaboration, *CDF II Technical Design Report*,
FERMILAB-PUB-96/390-E, Nov 1996, 318pp,
`http://www-cdf.fnal.gov/upgrades/tdr/tdr.html`

[2] Maybeck, Peter S.,
Stochastic Models, Estimation, and Control,
Vol. 1

[3] K. Anikeev, G. Bauer, I. Furic, S. Gromoll, A. Korn, I. Kravchenko, Ch. Paus, A. Rakitin, J. Tseng,
Calibration of Energy Loss and Magnetic Field using J/Ψ Events in Run II,
CDF/DOC/BOTTOM/CDFR/5958

[4] P. Azzi, G. Busetto, P. Gatti, A. Ribon,
Histogram Tracking in the COT,
CDF/DOC/TACKING/CDFR/5562

[5] A. Mukherjee,
CTC and VTX Tracking,
CDF/ANAL/TRACKING/PUBLIC/5490

[6] C. Lecci, S. Menzemer, K. Rinnert, M. Feindt,
Pre-Tracking PVz Finder,
CDF/DOC/TRACKING/PUBLIC/5988

Contents

Introduction

Res a parva origine orta.
The origin of all stuff is small.
(Marcus Tullius Cicero)

An important aim of physics is to find a consistent description of the world which explains the phenomena from very small scales (Planck scale $\approx 10^{-35}$ m) up to very large scales (such as the diameter of our universe $\approx 10^{26}$ m). Moreover the special interest of particle physics is to discover and to understand the fundamental laws and structures of the micro-cosmos. The key for this understanding is the reduction of all observed phenomena to a model that is as simple as possible. The most favored one in particle physics is currently the so-called Standard Model, which consists of a set of elementary particles and three interaction forces. It does not include the fourth known force, gravitation. But this force is negligible in particle physics due to the small masses of the particles involved. Also theoretically, there is no consistent description of gravitation within quantum field theory.

In order to test the Standard Model elementary particles such as protons, antiprotons, electrons and positrons are produced and accelerated to very high energies before beeing forced to collide with each other. The resulting collisions are then observed with huge, technically sophisticated detectors which serve as a kind of microscope to very small scales.
The LEP (Large Electron Positron collider) accelerator was located at CERN close to Geneva, where four experiments observed the collisions of electrons and positrons. There, the validity of the Standard Model has been essentially established and a lot of its parameters have been measured precisely. But after the shutdown of LEP in fall 2000, still some open questions remained. One elementary particle of the Standard Model, the Higgs boson, has not yet been discovered. In addition, there are many parameters such as the fermion masses, the mixing angles and coupling constants which are put into the model by hand. Therefore physicists are searching for a

more fundamental description which can explain the values of those parameters and which can also unify the three fundamental interactions of the Standard Model to one force. Due to the fact that the Standard Model works very well so far, the differences between the Standard Model predictions and those of a more fundamental theory, are expected to be very small or occur only at higher energies, which have not yet been studied. Therefore experiments with higher luminosities and higher center-of-mass energies have been constructed.

One of these experiments is the CDF II (Collider Detector at Fermilab) experiment, which is located at the Tevatron accelerator situated in Batavia/Illinois (USA). The experiment is performed by an international collaboration of about 600 physicists from 53 institutes in 11 countries. It restarted data taking after a five-year shutdown with a major accelerator and detector upgrade, in the fall of 2001. CDF II observes proton-antiproton collisions at a center-of-mass energy of 2 TeV. About 2 fb^{-1} of data are planned to be collected within the first two years. Another 15 fb^{-1} of data will be collected in a second data taking period.
One of the central detector devices of the upgraded experiment is the tracking system, which is located inside a magnetic field of about 1.4 T. It consists of three parts, the time of flight system, which is essential for particle identification, the outer drift chamber and an extended silicon vertex detector. The vertex detector consists of 6-7 double sided and one single sided layer of silicon strip detectors. They are sensitive to charged particles which represent about two thirds of all particles produced in the collisions. The resolution of a measurement in the CDF II silicon vertex detector is about 15 μm. Thus the combination of all measurements in the different silicon layers of the vertex detector precisely determine the helix parameters of tracks of particles in the detector. The silicon vertex detector is the detector device closest to the beam. Its innermost layer is directly mounted on the beampipe. Therefore the extrapolation distances to the decay vertices of the particles are very short and the resulting resolution of the reconstructed vertices is quite high. The precise reconstruction of primary and secondary vertices is an essential ingredient to many physics analyses e.g. for lifetime and oscillation measurements in b-physics as well as for identifying top quarks and low mass Higgs bosons, which predominantly decay into jets including long lived B mesons.

An r/ϕ view of a typical event as it is expected in the CDF II detector is shown in figure 10. Besides the physically interesting event, on average three underlying minimum bias events are expected per bunch crossing. Linking all hits of a particle to a track is a huge combinatorial issue. A well performing fitter is essential to distinguish between real tracks and arbitrary hit combinations, which accidentally form a helix. Only if those combinatorics are carefully resolved in the reconstruction software can the high precision vertex detector be fully exploited for physics analyses.

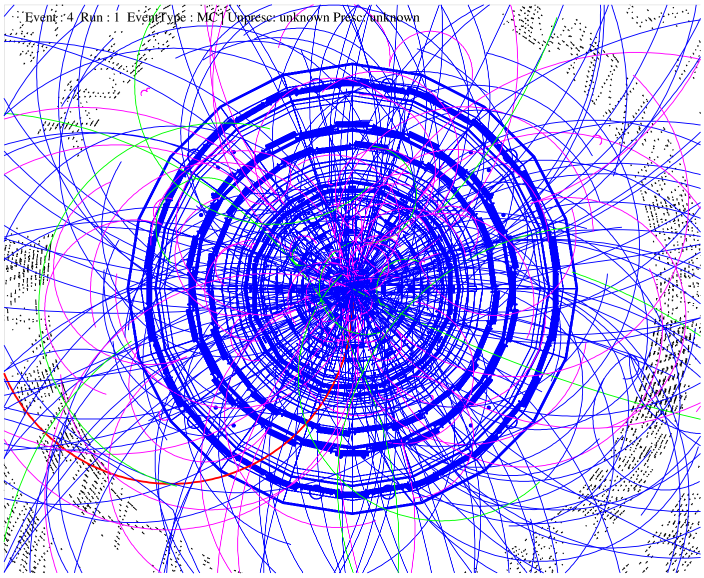


Figure 10: R/ϕ view of a simulated $b\bar{b}$ event with three underlying minimum bias events in the silicon vertex detector.

In addition there is a quite strict limit for the CPU time spent in the reconstruction to fulfill trigger conditions. At CDF II it is the first time in an hadronic environment that full event reconstruction is performed at trigger level. This e.g. makes it possible to efficiently select events with displaced vertices and thus to enrich the set of physically interesting events with long living particles such as B mesons.

The topic of this thesis is the track reconstruction (or "tracking") and fitting in the CDF II silicon vertex detector. TrackingKal, a reconstruction software package, which has been developed in Karlsruhe, is presented. The central part of this package is a Kalman fitter, which was an import from DELPHI, one of the LEP experiments. The fitter algorithm has been completely reorganized and optimized for the CDF II environment and implemented in C++. The fitter is the foundation of the tracking strategies of the TrackingKal package and also the official CDF II track fitter. It is used for refitting the tracks of all tracking strategies and for unpacking them from the PADs (compressed CDF II data format).

Two highly efficient tracking strategies have been developed inside the TrackingKal package. One is a so-called Outside-In tracking, which follows the tracks already found in the outer drift chamber and adds additional hits in the silicon to them. The second one is a Silicon Standalone strategy, which reconstructs tracks in the silicon without any additional information from other detector elements. In combination both tracking algorithms cover the whole region of geometrical acceptance of the silicon vertex detector.

The first chapter of this thesis gives an overview of the CDF II experiment and its physics aims e.g. the first time-resolved observation of $B_s^0\bar{B}_s^0$ oscillations, the discovery of the Higgs boson and precision top quark measurements. Then the silicon vertex detector is described in Chapter 2 in detail. All information which has to be taken into account during tracking and fitting are provided.
The concept of Kalman fitting is introduced in Chapter 3 and explained using a one dimensional example. This is then applied to the five dimensional case of track fitting in uniform magnetic fields. The performance of the Kalman fitter on Monte Carlo and on data is studied.
The concept of both strategies of the TrackingKal package is documented in Chapter 4. Besides the basic ideas of the algorithms also some technical insights are presented.
The strategies of the TrackingKal package are extensively validated on Monte Carlo and on data. The results are presented in Chapter 5.
An overview of some first promising physics results of the CDF II data is provided in Chapter 6. The results present the already advanced status of understanding of the Silicon Vertex Detector and the tracking algorithms on data.
Finally Chapter 7 summarizes the presented work.

The TrackingKal package is the central part of the CDF II silicon reconstruction software, thus it is essential to guarantee its maintenance, particularly concerning the planned upgrade of the silicon vertex detector in 2004. For future reference, necessary detailed documentation of the formulae and calculations used in the fitter and the tracking strategies is given in the Appendices.

Chapter 1

The CDF II Experiment

The CDF experiment is located at the Tevatron proton and antiproton collider in Batavia/Illinois. First events, running the collider at a center-of-mass energy of $\sqrt{\mathrm{s}} = 1.8$ TeV, have been detected in 1985 followed by 90 pb^{-1} of data samples in eleven years of data taking.
Some of the highlights of the analyses of this data are the first experimental evidence of the top quark provided by CDF [1] and a high accuracy measurement of its mass $\mathrm{m_t} = 176.1 \pm 6.6\ \mathrm{GeV/c^2}$ [2], precision electroweak measurements as for example the mass of the W boson $\mathrm{m_W} = 80.433 \pm 0.079\ \mathrm{GeV/c^2}$ [3], and the determination of the average lifetime for several B mesons [4].
Since the shutdown in 1996, the Tevatron and its detectors CDF and D0 have undergone major upgrades in order to be prepared for RUN II, which started end of 2001. The physical goals for RUN II such as $B_s^0 \bar{B}_s^0$ oscillations, Higgs discovery and precise analyses of rare physical processes, whose cross sections are several orders of magnitudes smaller than the inelastic $p\bar{p}$ cross section, require large and clean data samples. The major changes in order to obtain those goals are:

- The increase of the center-of-mass energy to $\sqrt{\mathrm{s}} = 2$ TeV. This does not affect the performance of the detector much but provides a major increase in the cross sections for several interesting processes, e.g. for $t\bar{t}$ production of about 40 %.

- The increase in integrated luminosity to about 2 fb^{-1} within the first two years (RUN IIA). This is mainly achieved by increasing the number of bunches per beam by a factor of six. The higher crossing frequency of the two beams is a huge challenge for the accelerator complex and the different detector components. Several devices such as the Main Ring - the former Tevatron preaccelerator - and the drift chamber of the CDF detector had to be replaced in order to achieve this. For RUN IIB the number of bunches per beam will be again increased by a factor of six, which results in 15 fb^{-1} of integrated luminosity.

- Increase in the detector acceptance and resolution mainly by implementing a larger central tracking system, which will be described later, and by closing caps in the RUN I muon system.

In the following chapter the Tevatron accelerator complex will be described briefly and also the different detector components of the CDF II detector.
Then an overview of the RUN II physics program is given.
A detailed description of the different detector devices and of the RUN II physics program is summarized in the CDF II Technical Design Report [5].

1.1 Accelerator Complex

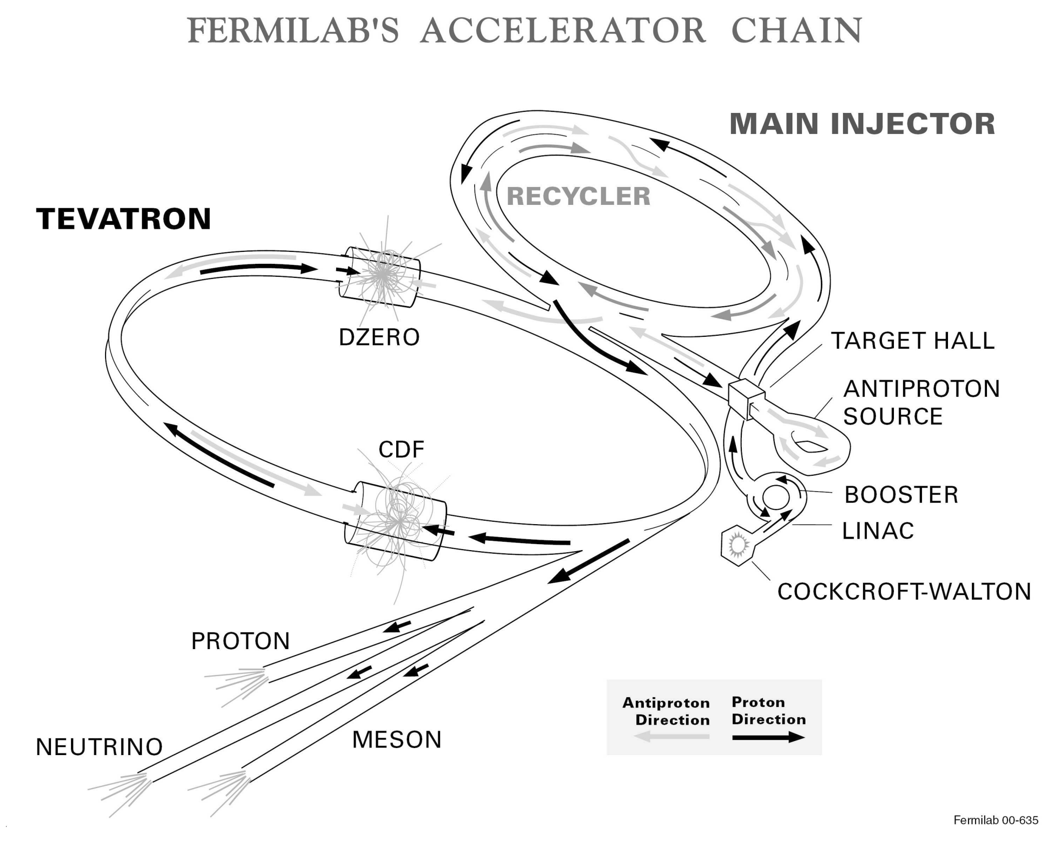


Figure 1.1: Fermilab's accelerator complex for RUN II.

The Fermilab accelerator complex is presented in figure 1.1. The first stage of acceleration is achieved at the **Cockcroft-Walton** preaccelerator. Negatively charged hydrogen ions are produced and introduced into a 150 m long linear accelerator (**Linac**) which they leave with an approximative energy of 750 keV. After being stripped of electrons, the protons enter the **Booster**, a synchrotron with a diameter of about 150 m. Here they are speeded up to about 8 GeV. The last acceleration stage before the protons enter the Tevatron is taking place in the **Main Injector**.
The **Main Injector** is also used for the production of antiprotons. 120 GeV protons are focused on a nickel target. Out of the wide range of collision products the antiprotons are collected, focused and then stored in the **Accumulator Ring**. Once a sufficient number of antiprotons has been produced, they are sent back to the **Main Injector** and are accelerated there.
The antiproton production is one of the limiting factors of the Tevatron efficiency. After the end of a store, 75 % of the antiprotons are expected to remain and are decelerated by the **Main Injector** back to the energy of 8 GeV and then stored for the next run in the so-called **Recycler Ring**.
The final acceleration is achieved by the Tevatron, a collider with a circumference of about six kilometers. There, the protons and antiprotons get their final energy of 1 TeV each, which leads to the center-of-mass energy for the two beams of 2 TeV.

1.2 CDF II Detector

As shown in figure 1.2, the tracking system of CDF II is placed inside a superconducting solenoid, which provides a uniform magnetic field of up to 1.4 T along the detector axis. The calorimeters and the muon system are outside the magnet.
We use a coordinate system where the polar angle θ is measured from the positive z direction (proton direction, which points east at the location of the CDF detector), the azimuthal angle Φ is measured from the Tevatron plane, and the pseudorapidity is defined as $\eta = -\ln(\tan(\frac{\theta}{2}))$.

1.2.1 Tracking System

The tracking system consists of three parts. The **Silicon Vertex Detector** is the detector device closest to the beam and covers a radial range of 1.6 to 29 cm with an η acceptance up to 2. It consists of 6-7 double and one single sided silicon layers and is responsible for the position and direction resolution of the tracking system. Different from RUN I, where the silicon provides only additional information for the tracks found

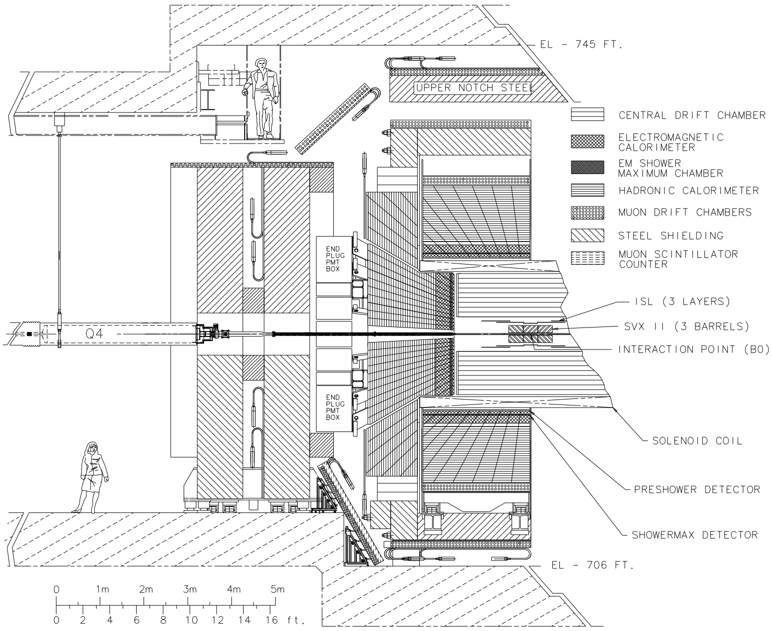


Figure 1.2: Elevation view of one half of the CDF II detector.

in the drift chamber, the extended **Silicon Vertex Detector** of RUN II has the challenging chance of standalone track reconstruction. The larger geometrical acceptance of the Silicon Vertex detector and its better resolution compared to RUN I increase the event yield and improve the background rejection and the resolution of a lot of analyses, which are mentioned in section 1.4.

A drift chamber called **Central Outer Tracker** (COT) covers the region from 40 cm to 138 cm and an η range up to 1. Although it has a much poorer position and direction resolution as the **Vertex Detector**, it provides a much better momentum resolution due to the larger radial extention and a higher purity due to lower track density than in the silicon. A detailed description of the inner tracking system and its performance is provided in chapter 2.

Additionally there is the **Time Of Flight System** (TOF), which is later fitted in the space between the COT and the solenoid (fig.: 1.3). The TOF system provides information for particle identification which in RUN I was only retrieved by dE/dx measurements in the COT. The particle identification power of the TOF system is presented in figure 1.4. This is a powerful tool for background rejection and as such of vital importance for the B physics analyses.

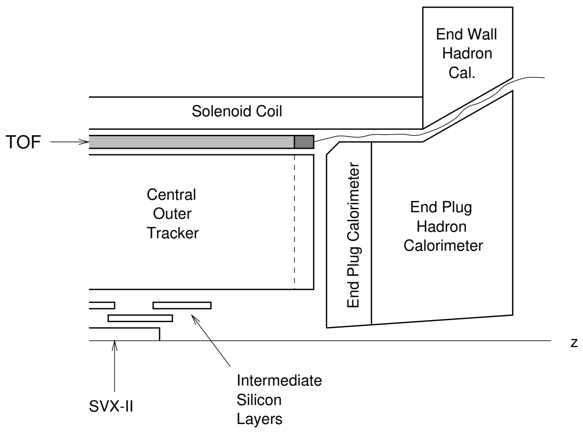


Figure 1.3: Location of the TOF system.

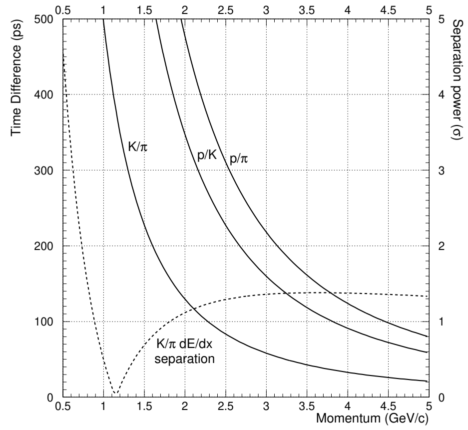


Figure 1.4: Time of flight differences as a function of particle type and momentum. The dashed line indicates the COT dE/dx separation power for K/π.

1.2.2 Calorimeters

The solenoid and the tracking volume are surrounded by the calorimeters, designed to measure the energy of photons, electrons and jets and thus to determine the missing transverse energy associated to neutrinos. There are altogether five calorimeter systems: central electromagnetic calorimeters, central hadron calorimeters, end-wall hadron calorimeters, end-plug electromagnetic and hadron calorimeters, covering the whole azimuth range and the pseudorapidity up to $|\eta| = 3.64$.

1.2.3 Muon Chambers

The outermost component of the CDF detector is a set of scintillators, drift tubes and steel absorbers, used for the detection of muons above ≈ 1.5 GeV. The muon systems are not able to take data within the RUN II inter bunch interval of 400 or 132 ns. But the low occupancy of the muon chambers allows integration over several events.
During RUN I, detection of muons have proven to be an important requirement, both for the analyses of several physics channels and for calibration. In RUN II the muon chambers will also contribute to several trigger decisions, e.g. in the di-muon trigger to select $J/\Psi \rightarrow \mu^+\mu^-$ events.

1.3 Data Acquisition and Trigger

The trigger plays an important role to efficiently extract the most interesting physics events from the large number of minimum bias and background events. A huge background rejection already at trigger level is essential to retrieve the high statistics needed especially for precision measurements in B physics.
The CDF II trigger is a three level system (fig.: 1.5), with each level providing a sufficient rate reduction for the processing of the next level. The design processing rates are L1/L2/L3 $\leq$ 50000/300/50 Hz. The typical event size is about 250-300 kB.

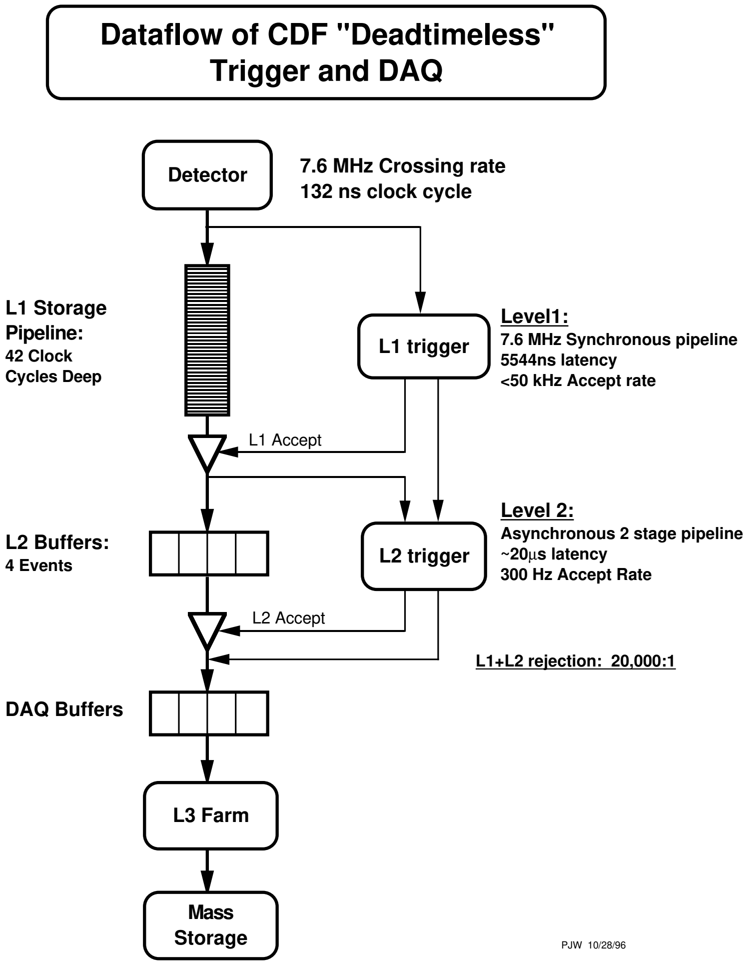


Figure 1.5: Block diagram of the CDF II trigger.

The first two triggers are hardware triggers, the last one is a software trigger running on a Linux PC farm.
The L1 triggers base their decisions on information of the calorimeters, the muon system, the forward detectors and the drift chamber (fig.: 1.6). The XFT (eXtremely Fast Tracker) reconstructs r/ϕ tracks in the COT with a transverse momentum resolution of $\Delta p_T/p_T^2 = 0.01651$ GeV^{-1} and an angular resolution of 5.1 mrad. About 40 L1 trigger conditions are typically active.
On the L2 the SVT (Silicon Vertex Tracker) trigger adds silicon r/ϕ hits to the L1 XFT tracks and selects events with two tracks with an impact parameter larger than 120 μm in order to identify secondary vertices. There are also calorimeter and muon based triggers at L2. Altogether there are about 80 L2 triggers.
On the L3 trigger farm, full event reconstruction takes place. Therefore the offline reconstruction software with slightly different tuning is used. This sets a strict limit of a about 1 CPUs for the average time available for event reconstruction. About 140 L3 triggers are installed.
The out coming events of L3 are sorted into 20 streams, and then after processing the data, they are split up into more specific calibration and physics datasets.
This complex trigger system is unique in a hadronic environment.

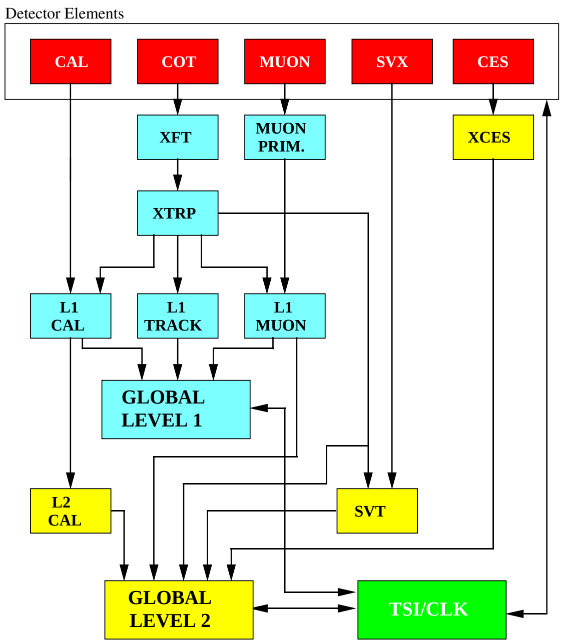


Figure 1.6: CDF II trigger system.

1.4 Physics Plans for Run II

The RUN II physics program is an extention of the RUN I program, taking advantage of the upgraded detector and accelerator complex. The combination of experience from RUN I analyses, the increase of luminosity by about a factor of 20, the higher center-of-mass energy ($\sqrt{s} = 1.8$ TeV $\rightarrow \sqrt{s} = 2$ TeV) and the wider geometrical acceptance of the different detector components give range to a hudge number of exciting analyses.
The main physics topics of RUN II are:

- top physics,
- electroweak physics,
- B physics,
- tests of perturbative QCD (precision measurement of α_s at large Q^2 and the determination of the parton distribution functions)
- and the search for new phenomena beyond the Standard Model

1.4.1 Top Physics

Until the start of LHC, the Tevatron is the only collider able to produce top quarks. Thus one of the major goals of CDF II is to determine the properties of the top quark. Due to the increase in the center of mass energy the cross section of the $t\bar{t}$ production increased by about 40 %.
The dominant decay modes of a $t\bar{t}$ pair are shown in figure 1.7. The t decays in $\geq$ 99 % of the cases via $t \rightarrow Wb$. The W is either decaying to a charged lepton and a neutrino or to two quarks. For the reconstruction of the $t\bar{t}$ event a good b-tagging efficiency to identify the b jets is required. High acceptance in the calorimeter and the muon chambers to reconstruct the W leptonic decays and good jet algorithms for the $q\bar{q}$ decays of the W are also necessary.
The upgraded CDF II detector and the higher luminosity increase the expected event yield for $t\bar{t}$ events (single-tagged lepton + jets) from some 10 events in RUN I to about 1000 events for 2 fb^{-1} in RUN IIA. Therefore the mass of the top quark can be measured with a precision of $\pm$ 4 GeV/c^2 instead of $\pm$ 6.6 GeV/c^2 in RUN I.
The $t\bar{t}$ production cross section will be determined with an accurancy of $\delta\sigma_{t\bar{t}}/\sigma_{t\bar{t}} \leq 7$ % (30 % in RUN I).

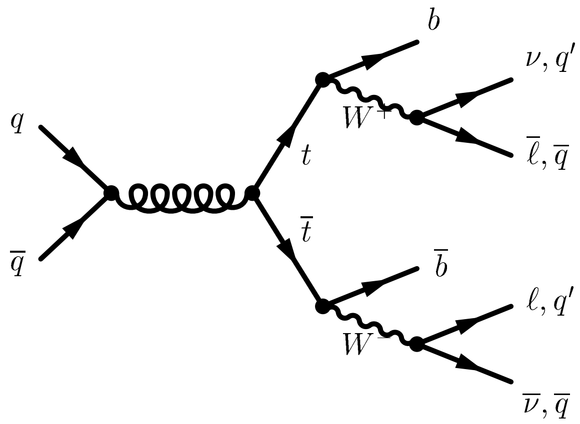


Figure 1.7: $t\bar{t}$ production and decay in di-lepton channel, lepton+jets or all hadronic.

Besides the dominate strong $t\bar{t}$ pair production there is also the electroweak single top production via the t-and s-channel shown in figures 1.8 and 1.9. The cross section of the t-channel is about two times higher as the one of the s-channel but has also larger theoretical uncertainties. Therefore the s-channel is preferred for precise measurements.

The single top production is of special interest due to the contribution from $|V_{tb}|$ to this process already in its production and its decay. Out of about 200 expected single top events the partial width $\Gamma(t \to Wb)$ can be determined to 26 %, and $|V_{tb}|$ can be determined with a precision of 13 %. This will be the first direct measurement of $|V_{tb}|$.

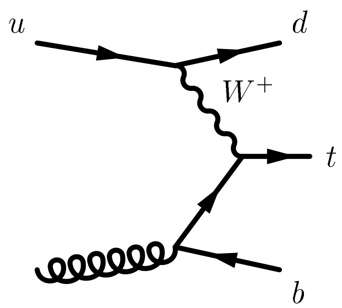


Figure 1.8: Single top production via the t-channel (Wg-fusion).

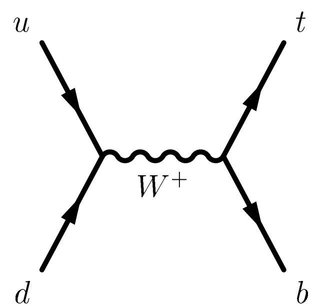


Figure 1.9: Single top production via the s-channel.

1.4.2 Electroweak Physics

The main issues of the electroweak physics program are the precision measurements of the mass, M_W, of the W boson and its width, $\Gamma(W)$.
The resolution of M_W is theoreticaly limited. The expected precision for the mass is $\sigma(M_W) \approx \pm 40$ MeV/c^2, which is comparable to the precision obtained at LEP.
The width of the W can be measured directly from the shape of the transverse mass distribution. Figure 1.10 shows the mass distribution for RUN IA data. There the width was measured to be $\Gamma_W = 2.11\pm$ 0.32 GeV. For RUN IIA the expected resolution is $\sigma(\Gamma(W)) \approx \pm$ 30 MeV. This is about a factor of 6 better than the resolution obtained at LEP 2.

The mass of the W and the Z bosons are linked in first order via the following relation:

$$M_W^2 = M_Z^2(1 - \sin^2(\theta_w)) \tag{1.1}$$

where θ_W is the Weinberg angle. Due to different higher order quantum corrections of the mass of the W and the Z boson, higher order correction terms have to be applied.

$$M_W^2 = M_Z^2(1 - \sin^2(\theta_w))(1 + \Delta\rho^{quarks} + \Delta\rho^{Higgs}) \tag{1.2}$$

where

$$\begin{aligned} \Delta\rho^{quarks} &\propto m_t^2 \\ \Delta\rho^{Higgs} &\propto \ln(\frac{M_{Higgs}}{v}) \end{aligned}$$

(v: electroweak vacuum expectation value)

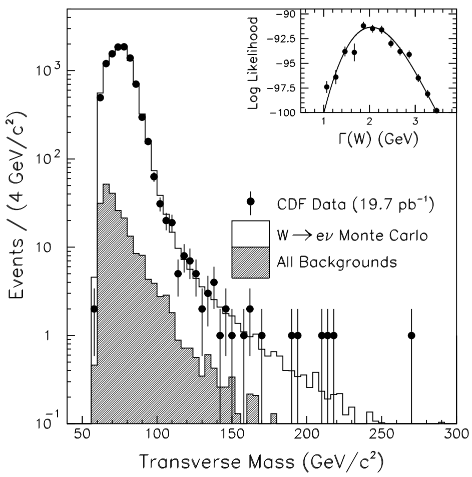


Figure 1.10: Transverse mass distribution (M_t^W) for $W \to e\nu$ candidates along with background and signal expectation. The inset is the fit to $M_T^W \geq 110$ GeV/c^2.

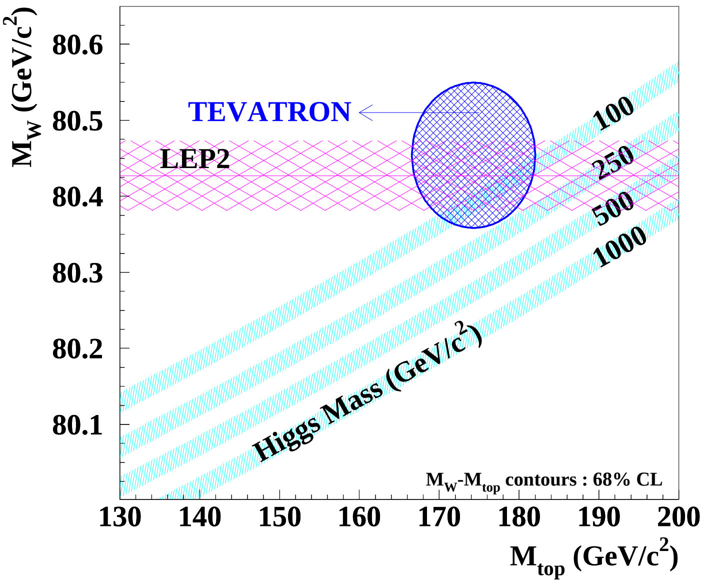


Figure 1.11: Higgs mass constraint from the electroweak fit [6] with M_W and M_t as input (combined data CDF/D0 from RUN I).

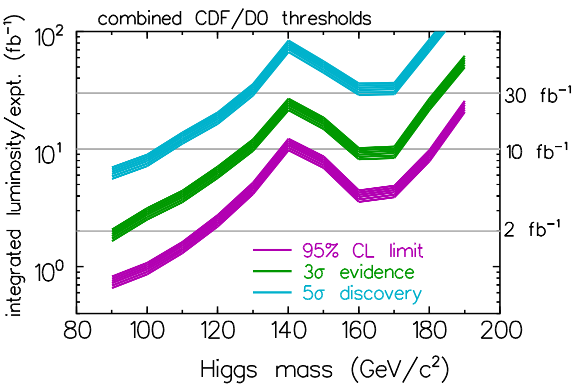


Figure 1.12: Combined CDF/D0 threshold for Higgs limits, evidences and discoveries.

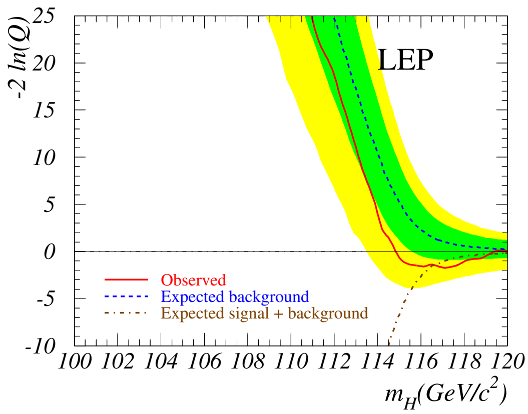


Figure 1.13: Comined data of all four LEP experiments, examined for their consistency with two hypothesis: (a) the Standard Model background, (b) the signal plus backround. The convidence levels of a Likelihood analysis as function of the Higgs boson mass are plotted. The dark/green and light/yellow shaded bands represent the 68 % and 95 % probability bands about the background expectation [7].

The Z mass and $\sin^2(\theta_w)$ have been measured very precicsely at LEP. Combining the W and the top mass, it is therefore possible to restrict the Standard Model Higgs mass m_H with a precision of $\approx 2m_H$ (fig.: 1.11). This is called elektorweak fit. The Tevatron is the only experiment which is able to measure both input parameters M_t and M_W.

Besides the indirect determination of the Higgs mass, there is also a small chance for a direct Higgs measurement at the Tevatron. The most sensitive channels are Higgs production in association with W and Z bosons.
With RUN IIA it will be possible to set a 95 % combined exclusion limit with CDF II and D0 data for a Higgs mass smaller than 120 GeV. With the RUN IIB data the exclusion limit can be pushed up to $m_H \approx 190$ GeV.
If the Higgs is really at the value where LEP has seen a deviation of about 2σ (fig.: 1.13), both experiments combined are able to make a 5σ discovery with RUN IIB. The combined CDF - D0 threshold for Higgs limits derived from RUN I extrapolation and MC studies are shown in figure 1.12.

The electroweak physics program also plans to check the Standard Model predictions for triple gauge boson couplings and to search for anomalous couplings in the $WW\gamma$ and WWZ couplings.

The forward-backward Z asymmetry will be studied to measure the Weinberg angle. Extrapolation of the A_{FW} from RUN I predicts a resolution of $\sigma_{A_{FW}} = 0.003$ and an uncertainty of $\sin^2\theta_W^{eff}$ of about 0.001.
Cross section measurements of Drell-Yan production can be used to get further constraints on the PDFs (patron distribution functions).

1.4.3 B Physics

Measurements with B hadrons can in principle be used to extract information on 5 of the 9 elements of the CKM (Cabibbo-Kobayashi-Maskawa) matrix, that relates the weak-interaction and mass eigenstates of quarks. It can be written as:

$$V_{CKM} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} \quad (1.3)$$

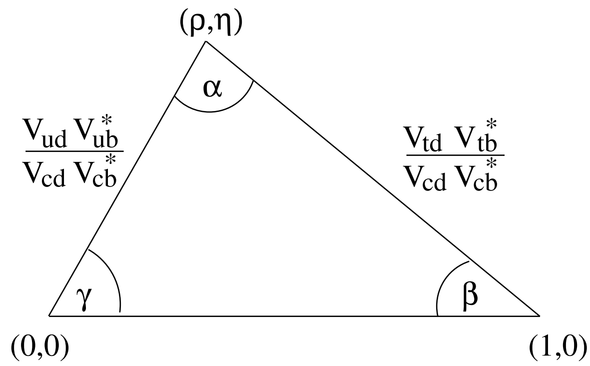


Figure 1.14: The unitarity triangle indicating the relationship between the CKM elements.

or in the Wolfenstein parametrization:

$$V_{CKM} \approx \begin{pmatrix} 1-\lambda^2/2 & \lambda & A\lambda^3(\rho - i\eta) \\ -\lambda & 1-\lambda^2/2 & A\lambda^2 \\ A\lambda^3(1-\rho-i\eta) & -A\lambda^2 & 1 \end{pmatrix} \tag{1.4}$$

given here to $O(\lambda^4)$, where $\lambda = \sin(\theta_{Cabibbo})$ and the other three parameters A, ρ and η (all of order $O(1)$) encode the remaining two weak mixing angles and the irreducible complex phase that introduces CP violation.
The unitarity of the CKM matrix requires the following relationship:

$$V_{tb}^* V_{td} + V_{cb}^* V_{cd} + V_{ub}^* V_{ud} = 0, \tag{1.5}$$

which can be displayed as a triangle in the complex plane, as shown in figure 1.14. The triangle is rescaled by $A\lambda^3$. This is the famous so called unitarity triangle[1]. The measurement of all the three angles α, β and γ and of the length of the sides is a probe of the Standard Model.

$B_s^0 \bar{B}_s^0$ Mixing

B_s^0 and $\bar{B}_s^0$ can mix via the box diagrams represented in figure 1.15. Let the quantum mechanical basis vectors be $\{|1>, |2>\} = \{|B_s^0>, |\bar{B}_s^0>\}$, then the Hamiltonian is:

[1]The other unitary relations can be visualized via triangles as well, but they almost collapse to a line, so that they are hard to overconstrain by the experimental measurements.

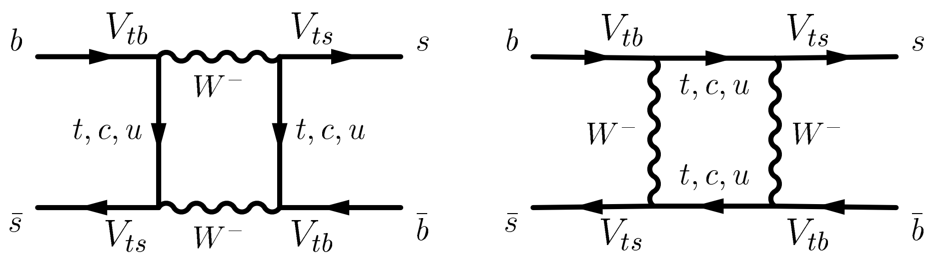


Figure 1.15: The two diagrams for B_s^0 mixing. The contributions of the t quark is dominating over the c and the u quark.

$$H = M - \frac{i}{2}\Gamma = \begin{pmatrix} M & M_{12} \\ M_{12}^* & M \end{pmatrix} - \frac{i}{2}\begin{pmatrix} \Gamma & \Gamma_{12} \\ \Gamma_{12}^* & \Gamma \end{pmatrix} \tag{1.6}$$

Diagonalising we have:

$$\Delta m_s = m_{B_{SH}} - m_{B_{SL}} = 2|M_{12}| \tag{1.7}$$

Here H refers to the heavier and L to the lighter of the two mass eigenstates.
The probability for a B_s^0 to oscillate into a $\bar{B}_s^0$ is given by:

$$P(t)(B_s^0 \to \bar{B}_s^0) = \frac{1}{2}\Gamma_s e^{-\Gamma_s t}[1 + cos(\Delta m_s t)] \tag{1.8}$$

If the final state of a decay is a CP eigenstate, then the decay of B_s^0 to the final state can occur in two ways, either directly or after the B_s^0 has oscillated into a $\bar{B}_s^0$. The amplitudes of these processes interfere, leading to a net asymmetry, which is a function of the decay time and can be expressed as:

$$A_{CP}(t) = \frac{N_f(t) - \bar{N}_f(t)}{N_f(t) + \bar{N}_f(t)} \propto \sin(\Delta m_s t), \tag{1.9}$$

where $N(t)(\bar{N}(t))$ are the number of particles created as a $B_s^0(\bar{B}_s^0)$ at time t=0, that decayed into the final state f at proper time t.

$B_s^0\bar{B}_s^0$ mixing has not yet been observed, but there is a lower limit on the mass difference, which is dominated by the LEP experiments [8], (fig.: 1.16).

$$\Delta m_s \geq 14.4 \text{ ps}^{-1} \text{ at } 95\ \%\ \text{CL}$$

The BELLE and BABAR experiments do not have sufficient center-of-mass energy to produce B_s^0 and $\bar{B}_s^0$, and in addition their time resolution, which is also related to the

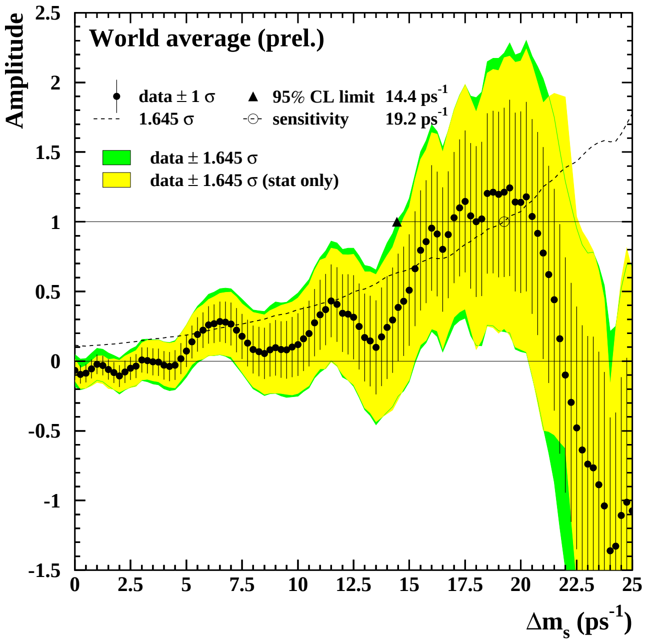


Figure 1.16: Combined B_s^0 oscillation amplitude versus Δm_s (ICHEP 2002).

center-of-mass energy, is too bad to resolve the $B_s^0\bar{B}_s^0$ oscillation frequency.
Until the start of the LHC the Tevatron is the only collider which is able to produce B_s^0 and $\bar{B}_s^0$ and thus has the chance to measure the $B_s^0\bar{B}_s^0$ oscillation frequency.
The most preferred decay modes for $B_s^0\bar{B}_s^0$ mixing at CDF in RUN II are fully exclusive final states such as $B_s^0 \rightarrow D_s\pi$ ($D_s \rightarrow \phi\pi, \phi \rightarrow KK$) [9]. These decay modes have been examined in RUN I already and achieved a lower limit of 6.3 ps^{-1} [10]. The $B_s^0\bar{B}_s^0$ mixing is possible to be observed up to $x_s = \frac{\Delta m_s}{\Gamma_s} = 60$ at 5σ significance at CDF in RUN IIA [11].
Together with x_d, the time dependent mixing fraction from the $B_d^0\bar{B}_d^0$ mixing, x_s can be used to determine $|V_{td}/V_{ts}|$. The determination of this ratio is theoretically cleaner to calculate than both matrix elements separately.

CP Asymmetry in $B_d^0 \rightarrow J/\Psi K_s$

$B_d^0\bar{B}_d^0$ flavor oscillations occur via the similar box diagrams as for the $B_s^0\bar{B}_s^0$ mixing, just replacing the s quark by a d quark. In case of the decay of B_d^0 to the final CP eigenstate $J/\Psi K_s$, the asymmetry is directly related to the angle β of the CKM

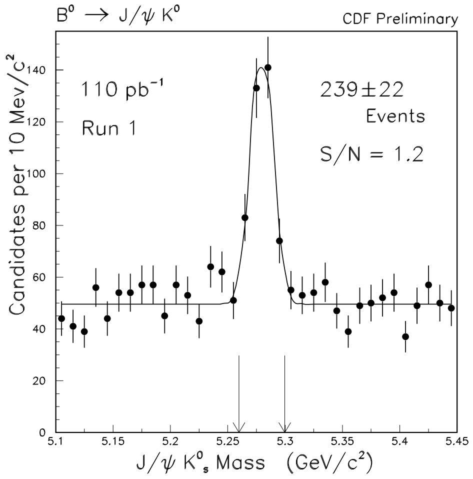


Figure 1.17: CDF's $B_d^0 \rightarrow J/\Psi K_s$ signal from RUN I.

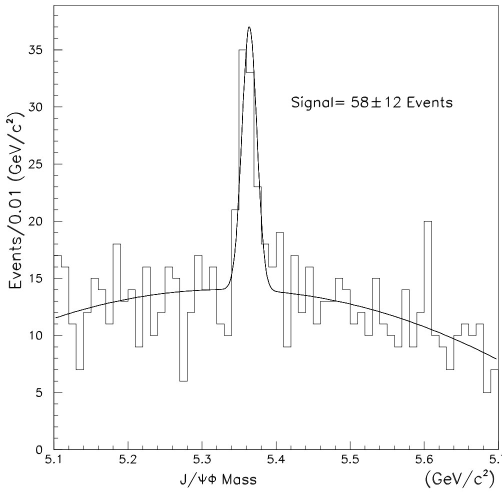


Figure 1.18: CDF's $B_s^0 \rightarrow J/\Psi\Phi$ signal from RUN I.

matrix:

$$A_{CP}(t) = \sin(2\beta)\sin(\Delta m_d t) \tag{1.10}$$

About 240 $B_d^0 \rightarrow J/\Psi K_s$ events have been reconstructed in RUN I (fig.: 1.17). About 15,000 events are expected for RUN IIA which results in a resolution of $\delta\sin(2\beta) \leq 0.07$ [11].

Recently (ICHEP 2002) BABAR [12] and BELLE [13] presented their measurements of $\sin(2\beta)$:

$$\begin{aligned}
\sin(2\beta) &= 0.741 \pm 0.067\ (\text{stat}) \pm 0.034\ (\text{sys})\ (\text{BABAR}) \\
\sin(2\beta) &= 0.719 \pm 0.074\ (\text{stat}) \pm 0.035\ (\text{sys})\ (\text{BELLE})
\end{aligned}$$

Due to the delay of the data taking, CDF will probably not be able to measure $\sin(2\beta)$ competitively.

CP Asymmetry in $B_d^0 \rightarrow \pi^+\pi^-$

The angle $\sin(2\alpha)$ can be measured via the asymmetry in $B_d^0 \rightarrow \pi^+\pi^-$ decays. The most challenging issue is to extract the signal from the physics background from $B_d^0 \rightarrow K\pi$, $B_s^0 \rightarrow K\pi$ and $B_s^0 \rightarrow KK$ decays. For this the particle identification in the time of flight system and via dE/dx in the COT and the vertex detector is essential. With a conservative estimate of $S/N = 1/4$ the expected overall uncertainty on $\sin(2\alpha)$ is of about 0.10.

CP Asymmetry in $B_s^0 \to J/\Psi\Phi$

While the CP asymmetry in $B_d^0 \to J/\Psi K_s$ measures the weak phase of the CKM matrix element V_{tb}, the CP asymmetry in $B_s^0 \to J/\Psi\Phi$ measures the weak phase of the CKM matrix element V_{ts}. This asymmetry is expected to be very small in the Standard Model but can be used in the context of testing it. Like the $B_s^0\bar{B}_s^0$ mixing, this analysis is currently unique to the Tevatron.
About 58 events have been reconstructed in RUN I (fig.: 1.18), 9,000 are expected from RUN IIA.
This decay also contributes to the measurement of $\Delta\Gamma_s/\Gamma_s$. A resolution of 0.05 can be achieved. Additionally to the $B_s^0\bar{B}_s$ mixing mentioned above, $\Delta\Gamma_s/\Gamma_s$ can be used to determine $|V_{td}/V_{ts}|$.

Further plans for the B physics program consider:

- the determination of the angle γ ($B_d^0 \to \pi^+\pi^-$, $B_s^0 \to K^+K^-$ [14]), which would complete the test of the unitarity of the CKM matrix,
- precision measurements of B hadron properties, such as masses and lifetimes,
- studies of heavier B hadrons, such as Λ_b and B_c, which are currently available at the Tevatron only and
- the observation of rare decays such as $B_d^0 \to \mu^+\mu^-K^{*0}$ and $B^\pm \to \mu^+\mu^-K^\pm$.

Chapter 2

Technical Basics for Tracking

The following chapter will provide all necessary technical information the fitter and the tracking strategies are based on. There is no complete detailed description of the silicon detector given, but all quantities and features, which are needed and have to be taken care of in the tracking algorithms and in the fitter, such as multiplexing, zigzag bonding and alignment issues are introduced. An overview of the geometry model used by the fitter is given. Also the class structure of the TrackingKal software package is briefly presented. Then the CDF track perigee parametrization and the internal track parametrization of the TrackingKal package are introduced.
Additionally, the silicon hit resolutions and drift models used in the simulation are presented.

2.1 The CDF II Vertex Detector in Detail

2.1.1 Shallow Angle Stereo and $90°z$ Layers

There are two different types of double sided layers used in the silicon vertex detector, the shallow angle stereo (SAS) and the $90°z$ layers. A schematic view of both types of layers is given in figures 2.1, 2.2.
Both types have strips parallel to the z axis in order to measure the ϕ coordinate of a hit. The $90°z$ layers measure in addition the z position via strips, which are perpendicular to the z axis. The SAS layers measure a stereo coordinate via strips, which are slightly ($\pm 1.2°$) turned versus the ϕ strips.

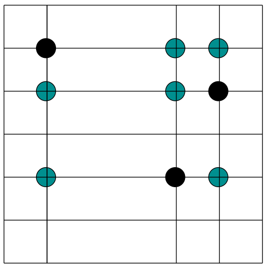


Figure 2.1: $90°z$ layer.

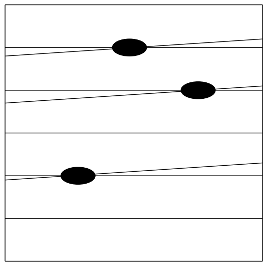


Figure 2.2: SAS layer.

The SAS strips are used to combine the r/ϕ measurement of a track with its z measurement. Figure 2.1 and figure 2.2 show how many possible combinations of pairs of strips on both sides of a layer can be reconstructed by three real hits. The combination of SAS and ϕ hits are unique but for the $90°z$ layers, there are a lot of additional ambiguities.
The resolution of the calculated z position of the ϕ and SAS measurements is about a factor 50 ($\approx \frac{1}{\sin(1.2°)}$) worse than the resolution of the z position measured in the $90°z$ layer due to the flat crossing of the strips on both sides.
During tracking, the SAS hits are picked up in combination with a ϕ hit found on the back side only. Both two dimensional information form a 3D hit, thus the SAS layers are often called 3D layers.

2.1.2 Detector Components

The silicon detector consists of three subdetectors. A r/z-cross view of them is shown in figure 2.3.
One of them is LAYER 00, a recently introduced radiation hard single sided layer of silicon. It is directly put on the beampipe and its ϕ measurements improve significantly the vertex resolution of the tracks.
The second part is the SVX II, which is an extention of the former SVX I of RUN I. It consists of five double sided layers of silicon. Two of them are 3D layers, the other three are $90°z$ layers.
The gap between the SVX II and the COT is filled by the ISL. The ISL consists of one 3D layer in the central region and two 3D layers in the forward region.
A 3D view of the silicon detector can be seen in figure 2.5.

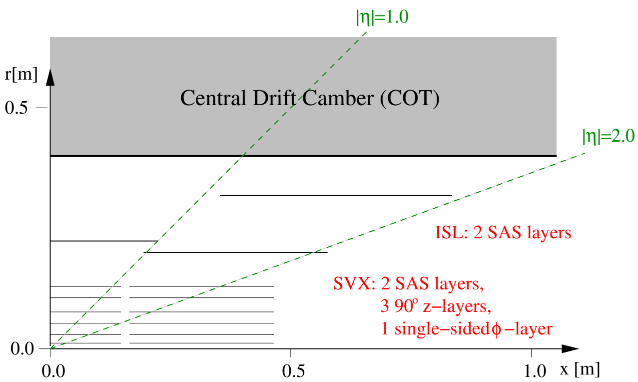


Figure 2.3: R/z-cross view of the inner tracking system.

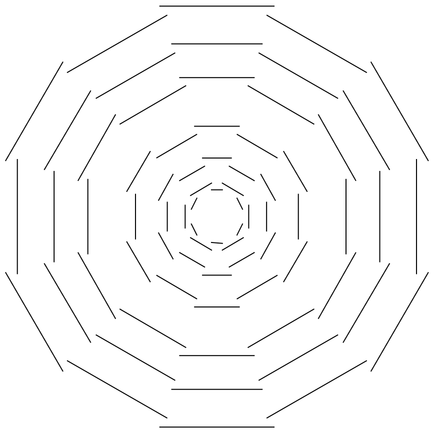


Figure 2.4: Transverse cross view of LAYER 00 and SVX II.

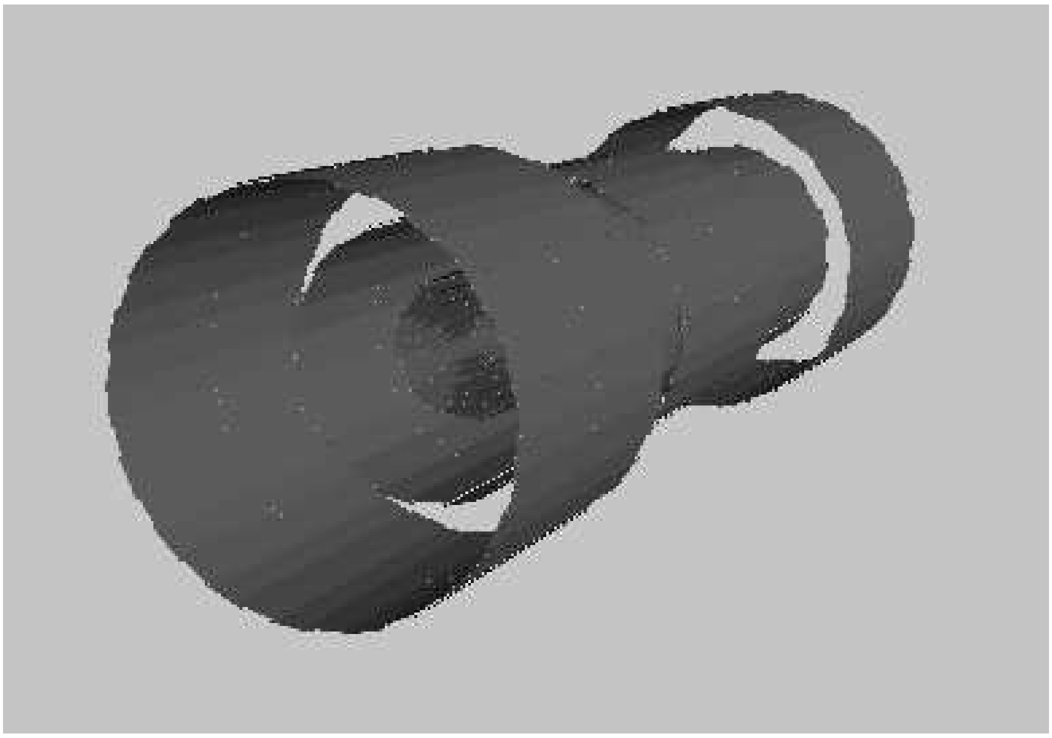


Figure 2.5: 3D view of the silicon tracker.

In table 2.1 the radii of the different silicon layers and their types are listed.
Due to the fact that the innermost single sided silicon layer has been added later to the silicon detector, the canonical numbering of the layers is 00, 0, 1, ... , 6. In the software the numbering is shifted and ranges form 0 to 7. In this thesis the canonical numbering has been used.
The canonical layers are split up in two (four) sublayers labeled with a,b (c,d). This corresponds to the odd and even r/ϕ segments (so-called wedges) of layers which are at slightly different radii, as shown in figure 2.4. The central and the forward region of LAYER 5 are at different radii, too. Thus this layer is split up in four sublayers.
During the tracking and the fitting the sublayers are treated as separate layers. Multiple hits in the overlapping region and calculation of the next intersection of a track with a layer are easier to handle within this concept.

	canonical name	internal name	radius (cm)	readout coordinate	nb. of wedges
	L00	layer 00a	1.35	ϕ	12
		layer 00b	1.62		
SVX II	L0	layer 0a	2.5325	ϕ,z	12
		layer 0b	2.9825		
	L1	layer 1a	4.1075	ϕ,z	12
		layer 1b	4.5575		
	L2	layer 2a	6.5075	ϕ, SAS	12
		layer 2b	6.9075		
	L3	layer 3a	8.2075	ϕ, z	12
		layer 3b	8.7075		
	L4	layer 4a	10.1325	ϕ, SAS	12
		layer 4b	10.6825		
ISL	L5	layer 5a	19.71	ϕ, SAS	24
		layer 5b	20.215		
		layer 5c	22.585	ϕ, SAS	28
		layer 5d	23.09		
	L6	layer 6a	28.585	ϕ, SAS	36
		layer 6b	28.99		

Table 2.1: Overview of the silicon layers.

properties	layer00a	layer00b	layer0	layer1	layer2	layer3	layer4	ISL
nb of ϕ strips	128	256	256	384	640	768	896	512
nb of z strips	-	-	256	384	640	512	896	512
ϕ strip pitch (μm)	50	50	60	62	60	60	65	112
z strip pitch (μm)	-	-	141	125.5	60	141	65	112
total width (mm)	8.43	14.83	17.140	25.594	40.3	47.86	60.17	59.26
total length (mm)	156.9	156.9	148.7	148.7	148.7	148.7	148.7	207.3/ 224.5

Table 2.2: Mechanical dimensions of the silicon detector.

The smallest unit in a layer are the so-called chips or wafers. Table 2.2 lists the technical dimension of the wafers of the different layers such as the size, the readout strip pitch (distance from the center of one strip to the center of the next one) and the number of readout channels on both sides. The pitch size is related to the resolution of the hits. For hits which consist of one-strip clusters the geometric resolution is $\approx \frac{1}{\sqrt{12}} \times$ pitch size. For more strip clusters the resolution is slightly better.
The number of readout channels is of the same order of magnitude for the ϕ and the z side. But the number of strips on the z side ($\frac{\text{length}}{\text{pitch}}$) of the 90°$z$ layers is much larger than the number of strips on the ϕ side ($\frac{\text{width}}{\text{pitch}}$). Therefore several z-strips are read out together in those layers. This is called multiplexing.

For all layers except LAYER 6 the ϕ strips are on the p side of the silicon semiconductor and the z/SAS strips on the n side. This has to be taken into account for cluster center corrections, due to the different drift velocity and drift directions of electrons and holes in the semiconductor.

The wafers are sorted for the SVX and the ISL on three barrels (east, center, west). On every barrel there are two readout units, so-called halfladders. The barrels are separated by a gap of about 1.5 cm in the SVX II, which causes tracking inefficiencies in those regions. The east and the west barrels of the ISL are shifted into a higher $|\eta|$ range as can be seen in figure 2.3 and 2.5.
Each halfladder in the SVX II consists of 2, in the ISL of 3 wafers. LAYER 00 consists of 2 barrels with 4 wafers each. The concept of halfladders does not exist in

LAYER 00.
Separate parameters for every wafer are forseen in the geometry description of the silicon in order to describe slight rotations or shifts between them due to misalignment. All alignment parameters concerning shifts at halfladder level are incorporated by calculating the global hit positions out of the local ones. They are therefore treated on hit level and the tracking does not have to take them into account. But corrections due to rotations or due to misalignment on wafer level have to be incorporated during or after tracking.

2.1.3 Multiplexing

Multiplexing introduces additional ambiguities for the positions of the 90°z hits. On layers with n times multiplexing channel 1, x+1, 2x+1, ..., (n-1)x+1 are read out together (fig.: 2.6). In LAYER 3 the channels are read out via a 2 times multiplexing, in LAYER 1 via 3 times multiplexing and in LAYER 0 via 4 times multiplexing. This means for every 90°z hit there are 2, 3 or 4 possible positions. This ambiguity can be resolved during pattern recognition stage in the tracking algorithms only.

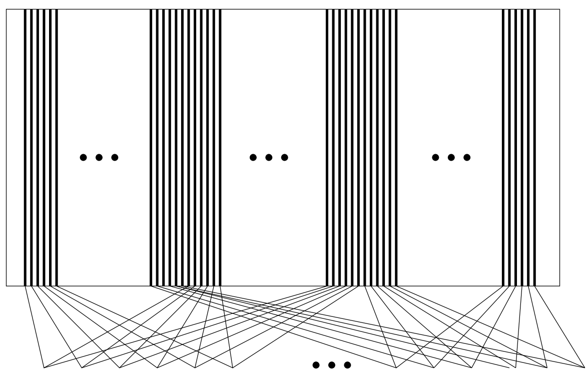


Figure 2.6: Read out via 3 times multiplexing as it is the case in LAYER 1.

2.1.4 Local and Global Coordinates

The clustering algorithm (e.g. described in [15]) calculates the position of the hits locally on the halfladder. In the geometry description on the level of halfladders the information of the center position and the directions of the local coordinate system shown in figure 2.7 is stored.
The global position $\vec{x}$ of a measurement on a halfladder can be determined via the

following formula:

$$\vec{x} = \vec{x}_{short\ direction} * local\phi + \vec{x}_{long\ direction} * localz + \vec{x}_{center\ position} \tag{2.1}$$

In the ideal case of a non-misaligned detector, $\vec{x}_{long\ direction}$ points in the direction of the z axis and $\vec{x}_{short\ direction}$ perpendicular, so its z coordinate is zero. In this situation the ϕ position of a hit can directly be calculated by the knowledge of $local\phi$ and the z position by the knowledge of $localz$:

$$\phi = \arctan\left(\frac{y_{short\ direction} * local\phi + y_{center\ position}}{x_{short\ direction} * local\phi + x_{center\ position}}\right) \tag{2.2}$$

$$z = z_{long\ direction} * localz + z_{center\ position} \tag{2.3}$$

But in case of slight rotations of the halfladder due to misalignment additional information about $localz$ for the calculation of the ϕ position of a hit is needed. And respectively additional information about $local\phi$ is needed to determine the z position. These information are retrieved from the helix parameters of the track, to which the hit should be added.

$$\phi = \arctan\left(\frac{y_{short\ direction} * local\phi + y_{long\ direction} * localz_{track} + y_{center\ pos.}}{x_{short\ direction} * local\phi + x_{long\ direction} * localz_{track} + x_{center\ pos.}}\right) \tag{2.4}$$

$$z = z_{short\ direction} * local\phi_{track} + z_{long\ direction} * localz + z_{center\ position} \tag{2.5}$$

The rotations of the halfladder due to misalignment are very small and so are the corrections on the global coordinates. During the pattern recognition stage the correction for the rotations are not taken into account, but they are incorporated for the final fit. The additional fine corrections for wafer alignment are also applied after tracking only.
The shifts of the halfladder due to misalignment are added to the center position of the halfladder and are therefore already corrected at hit level before the tracking.

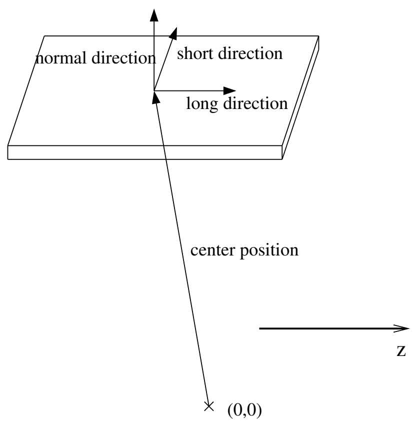


Figure 2.7: Geometrical parameters of a halfladder.

2.1.5 Zigzag-Bonding

In LAYER 6 from one wafer to the next, the SAS strips are not bonded parallel to the SAS strips but to the ϕ strips. The result is a stepwise shape for the SAS strips (fig.: 2.8). Although the calculation of the intersection of SAS and ϕ strips becomes a little bit more complicated by this strip shape, it does not introduce additional ambiguities. During the construction of the 3D hits the ambiguities are already solved. The z resolution of the 3D hits is about 1 mm, thus in an area of $\pm$ 1 mm around the bonding gap the ambiguities are potentially solved wrong. Compared to the whole size of the halfladder this uncertainty corresponds to less than 0.5 % of wrong association.

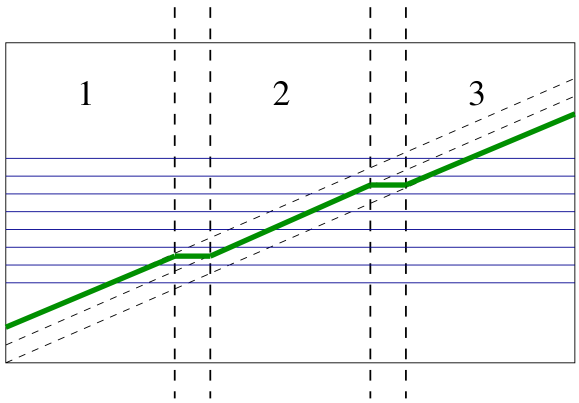


Figure 2.8: Zigzag bonding in LAYER 6.

2.2 Description of the Active and Passive Material

The most time consuming part of tracking and fitting is to look up which layer the track helix is intersecting with next, and then to compute this intersection. All the active material in the silicon, namely the sublayers (tab.: 2.1), can be sorted in a definite way by their radii. Every layer consists of two or four sublayers corresponding to the odd and even ϕ wedges. So the sorting in ϕ of every sublayers is definite as well, there are no overlaps in ϕ of the wedges of a sublayer. The last sorting criterion is z which is also unique. By organization of the active material in a tree, which first branches up in r, then in ϕ and then in z, the look up of the next intersection of a track helix is most efficiently structured.
Additional to the active silicon in the detector, there is a lot of passive material.

The first approach of the TrackingKal package to take the passive material also into account was to add additional material layers to the layer model described above (detailed subset detector). This method is limited to the pieces of passive material, which can be inserted in the list in r in a definite way. This is e.g. not possible for thin slides of the support structure covering several cm in radius in the forward regions. Only material with which the intersections of the helix can be calculated analytically can be incorporated. These are e.g. cylinders and planes parallel to the z axis. Another disadvantage of this approach is, that a lot of additional layers have to be added, which slows down the fitting and the tracking.
The most accurate treatment of the material would be to use the material integrator based on the entire GEANT geometry, which is also used in simulation. But this is just too slow for tracking, and as well for refitting.

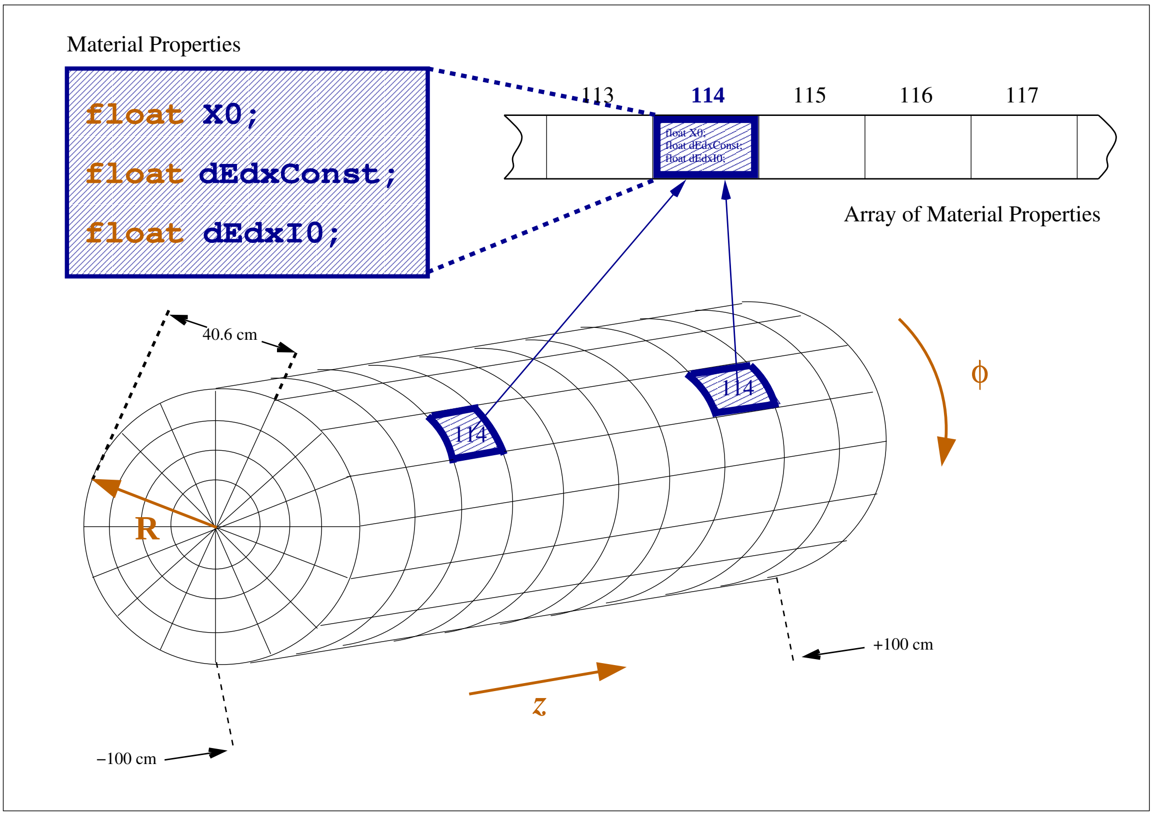


Figure 2.9: The concept of SiliMap is to produce a binned map of the passive material properties using GEANT. Each map bin holds the index of a description of the average material properties at the position of the bin. Due to the high symmetry of the silicon detector several bins have the same average material properties.

The solution provided by the TrackingKal package is the so-called SiliMap. The whole silicon tracking volume is binned in R, ϕ and z. The average material properties such as the dE/dx constant, the minimal ionization energy and the radiation length per bin are calculated using the GEANT based material integrator for a huge amount of tracks and averaging their corresponding quantities in each bin. They are stored in the SiliMap which has to be produced once for every frozen software release and then can be read in for every job using the fitter or the tracking (fig.: 2.9). Due to the large symmetry of the detector, there are a lot of identical bins. Therefore the size of the map can be reduced to 15 MB.
The binning in ϕ and z is equidistant, in r it is adapted to the active layers. There are bins in R for each canonical layer, for the port card (read out electronics) between the SVX II and the ISL and there is a layer for the COT inner wall and the beampipe. The spaces in between those layers are filled by one intermediated layer for each gap. An r/ϕ view of the binning in r is shown in (fig.: 2.10).
The SiliMap material integration can not be used for the Kalman Fitter of the TrackingKal package only but also by the SiTrackFitter, the second standard fitter of the CDF II offline software. The resolution of the fitter using the SiliMap is comparable to the one using the GEANT material integrator but about a factor of 100 faster.
A detailed description of the SiliMap concept can be found in [16].

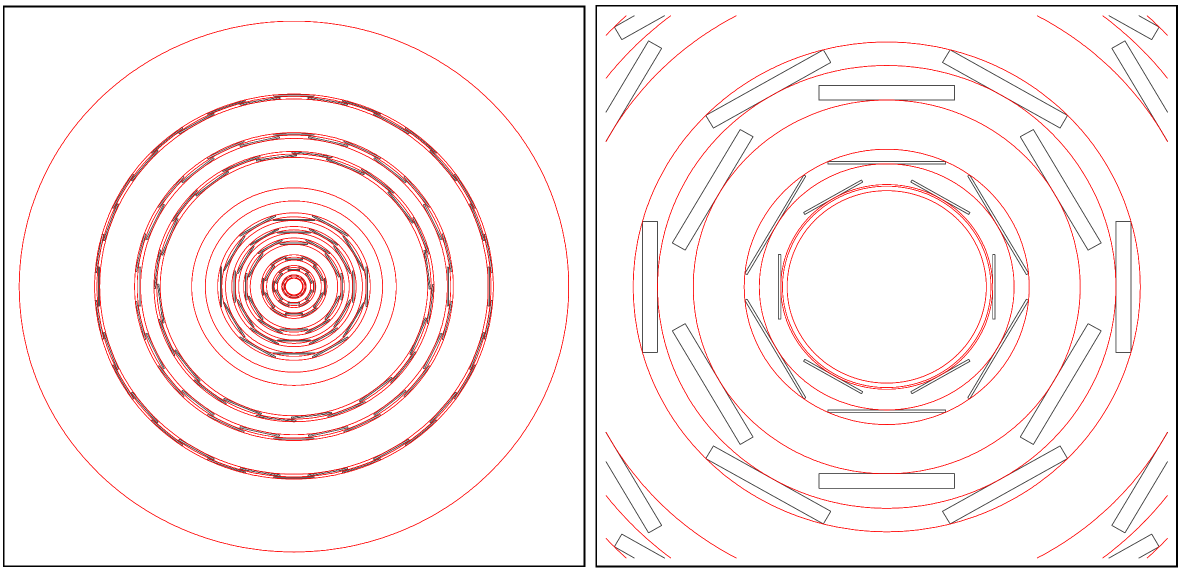


Figure 2.10: R/z-cross view of half ladders and the layer boundaries as used for the SiliMap. The (black) rectangles represent halfladders and the (red) circles correspond to the inner boundaries of the layers. The right hand side shows a zoomed view of LAYER 00 and the first SVX II layer.

2.3 Track Parametrization

In a homogeneous magnetic field, tracks of charged particles travel on a helix with the axis of the helix parallel to the magnetic field. At CDF the direction of the magnetic field is in negative z direction. For describing a helix three parameters are necessary to parametrize the circle in r/ϕ projection and two additional ones in order to parametrize the trajectory in z.
In this section the official CDF track parametrization will be introduced and also the internal track representation of the TrackingKal package. The formulae to transform one into the other are derived in appendix A.

2.3.1 Perigee Parametrization

At CDF [22] the following five parameters are used to describe the helix of a charged particle (fig. 2.11):

$$\vec{\alpha} = (\cot\theta,\ C,\ z_0,\ D,\ \phi_0) \tag{2.6}$$

where:

$\cot\theta$	:	cotangent of the polar angle at minimum approach to the origin,
C	:	half curvature (same sign as the charge of the particle),
z_0	:	z position at point of minimum approach to origin of helix,
D	:	signed impact parameter; distance between helix and origin at minimum approach,
ϕ_0	:	direction in r/ϕ of helix at point of minimum approach.

Figure 2.11 shows the projection of the track helix in to the x-y-plane. The curvature by definition has the same sign as the charge of the particle. If (X0,Y0) is the center of the circle, then the impact parameter is calculated in the following way: $D = Q \cdot (\sqrt{X0^2 + Y0^2} - \rho)$ where $\rho = \frac{1}{|2C|} = \frac{1}{2QC}$ is the radius of the circle and Q the charge of the particle. Figure 2.12 shows the four cases, where the absolute values of the tracks parameters are all the same, only the sign of the curvature and the sign of the impact parameter differ:

1. positively charged particle, impact parameter positive;
2. negatively charged particle, impact parameter positive;
3. positively charged particle, impact parameter negative;
4. negatively charged particle, impact parameter negative;

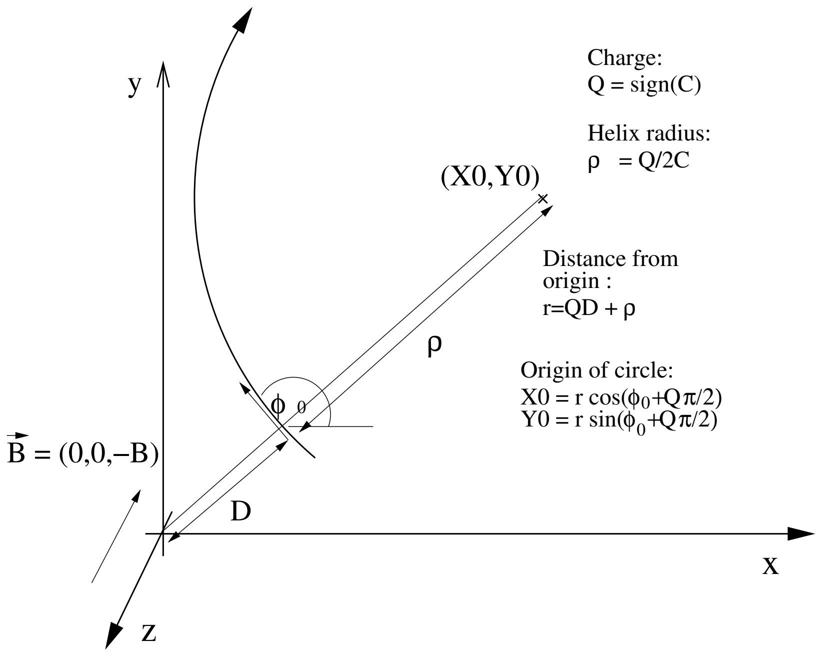


Figure 2.11: R/ϕ perigee parameters of track with negative charge.

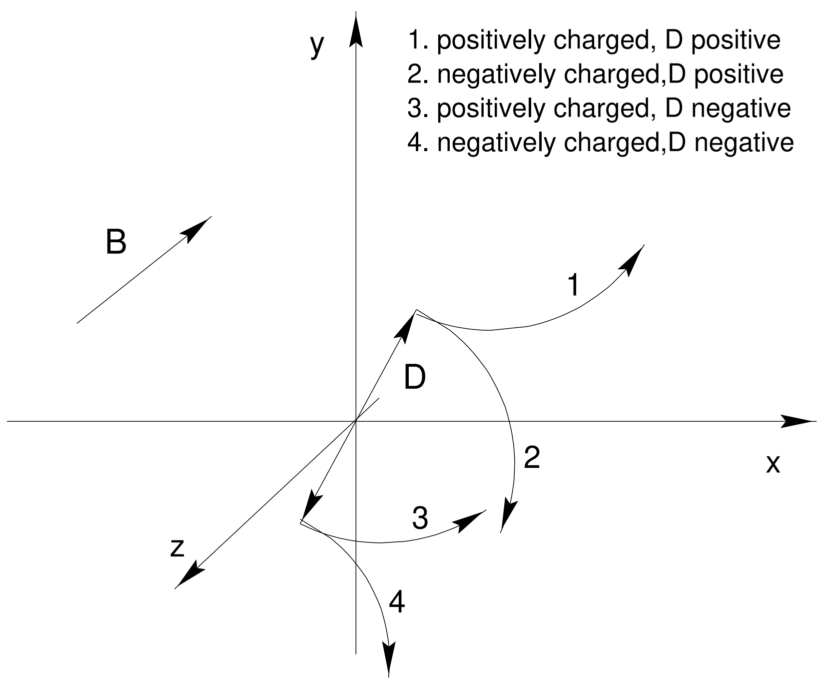


Figure 2.12: Tracks of a particle with positive (1,3)/negative (2,4) charge and positive (1,2)/negative (3,4) impact parameter.

2.3.2 TrackingKal Internal Track Parametrization

The code of the Kalman Fitter has been imported from the DELPHI experiment. At this time no comparable fitter existed in the CDF code. In order to use the full power of this fitter its appropriate parametrization had to be used. Therefore a second track parametrization is introduced inside the TrackingKal package. There a helix at a radius R is defined by the following parameters (fig. 2.13):

$$\vec{\alpha}(R) = (\phi(R),\ z(R),\ \theta(R),\ \beta(R),\ \kappa(R)) \tag{2.7}$$

where

$\phi(R)$	:	position in r/ϕ at R,
$z(R)$	:	z position at R,
$\theta(R)$	:	polar angle at R,
$\beta(R)$	:	angle difference between the position and the direction in r/ϕ at R,
$\kappa(R)$	:	curvature (same sign as the charge of the particle) at R.

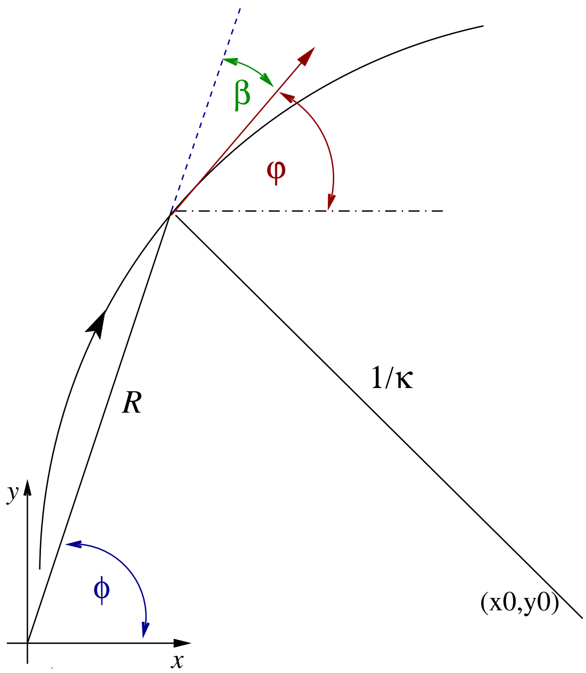


Figure 2.13: r/ϕ internal TrackingKal parametrization.

The perigee parametrization describes the trajectory of a particle as a single helix. This is only true in the ideal case of complete vacuum. Due to material interaction, such as multiple scattering and energy loss the real trajectory is a piecewise helix. In the perigee parametrization the information of the radius at which the set of five parameters correspond to the particle trajectory is missing. Although there might be additional assumptions, like every track has to be parametrized at the radius of its innermost measurement, a lot of confusing is still remaining. A COT track is parametrized at the COT inner wall radius of about 40 cm. But do the errors already include the multiple scattering to the perigee or not? Is the curvature already corrected for the energy loss of the material in the vertex detector?
Every time a hit in the detector is picked up the local or global z and ϕ coordinate of the track at the radius of intersection with the halfladder is needed. If the track parameters correspond to the measured parameters quite some extrapolation and transformation calculation for parameters and covariance matrices are saved which shows up in timing performance.
The R dependent track parametrization is more complex as the perigee parametrization. In principle, a set of parameters for every material the trajectory is passing has to be stored. This will for sure let the memory size of a track explode. But this information has not to be permanently stored as far as it can be reproduced at any time for every track (e.g. during vertexing).
In order to keep an interface for the tracking strategies in the TrackingKal package to the CDF code and the CDF track parametrization, a track converter has been introduced, which transforms parameters and covariance matrices from one parametrization into the other. It turned out to pay off even in timing performance to switch to the internal TrackingKal representation, although this add additional conversion calculations for every track.

2.4 TrackingKal Design and Software Environment

The reconstruction and analysis software of high energy physics experiments consists of millions of lines of code and several hundred authors contribute to it. Therefore it is essential to carefully design this huge software package.
The different concepts to design a program are called programming paradigms. Different programming languages support different paradigms. The one applied by recent high energy physics experiments is the so-called object-oriented approach. There the program consists of different objects which communicate with each other. Every object consists of a collection of information, which defines the status of the object, and

of methods, which allow to change its status. For example a track, which stores its helix parameters and a list of pointers to its silicon hits, can be described as an object. One of its methods can either be to add a hit to it or to calculate its parameters in a different parameter representation.
C++, which is used by CDF II, supports the object-oriented programming approach. But it also supports other paradigms such as procedural or structured programming. An overview of all programming paradigms is given in [17]. The most promising approach to design a software package is to use the Multi-Paradigm-Design [18], which is a combination of all available paradigms. Different issues fit to different paradigms and therefore a software using a combination of all of them is able to describe the problem in the most intuitive and adequate way. This is essential if several hundered authors and users are working with the code. Unecessary programming overhead is avoided, which also shows up in the timing performance.

The TrackingKal package is part of the CDF II offline software, which is e.g. documented in [19, 20]. It is a rewrite and an extension of the former silicon tracking software, which had become too large and too complex and which was not well suited for the implementation of a Kalman fitter. We focused the design of the TrackingKal package on being as simple and intuitive as possible (fig.: 2.14). We therefore chose the Multi-Paradigm-Design. A typical example of object-oriented design in the TrackingKal package is the concept of the different types of fit layers. They all inherite from `KalFitLayer`, which stores e.g. the intersection with the reference helix, which is needed by all types of fit layers. For material layers additional material properties such as minimum ionization energy or radiation length are needed to treat the multiple scattering and the dE/dx. For measurement layers the hit positions and its weight matrix have to be stored in addition. As opposed to the fit layers the `KalKalmanFitter` and the algorithms classes are typical examples for procedural programming inside the TrackingKal package.
The classes of the TrackingKal package can be divided into three different types:

- ➔ object classes
- ➔ container classes
- ➔ function/information server classes
  - ➔ algorithm classes
  - ➔ helper classes

There are the object classes, which describe tracks, hits and vertices. Those objects are stored in track, hit and vertex sets, which are implemented by the data container classes. The different tracking and vertexing algorithms use the track, hit and vertex sets as input and for writing out their results. The algorithms call functions of the Kalman fitter, which is implemented by a function server class and of the geometry

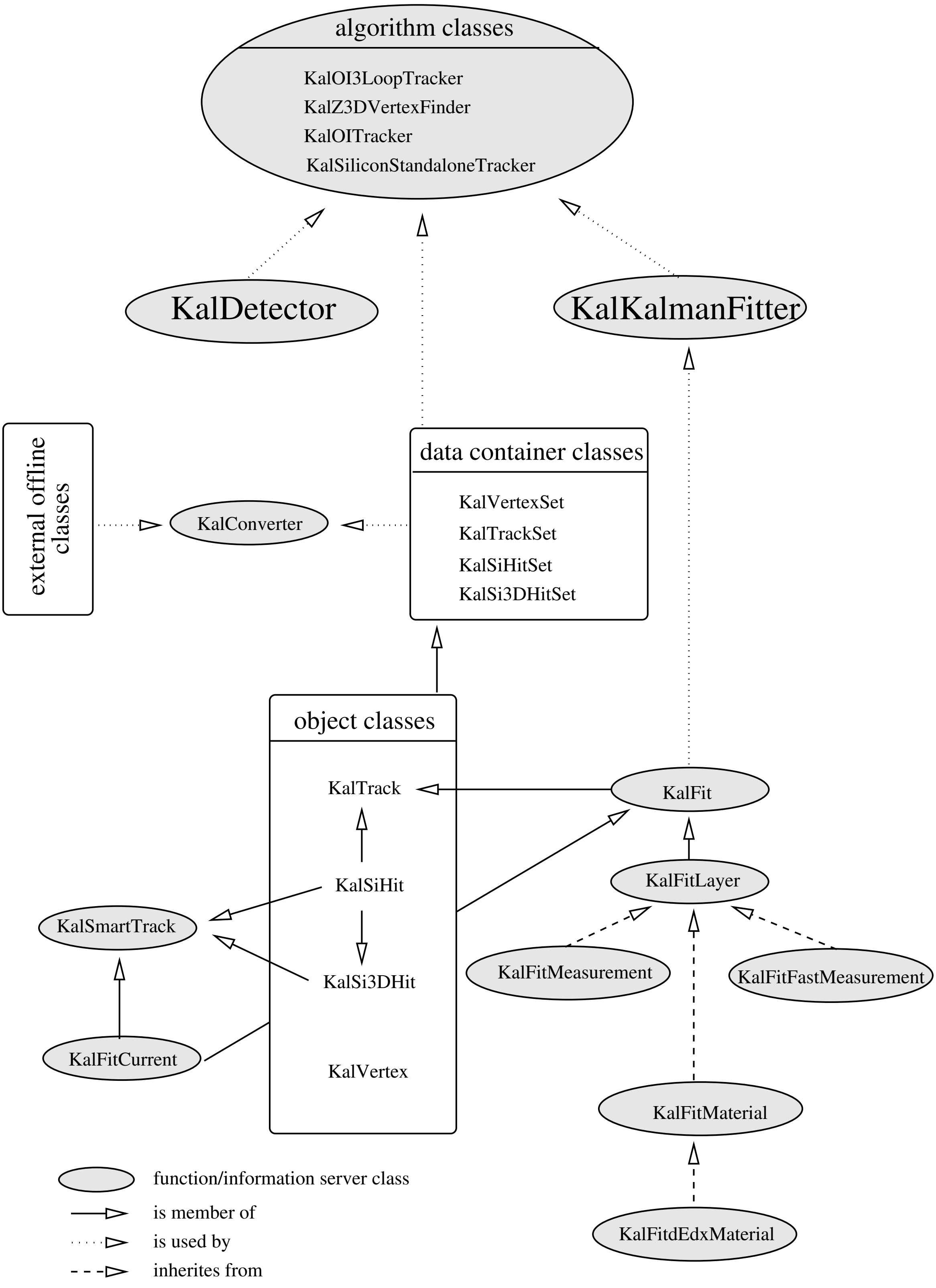


Figure 2.14: Class structure of the TrackingKal package.

description, which is implemented by an information server class. There are helper classes as well, which are mostly used for intermediate storage of information. The converter also is a helper class, which transforms the data stored in the offline classes into the TrackingKal representation and vice versa. This intermediate converter step is necessary in order to obey the existing interfaces and at the same time to avoid the complexity of the former implementation, which mainly arises from the pure object-oriented design.
Due to this converter step in between, it is also possible to compile and to run the TrackingKal package independently from other CDF II software packages, which allows a faster development cycle. The compile and link time for the TrackingKal package in the so-called standalone mode is about a few seconds, the one of the offline framework about a few minutes. More information of the standalone mode is provided in [17]. A detailed class documentation of the TrackingKal package is given in [21].

2.5 Hit Resolution and Hit Smearing

The geometrical resolution of a strip detector is $\frac{\text{pitch}}{\sqrt{12}} \approx \text{pitch} \times 0.288$. Test beam measurements with a prototype of the SVX II detector [23] obtained the following numbers depending on the cluster size: $0.21667 \times$ pitch for one-strip clusters, $0.18333 \times$ pitch for two-strip clusters and $0.38333 \times$ pitch for three-strip and larger clusters. So far this simplified model has been used in the reconstruction. First measurements on data, at least for the ϕ side, seem to be consistent with the resolution measured in the test beam. More detailed measurements depending on the halfladder, the side and the η of the track will be performed and implemented in the reconstruction software.
There are two different charge deposition models for the simulation. One is the geometric model, which is used for the simulations presented in this thesis. This model distributes the charge of a cluster over its strips according to the pathlength of the particle in the strip region. The physical model takes additional effects like Hall-effect or Lorentz-drift into account. It is more realistic than the geometric one, but not yet[1] reliably working.
The errors assigned to the hits in the clustering correspond to extrapolations of the test beam data. But they are not compatible to the hit resolution obtained by the geometric drift model.
The resolutions, depending on the number of strips, the layer and the type of measurement are shown for the simulation with the geometric model in tables 2.3, 2.4, the resolution expected for the data are given in tables 2.5, 2.6.

[1]software release 4.6.2

In order to obtain realistic resolutions and corresponding errors for the simulation the hit positions are smeared after the clustering. In case of better resolution in the simulation than on data, the hit position is smeared via a Gaussian distribution with the size $\sigma = \sqrt{\sigma_{data}^2 - \sigma_{geometric}^2}$. If the resolution expected from the data is better than the one of the simulation the error is blown up: $\sigma_{hit} = \sigma_{geometric}$.
The hit smearing is considered as a provisorium until the physical model is working reliably!

layer	resolution of 1-strip clusters [cm]	resolution of 2-strip clusters [cm]	resolution of ≥3-strip clusters [cm]
L00	0.00096	0.00044	0.00036
L0	0.0013	0.00050	0.00027
L1	0.0012	0.00049	0.00031
L2	0.0013	0.00050	0.00036
L3	0.0013	0.00048	0.00022
L4	0.0013	0.00054	0.00038
L5	0.0032	0.0022	0.0022
L6	0.0032	0.0022	0.0022

Table 2.3: Cluster resolution of the geometric model for ϕ clusters.

layer	resolution of 1-strip clusters [cm]	resolution of 2-strip clusters [cm]	resolution of ≥3-strip clusters [cm]
L0	0.0032	0.0014	0.0016
L1	0.0026	0.0013	0.0014
L2	0.0012	0.0005	0.0008
L3	0.0032	0.0014	0.0016
L4	0.0014	0.0006	0.0003
L5	0.0032	0.0022	0.0022
L6	0.0032	0.0022	0.0022

Table 2.4: Cluster resolution of the geometric model for SAS and z clusters.

layer	resolution of 1-strip clusters [cm]	resolution of 2-strip clusters [cm]	resolution of ≥3-strip clusters [cm]
L00	0.0011	0.0009	0.0019
L0	0.0013	0.0011	0.0023
L1	0.0013	0.0011	0.0024
L2	0.0013	0.0011	0.0023
L3	0.0013	0.0011	0.0023
L4	0.0014	0.0012	0.0025
L5	0.0024	0.0021	0.0043
L6	0.0024	0.0021	0.0043

Table 2.5: Cluster resolution on data for ϕ clusters.

layer	resolution of 1-strip clusters [cm]	resolution of 2-strip clusters [cm]	resolution of ≥3-strip clusters [cm]
L0	0.0031	0.0026	0.0054
L1	0.0027	0.0023	0.0048
L2	0.0013	0.0011	0.0023
L3	0.0031	0.0026	0.0054
L4	0.0014	0.0012	0.0025
L5	0.0024	0.0021	0.0043
L6	0.0024	0.0021	0.0043

Table 2.6: Cluster resolution on data for SAS and z clusters.

Chapter 3

The Kalman Fitter

In the following chapter the Kalman fitter used in the TrackingKal package will be presented.
First an introduction to the principle of Kalman fitting is given. The necessary ingredients and requirements are explained and the concept is illustrated by a one-dimensional example. Later this example is extended to the five-dimensional case of the track parameter space and the precise formulae are deduced.
For additional information about the Kalman fitter, reference [24] is recommended.
The good performance of the fitter implementation is proven by looking at MC and on the J/Ψ data samples.
The Kalman fitter presented here is already used during pattern recognition in the tracking strategies of the TrackingKal package. It is recommended as final fit by the tracking group for all tracks, e.g. propagating COT tracks through the silicon or refitting silicon tracks found by different algorithms. It is also used for the refit of the tracks after unpacking the PADs (compressed data format for storage).
The technical interfaces for accessing the Kalman fitter are described in appendix D.

3.1 The Concept of Kalman Fitting

3.1.1 What is a Kalman Fitter?

A Kalman fitter is, precisely spoken, a Kalman filter based fitter. The Kalman filter was first invented in electronic signal treating. It is a an optimal recursive filter algo-

rithm.
It is optimal in two ways. It incorporates all information and measurements available regardless of their precision to estimate the desired quantities, as in our case the track parameters. It fully exploits all information about the statistical description of the noise, the measurement errors and the uncertainties in the dynamic models (e.g. multiple scattering).
Secondly, the Kalman filter is optimal in obtaining the estimate of the output parameters. Due to the uncertainties of the input no filter can retrieve the precise values of them, but it can retrieve a conditional probability density, conditioned on knowledge of the actual data.
If the Bayesian viewpoint is adopted, it is possible to define the "optimal" estimate of a filter. Possible choices of such a definition would probably include the quality of the mean value, the modes and the median.
A Kalman filter calculates the probability density for problems, in which the system can be described through a linear model and in which system and measurement noises are white and Gaussian (to be explained shortly). Under these conditions, the mean, mode, median, and virtually any reasonable choice for an "optimal" estimate all coincide, so there is in fact a unique "optimal" estimate of the output parameters. The Kalman filter can be proven mathematically to be the best filter of any conceivable form. In other words, if we were to run a number of candidate filters many times for the same application, then the average results of the Kalman filter would be better than the average results of any other.

The word recursive in the previous description of the Kalman filter means that, unlike certain data processing concepts, the Kalman filter does not require all previous data to be kept in storage and to be reprocessed every time a new measurement is taken. This is of vital importance of the practicality of filter implementation (e.g. using the Kalman filter based fitter for tracking).

3.1.2 Basic Assumptions

The three basic assumptions for the Kalman filter are:

(1) linear propagation models (transport models)
(2) white and uncorrelated noise
(3) Gaussian error and noise distributions

These conditions seem to be very restrictive and most physical systems do not fulfill them. But many physical models can be described by linear ones (perturbation models, first order calculations ...). There are Kalman filter concepts to some nonlinear applications but these are considered only if linear models prove inadequate. As far as the Kalman filter is used for track fitting in the CDF environment the linear approximations are valid.
White noise is defined as a noise which is uniformly distributed in the frequencey spectrum. This definition only makes sense if electronic signals have to be treated by the Kalman filter. As far as the filter is used for track fitting the equivalent requirement is uncorrelated noise. This criterion is assumed to be fulfilled in the detector, although there are some small effects like coherent electronic noise which introduces correlations.
Most of the probability density distributions of signals in physics are only approximatively Gaussian. But most of the time only the first and second order statistics (mean and variance or standard deviation) of the measurement and the noise processes are known. Therefore there is no better probability density distriubtion ot assume than the Gaussian one. In practice, the assumption of Gaussian erros proves to be adequate.

3.1.3 A Simple Example

By choosing the example of a boat lost at sea during night the author follows the basic ideas of an example, which was given by Peter S. Maybeck [24] and has been already referenced in quite a lot of lectures about Kalman fitters. The reason for repeating it here again is to prepare the reader to understand the more complicated five-dimensional case described in the following section.

Suppose to be lost at sea during night and have no idea at all of your location. So you take a star sighting to establish your position. For the sake of simplicity, consider a one-dimensional location. At some time t_1 you measure your position z_1. Due to inaccuracies of the measurement device the result is somewhat uncertain. The uncertainty can be described by a Gaussian error distribution with the standard deviation σ_{z_1} (fig.: 3.1). So the mean of the probability distribution of the position is:

$$< x(t_1) >= z_1 \tag{3.1}$$

the corresponding standard deviation is:

$$\sigma_x(t_1) = \sigma_{z_1} \tag{3.2}$$

and the corresponding weight is:

$$\omega_x(t_1) = (\sigma_x^2(t_1))^{-1} \tag{3.3}$$

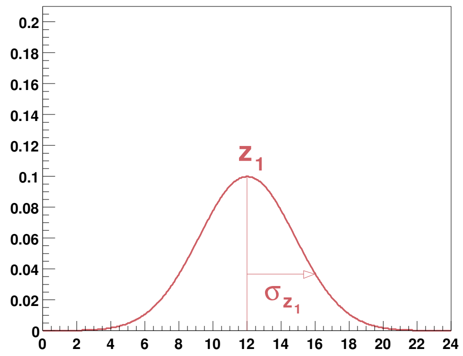


Figure 3.1: Probability density function of the first measurement at time t_1, Gaussian distributed with mean z_1 and standard deviation σ_{z_1}.

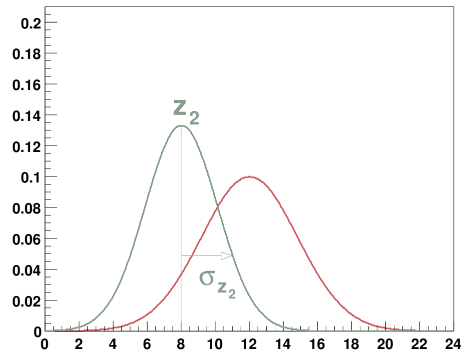


Figure 3.2: Probability density function of the second measurement at time t_2, Gaussian distributed with mean z_2 and standard deviation σ_{z_2}.

Now say a trained navigator friend takes an independent fix right after you do, at time $t_2 \approx t_1$ (so that the true position has not changed at all), and obtains a measurement z_2 with a standard deviation σ_{z_2}. Due to his higher skills the uncertainties on his measurement are smaller than yours (fig.: 3.2). Taking only the second measurement into accout, the probability density of the best estimate would have the mean value z_2 and the according standard deviation σ_{z_2}.
At this point, you have two measurements available for estimating the position. The best estimate which can be deduced by both measurements is the weighted mean of them (fig.: 3.3).

$$< x(t_2) > \; = \; \frac{z_1 * \omega_{z_1} + z_2 * \omega_{z_2}}{\omega_{z_1} + \omega_{z_2}} \tag{3.4}$$

$$\omega_x(t_2) \; = \; \omega_{z_1} + \omega_{z_2} \tag{3.5}$$

$$\sigma_x^2(t_2) \; = \; \omega_x(t_2)^{-1} \tag{3.6}$$

Note that ω_x is larger than either ω_{z_1} or ω_{z_2}, which is to say that the weight of the estimate of the position has been increased and the uncertainty in the estimate has been decreased by combining the two pieces of information. Even if one of both measurements has a very small weight, that means a huge uncertainty, it helps to improve the uncertainty of the combined estimate. The most important thing is that the errors correspond to the measurement and are not over- or underestimated. Otherwise the

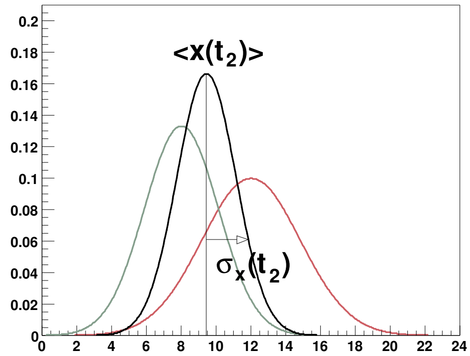


Figure 3.3: The probability distribution of the position after combining the first and the second measurement is again a Gaussian distribution with the mean $< x(t_2) >$ and the standard deviation $\sigma_x(t_2)$.

information can not be used in an adequate way for the determination of the best estimate.

The next step is adding dynamics to the problem. Suppose that you travel for some time before taking another measurement. Further assume that the best model you have of your motion is of the simple form

$$\frac{dx}{dt} = u + w \tag{3.7}$$

where u is a nominal velocity and w is a noise term used to represent the uncertainty in your knowledge of the actual velocity due to disturbances or off-nominal conditions, effects not accounted for in the simple first order equation. The "noise" w will be modeled as a Gaussian noise with a mean of zero and standard deviation σ_w.
As time progresses, the probability density travels along the x axis at the nominal speed u, while simultaneously spreading out about its mean, due to the uncertainties in the nominal speed. At the time t_3^- just before the next measurement is taken (at time t_3), the probability density of the position is as shown in figure 3.4, and can be expressed mathematically as a Gaussian density with mean, standard deviation and weight given by:

$$\begin{aligned} < x(t_3^-) > &= < x(t_2) > +u * [t_3^- - t_2] && (3.8)\\ \sigma_x^2(t_3^-) &= \sigma_x^2(t_2) + \sigma_w^2 * [t_3^- - t_2]) && (3.9)\\ \omega_x(t_3^-) &= \sigma_x^2(t_3^-)^{-1} && (3.10) \end{aligned}$$

Note that in this prediction z_1, z_2 and their corresponding standard deviations σ_{z_1} and σ_{z_2} and weights ω_{z_1} and ω_{z_2} do not enter explicitly but only indirectly by their contributions to $< x(t_2) >$, $\sigma_x(t_2)$ and $\omega_x(t_2)$.
At t_3 the next measurement is taken, and its value turns out to be z_3, its standard deviation is assumed to be σ_{z_3} and its weight ω_{z_3} (fig.: 3.5). As before, there are now two Gaussian densities available that contain information about the position, one encompassing all the information available before the measurement, and the other being the information provided by the third measurement itself. Again the weighted mean of both probability density functions is taken to obtain the optimal estimate of the position at t_3 (fig.: 3.6):

$$\begin{aligned} < x(t_3) > &= \frac{\omega_x(t_3^-) * < x(t_3^-) > + \omega_{z_3} * z_3}{\omega_x(t_3^-) + \omega_{z_3}} && (3.11) \\ \omega_x(t_3) &= \omega_x(t_3^-) + \omega_{z_3} && (3.12) \\ \sigma_x^2(t_3) &= \omega_x(t_3)^{-1} && (3.13) \end{aligned}$$

The basic rule for the Kalman fitter is taking the weighted mean every time different measurements became available for combination. This reduces the uncertainties and increases the weight. So just sum up the weights of both measurements to get the weight of the combination of them.
If there are any additional uncertainties in calculating the estimate, e.g. the uncertainties of the velocity, the uncertainty of the measurement decreases and the weight increases. So just sum up the squared standard deviations to get the resulting error.

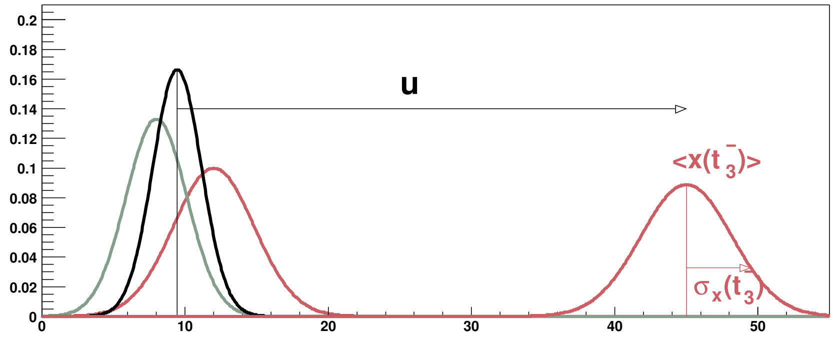


Figure 3.4: Propagation of the probability density of the position, while the boat is moving with a constant Gaussian distributed velocity u.

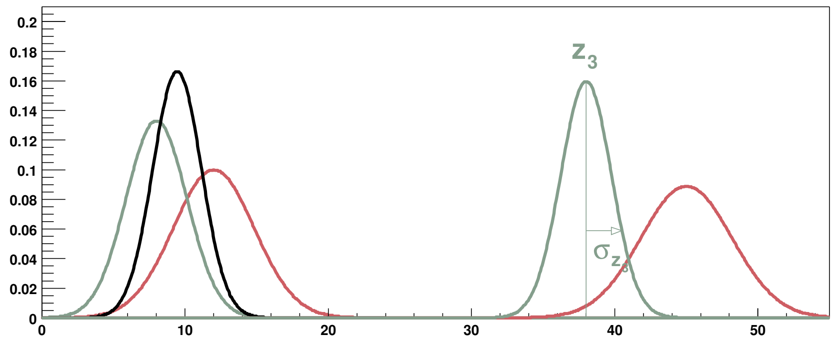


Figure 3.5: Third measurement of the position at time t_3, Gaussian distributed with mean z_3 and standard deviation σ_{z_3}.

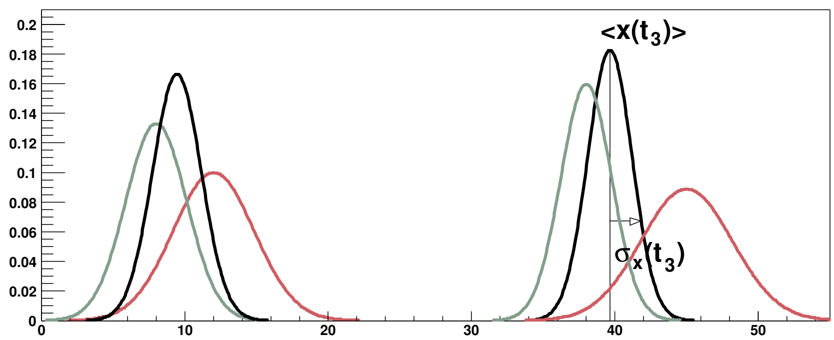


Figure 3.6: The probability distribution of the position after combining the propagated distribution of the best estimate and the third measurement is again a Gaussian distribution with the mean $< x(t_3) >$ and the standard deviation $\sigma_x(t_3)$.

3.2 Application for Track Fitting

The most obvious difference between the case of the sailing boat and track fitting is, that we do not like to determine one parameter as a function of time $x(t)$ but five track parameters as function of the radius R in order to describe the way of the particle through the detector (for the definition of the track parametrization see section 2.3.2):

$$\vec{x}(R) = \begin{pmatrix} x_1(R) \\ x_2(R) \\ x_3(R) \\ x_4(R) \\ x_5(R) \end{pmatrix} = \begin{pmatrix} \phi(R) \\ z(R) \\ \theta(R) \\ \beta(R) \\ \kappa(R) \end{pmatrix} \tag{3.14}$$

In case of the sailing boat, the best estimate of the position $x(t)$ has been expressed by the mean $< x(t) >$ and the standard deviation $\sigma_x(t)$. In the five dimensional case, the probability density distribution of the best estimate of the five parameters is described by the mean $< \vec{x}(R) >$ and by a 5×5 covariance matrix $\Sigma^2_{\vec{x}}(R)$:

$$\Sigma^2_{\vec{x}}(R) = \begin{pmatrix} \sigma^2_\phi(R) & \sigma^2_{\phi z}(R) & \sigma^2_{\phi\theta}(R) & \sigma^2_{\phi\beta}(R) & \sigma^2_{\phi\kappa}(R) \\ \sigma^2_{z\phi}(R) & \sigma^2_z(R) & \sigma^2_{z\theta}(R) & \sigma^2_{z\beta}(R) & \sigma^2_{z\kappa}(R) \\ \sigma^2_{\theta\phi}(R) & \sigma^2_{\theta z}(R) & \sigma^2_\theta(R) & \sigma^2_{\theta\beta}(R) & \sigma^2_{\theta\kappa}(R) \\ \sigma^2_{\beta\phi}(R) & \sigma^2_{\beta z}(R) & \sigma^2_{\beta\theta}(R) & \sigma^2_\beta(R) & \sigma^2_{\beta\kappa}(R) \\ \sigma^2_{\kappa\phi}(R) & \sigma^2_{\kappa z}(R) & \sigma^2_{\kappa\theta}(R) & \sigma^2_{\kappa\beta}(R) & \sigma^2_\kappa(R) \end{pmatrix} \tag{3.15}$$

$\sigma_i, i = 1, ..., 5$ are the standard deviations of x_i and $\frac{\sigma^2_{ij}}{\sigma_i * \sigma_j}, i = 1, ..., 5, j = 1, ...5, i \neq j$ are the correlations between the track parameters x_i and x_j.
The weight matrix of the track parameters is the inverse of the error matrix:

$$\Omega_{\vec{x}}(R) = (\Sigma^2_{\vec{x}}(R))^{-1} \tag{3.16}$$

The CDF detector measures z and ϕ positions only, thus the measurement parameters are $z_m(R)$ and $\phi_m(R)$ with the standard deviations $\sigma_{z_m}(R)$ and $\sigma_{\phi_m}(R)$ corresponding to the approximatively Gaussian distributed errors of the measurements.
The transport model of the boat was a linear function of the estimated position at time t_2, $< x(t_2) >$ and of the time t_2 and t_3 (equ. 3.8). $< \vec{x}(R') >$ can be calculated as a function of $< \vec{x}(R) >$, R and R'. But unfortunately this is not at all a linear function. Thus the Kalman fitter algorithm is not valid for fitting the track parameters.

But a trick helps out. In order to express the transport model via a linear first order approximation the fit parameters have to be relatively small. Therefore the track parameters are split up into a reference trajectory and small deviations from this reference:

$$< \vec{x}(R) >= \vec{x}_{ref}(R)+ < \delta\vec{x}(R) >= \begin{pmatrix} \phi(R)_{ref} \\ z(R)_{ref} \\ \theta(R)_{ref} \\ \beta(R)_{ref} \\ \kappa(R)_{ref} \end{pmatrix} + \begin{pmatrix} < \delta\phi(R) > \\ < \delta z(R) > \\ < \delta\theta(R) > \\ < \delta\beta(R) > \\ < \delta\kappa(R) > \end{pmatrix} \quad (3.17)$$

The covariance and the weight matrix of the deviations are the same as for the parameters themselves:

$$\begin{aligned} \Sigma^2_{\delta\vec{x}}(R) &= \Sigma^2_{\vec{x}}(R) && (3.18) \\ \Omega_{\delta\vec{x}}(R) &= \Omega_{\vec{x}}(R) && (3.19) \end{aligned}$$

$< \delta\vec{x}(R^{'}) >$ is still not a linear function of $< \delta\vec{x}(R) >$, but due to the smallness of the components of $< \delta\vec{x}(R) >$ all second and higher order terms in the Taylor expansion of this function can be neglected:

$$\mathcal{T}(R, R^{'}) = \begin{pmatrix} \frac{d\phi(R^{'})}{d\phi(R)} & \frac{d\phi(R^{'})}{dz(R)} & \frac{d\phi(R^{'})}{d\theta(R)} & \frac{d\phi(R^{'})}{d\beta(R)} & \frac{d\phi(R^{'})}{d\kappa(R)} \\ \frac{dz(R^{'})}{d\phi(R)} & \frac{dz(R^{'})}{dz(R)} & \frac{dz(R^{'})}{d\theta(R)} & \frac{dz(R^{'})}{d\beta(R)} & \frac{dz(R^{'})}{d\kappa(R)} \\ \frac{d\theta(R^{'})}{d\phi(R)} & \frac{d\theta(R^{'})}{dz(R)} & \frac{d\theta(R^{'})}{d\theta(R)} & \frac{d\theta(R^{'})}{d\beta(R)} & \frac{d\theta(R^{'})}{d\kappa(R)} \\ \frac{d\beta(R^{'})}{d\phi(R)} & \frac{d\beta(R^{'})}{dz(R)} & \frac{d\beta(R^{'})}{d\theta(R)} & \frac{d\beta(R^{'})}{d\beta(R)} & \frac{d\beta(R^{'})}{d\kappa(R)} \\ \frac{d\kappa(R^{'})}{d\phi(R)} & \frac{d\kappa(R^{'})}{dz(R)} & \frac{d\kappa(R^{'})}{d\theta(R)} & \frac{d\kappa(R^{'})}{d\beta(R)} & \frac{d\kappa(R^{'})}{d\kappa(R)} \end{pmatrix} \quad (3.20)$$

$$\begin{aligned} < \delta\vec{x}(R^{'}) > &\approx \mathcal{T}(R, R^{'})* < \delta\vec{x}(R) > && (3.21) \\ \Sigma^2_{\delta\vec{x}}(R^{'}) &\approx \mathcal{T}(R, R^{'}) * \Sigma^2_{\delta\vec{x}}(R) * \mathcal{T}^T(R, R^{'}) && (3.22) \end{aligned}$$

We now do not longer fit the track parameters but their deviation from the reference. Their transport model can be considered as approximatively linear, thus the use of the Kalman fitter is valid. The precise form of the transport matrix $\mathcal{T}$ is derived in appendix B.

The measurements, which are incorporated in the Kalman fitter, have to be expressed as deviation from the reference as well, and not as absolute values z_m and ϕ_m:

$$< \delta z_m(R) > \; = \; < z_m(R) > - z_{ref}(R) \tag{3.23}$$
$$< \delta\phi_m(R) > \; = \; < \phi_m(R) > - \phi_{ref}(R) \tag{3.24}$$
$$\sigma_{\delta z_m}(R) = \sigma_{z_m}(R) \tag{3.25}$$
$$\sigma_{\delta\phi_m}(R) = \sigma_{\phi_m}(R) \tag{3.26}$$

Once we have defined the set of parameters for which the Kalman fitter is valid, the incorporation of a new measurement and the calculation of the best estimate correspond exactly to the one-dimensional case.

Let us assume that we have a series of ϕ and z measurements sorted by their radii. In addition there is a reference trajectory $\vec{x}_{ref}(R)$. This can either be the parameters of the track, which have been already found in the drift chamber outside of the silicon or a rough estimate, precalculated by some of the ϕ and z measurements, which we assume belonging to the track.

We start to fit the track parameters at the first radius R_1, where a z and a ϕ measurement are available. At this stage we do not yet have taken any measurement into account. We start with the estimate $< \delta\vec{x}(R_1^-) >= \vec{0}$. This estimate has so far no weight: $\Omega_{\delta\vec{x}}(R_1^-) = O$[1].

[1]In fact in the algorithms the diagonal values of $\Omega_{\delta\vec{x}}(R_1^-)$ are set to very small numbers, which are orders of magnitudes smaller than the weights expected from the measurements. This is done to guarantee the weight matrix to be invertible, which is needed for calculations of the weighted mean. The impact of this manipulation of the weight matrix to the final fit result is negligibly small.

First the weighted mean between the ϕ measurement and the estimate is calculated.

$$< \delta\vec{x}_m(R_1) > \; = \; \begin{pmatrix} < \phi_m(R_1) > -\phi_{ref}(R_1) \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.27}$$

$$\Omega_{\delta\vec{x}_m}(R_1) \; = \; \begin{pmatrix} \frac{1}{\sigma^2_{\phi_m}(R_1)} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{3.28}$$

$$< \delta\vec{x}(R_1) > \; = \; \frac{\Omega_{\delta\vec{x}}(R_1^-) * < \delta\vec{x}(R_1^-) > + \Omega_{\delta\vec{x}_m}(R_1) * < \delta\vec{x}_m(R_1) >}{\Omega_{\delta\vec{x}}(R_1^-) + \Omega_{\delta\vec{x}_m}(R_1)} \tag{3.29}$$

$$= \; \begin{pmatrix} < \phi_m(R_1) > -\phi_{ref}(R_1) \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\Omega_{\delta\vec{x}}(R_1) \; = \; \Omega_{\delta\vec{x}}(R_1^-) + \Omega_{\delta\vec{x}_m}(R_1) \tag{3.30}$$

$$= \; \begin{pmatrix} \frac{1}{\sigma^2_{\phi_m}(R_1)} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The probability distribution of the best estimate of the track parameters after the first measurement corresponds exactly to the probability distribution of the first measurement, as in the one-dimensional equation (equ. 3.1, equ. 3.3). Then the z measurement at the same radius $R_2 = R_1$ is incorporated in the following way:

$$< \delta\vec{x}_m(R_2) > \; = \begin{pmatrix} 0 \\ < z_m(R_2) > - z_{ref}(R_2) \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.31}$$

$$\Omega_{\delta x_m}(R_2) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sigma^2_{z_m}(R_2)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{3.32}$$

$$\begin{aligned} < \delta\vec{x}(R_2) > &= \frac{\Omega_{\delta\vec{x}}(R_2^-) * < \delta\vec{x}(R_2^-) > + \Omega_{\delta\vec{x}_m}(R_2) * < \delta\vec{x}_m(R_2) >}{\Omega_{\delta\vec{x}}(R_2^-) + \Omega_{\delta\vec{x}_m}(R_2)} \qquad (3.33) \\ &= \begin{pmatrix} < \phi_m(R_1) > - \phi_{ref}(R_1) \\ < z_m(R_2) > - z_{ref}(R_2) \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} \Omega_{\delta\vec{x}}(R_2) &= \Omega_{\delta\vec{x}}(R_2^-) + \Omega_{\delta\vec{x}_m}(R_2) \qquad (3.34) \\ &= \begin{pmatrix} \frac{1}{\sigma^2_{\phi_m}(R_1)} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sigma^2_{z_m}(R_2)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{aligned}$$

Now both measurements are incorporated. Due to the fact that this has been the first z measurement the probability density of the estimate of the z position after the measurement corresponds to the probability distribution of the measurement.
Before the best estimate can be propagated to the radius of the next measurement multiple scattering and dE/dx effects due to the material of the wafer at radius $R3 \approx R2$ have to be added. Multiple scattering is a symmetric effect, so it does not change the mean value of the parameters but introduces errors in the two parameters, which describe the track direction: β and θ. The total momentum of the track stays the same for multiple scattering but due to the change in θ the relation between p_z and p_T changes. Thus the change in θ introduces a change in κ. The size of those errors depends on the path length in the material, material properties and the transverse momentum of the particle:

$$< \delta\vec{x}(R_3) > \quad = \quad < \delta\vec{x}(R_3^-) > \tag{3.35}$$

$$\Sigma^2_{ms} \quad = \quad \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{ms} & 0 & \frac{\delta\kappa}{\delta\theta} * \sigma^2_{ms} \\ 0 & 0 & 0 & \sigma^2_{ms} & 0 \\ 0 & 0 & \frac{\delta\kappa}{\delta\theta} * \sigma^2_{ms} & 0 & (\frac{\delta\kappa}{\delta\theta})^2 * \sigma^2_{ms} \end{pmatrix} \tag{3.36}$$

$$\Sigma^2_{\delta\vec{x}}(R_3) \quad = \quad \Sigma^2_{\delta\vec{x}}(R_3^-) + \Sigma^2_{ms} \tag{3.37}$$

The curvature change (dE/dx-correction) implies a correction of the mean value of κ and introduce an error on κ. The values of the correction and the error are again functions of the path length in the material, material properties and the transverse momentum of the particle. The influence of the best estimate due to dE/dx curvature change at the wafer at $R_4 \approx R_3$ are:

$$< \delta\vec{x}_{dE/dx}(R_4) > \quad = \quad \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ < \delta\kappa_{dE/dx} > \end{pmatrix} \tag{3.38}$$

$$< \delta\vec{x}(R_4) > \quad = \quad < \delta\vec{x}(R_4^-) > + < \delta\vec{x}_{dE/dx}(R_4) > \tag{3.39}$$

$$\Sigma^2_{dE/dx} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{dE/dx} \end{pmatrix} \tag{3.40}$$

$$\Sigma^2_{\delta\vec{x}}(R_4) = \Sigma^2_{\delta\vec{x}}(R_4^-) + \Sigma^2_{dE/dx} \tag{3.41}$$

Relative to σ^2_κ, $\sigma^2_{dE/dx}$ is negligibly small (order of 10^{-3}) but hard to calculate. Therefore it is not taken into account in the Kalman fitter of the TrackingKal package. The precise formulae for the multiple scattering and the dE/dx correction are derived in appendix B.
Now all material is incorporated correctly and the best estimate can be propagated to the radius R_5, where the next measurement and/or material is located ($\mathcal{T}$ is defined in equ. 3.20):

$$< \delta\vec{x}(R_5) > = \mathcal{T}(R_4, R_5) * < \delta\vec{x}(R_4) > \tag{3.42}$$

$$\Sigma^2_{\delta\vec{x}}(R_5) = \mathcal{T}(R_4, R_5) * \Sigma^2_{\delta\vec{x}}(R_4) * \mathcal{T}^T(R_4, R_5) \tag{3.43}$$

$$\Omega_{\delta\vec{x}}(R_5) = \mathcal{T}^T(R_5, R_4) * \Omega_{\delta\vec{x}}(R_4) * \mathcal{T}(R_5, R_4) \tag{3.44}$$

Now all fit actions, which are needed for using the Kalman filter as fitter for track parameters, have been introduced: incorporating a measurement, incorporating dE/dx and multiple scattering effects of the traversed material and propagating the best estimate to another radius. Those actions just have to be repeated at every relevant radius R_i until the last radius is reached.

3.2.1 Forward and Backward Fit

There are two possible fit directions for the Kalman fitter. One is the forward direction, which corresponds to the flight direction of the particles assuming they are originated in the beampipe. The other one is the backward direction, fitting from the outermost measurement to the innermost one.
The forward fitter provides the best estimate of the track parameters outside of the silicon detector (fig.: 3.7). This is useful for extrapolating tracks found in the vertex detector to the outer drift chamber.
The backward fitter provides the best estimate of the track parameters at their origin (fig.: 3.8). This is the most interesting issue for physics analyses. In general the best estimate at a given radius is the weighted mean of the forward fitter $\vec{x}_F(R)$,

incorporating all fit actions from the beampipe up to this radius, and the backward fitter $\vec{x}_B(R)$ coming from the outermost measurement up to this radius (fig.: 3.9):

$$< \delta\vec{x}(R) > \;=\; \frac{\Omega_{\delta\vec{x}_F}(R) * < \delta\vec{x}_F(R) > + \Omega_{\delta\vec{x}_B}(R) * < \delta\vec{x}_B(R) >}{\Omega_{\delta\vec{x}_F}(R) + \Omega_{\delta\vec{x}_B}(R)} \tag{3.45}$$

$$\Omega_{\delta\vec{x}}(R) \;=\; \Omega_{\delta\vec{x}_F}(R) + \Omega_{\delta\vec{x}_B}(R) \tag{3.46}$$

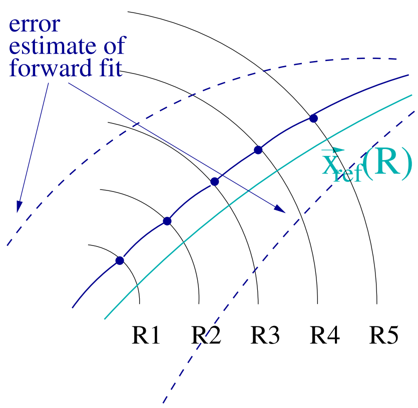


Figure 3.7: The forward fit provides the best estimate of the track parameters outside of the silicon.

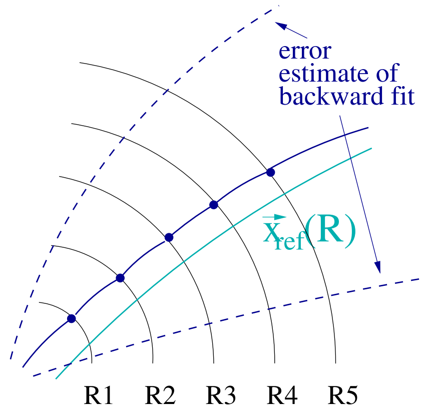


Figure 3.8: The backward fit provides the best estimate of the track parameters inside the beampipe.

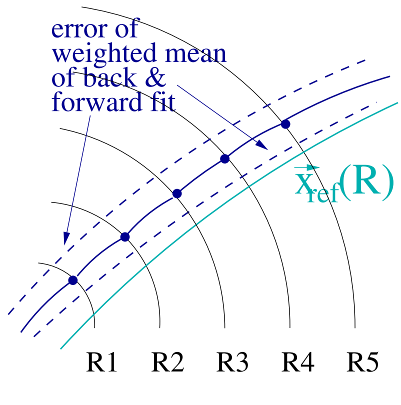


Figure 3.9: The weighted mean of the forward and backward fit at every radius is the best available estimate.

3.2.2 Kalman Fitter Applied for Electron Identification

Electrons passing through material emit Bremstrahlung which causes energy loss. Therefore it is expected to observe a difference in the curvature of the backward and the forward fit of an electron track at material layers (fig.: 3.10). A so-called constraint fit is introduced to determine the curvature change ($\Delta\kappa$) and its error. The constraint fit uses the parameters of the forward and the backward fit as input and determines the ϕ, β and κ of the track under the assumption, that β and ϕ before and after the material are the same but the curvature changes. While β, ϕ and κ are determined directly from the measurements, $\Delta\kappa$ is only determined from the correlations.

The parameters of the backward fit are β_b, ϕ_b and κ_b and the one of the forward fit are β_f, ϕ_f and κ_f. Ω_b and Ω_f are the corresponding 3×3 weight matrices. The parameters which have to be fitted are $\delta\beta_b$, $\delta\phi_b$, $\delta\kappa_b, \delta\beta_f$, $\delta\phi_f$, $\delta\kappa_f$ and $\Delta\kappa$, where δx are the differences between the measured and the true parameters. There are several constraints $f_i, 1 \leq i \leq 3$, for this fit:

$$f_1 = \phi_f + \delta\phi_f - \phi_b - \delta\phi_b = 0 \tag{3.47}$$

$$f_2 = \beta_f + \delta\beta_f - \beta_b - \delta\beta_b = 0 \tag{3.48}$$

$$f_3 = \kappa_f + \delta\kappa_f + \Delta\kappa - \kappa_b - \delta\kappa_b = 0 \tag{3.49}$$

The according Lagrange function is:

$$\begin{aligned} \mathcal{L} &= \Delta y^T \mathcal{W} \Delta y + 2 \sum_{1 \leq i \leq 3} \lambda_i f_i \\ &= \Delta y^T \mathcal{W} \Delta y + 2\lambda^T (\mathcal{A}\Delta\kappa + \mathcal{B}\Delta y - c) \end{aligned} \tag{3.50}$$

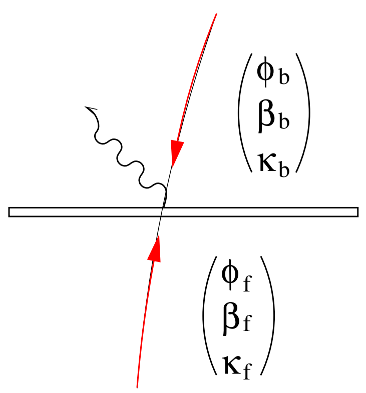


Figure 3.10: Bremsstrahlung emitted by electrons passing through material.

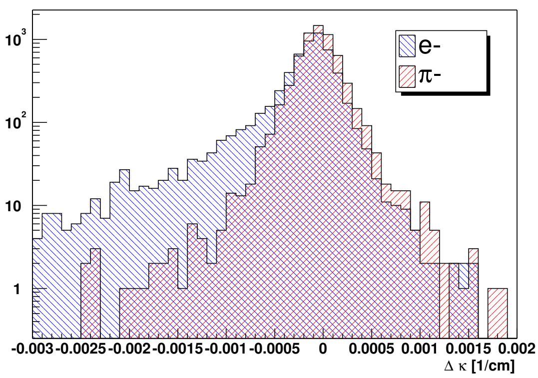


Figure 3.11: Fitted curvature change ($\Delta\kappa$) between backward and forward fit for electrons and pions at the port card.

where

$$
\begin{aligned}
\Delta y^T &= (\delta\phi_f,\ \delta\beta_f,\ \delta\kappa_f,\ \delta\phi_b,\ \delta\beta_b,\ \delta\kappa_b) \\
\mathcal{W} &= \begin{pmatrix} \Omega_f & 0 \\ 0 & \Omega_b \end{pmatrix} \\
\mathcal{A} &= \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \\
\mathcal{B} &= \begin{pmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{pmatrix} \\
\lambda^T &= (\lambda_1,\ \lambda_2,\ \lambda_3) \\
c &= \begin{pmatrix} \phi_b - \phi_f \\ \beta_b - \beta_f \\ \kappa_b - \kappa_f \end{pmatrix}
\end{aligned}
$$

The minimization of this function corresponds to the minimization of the χ^2 function ($\chi^2 = \Delta y^T \mathcal{W} \Delta y^T$) under the conditions $f_i = 0,\ 1 \le i \le 3$. Therefore $\mathcal{L}$ has to be derived with respect to all $y_i,\ \lambda_i$ and to $\Delta\kappa$. This gives the following set of equations:

$$
\begin{pmatrix} \mathcal{W} & 0 & \mathcal{B}^T \\ 0 & 0 & \mathcal{A}^T \\ \mathcal{B} & \mathcal{A} & 0 \end{pmatrix} \begin{pmatrix} \Delta y \\ \Delta\kappa \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ c \end{pmatrix} \tag{3.51}
$$

The solution for this set of equations can be e.g. found in [25], where also the covariance matrix of the fit parameters is determined.

The thickest material in the silicon detector and therefore the most promising one to observe the Bremsstrahlung for electrons is the port card, a layer of readout electronics between the SVX II and the ISL. The $\Delta\kappa$ distributions at the port card for electrons and pions of about 2 GeV are shown in figure 3.11. The power of identification of electrons by this method is weak but it can be used as additional and completely independent information to be combined with dE/dx measurements in the COT and in the time of flight system, for example as input to a neural net.
A study of electron identification using the Kalman fitter can be found in [26].

3.2.3 Validity Of Linear Approximation

One of the basic assumptions of the Kalman fitter is the linearity of the transport model. In the case of track fitting the nonlinear problem has been approached by a linear approximation which is based on small deviations of the track parameters from the reference trajectory. This smallness can not always been guaranteed by using one single helix as reference. E.g. if the COT track parameters are used as reference helix the deviations in the inner-more silicon layers can become too large. This results in a bias towards the reference parameters of some microns in r/ϕ and z for low energetic tracks. Compared to the hit resolution of about 15 μm this bias have no impact on the pattern recognition.
But for the final fit a dependency of the reference has to be avoided definitely. Therefore a piece wise reference helix has been introduced which is automatically adapted to the track parameters during fitting. Every time a measurement is incorporated to the fit, the reference is set to the new best estimate and the corresponding deviations are set to zero. The weight matrices are kept as they were before.

$$\vec{x}^{\prime}_{ref}(R) = \vec{x}_{ref}(R) + < \delta\vec{x}(R) > \tag{3.52}$$

$$< \delta\vec{x}^{\prime}(R) > = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.53}$$

$$\Sigma^2_{\delta\vec{x}^{\prime}}(R) = \Sigma^2_{\delta\vec{x}}(R) \tag{3.54}$$

$$\Omega_{\delta\vec{x}^{\prime}}(R) = \Omega_{\delta\vec{x}}(R) \tag{3.55}$$

For the one-dimensional case of the sailing boat this switch of the reference corresponds to a shift of the origin of the coordinate system. This does not at all influence the fit result. But in case of track fitting it guarantees the smallness of the deviations of the track parameters from the reference and therefore the validity of the linear approximation. So the fit does no longer depend on the choice of the reference and the reference is optimal at every radius.
For calculation of the best estimate of forward and backward fit, it should be kept in mind that there are now different references for both fit directions. All calculations have to be done in absolute track parameters and not in terms of deviations from the reference any more.

3.3 Validation of the Fitter

3.3.1 Validation on Monte Carlo

Two samples of 10^5 single muons each have been used for the following validation of the fitter. One contains μ^-, the other one μ^+ events, all simulated with a flat distribution in transverse momentum (0.5 GeV $\leq |p_T| \leq$ 50 GeV) and in pseudo-rapidity space (-2 $\leq \eta \leq$ 2).
The major ingredients for the fitter are the hit positions and their errors. Only if the hit errors correspond precisely to the hit resolution any evaluation of the fitter can be performed. Therefore and in order to obtain a realistic estimate of the resolution expected on data, the hit smearing explained in section 2.5 has been used for the simulation of the Monte Carlo (MC) files.
There are two major cross checks of the fitter. The pulls (difference between the reconstructed parameters and the MC_{truth} parameters divided by the fit errors) are expected to be Gaussian distributed with a σ of one. Otherwise either the input errors of the measurements are not described correctly, or the fitter has a bug in the calculation or propagation of its covariance matrices. Additionally, the pulls have to be centered at zero in order to exclude any bias. Especially the mean of the curvature pull is of interest, because it depends on the incorporated amount of material and the corresponding dE/dx corrections.
If those cross checks are consistent the quality of the fitter is measured by the parameter resolution (size of the residuals, difference between the reconstructed parameters and the MC_{truth}).

Parameter Pull Plots

The size of the pulls of the Kalman fitter in dependence of the tracks transverse momentum for the μ^- sample and for silicon only tracks are shown in fig. 3.12. They look the same for tracks with additional COT information and for the μ^+ sample.
The parameters in the r/z plane are normalized over the whole range of transverse momentum as they should be.
The ϕ_0 and D pulls are slightly too large, but the pulls are flat over the whole range of transverse momentum, therefore this effect is most probably not dominated by multiple scattering. The pulls in the r/ϕ plane are more sensitive to wrong hit errors than the one in z, because the number of r/ϕ hits and therefore the r/ϕ resolution is higher. Although the hit smearing has been used the hit errors are still not independent of the pseudo-rapidity of the track and of the wafers they are on. The hit smearing has just introduced a rough normalization at layer level depending on the cluster size of the hits.

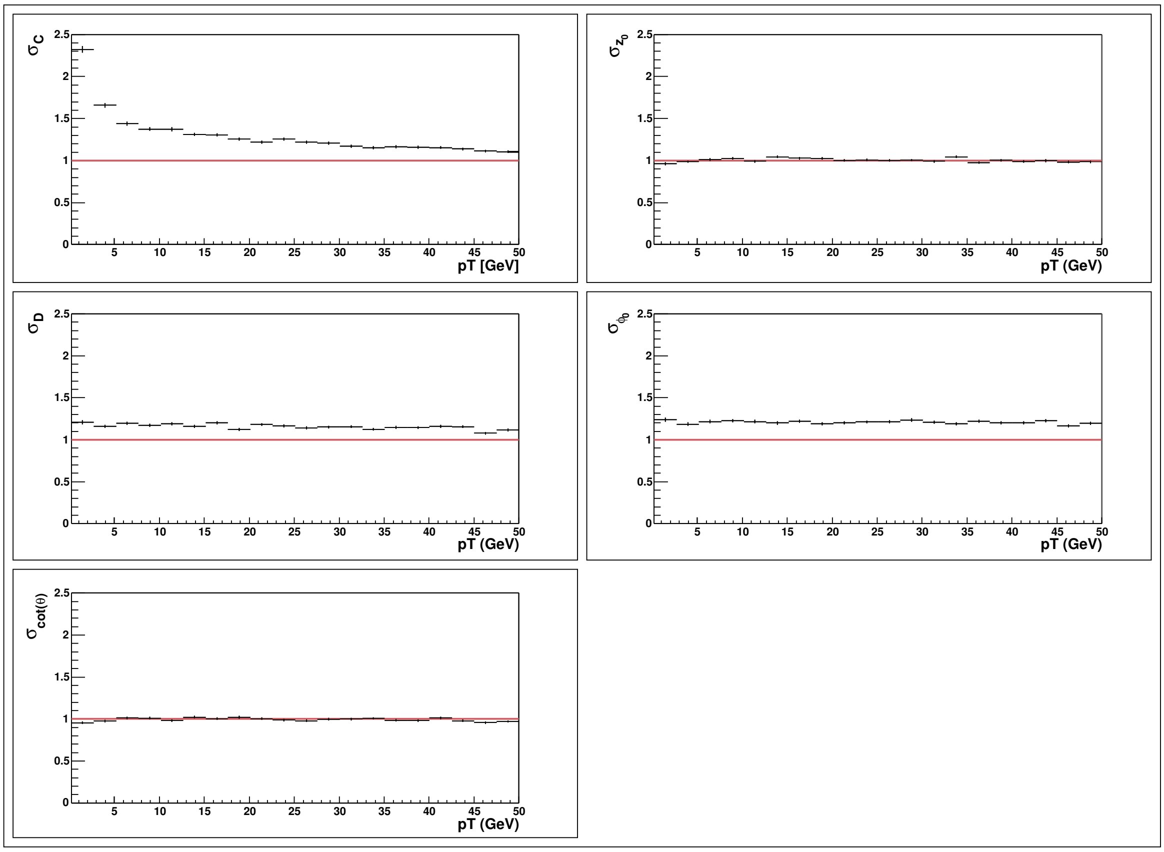


Figure 3.12: Size of the pulls of silicon only tracks fitted with the Kalman fitter.

The effect of material effects on r/ϕ parameters can be seen in the decrease of the size of the curvature pulls for low transverse momentum. Multiple scattering for low momentum particles can be approximatively described by a Gaussian distribution only. This description is valid for about 98 % of the cases, but for the remaining 2 % of the tracks hard scatters are expected, which are not incorporated in the fitter. The curvature pulls are more sensitive to any non Gaussian effect, because they are measured in the most indirect way compared to the other track parameters. z and ϕ are measured directly, $\cot(\theta)$ and the impact parameter can be obtained out of two measurements but the curvature needs at least three measurements.
Although the sizes of the pull plots of the fitter are slightly larger than the size of a normal Gaussian, they are good enough to exclude any major bug in the fitter. Correspondingly to the more and more detailed description of the hit errors obtained on data the size of the pulls will vary. But in longer terms, normalizing factors for the fit errors of the Kalman fitter will be introduced in order to obtain normalized pulls.

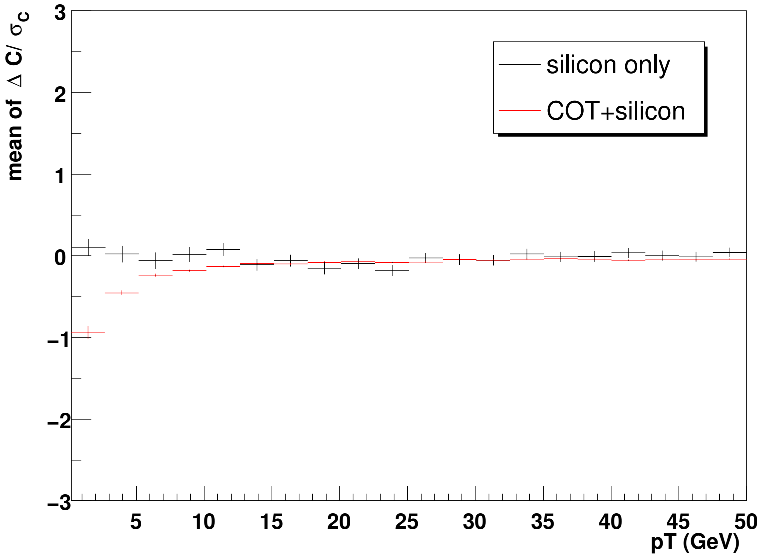


Figure 3.13: Mean value of curvature pull for silicon only and COT+silicon tracks.

dE/dx Correction

The mean of the curvature pull as a function on the transverse momentum for negatively charged muons is shown in figure 3.13.
The plot shows a correct mean value for the silicon only tracks. But there is a slight shift of about one standard deviation towards low momenta for the tracks with COT and silicon information. For the μ^+ sample the shift is in the opposite direction. This indicates too few dE/dx corrections most probably inside the drift chamber. This is outside the influence of the Kalman fitter.
The material inside the silicon seems to be incorporated correctly.

Parameter Resolution

The parameter resolutions are calculated for different transverse momenta (fig. 3.14 - 3.18). In the high p_T limit the parameter resolutions for silicon only tracks are:

- $\frac{\delta p_T}{p_T^2} \approx 3.5 \times 10^{-3}$ GeV^{-1}
- $\delta\phi_0 \approx 0.2$ mrad
- $\delta D \approx 10\ \mu$m
- $\delta z_0 \approx 50\ \mu$m
- $\delta \cot(\theta) \approx 0.9 \times 10^{-3}$

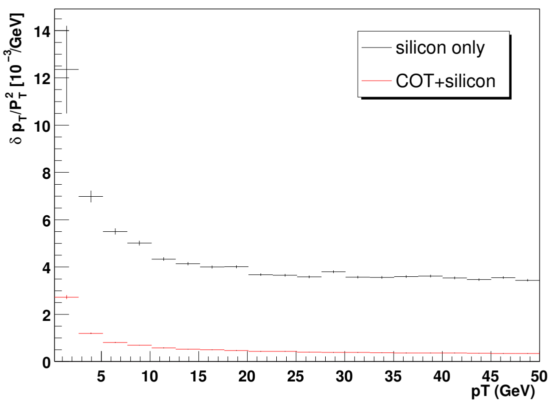


Figure 3.14: p_T resolution of the fitter.

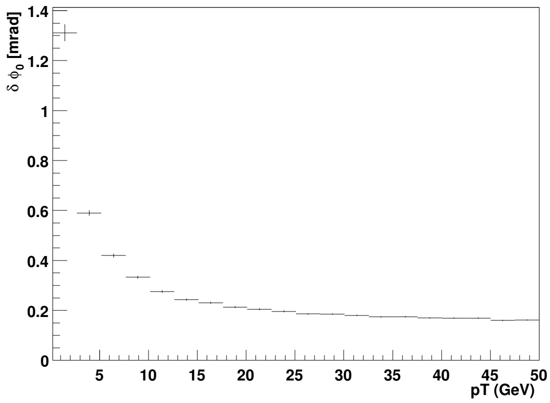


Figure 3.15: ϕ_0 resolution of the fitter.

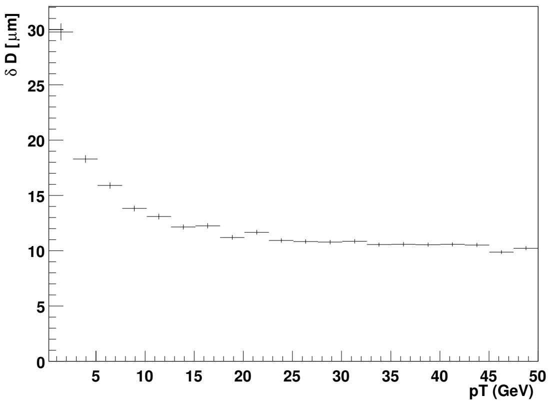


Figure 3.16: D resolution of the fitter.

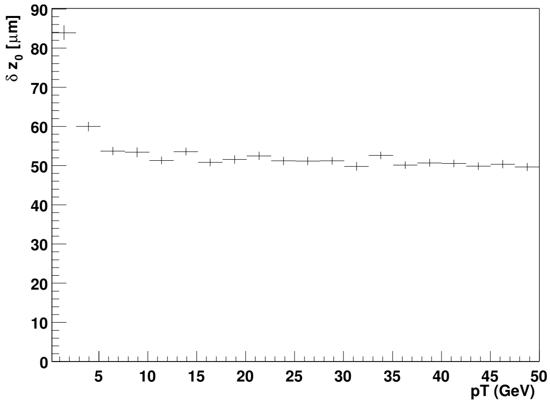


Figure 3.17: z_0 resolution of the fitter.

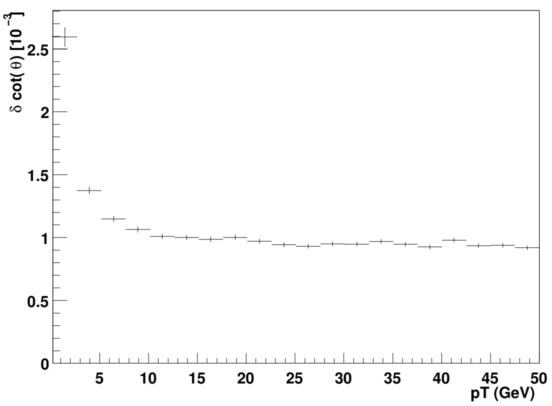


Figure 3.18: $\cot(\theta)$ resolution of the fitter.

While the position and direction parameter resolution remains the same wether or not COT information is added to the fitter, the p_T resolution improves significantly (fig. 3.14):

- $\frac{\delta p_T}{p_T^2} \approx 0.5 \times 10^{-3}$ GeV^{-1}

Those resolutions are better than the design values. But it has to be taken into account, that these are single particle events in a fully efficient detector. No shared hits are incorporated in the fit as it will happen in a populated event.

3.3.2 Validation on Data

The reconstruction of the J/Ψ mass is a good method to validate and to calibrate the fitter on data for several reasons: Its mass is precisely known from other experiments, the resonance is narrow and there are quite some J/Ψ mesons already detected.
The J/Ψ mass peak has been reconstructed with two different fitters (fig: 3.19) from 37 pb^{-1} of the data of the $J/\Psi \to \mu^+\mu^-$ trigger sample. One plot has been obtained by

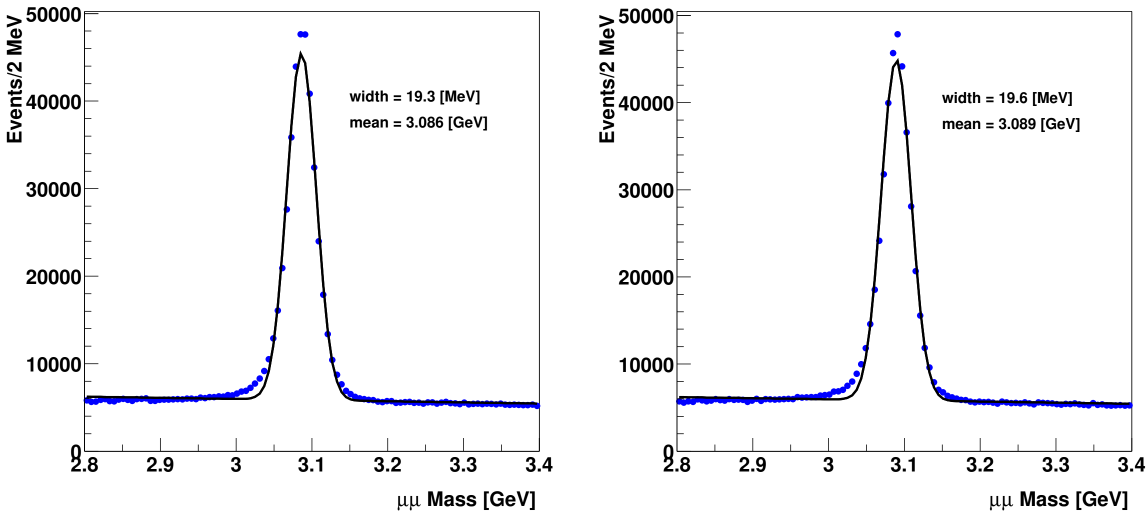


Figure 3.19: J/Ψ peak reconstructed with the Kalman fitter of the TrackingKal package using the detailed subset detector (left) and reconstructed with the SiTrackFitter using the GEANT integrator (right).

refitting the tracks with the SiTrackFitter using the GEANT material integrator. The second plot uses the Kalman fitter of the TrackingKal package using the detailed subset detector[2] for refitting.
Two different fitters with different material integrators are compared, therefore the resolution is the product of two components: the precision of the fitter and the precision of the material description. The resolution obtained by the Kalman fitter ($\sigma =$ 19.3 MeV) is slightly better than the SiTrackFitter using the entire GEANT material description for the material integration ($\sigma = 19.6$ MeV). The material description of the Kalman fitter of the TrackingKal package can by definition not be better than the GEANT description but the Kalman fitter proves a better performance in terms of resolution. The mean value of the peak obtained by the Kalman fitter is a little bit lower as the one of the SiTrackFitter because the detailed subset geometry description contains less material than the GEANT description.
But also the mean of peak obtained with the GEANT integrator is lower as the PDG value of the J/Ψ mass of 3.096 GeV [27]. Therefore the B group has developed a calibration [28] (fig: 3.20). It uses the J/Ψ mass as input. An additional material layer of 0.560 g/cm^2 (≈ 4 % X_0) has been added under the inner wall of the COT to correct for the energy dependent slope of the J/Ψ mass. This additional material in the detector compared to the simulation is consistent with photon conversion plots obtained from data. In addition a B field correction (B = 1.41348 T instead of 1.4116 T) has been applied to shift the peak towards the PDG value. This calibration has been cross checked with other particle signals, e.g. the D^0 mass (fig.: 3.21) and the Y resonances (fig.: 3.22). Although this calibration has been performed with the SiTrackFitter, the Kalman fitter can use it as well once the Silimap is reliable working. The additional material layer has to be added into the GEANT description used for the scan. The B field correction can easily be read in by the Kalman fitter as well.

[2]The SiliMap integrator does currently not show consistent results (status of end of 2002). The problem is under investigation.

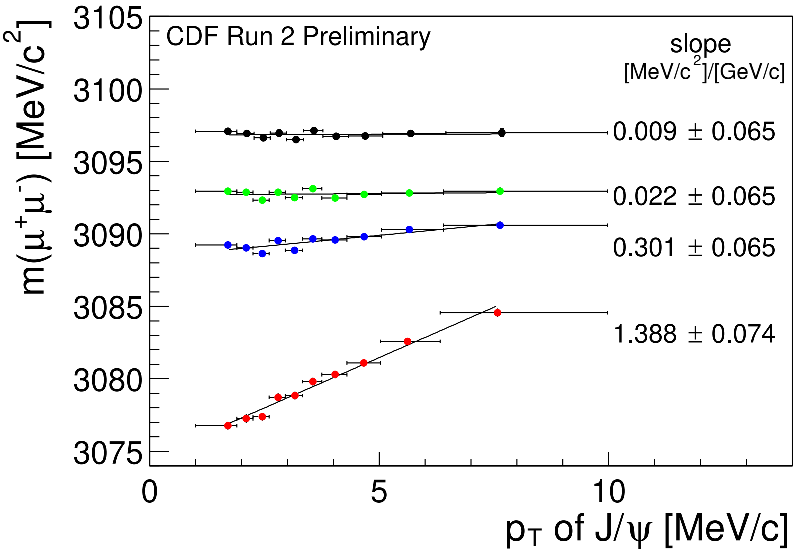


Figure 3.20: Fitter and tracking calibration using the J/Ψ sample, red (lowest) curve without any material correction, blue (next lowest) curve with corrections for all known material, green (second highest) curve with additional material layer, black (top) curve with additional B field correction applied.

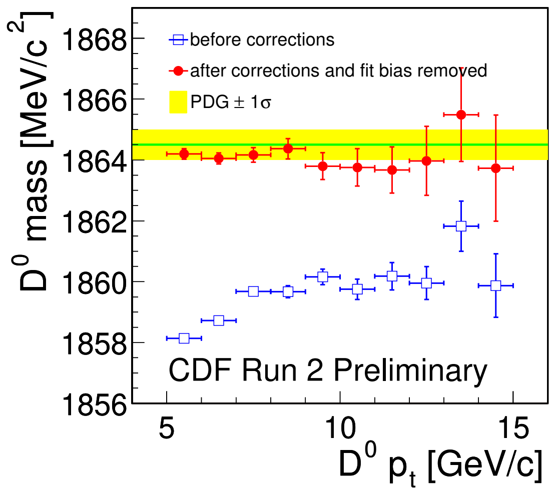


Figure 3.21: Cross check of fitter calibration with the D^0 mass.

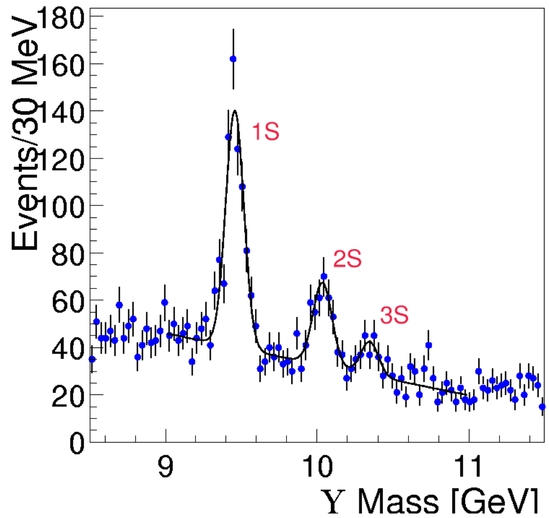


Figure 3.22: Cross check of fitter calibration with Υ resonances.

Chapter 4

Silicon Tracking Algorithms

This chapter first gives an overview of the CDF track reconstruction software of the tracking system. The TrackingKal package is a central part of this software. It provides one of two Outside-In algorithms used for processing the data. Additionally, it provides the only working Silicon Standalone algorithm, which is used for the standard data processing as well. These algorithms are then described in detail.
The main feature of TrackingKal strategies is the use of the Kalman fitter described in the last chapter already during the pattern recognition. This is only possible due to the fast and efficient implementation of the fitter and the according geometry description. Using an optimal fitter already for pattern recognition means to base the decision between different possible hits, especially in dense environments, on the best information available at this stage. This is essential in order to provide pure tracks with low wrong hit usage.
In addition, the tracks coming out of the TrackingKal strategies are not only containers of a list of silicon hits but they are already completely fitted tracks. They provide the best estimate of the track parameters and the corresponding covariance matrix. They can be used for trigger decisions and analyses directly without any refit. This is an unique property of the algorithms of the TrackingKal package.

4.1 Overview of the CDF tracking algorithms

The track reconstruction software of the CDF tracking system consists of a combination of several algorithms.
The drift chamber is at larger radii than the silicon tracker, thus the relative track

density is lower there than in the vertex detector. The tracks are more isolated, which results in less combinatorics for the track reconstruction. Thus the tracking in the COT is purer and faster as in the silicon. Therefore first the tracking in the drift chamber is performed. The different kinds of tracking algorithms in the COT are described e.g. in the references [15], [29] and [30].

The next step is the so-called Outside-In (OI) algorithm. The tracks reconstructed in the COT are extrapolated into the silicon vertex detector, where additional silicon hits are attached to them. Every track passed from the COT to the OI algorithm can be considered as reasonable track. So the OI tracking has only to decide which hits belong to the track, but not to distinguish between fake or real tracks. Besides the OI algorithm in the TrackingKal package, there is a second OI algorithm which is an extension of the algorithm used in RUN I. A description of this strategy can be found in reference [31].

The Silicon Standalone algorithm reconstructs tracks with the information in the silicon detector only, without considering any additional detector devices. The main challenge of the Standalone tracking is to distinguish between arbitrary combinations of hits accidently forming a possible track and real tracks. The hits already used by the OI tracking are flagged, which helps to reduce the combinatorics of the Standalone tracking and to avoid to reconstruct ghost tracks[1]. Due to the larger η acceptance of the silicon up to $|\eta| \leq 2$ compared to $|\eta| \leq 1$ for the COT, the main issue of the Silicon Standalone tracking is the reconstruction of tracks in the forward region. But also inefficiencies of the COT/OI tracking in the central region can be recovered.

The last pattern recognition step should be an Inside-Out (IO) strategy, which picks up some COT segments in the forward region to improve the Standalone tracks. Those COT segments have given too little information to construct a reliable COT track. But with the additional information of the silicon they could be associated to tracks. An IO tracking is not yet implemented in the CDF software, but it is under investigation.

4.2 The Outside-In Algorithm

4.2.1 The Three Loop Version

The OI algorithm uses COT tracks as seed. The COT track serves as reference for the fit of the OI track. The COT parameters and their covariance matrix are also the first incorporated measurements for the fit. The track is then extrapolated to

[1] A couple of ghost tracks are two tracks which contain hits of the same particle but are accidently considered as two different tracks in the pattern recognition.

the radius of the interaction with the outermost silicon layer. All ϕ hits in a several standard deviations large window around the calculated intersection are considered as possibly belonging to the track. For every hit in the window the track is copied, and one of the hits is added to each copy. This procedure is called cloning. In order to take care of the potential hit inefficiencies in the detector one clone without a hit in the layer is build, too. For every clone the track is fitted with the Kalman fitter including the new hit, and the best estimate is transported to the next active silicon wafer. There the hit search is performed for every clone. The clones are branched up and fitted corresponding to the number of hits in their windows. This procedure is continuing till the innermost layer is processed (fig.: 4.1).
In the end the best clone has to be chosen. The selection criterion is a combination of χ^2 per degree of freedom[2] (dof) of the fit and the number of hits picked up.
Once a r/ϕ clone is chosen SAS hits are added to this track. The best r/ϕ track is used as seed for this second path. The hit search starts again from the outside to the inside. Several standard deviations large windows around the best estimate of the fit for the z position are examined. The tracks are branched up, if several hits are in the window, and one clone without a SAS hit in the corresponding layer is kept as well.
The best SAS clone is chosen by the same criteria as in the r/ϕ path. The minimum requirement for SAS hits is zero. Even if no SAS hit has been added to the r/ϕ track, the track is still kept and used as seed to the $90°z$ path.
This last loop is performed similar to the r/ϕ and SAS path. No $90°z$ hit is required to be found. Due to the three loops, r/ϕ, SAS and then $90°z$, this OI strategy is called three loop version.

4.2.2 The Two Loop Version

The two loop version is essentially the same as the three loop version. But in the 3D layers the ϕ and SAS hits are picked up together. The z measurement of the SAS hits is heavily correlated to the position of the ϕ hit. Thus picking up the hits together provides an additional information about the reliability of the ϕ hit. But on data, due

[2]The dof of a fit is the sum of the dof in the r/ϕ and in the z space. The dof in each parameter subspace is the difference of the number of measurements of parameters of this subspace incorporated in the fit and the number of parameters needed to describe the helix in this subspace. Three parameters are needed to describe the helix in r/ϕ and two to describe it in z. If there are less measurements than parameters in one subspace the fit is under-determined and the corresponding dof is zero. Due to the COT measurement, which is included to the fit of every OI track, all parameters are determined already before the track enters the silicon. Thus the dof of the OI tracks corresponds directly to the number of hits picked up in the silicon.

to the fact that the z alignment is not yet[3] in place, this correlation can negatively influence the ϕ hit search. Both versions have about the same performance. Cross checking the result of the two loop version with the three loop version is done to exclude any systematics. In this thesis all results are shown for the three loop version. The results of the two loop version are not significantly different.

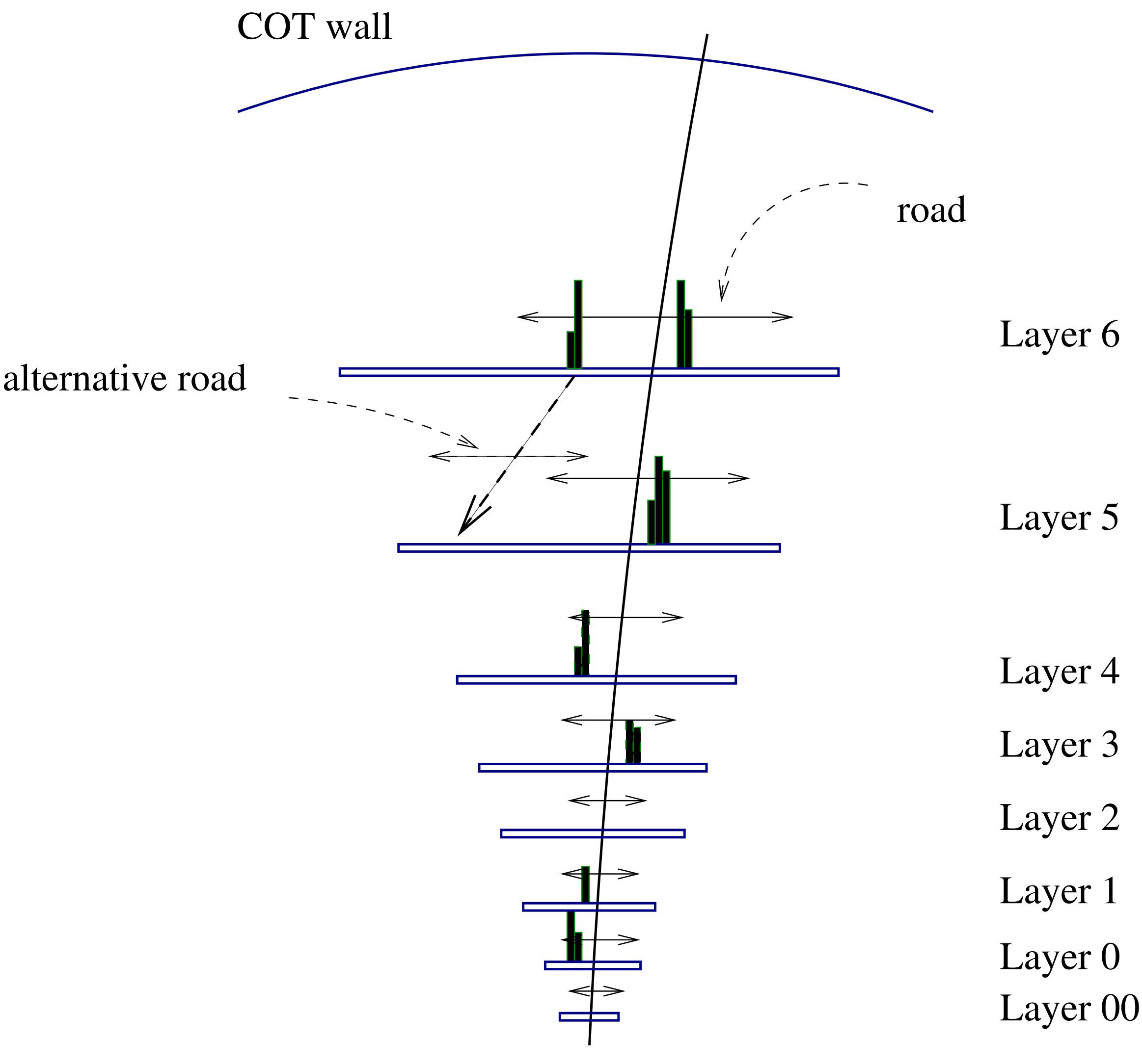


Figure 4.1: Following a track from the outside into the SVX II.

[3]software release 4.6.2

4.3 The Silicon Standalone Algorithm

The basic idea of picking up hits in the Standalone tracking is similar to the OI tracking. The significant difference is that the seed tracks have to be obtained first. In case of the OI tracking there are the COT tracks, for the Standalone tracking they have to be constructed from hits in the silicon.
For the OI tracking there are up to 200 seed tracks depending on the event type. These tracks have been already well established by the COT tracking. It is expected to find an OI track for each of them. For the Standalone Tracking there are up to 50.000 seed tracks and only a few of them are belonging to real tracks. The problem of the Silicon Standalone strategy is to distinguish between real and fake seed tracks.

4.3.1 Nominating Track Candidates

In order to construct a helix three points in r/ϕ and two points in r/z are required. The track candidates of the Standalone tracking are calculated out of two 3D hits in two different SAS layers and the beam position $(\mathrm{x}_0, \mathrm{y}_0)$[4]. By using the beam position to calculate the reference of the track candidate, the Silicon Standalone strategy is limited to tracks originated inside the beam pipe. It should be stressed that the beam position is not incorporated into the fit as measurement. But due to the linear transport model used for the Kalman fitter it is only possible to find tracks within a a several milimeter wide road around the reference.
The Silicon Standalone tracking is planned to run on the remaining unused hits after the OI tracking has been performed. Thus the main issue of the Silicon Standalone strategy is to find tracks in the forward region, where four SAS layers are located. Two out of four 3D hits are required to be available.
In order to reduce the amount of couples of 3D hits, which later form track candidates, additional cuts are applied. The two 3D hits have to be consistent in ϕ within $\Delta\phi = \pm\frac{\pi}{20}$. This corresponds to a weak cut on the transverse momentum of smaller than 0.1 GeV, to exclude curling low momentum tracks.
The straight line in r/z constructed from the two 3D hits has to be consistent with one of the primary vertices (PV) found by the pre-tracking primary vertex finder within $\pm$ 5 cm. The pre-tracking PV finder is also developed inside the TrackingKal package. It is documented in reference [32]. Therefore the efficiency of the Silicon Standalone Strategy is directly correlated to the efficiency of the pre-tracking PV finder. Every inefficiency is inherited by the tracking. The influence of efficiency, fake rate and timing performance of the PV finder on the Standalone strategy will be discussed in the next chapter.

[4] The precise formulae for calculating the parameters of the reference are derived in appendix C

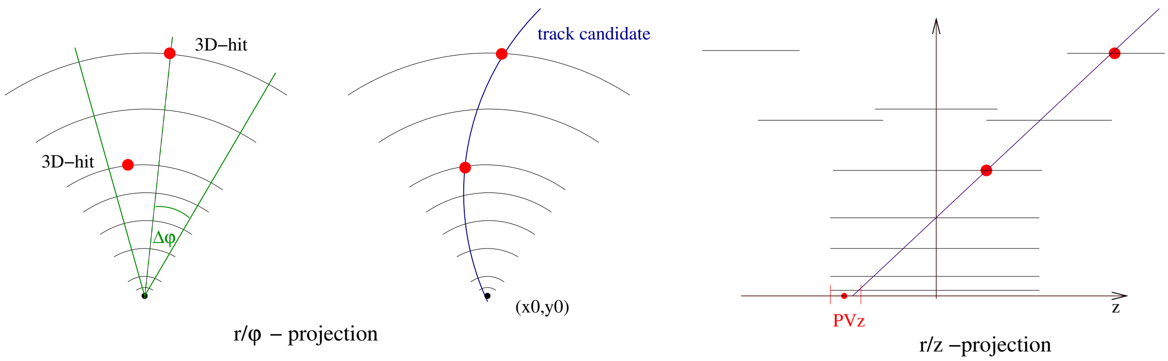


Figure 4.2: Nomination and creation of track candidates for Silicon Standalone tracking.

Once the helix parameters of the reference are calculated a stricter cut on the z-consistency with the PV z positions is applied: $|\Delta z| \leq 0.8$ cm.
In order to avoid ghost tracks only couples of unused 3D hits (ϕ and SAS hits both unused) are considered. The above described procedure of nominating and creating tracks is shown in figure 4.2.

4.3.2 Ranking of the Track Candidates

The pattern recognition step similar to the OI tracking is performed for one track candidate after the other. Once a track is accepted, its hits are flagged as used. Every other doublet of 3D hits which consist of one of those hits is automatically erased from the candidate list. One can think about testing first all combinations and then choose the best ones. But this is too CPU consuming. Thus the Silicon Standalone Tracking heavily depends on the order in which the track candidates are processed. The list of track candidates is sorted, so first the candidates which are trusted most are processend and then the more critical ones.
There are two ranking criteria considered. The first one is the transverse momentum of the candidate. The full helix is not calculated for every couple of two 3D hits when it is nominated, because this is a quite CPU consuming calculation. A lot of the candidates are already erased from the list before they are used for the pattern recognition. Thus the transverse momentum is first approximatively calculated in advance for every doublet of 3D hits[5].
The second ranking criterion is the combination of layers, in which the two 3D hits are located. There are preferred combinations of SAS layers for the candidates, which is related to the resolution of the reference. The wafers which are considered for the

[5] The according formulae are given in appendix C.

search of hits during the pattern recognition are obtained by looking in a certain window around the reference. The z precision of the reference is quite low. The 3D hits have a z resolution of about 2.5 mm in the ISL and 1.5 mm in the SVX II. For candidates constructed out of 3D hits in LAYER 2 and LAYER 4 (both in the SVX II), the z resolution is higher especially in the innermost layers compared to a candidate constructed out of two 3D hits in the ISL. If the z position of the candidate is more reliable, the z window size for the search of the active wafers, can be reduced. The r/ϕ search can thus be restricted to one wafer per layer instead of two wafers with the same ϕ range but different z ranges. The combinatorics in the dense inner layers is reduced, which results in lower wrong hit usage and lower fake rate.
So first all candidates of 3D hits in LAYER 2 and LAYER 4 are used for the pattern recognition starting with the high transverse momentum candidates. Then the candidates of LAYER 2 and LAYER 5 and LAYER 4 and LAYER 5 and in the end all combinations including LAYER 6 are processed.
Several loops over each candidate list are performed. In the first loop, candidates which have found hits in at least 80 % of their active layers are accepted, in the second loop 60 % are required and in the last loop 40 %. The idea behind this is again the same: We first want to reconstruct the most reliable tracks.
Picking up wrong hits affects one track in the OI algorithm. In the Silicon Standalone algorithm, this can also cause inefficiencies for other tracks. Therefore the ranking of the candidates is essential.
It is also recommended to perform the Silicon Standalone tracking in the whole range of transverse momentum greater than 0.2 GeV, even if we are not interested in low energetic tracks. But 3D hits eliminated by reasonable low momentum tracks reduce the fake rate of the high momentum tracks, too.

4.3.3 Forward Fit in the ISL

The candidates which have been chosen to be processed by the Standalone tracking consist of a reference only with no measurement taken. Thus there is no reliable weight matrix available before the first three ϕ measurements are incorporated to the fit. Therefore the road search windows are fixed windows until the covariance matrix is fully determined. For the hits in the ISL, the fitter can not distinguish between the correct hits and wrong ones, because χ^2 or any other quantity to cut on are not yet available. Due to the large extrapolation distances from the ISL in the SVX II and the relatively thick material contribution from the port card, the influence of the hits in the ISL to the final χ^2 of the fit is very small. This causes a relatively high wrong hit usage in the ISL.
Therefore a cross check of the ISL hits is implemented. Once a r/ϕ clone is chosen, its ISL hits are eliminated and a forward fit incorporating the remaining SVX II hits is performed. A hit search from inside to the outside is processed in the ISL. This procedure reduces the wrong hit usage in the ISL significantly.

4.4 Tuning Parameters

There are several parameters in the strategies such as road search window sizes, cut on χ^2 and the minimum number of required hits. The choice of the value of those parameters heavily depends on the status and our understanding of the detector, the quality of the alignment and the correct estimate for the resolution of the measurements.
The current default values of the tuning parameters are chosen in order to get an optimal result on data. Every improvement of the understanding of the detector will require a new tuning.
The name of the variables of the tuning parameters used in the code of the strategies are written in typewriter style. The interfaces for setting the tuning parameters are described in appendix D.

4.4.1 Tuning of the OI Tracking

Road Search Window Sizes

The size of the road search window is correlated to the precision of the best estimate of the fitter. In order to get an idea of this size, the distance in r/ϕ of the best estimate of an OI track to its associated hit is plotted in figure 4.3 for the different silicon layers. The fit includes all silicon hits of the track in more outer ward layers.
Due to the large radial coverage of the COT compared to the silicon the resolution of the curvature of the OI tracks is mostly dominated by the COT measurements. But the location and direction in r/ϕ space are more precisely measured in the silicon due to its better intrinsic hit resolution. Therefore at least two additional ϕ hits in the silicon are needed to obtain a good r/ϕ resolution. Most of the OI tracks do not intersect with LAYER 6, because it covers the forward regions only, while the COT tracks are reconstructed in the central region. Therefore the fit estimate has for the first time in LAYER 3 a good resolution. Going further inward, the resolution does not improve, because it is dominated by multiple scattering and by the intrinsic hit resolution.
The same plots divided by the according errors are show in figure 4.4. They are normalized in the inner silicon layers. But in the ISL, they are much too small. This is related to the incorrect error estimate of the COT tracks[6]. Once a hit is picked up the r/ϕ resolution is dominated by the silicon measurements.

[6]There is work going on to understand the COT covariance matrix, thus soon normalized COT pulls should be available.

In figure 4.5 and 4.6 the same quantities are shown for z. The z resolution of the COT and of the SAS layers in the silicon is relatively low, thus two $90°z$ hits are needed to obtain a good z resolution. The z errors are also overestimated by the COT. Therefore the pulls are also not normalized in the ISL but in the SVX II.

According to those plots road search windows of about 4σ seem to be reasonable. But the optimal road search window size depends, as already mentioned, on the understanding of the detector. The following road search windows have proven to work best on the current data[7]:
The road search window of the r/ϕ path is `phiSigmaWindow1` $= 8 \times \sigma$, if there are not at least two hits attached to the clone. This large window is chosen because the ISL is not yet aligned, and due to a small rotation between the COT and the silicon. If a clone does not pick up a hit in the ISL, there is almost no chance to pick up the correct hits in the SVX II.
The road search window decreases to `phiSigmaWindow2` $= 6 \times \sigma$, once two hits are attached to the clone. Once the clone has picked up at least five hits, or we are looking for hits in LAYER 00 the road search window is reduced to `phiSigmaWindow3` $= 3 \times \sigma$. For the SAS and the $90°z$ path, the default road search windows are `zSigmaWindow` $=$ `sasSigmaWindow` $= 4 \times \sigma$. So far there is no z alignment at all, and therefore no reasonable tuning of the SAS and z path of the strategy. The default windows are currently the ones obtained from studying Monte Carlo samples.
For the hits added to the strategy, there is no additional χ^2 cut besides the one which is indirectly introduced by the choice of the road search windows.

Parameters for the Choice of the Best Clone

The clones have different number of hits and different χ^2 of their fits. Clones with more hits are preferred, if they have a reasonable χ^2.
The optimal way to define a reasonable χ^2 is the fit probability (In case of Gaussian error distribution the fit probability is called χ^2 probability). The fit probability is a function of χ^2_{fit} and number of dof of the fit. It gives the probability for a fit with a given dof and χ^2_{fit} that on average a fit with the same dof have a χ^2 of at least χ^2_{fit}. Consider for example a fit probability of 0.7 for a fit with a certain χ^2_{fit} and a certain dof: This means that on average 70 % of all fits with the same dof have a larger χ^2 and therefore a smaller fit probability than 0.7. Thus this function is by definition flatly distributed between 0 and 1.

[7]status of summer 2002

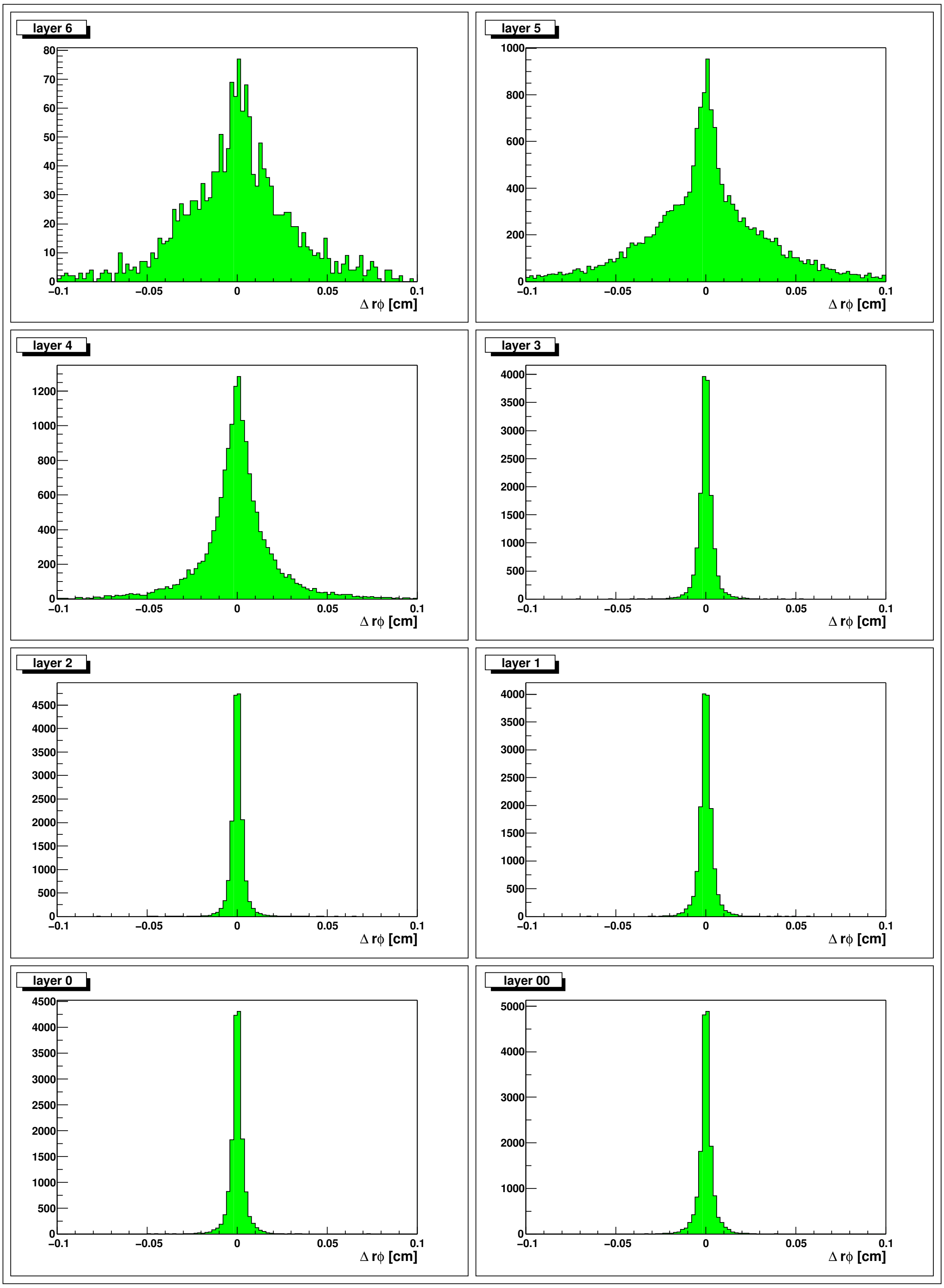


Figure 4.3: Distance of best estimate of the fit to the hit in r/ϕ for an OI track.

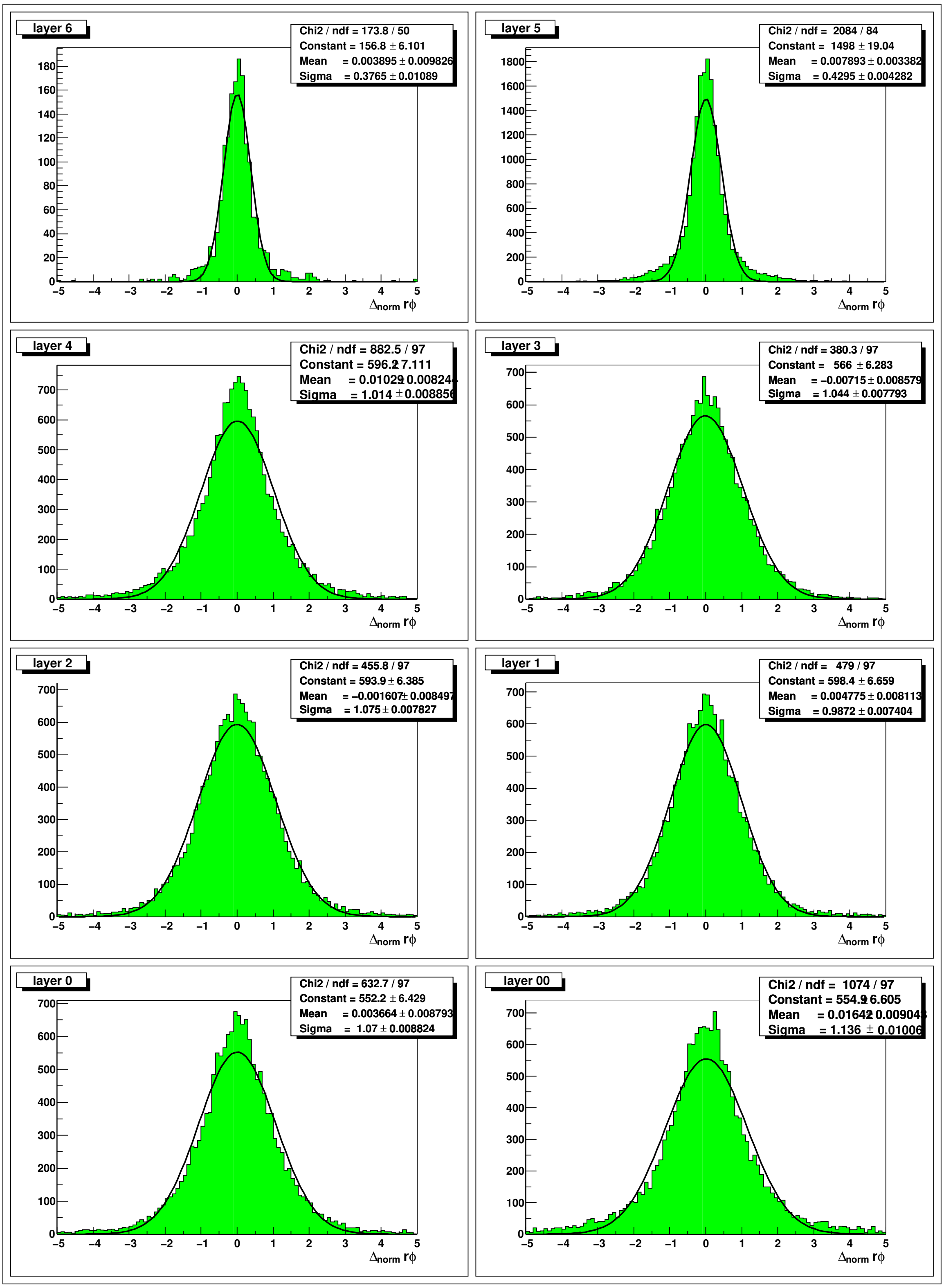


Figure 4.4: Distance of best estimate of the fit to the hit in r/ϕ for an OI track, normalized by the combined error of the hit and the fit.

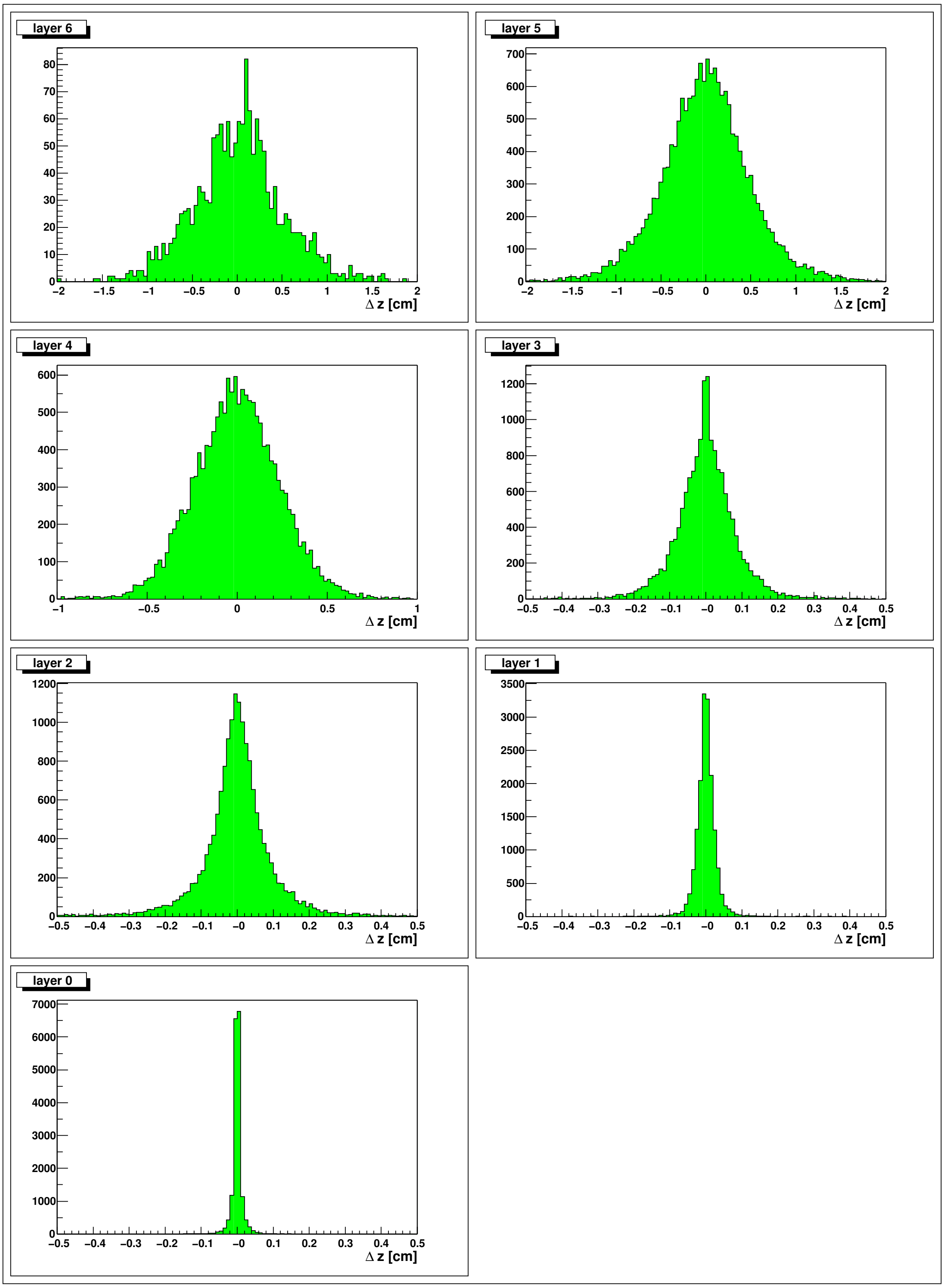


Figure 4.5: Distance of best estimate of the fit to the hit in z for an OI track.

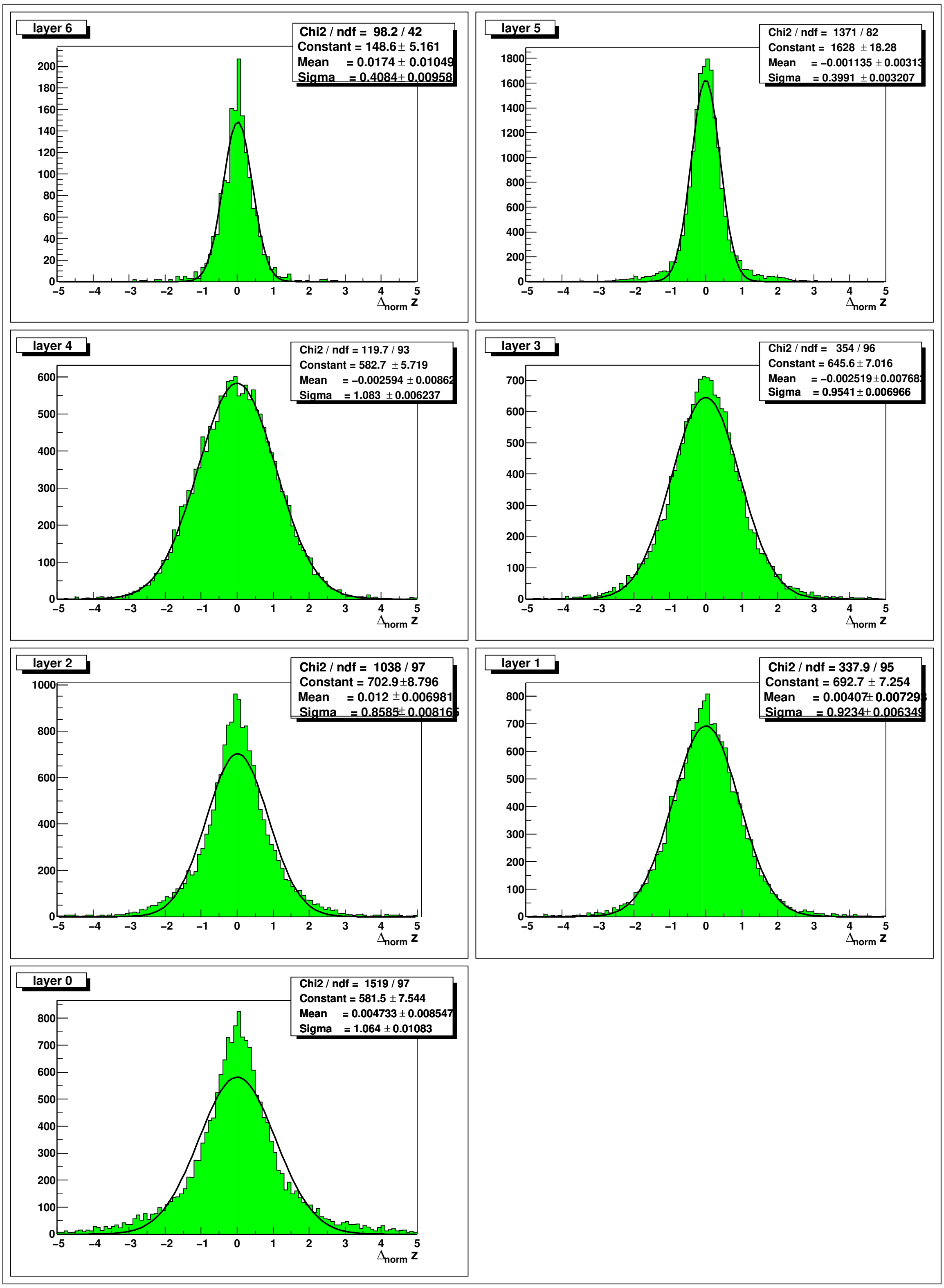


Figure 4.6: Distance of best estimate of the fit to the hit in z for an OI track, normalized by the combined error of the hit and the fit.

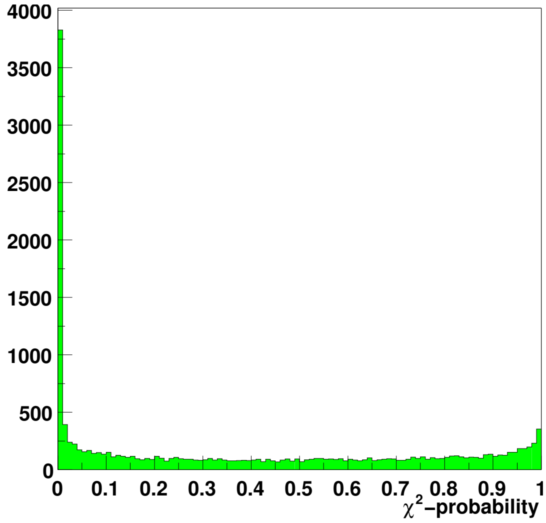


Figure 4.7: χ^2 probability distribution of perfect OI tracks.

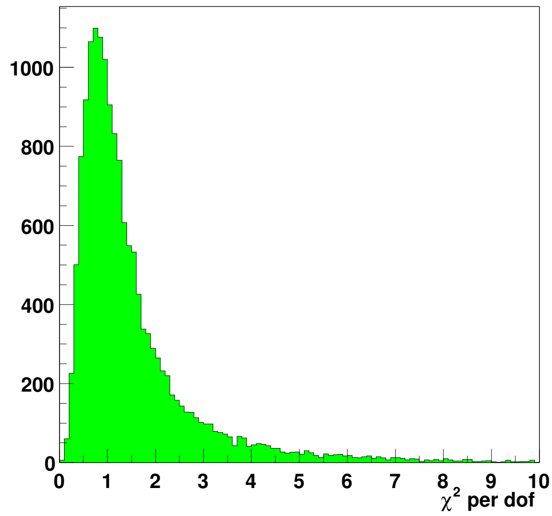


Figure 4.8: χ^2 per degree of freedom for perfect OI tracks.

The χ^2 probability function as it is implemented in the standard high energy physics libaries assumes Gaussian error distribution as input. This function is used to calculate the fit probability of OI tracks with correct hits attached to them (fig.: 4.7). It is very sensitive to non Gaussian input distributions especially for fits with a high number of dof. The tails from the multiple scattering in the pull distributions give rise to the large peak at a fit-probability of ≤ 1 %. It is not possible to use this χ^2 probability function. This is not related to the quality of the fitter, but just proves the non Gaussian error distributions of the hits and the scattering.
Therefore the fit probability has to be recalculated for the given error distributions. This can e.g. be done by introducing a cut on χ^2 per dof depending on the number of dof. The χ^2 per dof distribution is presented for all dof in figure 4.8.
The requirement of the χ^2 per dof in the r/ϕ path of the OI strategy is `chi2Cut1` $= 40$ for clones with 3 hits, `chi2cut2` $= 30$ for clones with 4 hits, `chi2cut3` $= 20$ for clones with 5 hits and `chi2cut4` $= 10$ for the clones with more than 5 hits.
For the SAS and $90°z$ path, there is no minimum hit requirement and the cut on the χ^2 per dof is set to `chi2cutz` $= 40$.

4.4.2 Tuning of the Silicon Standalone Tracking

For the Silicon Standalone tracking, there are first the cuts on the consistency with one of the primary vertices. The calculated straight line out of the both 3D hits has to

be consistent within PVz1 $= 5$ cm. The precise calculated z_0 position of the helix has to be consistent with one of the primary vertices within PVz2 $= 0.8$ cm. The resolution of the vertices found by the pre-tracking PV finder is 1 mm. The z_0 resolution of the track candidates is about 2.5 mm. A cut on 0.8 cm correspond to $3 \times$ the combined standard deviation of both resolutions.
Till the covariance matrix is not fully determined the r/ϕ window is limited to phiWindow $= 0.05$ cm and the z window to zWindow $= 2$ cm. These windows seem to be quite small. But we know that the reference is build out of two 3D hits, even if they are in more inward layers and not yet incorporated to the fit. This window sizes have proven to work the best on Monte Carlo and on data.
Once the covariance matrix is fully determined, the road search windows for the hit search are sigmaPhiWindow = sigmaStereoWindow = sigmaZWindow $= 6 \times \sigma$.
In the Silicon Standalone tracking, the SAS and the r/ϕ hits are picked up together. It does not make any sense to implement a three loop strategy as for the OI tracking. The correlation between the r/ϕ and z space is anyhow present, due to the way of creating the track candidates.
In order to accept a clone after the ϕ/SAS path at least minPhiHits $= 4$ are required. For candidates intersecting with LAYER 6, minPhiHits $+ 1$ hits are required. Asking for 4 to 5 ϕ hits is a quite low criteria keeping in mind that 3 ϕ hits are required to determine a helix. For Monte Carlo studies a requirement of 5 to 6 hits has been proven to yield the same efficiency but a lower fake rate and a higher purity. But on data this number of hits has shown to be too restrictive, especially on the pre shutdown data (june 2002) without the ISL LAYER 5.
In addition the best clone has to have a χ^2 per dof smaller than chi2pdof $= 20$. The same cut on the χ^2 per dof is applied for the choice of the best clone of the forward search in the ISL and the $90°z$ path. In both cases, no minimum number of hits is required.

4.5 Implementation Issues

4.5.1 Clones are Smart Tracks

In the sense of programming a track is a quite sizeable object. In case of an OI track it consists first of the weight matrix of the COT and the COT parameters, namely 15 double precision variables and 5 double precision parameters. Additionally, there are for each intersection with silicon layers the reference, the weight matrix and the parameters of the fit before and after incorporating the fit action of this layer. In

case of material layers, the material properties such as radiation length or minimal ionization energy and for measurement layers the five measurement parameters and their weight matrix have to be stored, too. The average number of intersections of a track with the different fit layers is about 40. This number is that high, because there are not only the 15 active sublayers in the silicon but also their material, the support structure, the port card, the COT inner wall and the beampipe to be taken into account in order to incorporate the multiple scattering and the dE/dx correctly into the fit.
Therefore cloning a track is copying a lot of information. If this is done the dominant part of the time used for the algorithms will be spent by copying the tracks. This is especially true for the Standalone algorithm, where the road search windows are larger as for the OI tracking, due to the missing COT information. The number of clones increases exponentially with the number of active layers and so does the number of times a track has to be copied.

The Kalman fitter is a progressive fitter. All measurements incorporated in the fit are no longer needed explicitly. This information is implicitly stored in the best estimate of the fit. Therefore there is no need in copying all this information for every clone. Just the current status of the fit and a list of hits picked up are needed in order to reproduce the track once the best clone has been chosen.
So the clones used in the TrackingKal Strategy are no real tracks but so called smarties (smart tracks). There is one real track per candidate, which keeps the reference and the information about the different fit layers. These are the parameteres of the reference at each layer and the material properties of the material layers. Every time a hit is added to a clone, the current fit status of the clone is copied to this one track, the fit action is performed and the new status of the fit is copied back to the clone. In addition, the hit is stored in the hit list of the clone. Then the fit status of the next clone is copied to the track and the hit search is performed.
There are about 100 clones per track candidate. They all pick up hits in one layer, then they are all extrapolated to the next layer, ... Therefore they are existing in parallel. By introducing the smart tracks the memory consumption of the strategy is reduced from the size of about 100 tracks to the size of one track and about 100 smarties. In addition the time spend for copying information is reduced by the same factor. This shortcut is only possible by using a progressive fitter. And it is absolutely necessary in order to perform the strategies within a realistic time and memory scale.
If the only interesting result of the tracking strategies are the optimal fitted perigee parameters of the reconstructed tracks, the final track does not have to be refitted but can just use the final fit status of the corresponding smartie. The intermediate fit results at every layer can be recalculated easily, once they are needed for alignment or validation purposes. The CPU time for this computation is by orders of magnitudes smaller than the gain in CPU time by not copying this information for every clone.

In order to minimize the memory, which has to be copied for every clone, there are different smarties for the different pattern recognition passes. E.g. a smartie in the $90°z$ does not need to copy the list of 3D hits because it will not pick up any. Instead it has to know the list of the multiplexing indices for the picked up $90°z$ hits which is not of interest in the r/ϕ path.

4.5.2 Material List Provided by the Reference

One of the most time consuming part of the strategy is the search for the next fit layer, either active of just passive material. All clones for one candidate differ only slightly from the seed track. So the list of material the track is intersecting with is computed only once for the reference of the seed track. All material, which is within a 2 cm window in z and a 0.15 rad window in ϕ around the reference, is taken into account for the fit of each clone. If there are two wafers in one layer in the window, both are taken into account for the hit search in the pattern recognition, but certainly only one of them is used for incorporating the material. Two wafers per layer in a window can happen in z only due to the small gaps between the half ladders. Due to the shifted radii of the odd and the even ϕ wedges such ambiguities never happen in r/ϕ.
Those window are large enough to guarantee not to miss an active layer due to deviations between a clone and its seed track. But in some cases, material with which the clone does not intersect, is incorporated in the fit. Our studies have shown that these are negligibly small edge effects.

The fitter calculates the weighted mean of its best estimate and of the hit position at a given radius. But in r/ϕ view the layers do not have the shape of a piece of a circle but of a box. A small deviation of the clone and the picked up hit from its reference in r/ϕ results in a slightly different radius. This effect is corrected for in the strategies. For incorporating a measurement to the fitter the reference is set to the radius, at which the hit is taken. This is possible for ϕ and 3D hits only. The $90°z$ hits do not have the information of their precise radius. If there is a ϕ measurement on the same wafer, this radius is used, otherwise the radius of the reference is the available one at this layer.
As already mentioned the search for the next active or passive layer is quite a time consuming procedure. For the Silicon Standalone algorithm, there are several passes over the seed tracks. First a hit is required in 80 % of the passed active layers, later this criterion is lowered. In the first pass, the seed tracks are most of the time already rejected after one or two layers without a hit. Thus there is no need to calculate all

the intersections of the seed track with all layers but just with the next one step by step. If the seed track is rejected in between a lot of computation have been saved.
In the second pass, the already computed intersections are reused, and again only those intersections which are really needed are calculated. This procedure makes the code a little bit more complicated but saves a lot of CPU time. A temporal memory usage of up to 200 MB can be necessary in very rare events on data. This is due to the fact, that the already calculated intersections with the different active and passive layers are stored for every seed track over several passes. But the memory is free to use after these large events are processed. CPU time is paid by memory space.

4.5.3 Limit and Clean up of the Clone List

If there would be no limit on the size of the list of clones, the strategies risk to explode in memory space and time consumption, once there is an unreasonable noisy event with a lot of noisy wafers in the data. The maximum size of the clone list is chosen such that no reasonable physical event reaches it (2,000 clones for the r/ϕ path, and 200 for the SAS and the z path).
Besides the extreme case of an exploding clone list, for sure the number of clones, which has to be checked and treated is responsible for the time spent in the pattern recognition. Due to the fact, that a clone without a hit in a layer is kept as well, the spectrum of number of hits used for the clones is quite huge in the clone list. A lot of them can be already identified during the pattern recognition part as unsuccessful in the sense that they will not be chosen as the best clone in the end. So the clone list is cleaned up after every layer. Only the ones with the most hits (n_{max}) picked up and the ones with $n_{max} - 1$ are kept. In case of a small list of clones (≤ 10) and not well determined tracks ($n_{max} \leq 2$), the candidates with $n_{max} - 2$ hits are also kept. This cleaning up of the clone list does not affect at all the efficiency and purity neither on MC nor on data, but it gains a factor of about two in the timing performance for both OI and Standalone tracking.

Chapter 5

Validation of the Silicon Tracking

There are several quantities which characterize the quality of a tracking algorithm. Those are efficiency, purity, resolution and timing and memory performance.
Efficiency measures the fraction of reconstructed and reconstructible tracks. It therefore heavily depends on the definition of reconstructed and reconstructible. But anyhow, it is the quantity which relates the number of physical interesting events taking place in the detector and the rate of corresponding reconstructed signals.
Purity characterizes the reliability of the out-coming tracks of a strategy. The resolution of the track parameters is related to the intrinsic resolution of the hits in the silicon and to the quality of the fitter. But it is also determined by the number of hits picked up during pattern recognition. Purity and resolution are both important to perform separation between signal and background, e.g. for the reconstruction of displaced vertices.
A reasonable timing and memory performance are essential for running the algorithms e.g. at trigger level and in production. The best algorithm is useless if it is not able to process the incoming data in almost real time.
Every algorithm is a trade-off between those four quantities.
This chapter presents the validation of the OI and the Standalone algorithm of the TrackingKal package on Monte Carlo and a first look on their performance on data is given.
Tools for comparison of the performance of the TrackingKal and the RUN I like OI algorithm have been developed by the tracking group and are documented in reference [33]. The results of this study are summarized in reference [34].

5.1 Validation on Monte Carlo Simulation

The results presented in this section are all obtained by studying two different Monte Carlo samples produced in the integration release 4.6.2 of the CDF II offline software: a $t\bar{t}$ sample with 3 Poisson distributed underlying minimum bias (mbr) events as expected for the luminosity of RUN IIA and a $b\bar{b}$ sample with the same number of minimum bias events. The $t\bar{t}$ samples contains 500 events, the $b\bar{b}$ sample contains 5,000 events. The total number of tracks in 500 $t\bar{t}$ and 5,000 $b\bar{b}$ is comparable and sufficiently high to provide significant results concerning efficiency and purity studies. Reasonable physics events similar in population to a $t\bar{t}$ event in data are quite rare (about 10,000 are expected in RUN IIA). They are examined as a kind of worst case study. Most of the events have a $b\bar{b}$ like event structure.
The validation is done for the strategy using the default values of tuning parameters. They are not optimal for MCs but provide a realistic estimate for the performance on data. The used MC are simulated with the geometric drift model and later hit smearing described in section 2.5.

5.1.1 Efficiency Studies

Efficiency of the Outside-In Tracking

For this study so-called perfect COT tracks are used as input. Perfect COT tracks are tracks which are not found by pattern recognition in the COT, but which are constructed by linking the correct hits with the help of the MC information. The basic idea of this is to make sure that the input tracks are reliable and all inefficiencies observed are due to the pattern recognition in the silicon itself. The same study with real COT tracks as input has been performed, and the results do not differ significantly.

Efficiency Denominator

In the denominator of the efficiency are all tracks which fulfill the following conditions:

- a perfect COT track is available with ≥ 70 (out of 96) COT hits,
- ≥ 6 silicon ϕ hits are available,
- $p_T \geq 0.5$ GeV.

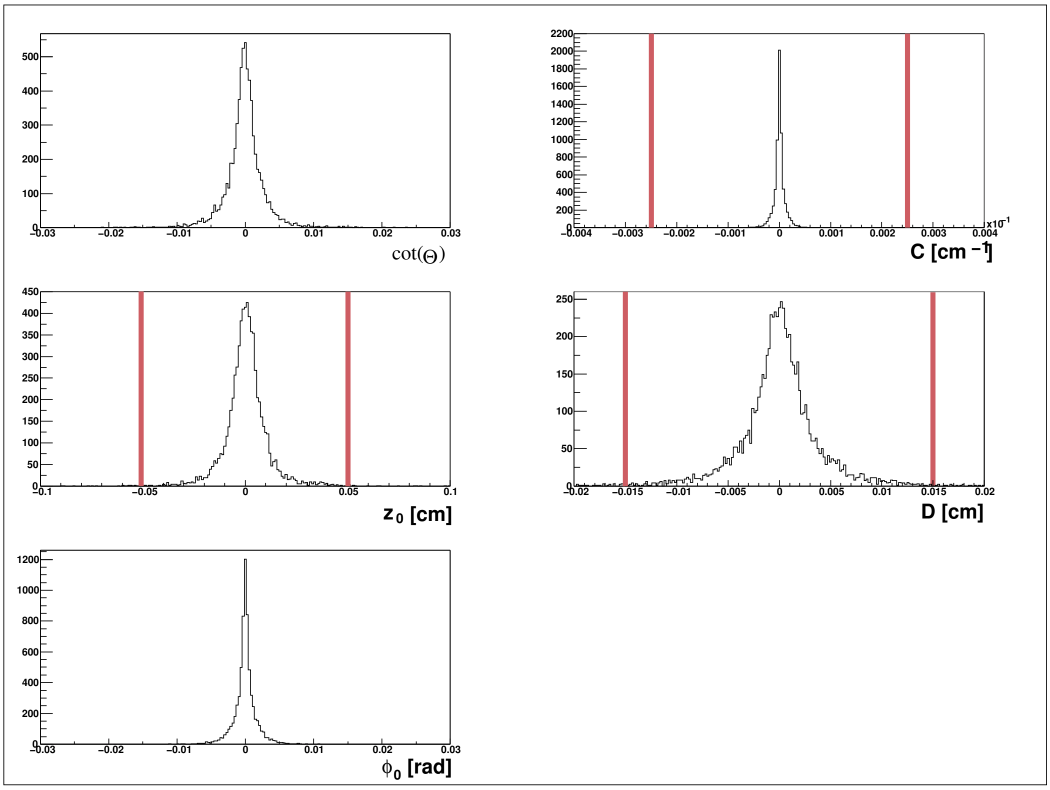


Figure 5.1: Residuals of the track parameters: perfect OI tracks - MC_{truth}. The red (grey) lines show the paramterer cuts for the definition of the efficiency numerator.

Efficiency Numerator

For the numerator the tracks are not only required to be reconstructed but also to be consistent with the MC parameters within some range. Figure 5.1 shows the residuals of the parameters of the perfect OI tracks[1] and the MC_{truth} parameters for the tracks which count for the denominator of the $t\bar{t}$ sample. This is a measure of the best resolution which can be obtained.
The cuts on the residuals (in perigee parametrization defined in section 2.3.1) chosen by the tracking group for the definition of the efficiency are:

- $\Delta C \leq 0.00025$ cm^{-1},
- $\Delta D \leq 0.015$ cm,
- $\Delta z_0 \leq 0.05$ cm.

[1] Perfect OI tracks are perfect COT tracks plus all available silicon hits associated to the particle.

Efficiency Result

type of event	no cuts	cuts on C and D	cuts on C, D and z_0
$t\bar{t} + 3$ mbr	98.2 %	92.1 %	86.2 %
$b\bar{b} + 3$ mbr	97.0 %	92.5 %	88.8 %

For both types of events more than 97 % of the denominator tracks have an associated OI track. The additional cuts on C, D and z_0 reject these tracks, which have either picked up wrong or not enough hits to obtain the required resolution. The remaining total efficiency is above 86 % for both samples.
Low energetic tracks are harder to reconstruct due to the larger uncertainties introduced by multiple scattering. The average transverse momentum for $b\bar{b}$ is lower than for $t\bar{t}$ events. Therefore the total rate of found tracks without any quality cut in $b\bar{b}$ events is lower than in $t\bar{t}$ events.
The decrease of efficiency for applying the cuts on the parameters is larger for the more populated $t\bar{t}$ events. This arises from the fact that in a dense environment the rate of wrongly picked up hits is higher. Wrongly picked up hits, at least in the innermost layers, influence the fitted track parameters such that they end up in the tails of the residuals.
The efficiency distribution depending on the transverse momenta is shown in figure 5.2. The efficiency is lower for low momentum tracks and converges against a quite high level for higher transverse momenta. Most of the tracks in physical events are low energetic, thus the total efficiency suffers by the low efficiency in the low energy region.

Wrong Hit Usage of the Efficiency Denominator Tracks

Figure 5.3 shows the fraction of wrong hits on reconstructed denominator tracks (In this plot the numbering of the layers ranges from 0-7 instead of 00-6). This can not be considered as purity measurement of the strategy. The minimum of wrong ϕ hit usage is in LAYER 3 (4). There the resolution of the tracks is already quite good and the hit density no yet too high. The wrong hit usage increases towards LAYER 00 (0). The dense environment of the $t\bar{t}$ sample is much more challenging than the relatively low populated $b\bar{b}$ sample. The SAS hits are most of the time picked up correctly due to their strong correlation to the ϕ measurement. The wrong hit usage in the $90°z$ layers is quite high especially for the $t\bar{t}$ events. This is related to the multiplexing, already mentioned in section 2.1.3. The multiplexing introduces additional two to four fold ambiguities for the hits in the $90°z$ layers. Before the first $90°z$ hit is picked up, the resolution in z of the fit estimate is determined by the relatively weak COT and the SAS z measurements only. The hit density amplified by the multiplexing in the $90°z$ layers is too high compared to the resolution of the best estimate of the fit.

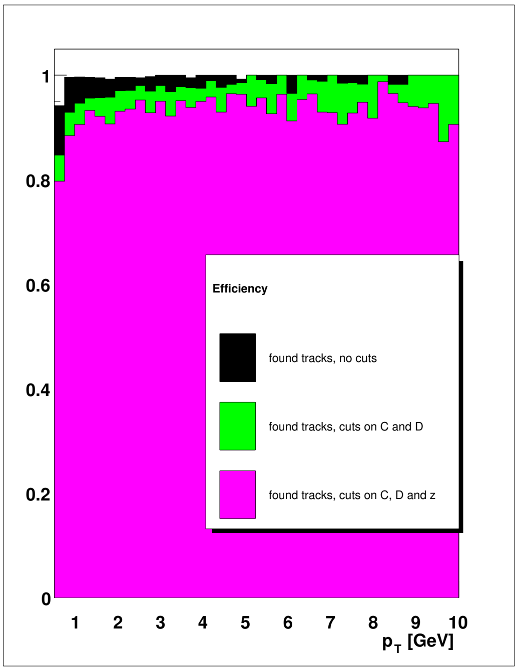


Figure 5.2: Efficiency in dependence of the transverse momentum of the track for $t\bar{t}+$ 3 mbr events.

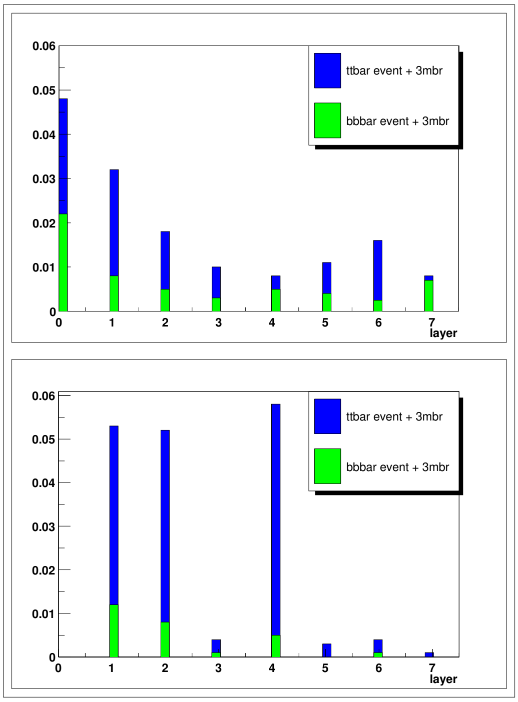


Figure 5.3: Wrong hit usage for denominator tracks of $t\bar{t}$ + 3 mbr and $b\bar{b}$ + 3 mbr events for ϕ layers (upper plot) and z/SAS layers (lower plot).

Efficiency of the Silicon Standalone Tracking

Efficiency Denominator

The reconstruction possibility of the Standalone Tracking is limited to tracks which are consistent with a reconstructed primary vertex within ± 1 cm. In addition, the tracks are supposed to come from the beam line and at least 2 3D hits are necessary to create a track candidate. Therefore the following requirements are added to the definition of the denominator for the Standalone tracking:

- $\geq$ 2 3D hits in different SAS layers,
- ≥ 6 ϕ hits in the silicon,
- consistent in z with one of the reconstructed primary vertices within ± 1 cm,
- originated from inside the beampipe (D $\leq$ 1.2 cm),
- and $p_T \geq 0.5$ GeV.

There are two validation modes for the Silicon Standalone tracking. The first one is running on the whole combinatorics. This is a cross check of the principle of the strategy and also the only possibility to measure the efficiency of the Standalone tracking on data later. The standard mode is to be processed on the remaining hits after OI tracking. For this second mode, there is an additional denominator cut:

- not yet found by OI tracking.

Efficiency Numerator

Tracks are associated to a MC particle, if at least 70 % of the hits of the track are associated to this particle. The other tracks are considered to be fakes.
As for the OI tracking there are additional cuts on the residuals of the reconstructed parameters and the MC_{truth}. They are not as strict as for the OI tracking, due to the worse resolution of standalone tracks. They are:

- $\Delta D \leq 0.03$ cm,
- $\Delta z_0 \leq 0.1$ cm.

Efficiency Result

The tracks coming out of the Silicon Standalone algorithm are divided into two quality classes, which yield different results in efficiency and fake rates. Tracks, which have at least 6 ϕ in case of intersection with LAYER 6, otherwise at least 5 ϕ hits, are of quality one. Quality two tracks are low quality tracks, which have only 4-5 ϕ hits. The qualities are written out and can be used to classify the tracks in an analysis later. This information is not redundant with the number of ϕ hits on the tracks, because this can be reduced by later clean up of suspicious hits, as it will be described later.
The efficiencies obtained for the different running modes and quality criteria are shown in tables 5.1-5.4.
As for the OI tracking the $t\bar{t}$ sample has a lower efficiency and a higher fake rate. Its efficiency suffers also more from the cut on the parameters, especially on z_0. The dense environment is entered with less information than in case of the OI tracking due to the missing COT information.
The efficiency for the quality 1 tracks are almost the same as for quality 1+2 tracks together. But the fake track rate for quality 2 tracks is much higher.
Figures 5.4-5.7 show the total number of reconstructed tracks of the different qualities for both modes. The tracks shown are all tracks coming out of the strategies, which must be associated to MC particles. But even in the total number of tracks, there is no significant gain by taking quality 2 tracks into account. This is different from

running on data as can be seen later.
Running on the full combinatorics, the Standalone tracking is almost as efficient as the OI tracking at least for the $b\bar{b}$ events. Its fake rate is lower compared to mode two. This is related to the fact that the tracks in the central region have a better resolution especially in z compared to high $|\eta|$ tracks in the forward regions. It is more challenging to reconstruct tracks in the forward regions.
Running after the OI tracking, the Standalone tracking does not only pick up tracks in the forward region but also some left over tracks in the central region, which either the COT or the OI tracking missed. The efficiency is flatly distributed in η for both modes and the reconstruction ability of the Standalone tracking in the vertex detector is up to $|\eta| \leq 2.5$.

type of event	no cuts	cut on D	cuts on D and z_0	fakes
$t\bar{t}$ + 3 mbr	83.8 %	78.6 %	65.8 %	14.6 %
$b\bar{b}$ + 3 mbr	92.2 %	88.8 %	80.2 %	3.0 %

Table 5.1: Efficiency of Silicon Standalone tracking, quality 1, running in mode two.

type of event	no cuts	cut on D	cuts on D and z_0	fakes
$t\bar{t}$ + 3 mbr	84.2 %	78.6 %	66.0 %	23.0 %
$b\bar{b}$ + 3 mbr	93.5 %	89.5 %	80.7 %	8.6 %

Table 5.2: Efficiency of Silicon Standalone tracking, quality 1+2, running in mode two.

type of event	no cuts	cut on D	cuts on D and z_0	fakes
$t\bar{t}$ + 3 mbr	87.1 %	82.8 %	70.7 %	8.2 %
$b\bar{b}$ + 3 mbr	95.3 %	93.5 %	87.5 %	1.4 %

Table 5.3: Efficiency of Silicon Standalone tracking, quality 1, running in mode one.

type of event	no cuts	cut on D	cuts on D and z_0	fakes
$t\bar{t}$ + 3 mbr	87.3 %	82.9 %	70.8 %	11.6 %
$b\bar{b}$ + 3 mbr	95.8 %	93.8 %	87.7 %	3.8 %

Table 5.4: Efficiency of Silicon Standalone tracking, quality 1+2, running in mode one.

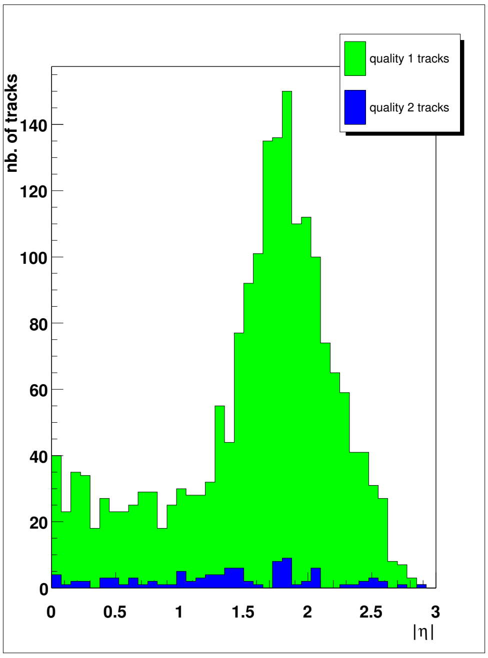


Figure 5.4: Total number of tracks dependent on $|\eta|$ for 500 $b\bar{b}$ + 3 mbr events for Standalone tracking in mode two.

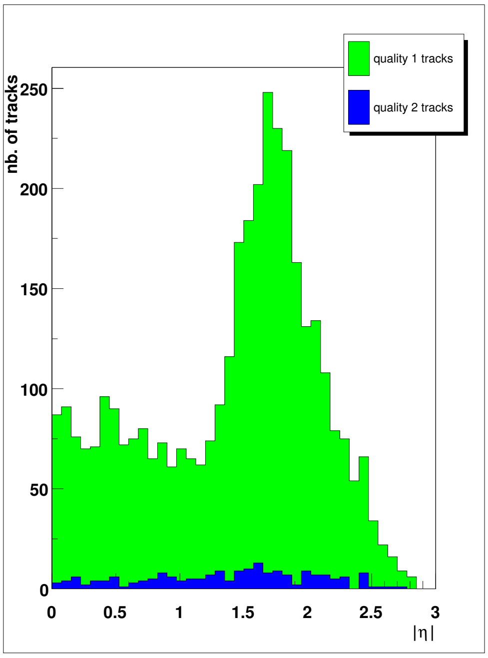


Figure 5.5: Total number of tracks dependent on $|\eta|$ for 500 $t\bar{t}$ + 3 mbr events for Standalone tracking in mode two.

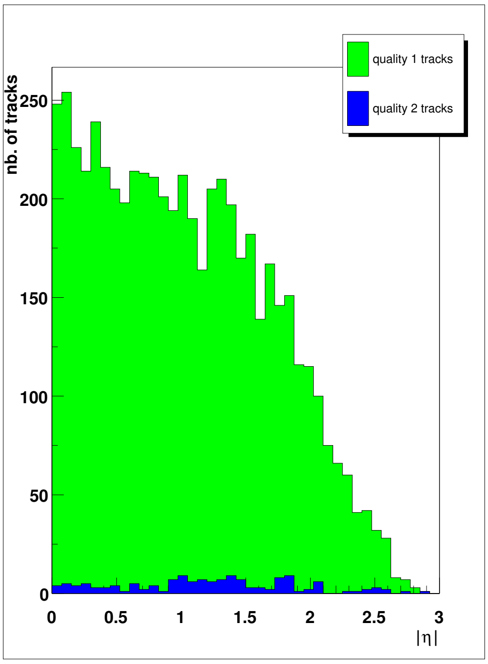


Figure 5.6: Total number of tracks dependent on $|\eta|$ for 500 $b\bar{b}$ + 3 mbr events for Standalone tracking in mode one.

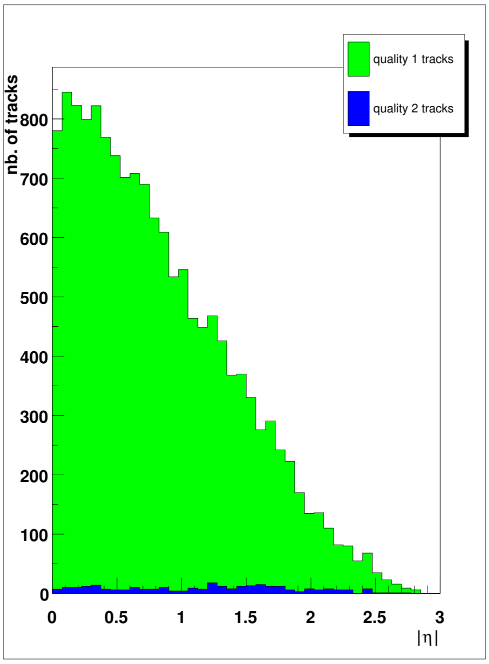


Figure 5.7: Total number of tracks dependent on $|\eta|$ for 500 $t\bar{t}$ + 3 mbr events for Standalone tracking in mode one.

5.1.2 Average Hit Usage

Tracks in the central region of the vertex detector intersect on average with three SAS layers, three $90°z$ layers and the single side ϕ layer LAYER 00. Therefore they are expected to have 13 silicon hits on average: 7 ϕ hits, 3 $90°z$ hits and 3 SAS hits. This corresponds to the average hit usage of the reconstructed OI tracks (fig.: 5.8). The tails towards higher hit usage arises from tracks passing through overlap regions. The tails toward lower hit usage indicate either gaps in the geometry or refused correct hits in the pattern recognition.
The hit usage of the tracks found by the Standalone algorithm for running on the whole combinatorics is similar to the hit usage of the OI tracking. It can be seen in figure 5.9.
The average hit usage of tracks running Standalone after the OI tracking is a combination of several effects. There are the tracks, which point in forward direction and pass the ISL. Most of them have about 15 hits. There are also the tracks in the central region with about 13 or 11 hits on average (depending if their ISL hit has been accidently already marked by the OI tracking) and those in the very high $|\eta|$ regions which do not intersect with the ISL at all and thus have about 11 hits in average. These three peaks can be seen in the distribution of the hit usage in fig. 5.10. This also explains the different shapes in ϕ and SAS hit usage.

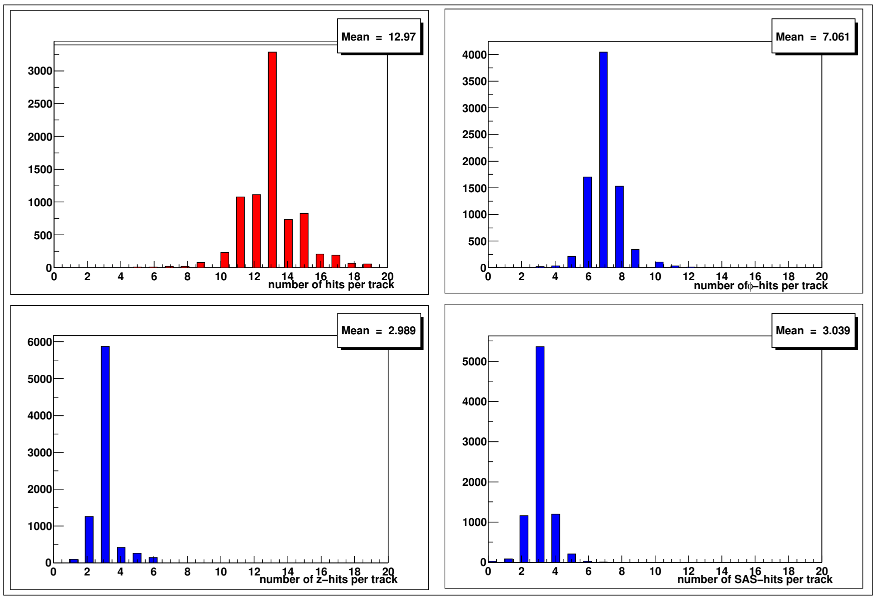


Figure 5.8: Hit usage of OI tracking for $t\bar{t} + 3$ mbr events.

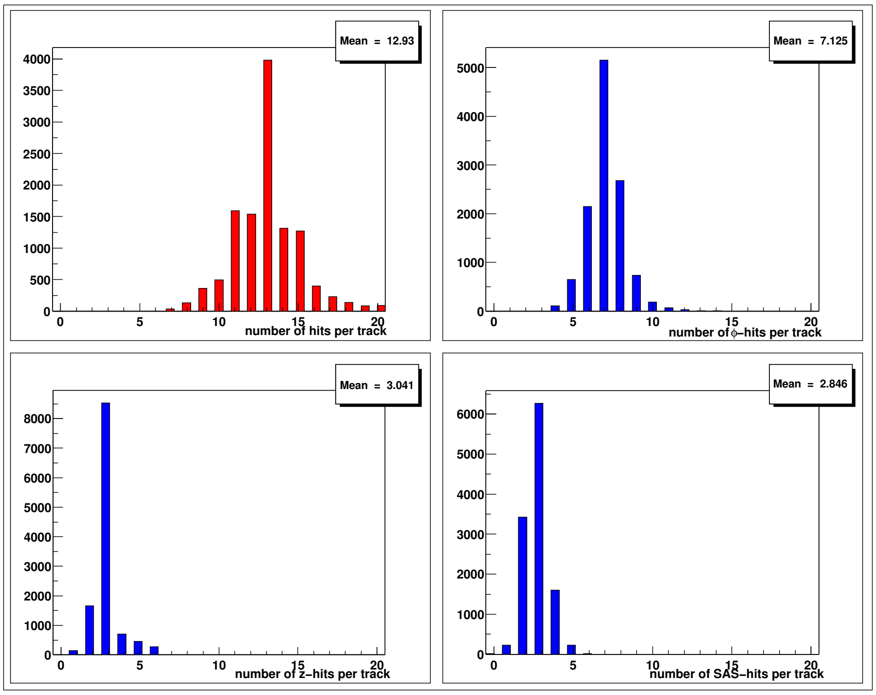


Figure 5.9: Hit usage of Standalone tracking mode one for $t\bar{t} + 3$ mbr events.

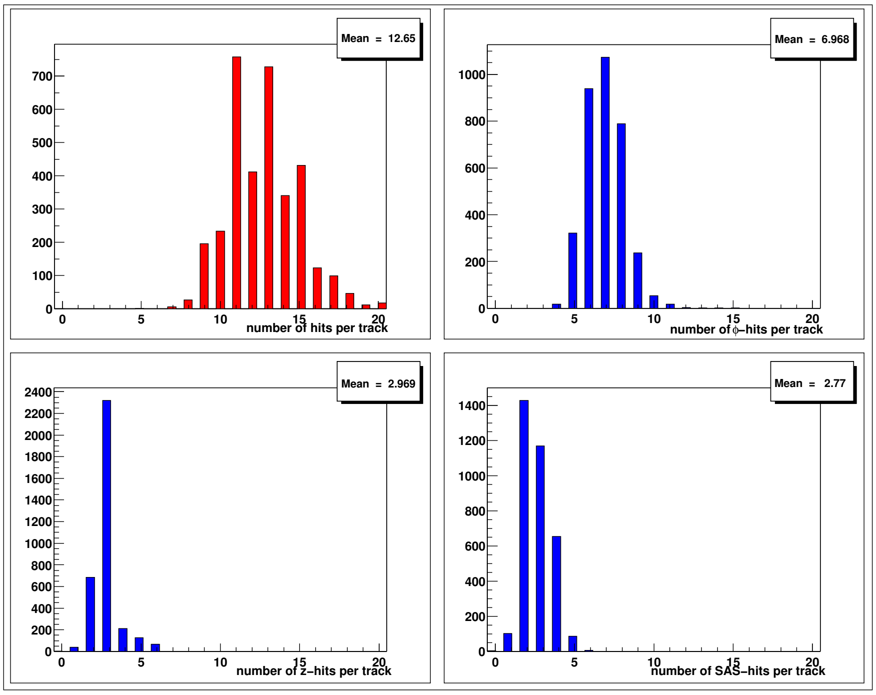


Figure 5.10: Hit usage of Standalone tracking mode two for $t\bar{t} + 3$ mbr events.

5.1.3 Purity Studies

Contrary to the efficiency, there is only one valid denominator for the purity: these are all tracks coming out of the tracking algorithms without any additional cut.
For the Standalone tracking, the denominator of the purity has to be reduced to those tracks which can be associated to a MC particle. These are all tracks which are not listed as fakes in the efficiency studies. Due to the fact that the OI tracks are matched by their parent COT track, there are by definition no fake tracks for the OI tracking. The wrong hit usage of all the tracks coming out of the OI strategy are shown in figure 5.11. The wrong hit usage has the same shape as the wrong hit usage of the denominator tracks shown in figure 5.3, but is much worse. This results from the fact that there are about 1.5 times more tracks coming out of the OI tracking, which are not in the denominator, this means, which do not have at least 6 ϕ hits in the silicon. Those tracks are passing through gaps in the geometry. This results in worse resolution and this is fatal in dense environments like e.g. $t\bar{t}$ + 3 underlying minimum bias events.
The same is valid for the wrong hit usage of the Standalone tracks (fig.: 5.12, 5.13).

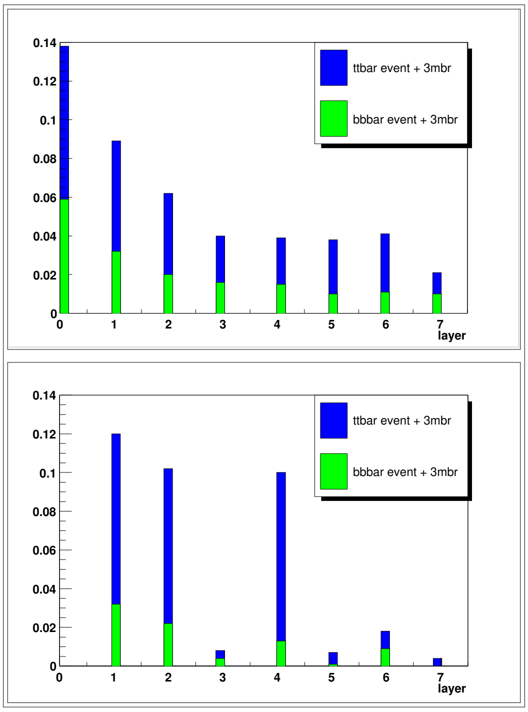


Figure 5.11: Wrong hit usage for all OI tracks of $t\bar{t} + 3$ mbr and $b\bar{b} + 3$ mbr events for ϕ layers (upper plot) and z/SAS layers (lower plot).

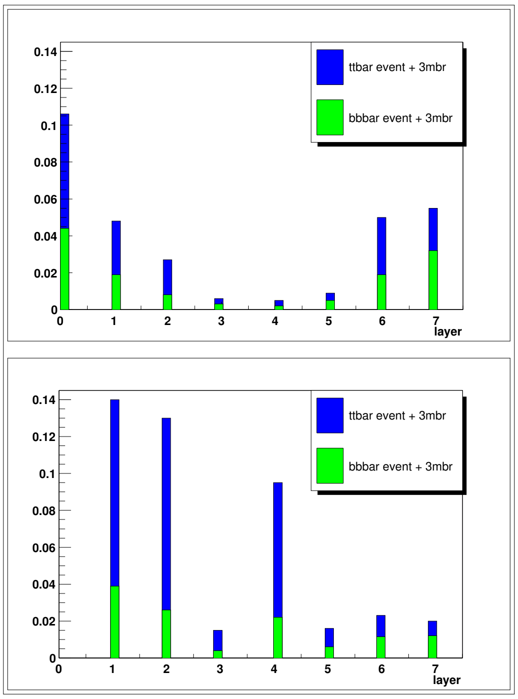


Figure 5.12: Wrong hit usage for all Standalone tracks (mode one) of $t\bar{t}+3$ mbr and $b\bar{b}+3$ mbr events for ϕ layers (upper plot) and z/SAS layers (lower plot).

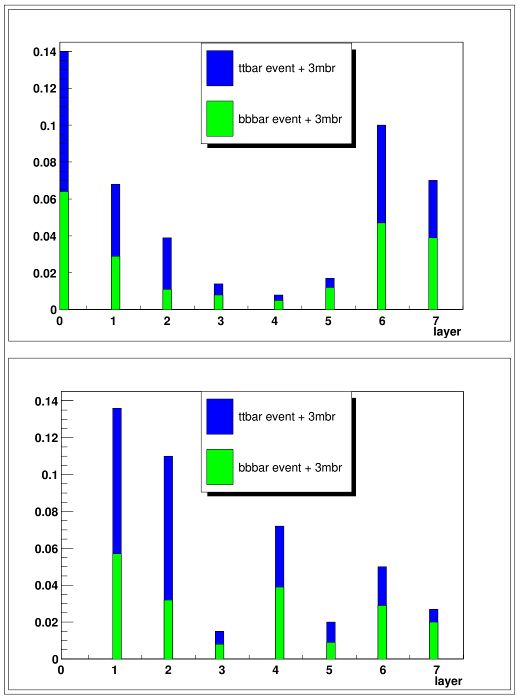


Figure 5.13: Wrong hit usage for all Standalone tracks (mode two) of $t\bar{t}+3$ mbr and $b\bar{b}+3$ mbr events for ϕ layers (upper plot) and z/SAS layers (lower plot).

In the following, a "clean up mechanism" to reduce the wrong hit usage and so to increase the purity for OI and Standalone tracking is presented. It is based on identifying a dense environment and on avoiding to pick up wrong hits in critical dense situations. Density is a relative quantity, relative to the resolution of the fit estimate. The OI tracks compared to the Standalone tracks have a better resolution, due to the additional measurements in the COT. The same event population appears worse for standalone than for OI tracks. Therefore the "clean up mechanism" yields better results for OI than for Standalone tracking.

Clean up of Wrong Hit Usage

When a hit is picked up during pattern recognition, the resolution of the estimated track parameters and the hit density in the current layer are known. In situations where a decision between two hits has to be taken, which can not be separated by the current resolution, we risk to pick up the wrong one. In such situations it is better not to incorporate a hit at all than to risk to pick up the wrong one. In order to quantify

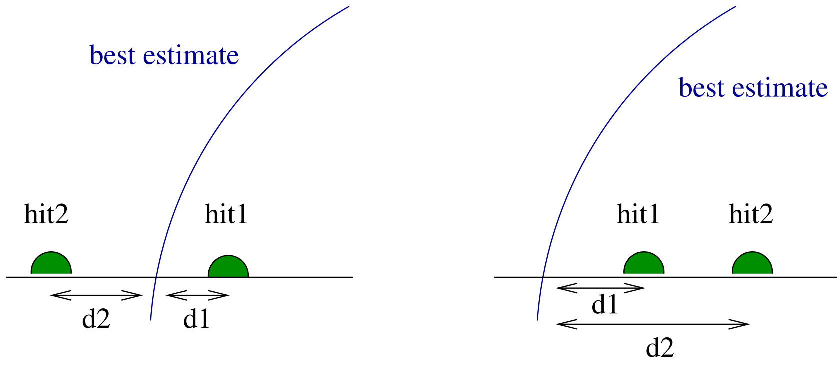


Figure 5.14: Schematic view for the definition of relative hit density.

the relative hit density the following quantity is introduced:

$$\Delta_{norm} = \frac{||d1| - |d2||}{\sqrt{\sigma_{fit}^2 + \sigma_{hit}^2}}, \tag{5.1}$$

where $d1$ and $d2$ are the distance between the best estimate and the closest and second closest hit (fig.: 5.14) and $\sigma_{fit}/\sigma_{hit}$ are the resolutions of the fit/hit.

Clean up of Wrong Hit Usage for OI tracking

Figure 5.15 shows Δ_{norm} of r/ϕ hits picked up in the OI tracking and their next closest hit. Figure 5.16 shows Δ_{norm} for OI tracks with wrongly and correctly picked up $90°z$ hits. On those plots, it can be seen that the relative hit density is a quantity to recognize potential critical situations.
Only ϕ hits, which have no nearby hit ($\rightarrow \Delta_{norm} \geq 3$) in LAYER 00, LAYER 0 and LAYER 1 are taken. Those layers are the ones with the highest rate of wrong hit usage. For the $90°z$ layers a cut on $\Delta_{norm} \geq 3$ in LAYER 0 and LAYER 1 is applied. In LAYER 3 the cut is $\Delta_{norm} \geq 1$.
The rate of correctly identified wrong hits and misidentified correct hits are listed in tables 5.5 and 5.6 for the $t\bar{t}$ and in 5.7 and 5.8 for the $b\bar{b}$ sample.
The cuts are not applied during pattern recognition but later. This means the potential wrong hits are picked up and incorporated in the fitter but marked as suspicious during the pattern recognition. They are eliminated once a clone is chosen for the final fit.
Keeping in mind that a couple of per cent of accidently misidentified correct hits are in absolute numbers still 3 times more hits than e.g. a third of the wrong hits, this is quite obviously. If it is decided not to take a hit during pattern recognition the resolution gets worse and the window for the hit search in the next layer increases thus additional wrong hits are picked up there.

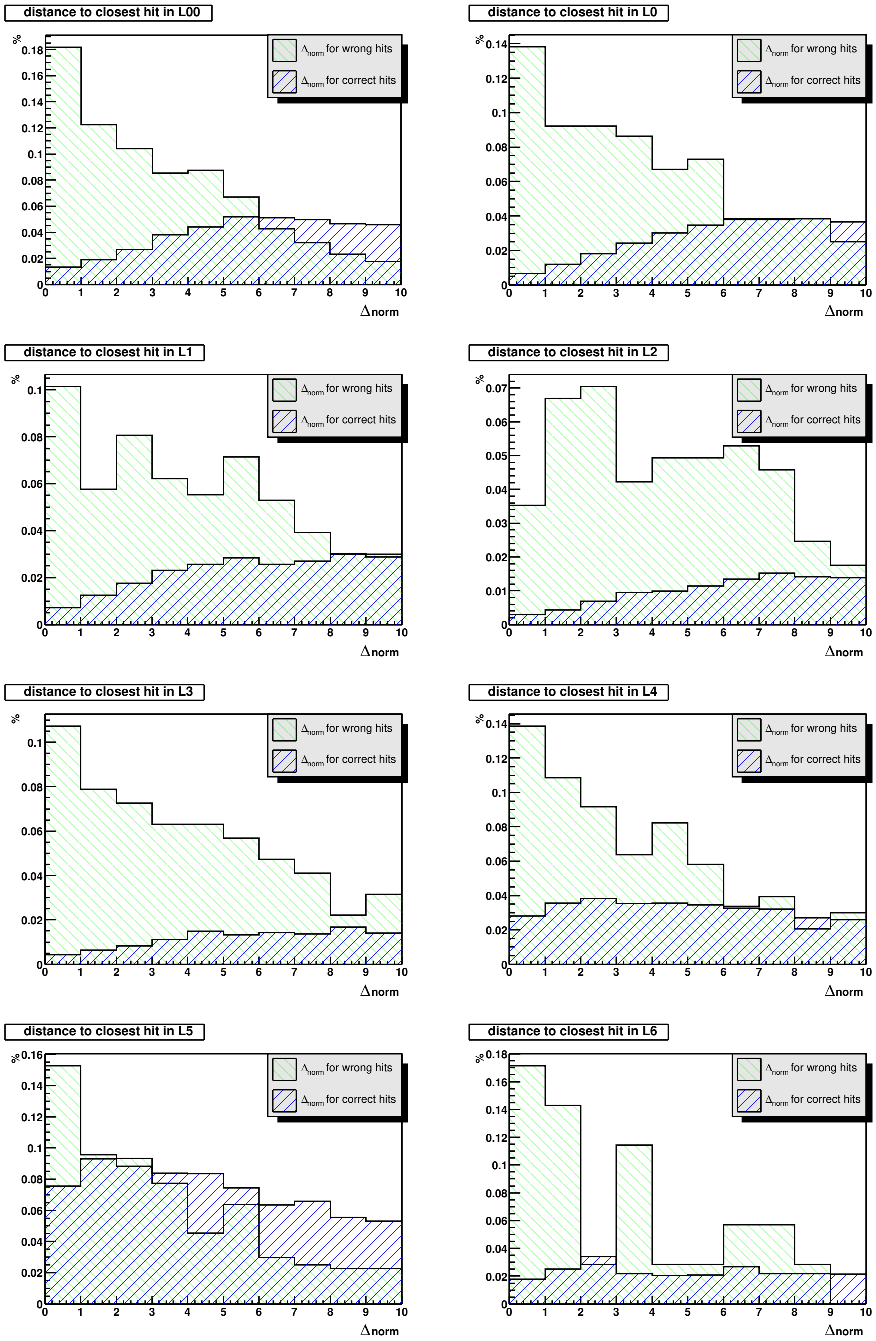


Figure 5.15: Relative hit density for correctly and wrongly picked up ϕ hits for a $t\bar{t}$ + 3 mbr events.

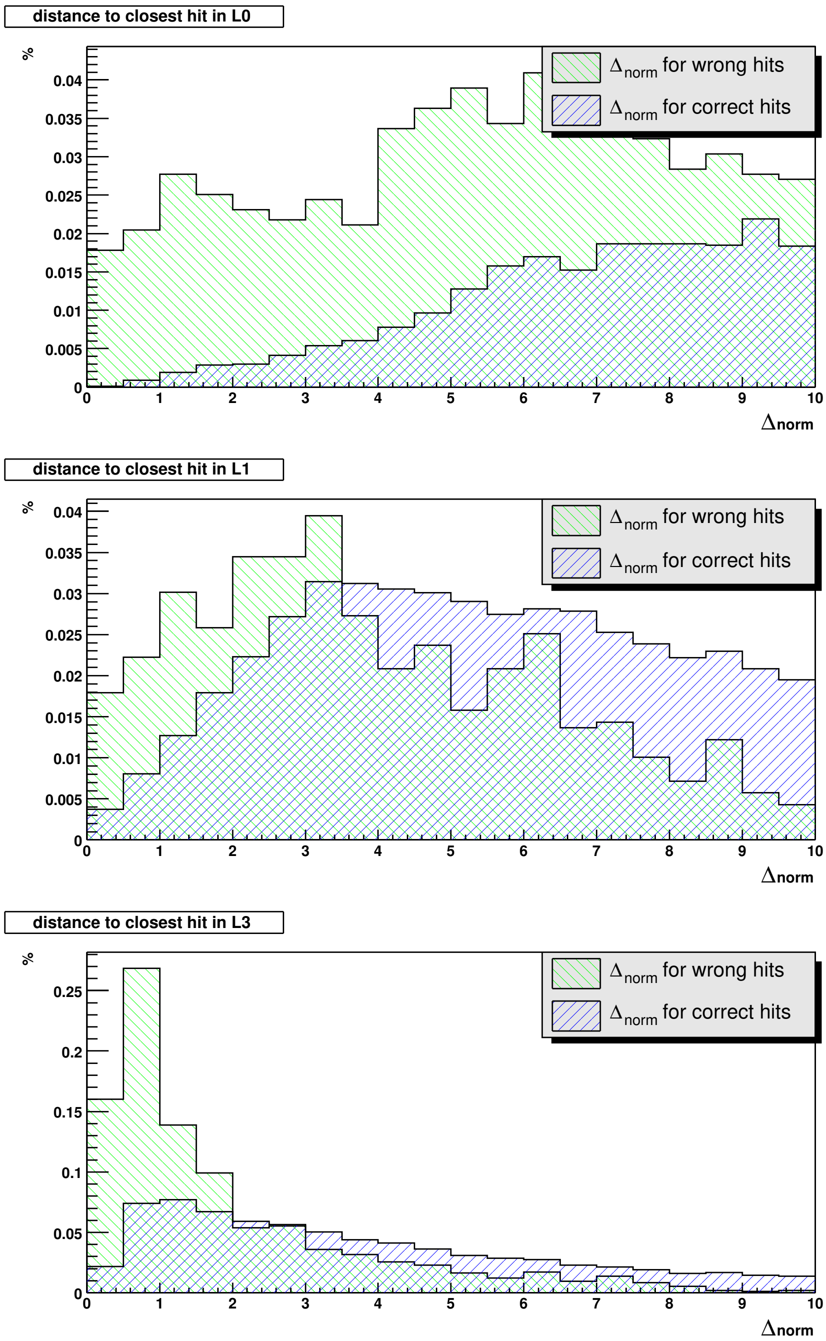


Figure 5.16: Relative hit density for correctly and wrongly picked up $90°z$ hits for a $t\bar{t}$ + 3 mbr events.

layer	identified wrong hits	misidentified correct hits
LAYER 00 (3σ)	40 %	5 %
LAYER 0 (3σ)	32 %	3.5 %
LAYER 1 (3σ)	24 %	5 %

Table 5.5: Influence of cut on Δ_{norm} on ϕ hit usage for $t\bar{t}$ events + 3 mbr .

layer	identified wrong hits	misidentified correct hits
LAYER 00 (3σ)	14 %	1.5 %
LAYER 0 (3σ)	16 %	9 %
LAYER 1 (1σ)	42 %	8 %

Table 5.6: Influence of cut on Δ_{norm} on $90°z$ hit usage for $t\bar{t}$ events + 3 mbr.

layer	identified wrong hits	misidentified correct hits
LAYER 00 (3σ)	35 %	3 %
LAYER 1 (3σ)	27 %	2 %
LAYER 2 (3σ)	26 %	1.5 %

Table 5.7: Influence of cut on Δ_{norm} on ϕ hit usage for $b\bar{b}$ events + 3 mbr.

layer	identified wrong hits	misidentified correct hits
LAYER 00 (3σ)	25 %	1 %
LAYER 0 (3σ)	22 %	2.5 %
LAYER 1 (1σ)	31 %	2 %

Table 5.8: Influence of cut on Δ_{norm} on $90°z$ hit usage for $b\bar{b}$ events + 3 mbr.

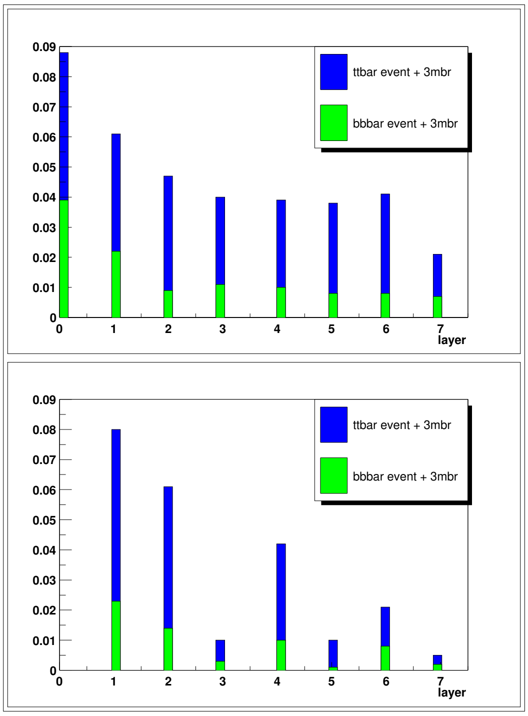


Figure 5.17: Wrong hit usage of OI tracks after applying the clean up mechanism for ϕ layers (upper plot) and z/SAS layers (lower plot).

The improvement for OI tracks achieved by the hit cleaning can be observed in the rate of wrong hits on the tracks shown in figure 5.17.
The efficiency of the OI tracking is almost not affected ($\pm$ 0.5 %) by the wrong hit cleaning. The price for the higher purity is a lower resolution. By this method we are not able to replace the wrong hit by the correct one but just to erase it. But correct hits are erased as well. The average hit usage after the hit cleaning is shown in figure 5.18 and 5.19. The impact on the $t\bar{t}$ events is larger than for the $b\bar{b}$ events, due to their higher hit density.
Most of the events expected from data are $b\bar{b}$ like events and high purity is essential for all the planed b physics analyses. Instead the top analyses do not rely so much on vertex resolution but deal more with calorimeter and jet information. Keeping this in mind in the author's opinion the loss of resolution is worth to pay for the gain in purity.
The clean up of the wrong hit usage can be switched off by the user for r/ϕ and z separately (`cleanUpPhi, cleanUpZ`). The interfaces for this switches are explained in appendix D.

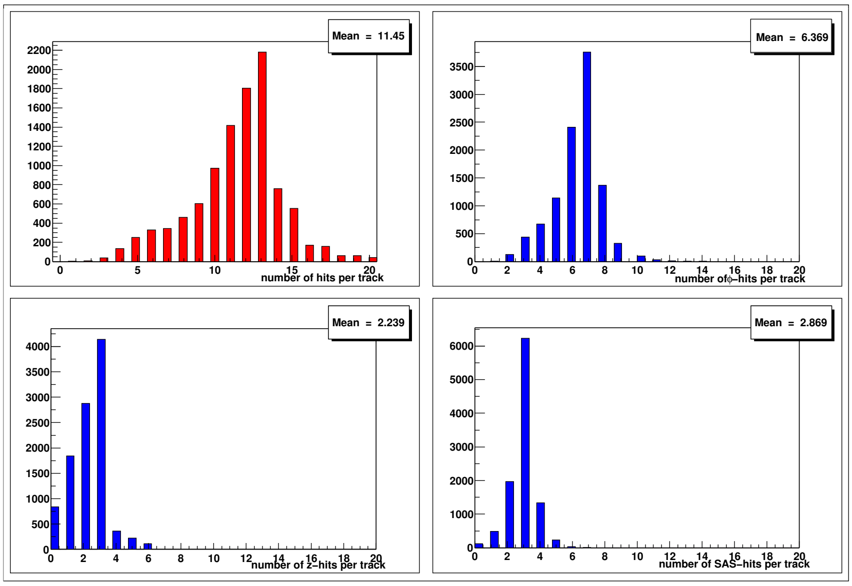


Figure 5.18: Average number of hits on OI tracks for a $t\bar{t} + 3$ mbr events including clean up of wrong hits.

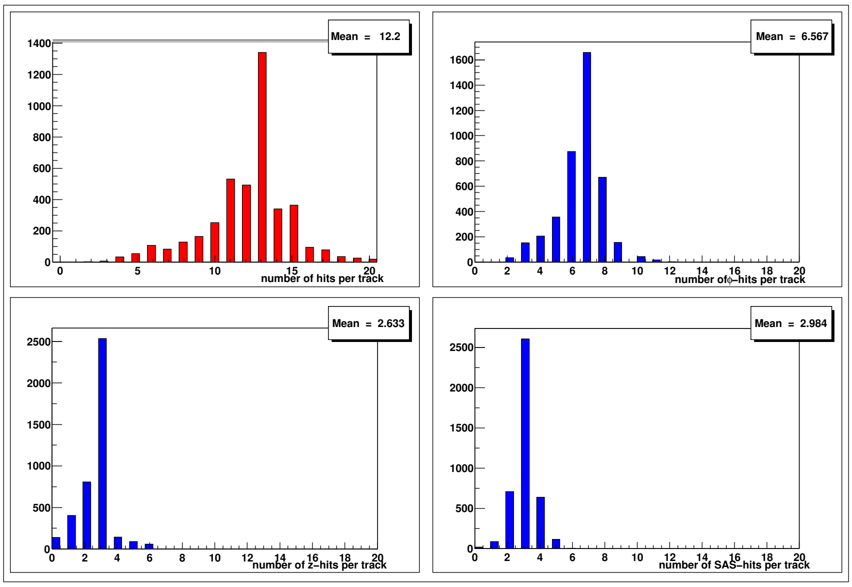


Figure 5.19: Average number of hits on OI tracks for a $b\bar{b} + 3$ mbr events including clean up of wrong hits.

Clean up of Wrong Hit Usage for Standalone Tracking

For the Standalone tracking, we have chosen to eliminate the ϕ hits in LAYER 00 ($\Delta_{norm} < 3$) only. Otherwise we risk to eliminate too many hits and the helix of the track is no longer fully determined. This cut reduces the wrong hit usage in LAYER 00 for both event samples and both modes by about a third.
In z a cut on ($\Delta_{norm} < 1$) is applied for all 90°z layers. This reduces the wrong hit usage in LAYER 0 and LAYER 1 by about 40 % in all cases. The wrong hit usage in LAYER 3 is reduced by 65 % (fig.: 5.20, 5.21).
For the Standalone tracking, the clean up of the wrong hit usage has a non-negligible impact on the efficiency. This is mostly related to the huge uncertainties in z. In z there are so many suspicious hits which are eliminated, that even some of the correct tracks do no longer fulfill the parameter cut on z for the efficiency definition.
The efficiency for both modes with the hit cleaning are shown in tables 5.9 and 5.10. The total hit usage of the tracks after the hit cleaning is shown in figures 5.22-5.25.
The cleaning-up of the wrong hit usage can be switched on and off for the ϕ and the 90°z layers separately (`cleanUpPhi, cleanUpZ`). The switches are explained in the appendix D. The default is to use the clean up in ϕ but not in z. This does not affect the efficiency significantly (± 0.5 %).
The wrong hit usage of the Silicon Standalone strategy running in mode one is comparable to the OI tracking. But it is slightly worse in the more problematic forward regions.

type of event	no cuts	cut on D	cuts on D and z_0
$t\bar{t} + 3$ mbr	85.4 %	78.9 %	60.9 %
$b\bar{b} + 3$ mbr	93.6 %	89.4 %	77.7 %

Table 5.9: Efficiency of Silicon Standalone tracking with hit cleaning, quality 1+2, running in mode two.

type of event	no cuts	cut on D	cuts on D and z_0
$t\bar{t} + 3$ mbr	88.0 %	83.0 %	63.6 %
$b\bar{b} + 3$ mbr	95.85 %	93.8 %	85.3 %

Table 5.10: Efficiency of Silicon Standalone tracking with hit cleaning, quality 1+2, running in mode one.

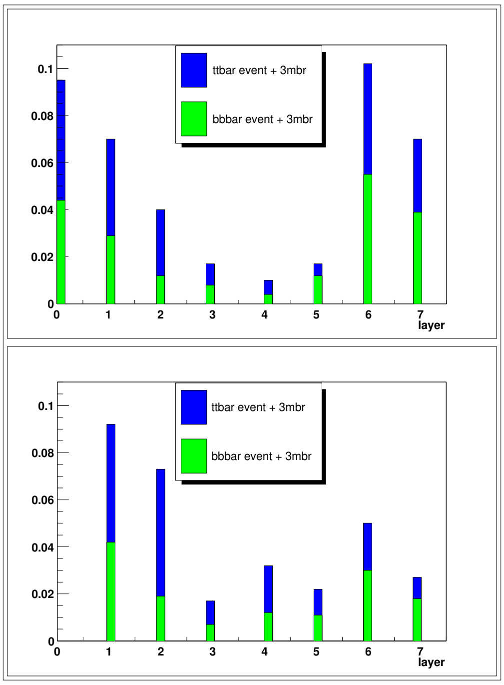


Figure 5.20: Wrong hit usage of quality 1+2 Standalone tracks processed in mode two for ϕ layers (upper plot) and z/SAS layers (lower plot) including hit cleaning.

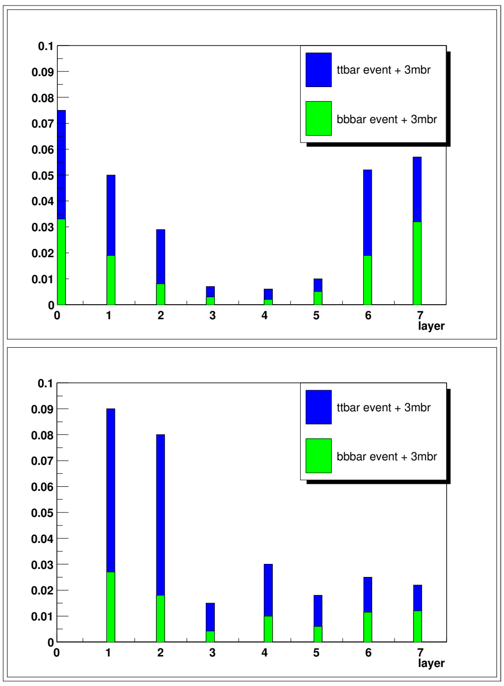


Figure 5.21: Wrong hit usage of quality 1+2 Standalone tracks processed in mode one for ϕ layers (upper plot) and z/SAS layers (lower plot) including hit cleaning.

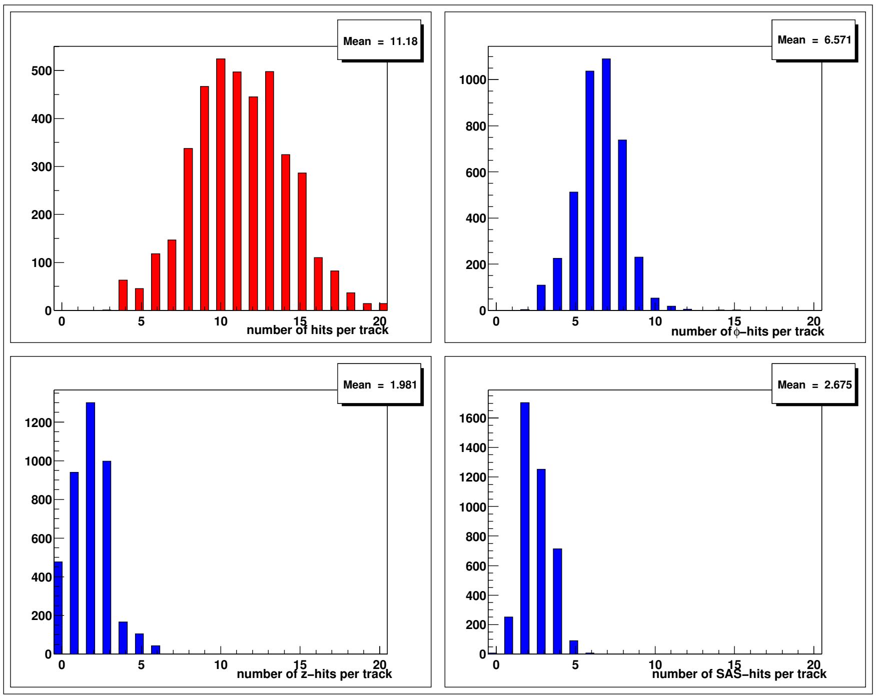


Figure 5.22: Total number of hits on the tracks ($t\bar{t}$ + 3 mbr) after clean up of the wrong hit usage for the Standalone algorithm running in mode two.

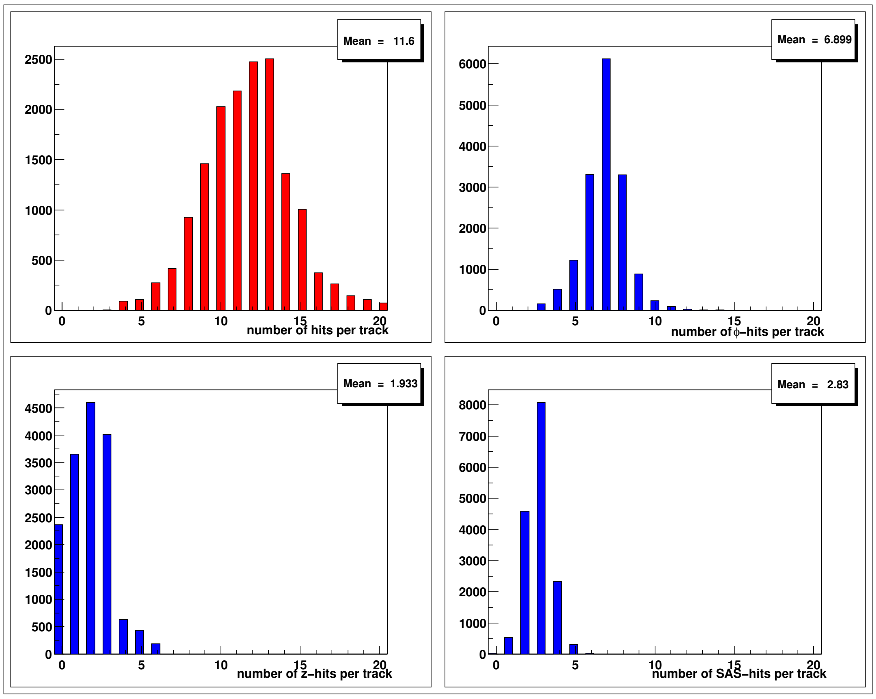


Figure 5.23: Total number of hits on the tracks ($t\bar{t}$ + 3 mbr) after clean up of the wrong hit usage for the Standalone algorithm running in mode one.

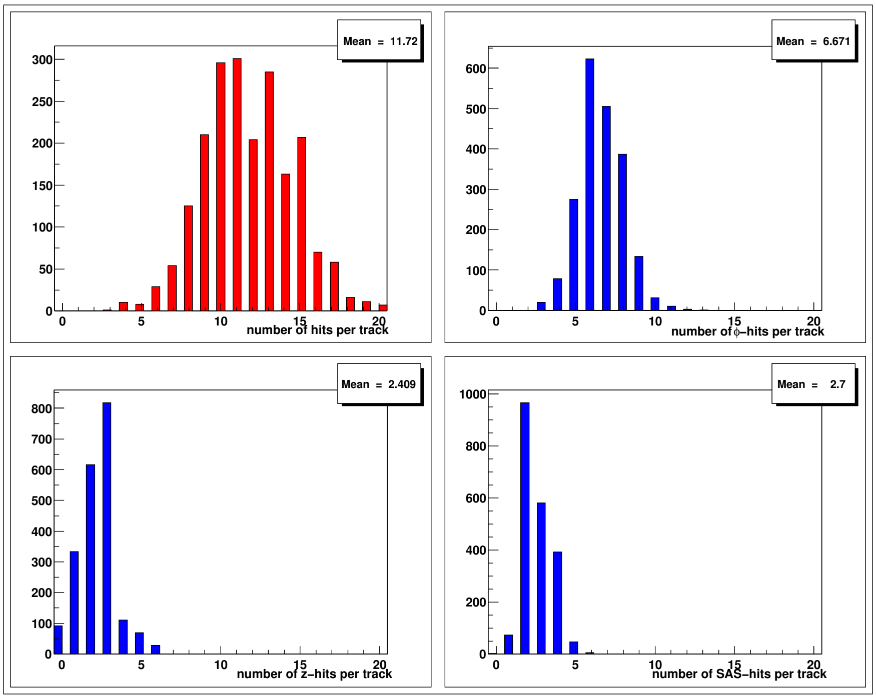


Figure 5.24: Total number of hits on the tracks ($b\bar{b}$ + 3 mbr) after clean up of the wrong hit usage for the Standalone algorithm running in mode two.

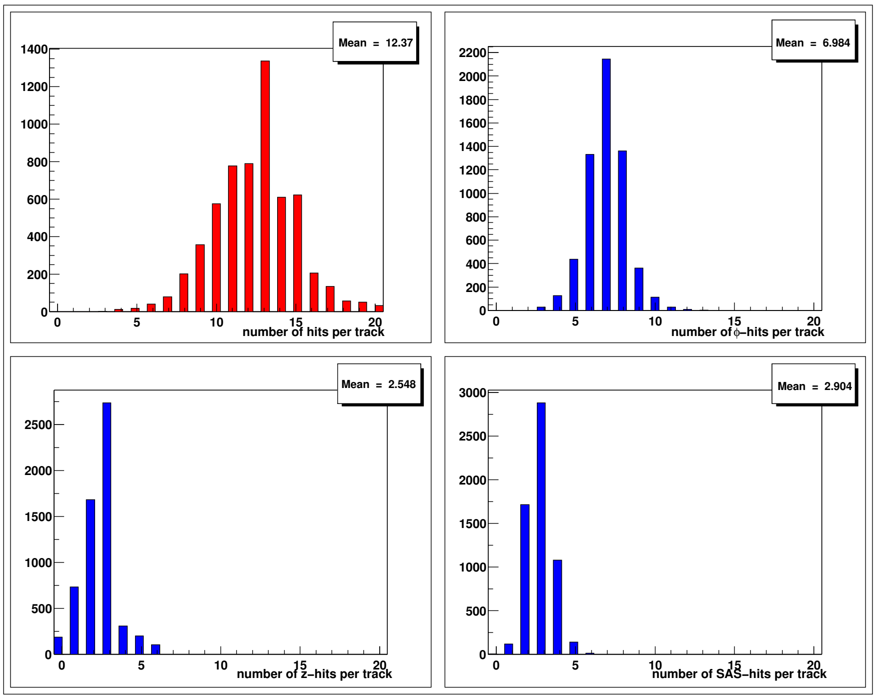


Figure 5.25: Total number of hits on the tracks ($b\bar{b}$ + 3 mbr) after clean up of the wrong hit usage for the Standalone algorithm running in mode one.

5.1.4 Impact of the Pre-Tracking Primary Vertex Finder

Every inefficiency of the pre-tracking PV finder is directly inherited from the Standalone algorithm. The current implementation of the PV finder yields an efficiency of about 89 % on $t\bar{t}$ + 3 mbr events and about 95 % for $b\bar{b}$ + 3 mbr events. A detailed description of the pre-tracking PV finder algorithm and its performance can be found in [32].
Besides the efficiency, the influence of the number of fake vertices found by the pre-tracking PV finder on the performance of the Standalone tracking is interesting. For this study the MC_{truth} vertices have been passed to the strategy and different numbers of fake vertices have been added. The z positions of the fake vertices have been randomly simulated following a Gaussian distribution with σ_z = 30 cm and a mean of z = 0 cm. The results are presented in tables 5.11 and 5.12 for the Standalone tracking running in mode two. The efficiency and fake rates are slightly better compared to the ones presented in the last section, because the true vertices are directly taken from MC and not smeared by the resolution of the PVz finder. The additional fake vertices have almost no effect on the efficiency and the fake rate. The time[2] needed for processing the events increases by about 15 % – 20 % by adding additional 70 % of fake vertices.
The current fake rate of the pre-tracking PV finder is less than 70 %. Therefore from the point of view of the Standalone tracking the seed vertices are of good quality.

	0 fake PVs	+30 % fake PVs	+50 % fake PVs	+70 % fake PVs
eff. no cuts	87.2 %	87.0 %	86.9 %	86.7 %
cut on D	81.4 %	81.2 %	81.1 %	80.9 %
cut on D, z_0	69.1 %	68.7 %	68.4 %	68.0 %
fake	19.9 %	20.3 %	20.3 %	20.4 %
time (100 ev.)	33.1 s	7.1 s	39.1 s	40.7 s

Table 5.11: Impact of the fake rate of the seed of PVs for $t\bar{t}$ + 3 mbr events.

	0 fake PVs	+30 % fake PVs	+50 % fake PVs	+70 % fake PVs
eff. no cuts	94.1 %	93.9 %	93.8 %	93.8 %
cut on D	89.6 %	89.4 %	89.3 %	89.2 %
cut on D, z_0	82.9 %	82.9 %	82.82 %	82.8 %
fake	7.5 %	7.7 %	7.8 %	7.8 %
time (100 ev.)	10.2 s	11.5 s	11.8 s	12.5 s

Table 5.12: Impact of the fake rate of the seed of PVs for $b\bar{b}$ + 3 mbr events.

[2]measured on a 1.4 GHz Athlon with gcc version 3.0.1 using optimization level 2 ("-02")

5.2 Timing and Memory Performance

The following timing measurements are made on a 1.4 GHz Athlon with KCC version 4_0f compiled without any optimization corresponding to the currently running code on the trigger and production farms. The following numbers (tab.: 5.13) are obtained by using the standard Unix command "time". Considering the large size of the test samples (500 $t\bar{t}$ + 3 mbr events, 5,000 $b\bar{b}$ + 3 mbr and 5,000 $b\bar{b}$ events) this provides reliable results.
Once CDF collects the planed amount of data, the average events and therefore the expected timing performance correspond to the one of the $b\bar{b}$ + 3 mbr events. Due to the low luminosity the typical event size and time consumption are currently more similar to the $b\bar{b}$ events.
There are some rare events on data (1:1,000), which suffer from back scattering of particles in the very forward direction. They have order of magnitudes more hits than a typical $t\bar{t}$ + 3 mbr event. They are of no physical interest but have to be filtered out in order not to exceed the timing limits on the farms due to the exploding combinatorics in the silicon tracking.
The memory consumption of the TrackingKal tracking algorithms for $t\bar{t}$ + 3mbr events never exceeds 200 MB, which are temporarily needed for the event. The memory consumption of the production executable on the farms is limited to 1.2 GB. The memory consumption especially for the Standalone tracking scales with combinatorics. Therefore it is also of interest to filter out the above described back scatter events on data.
The timing and memory performance of the silicon tracking is, from the point of view of trigger and production issues, in very good shape. It is the first time at a hadron collider that the full event reconstruction is fast enough to take place at trigger level.

	$t\bar{t} + 3$ mbr	$b\bar{b} + 3$ mbr	$b\bar{b}$
COT tracking	0.84 s/ev.	0.29 s/ev.	0.10 s/ev.
silicon clustering	0.05 s/ev.	0.04 s/ev.	0.02 s/ev.
pre-tracking PV finder	0.11 s/ev.	0.04 s/ev.	0.02 s/ev.
OI tracking	0.50 s/ev.	0.12 s/ev.	0.04 s/ev.
Standalone tracking (mode two)	0.70 s/ev.	0.22 s/ev.	0.04 s/ev.
Standalone tracking (mode one)	1.12 s/ev.	0.55 s/ev.	0.06 s/ev.

Table 5.13: Timing performance of the tracking.

5.3 First Look at Data

5.3.1 Status of the Silicon System

In the mean time[3] the silicon system (SVX II + ISL) is stably working and getting good data from 80 % of its modules (fig.: 5.26). This value is still below the quality which is needed for RUN IIA physics goals. The following problems in the silicon each causes about a third of the failure modes:

- There are damaged ISL and SVX II ladders, which can be repaired, but most of them require to open the detetor. A shutdown is planed for january 2003.
- There are still two ISL lines with cooling blockages and one line with insufficient cooling. In order to fix this problem detector access is required as well.
- There are several ladders of the ISL and of the SVX II, which have been irrecoverably damaged during operation (e.g. beam incident of 3/30/02).

LAYER 00 is not yet fully working mostly due to pedestal subtraction, noise suppression and clustering issues, but work is going on to improve its operation.
The SVX II is completely aligned at wafer level in r/ϕ, the ISL central barrel as well. The alignment of the ISL forward region and the overal z alignment is under investigation.

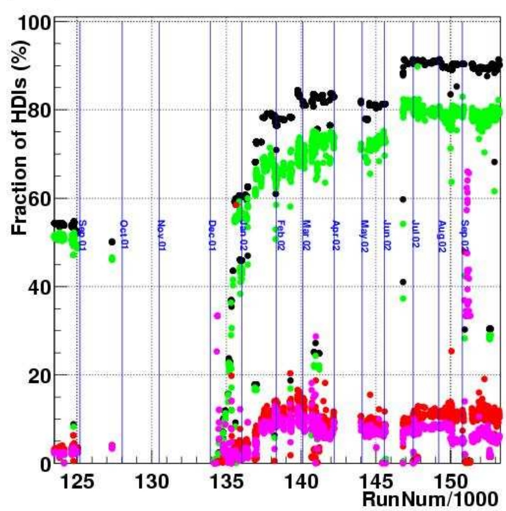


Figure 5.26: Progress in the silicon commissioning: powered ladders are shown in black, good ladders in green (light grey), bad ladders in red (dark grey) and error rate in pink (middle grey).

[3]status of 10/2002

5.3.2 Validation of the Tracking Algorithms

The validation of the tracking algorithms on data is a much more complicated task than on Monte Carlo, because there is no known reference to compare with. The only reasonable way to validate the strategies is by studying the quality of the physics analyses based on them. An overview of the first CDF II analyses based on silicon tracking are presented in the next chapter.
In order to get at least a feeling of the tracking performance on data the COT tracks are used as efficiency denominator for the OI tracking and the already found OI tracks are used as denominator for the Standalone tracking running on the full combinatorics. This study has been performed using the j-and g-stream data of run #148824, which is considered as good silicon run.
For the OI tracking all COT tracks without any quality cut are used as denominator and every reconstructed OI track is counted as numerator track. With this definition, the OI tracking is yielding an efficiency of 85 % on both used samples, which is a better result as has ever been obtained in RUN I. Anyhow, an improvement is expected once the alignment is completely in place and the strategy is tuned to the higher level of understanding of the detector.
For the Standalone tracking, all OI tracks, which are compatible with one of the reconstructed PVz positions within $\pm$ 1 cm, and which have at least 4 ϕ and 2 SAS hits, are in the denominator. All reconstructed Standalone tracks, which share at least 70 % of their hits with an OI track are counted for the numerator. By this definition an efficiency of 76 % for all Standalone tracks (quality1+2) is achieved. Taking only quality 1 tracks into account the efficiency decreases to 72.5 %. If the additional denominator cut on the number of hits on the OI tracks are skipped, the efficiencies go down to 58 % and 55.5 %, respectively. If the OI track was not able to pick up two SAS hits because they are shifted in z due to misalignment, then the Standalone strategy can not build a reasonable candidate out of them.
It is only possible to measure the efficiency of the Standalone tracking in the central region, where the OI tracks are available. If the experience from MC is extrapolated to the data, a lower efficiency is expected in the more challenging forward regions.
In figure 5.27, the hit usage of the OI tracking is presented. Figure 5.28 shows the hit usage of the Standalone tracking. Both total average hit usages are about 3.2 hits lower than on MC. The deficit is equally distributed over the ϕ, SAS and z hit usages. Figure 5.29 shows the fraction between the number of times a track is intersecting a layer and picking up a hit there. This fraction is between 60 % - 80 % in the SVX II and the ISL. It is slightly lower in LAYER 00. If those numbers are rescaled by the 80 % of reliably working silicon modules and by the still existing LAYER 00 problems, a quite high average hit yield is obtained.

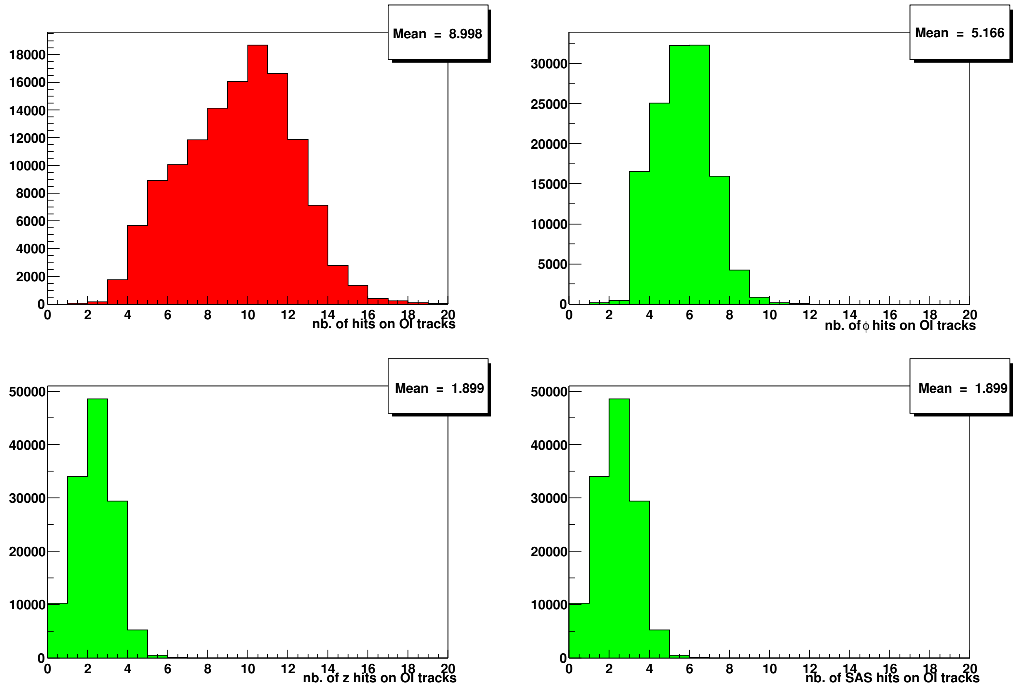


Figure 5.27: Hit usage of OI tracks on data.

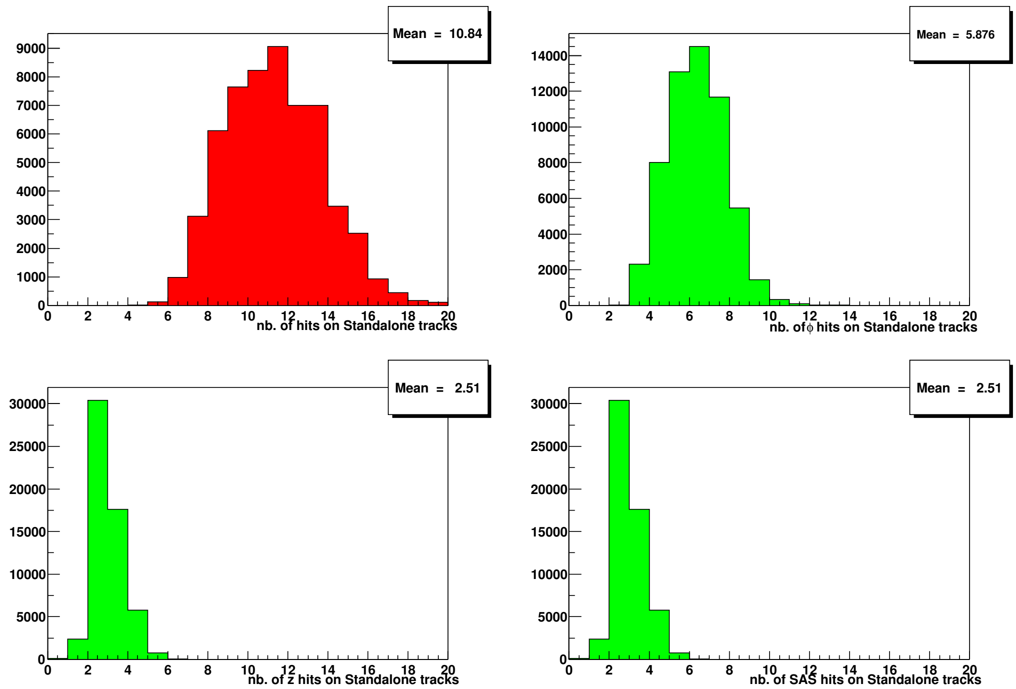


Figure 5.28: Hit usage of Standalone tracks (mode two) on data.

The total amount of tracks on the j-stream sample is shown in figure 5.30. The fraction of Standalone tracks to OI tracks is about a factor of 2 lower than on MC. This indicates problems in the forward regions, which are most probably related to the missing z alignment and the failure mode of the modules of the ISL. The fraction of the quality 2 silicon tracks is higher than on MC but they still contribute very little to the efficiency ($\approx 3\%$). Considering the high fake rate of the quality 2 tracks on MC it is not recommended to trust them at least not if they are not confirmed by some signals e.g. in the calorimeters.
After this snapshot on the tracking performance on data, it can be concluded that the silicon tracking is already well performing on data but the monitoring and the tuning of the strategies have to go on until the silicon detector is completely aligned and fully working.

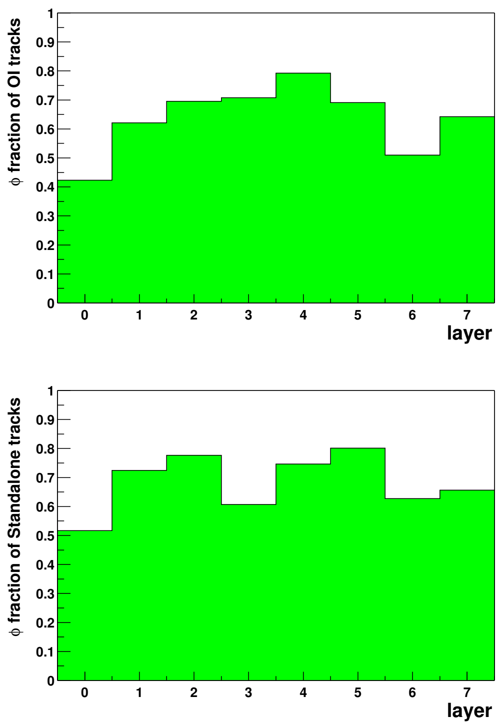


Figure 5.29: Fraction of tracks which picked up hits in a layer (top OI tracks, bottom standalone tracks (mode two)).

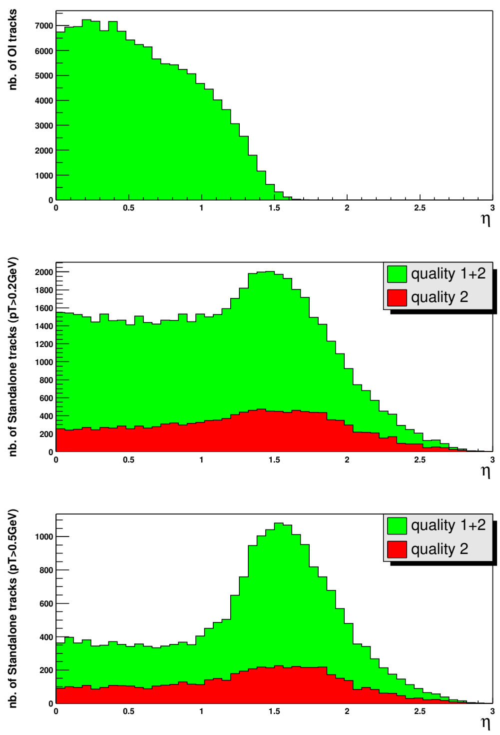


Figure 5.30: Total number of reconstructed tracks (top OI tracking, middle Standalone tracking (mode two) $p_T \geq 0.2$ GeV, bottom Standalone tracking (mode two) $p_T \geq 0.5$ GeV).

Chapter 6

First Physical Results

The data collected so far has been used for two main issues.
One is the improvement in the understanding of the detector. The data is used e.g. for the alignment of the silicon vertex detector, for determination of the tracking momenta scale and for characterizing the detailed detector performance.
But there are also already real physics measurements with complete analysis including a full estimate of systematic errors. Although the statistics is still limited, some results are already competitive with the best measurements currently available.
In this chapter, an overview of some early CDF II results is given. It is concentrating on results, which crucially depend on the tracking in the vertex detector in order to prove the good performance of the silicon system. All those analyses are obtained by using the Kalman fitter and the tracking strategies of the TrackingKal package, presented in the last chapters.

6.1 The $J/\Psi \to \mu^+\mu^-$ Trigger Sample

The current size of the $J/\Psi \to \mu^+\mu^-$ trigger sample is 37 pb^{-1} and about 350.000 J/Ψ mesons have been reconstructed. This number is reduced by about a factor two when silicon information is required for both muon tracks. This rate and the J/Ψ mass resolution (21 MeV/c^2 COT only muon tracks, 15 MeV/c^2 silicon tracks for both muons) are consistent with the expectations.

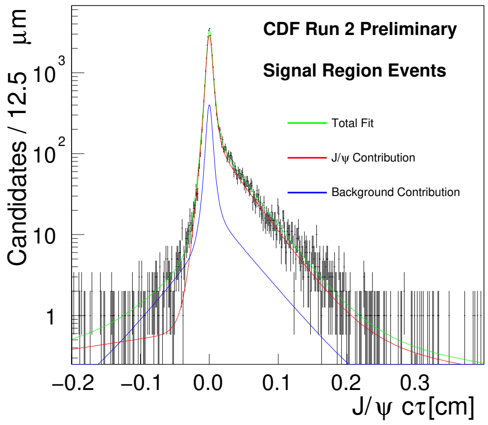


Figure 6.1: Inclusive lifetime fit for $B \to J/\Psi + X$.

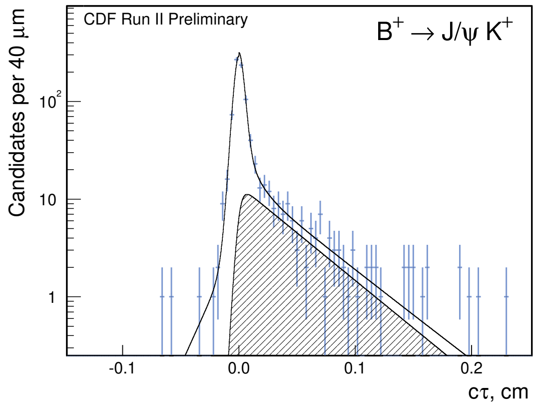


Figure 6.2: The lifetime distribution of $B^+ \to J/\Psi K^+$.

While most J/Ψ mesons produced at the Tevatron are prompt, a fair fraction originates from decays of B mesons. The sample is large enough to separate statistically the two components and to measure the inclusive B hadron lifetime and the fraction of prompt J/Ψ. The measured lifetime is [35] (fig.: 6.1):

$$c\tau_{B\ incl} = 458 \pm 10\ (\mathrm{stat}) \pm 11\ (\mathrm{sys})\ \mu\mathrm{m}$$

This result is consistent both with the PDG value [27] and with previous CDF measurements [36]. This is a good indication that most of the tracking systematics are well under control.
The fraction of prompt J/Ψ mesons has been determined to be 85 %. They are useful to test the quality of our understanding of the tracking errors. The standard deviation of the observed transverse decay length distribution is just 7 % larger than the calculated errors. This shows that residual misalignments and additional systematics, not modelled in the Monte Carlo already, are at a very low level.

The exclusive lifetime of the B^+ meson has also been reconstructed [37] (fig.: 6.2):

$$c\tau_{B+} = 446 \pm 43\ (\mathrm{stat}) \pm 13\ (\mathrm{sys})\ \mu\mathrm{m}$$

Relying on the momentum calibration based on the J/Ψ mass peak[1] [28], the masses of the three B meson ground states are reconstructed [38] in their exclusive J/Ψ modes: $B^+ \to J/\Psi K^+$, $B^0 \to J/\Psi K^{*0}$ and $B^0_s \to J/\Psi\phi$ (fig.: 6.3-6.5).

$$\begin{aligned} m(B^+) &= 5280.6 \pm 1.7\ (\mathrm{stat})\ \pm 1.1\ (\mathrm{sys})\ \mathrm{MeV/c^2} \\ m(B^0) &= 5279.8 \pm 1.9\ (\mathrm{stat})\ \pm 1.4\ (\mathrm{sys})\ \mathrm{MeV/c^2} \\ m(B^0_s) &= 5360.3 \pm 3.8\ (\mathrm{stat})\ ^{+2.1}_{-2.9}\ (\mathrm{sys})\ \mathrm{MeV/c^2} \end{aligned}$$

As a higher statistics monitor, the Ψ' mass from $\Psi' \to J/\Psi\pi\pi$ was used.

$$m(\Psi') = 3696.43 \pm 0.54\ (\mathrm{stat})\ \mathrm{MeV/c^2}$$

[1] The momentum calibration has been already described in detail in chapter 3.

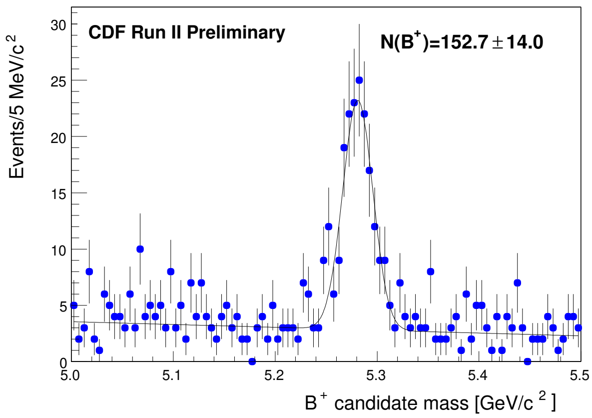


Figure 6.3: Reconstructed B^+ candidates.

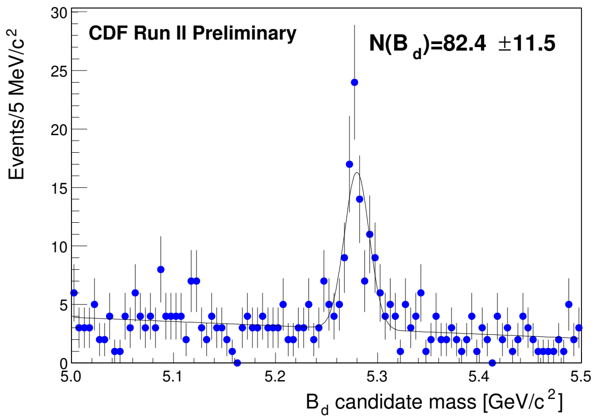


Figure 6.4: Reconstructed B^0 candidates.

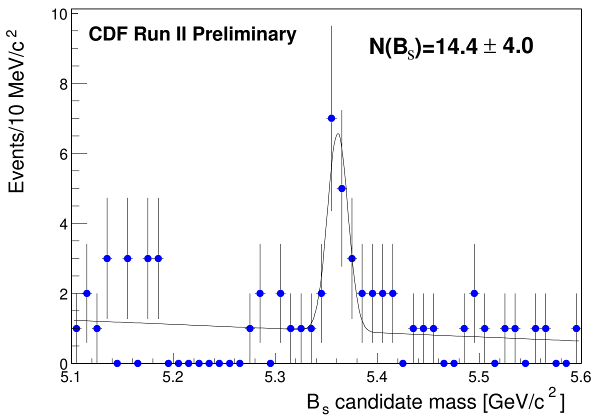


Figure 6.5: Reconstructed B_s^0 candidates.

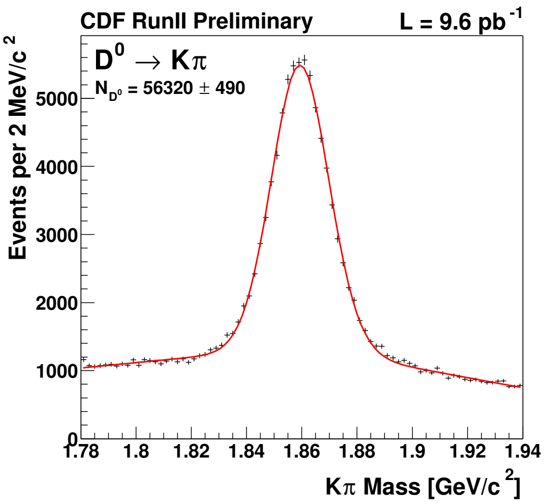


Figure 6.6: $D^0 \to K^\pm\pi^\pm$ candidate invariant mass distribution.

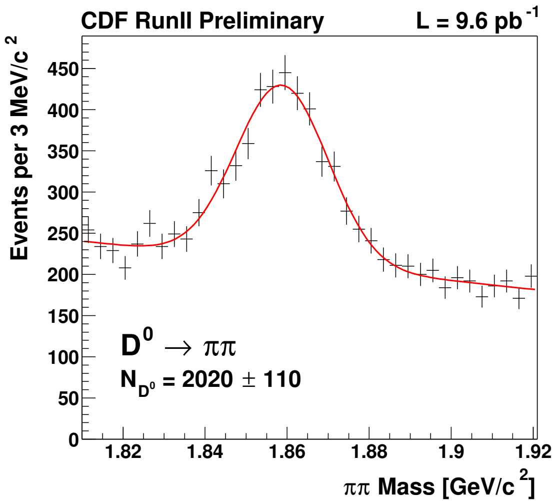


Figure 6.7: $D^0 \to \pi^-\pi^+$ candidate invariant mass distribution.

6.2 The Secondary Vertex Trigger Sample

The secondary vertex trigger not only depends on the momentum resolution as the J/Ψ trigger but as well on a good imapct paramter resolution in r/ϕ and in z in order to select heavy quark events and to reject the huge background from u,d and s quarks. Although the lifetime of B hadrons is longer, the huge production cross section of charm quarks makes the secondary vertex trigger especially rich in D mesons. The event yield of $D^0 \to K^\pm\pi^\pm$ (fig.: 6.6) is comparable to that of the J/Ψ mesons. But also large amounts of Cabibbo suppressed D^0 decays into K^+K^- and $\pi^+\pi^-$ (fig.: 6.7) are observed.
Given this large charm yield CDF II is in the position to perform statistically competitive measurements of charm production and decays even with little integrated luminosity. But the systematics are a challenging issue at this early stage of the experiment.
Therefore the relative branching ratios are particular interesting, because many of the systematics associated with trigger and reconstruction efficiencies cancel to first order. The following results are obtained [39] based on 10 pb^{-1}:

$$\frac{\Gamma(D^0 \to K^+K^-)}{\Gamma(D^0 \to K^\pm\pi^\pm)} = 11.17 \pm 0.48 \text{ (stat)} \pm 0.98 \text{ (sys) } \%$$
$$\frac{\Gamma(D^0 \to \pi^+\pi^-)}{\Gamma(D^0 \to K^\pm\pi^\pm)} = 3.37 \pm 0.20 \text{ (stat)} \pm 0.16 \text{ (sys) } \%$$

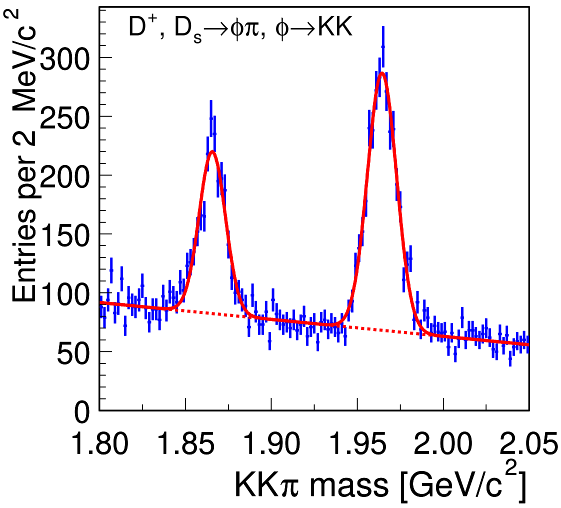


Figure 6.8: $D_s^+/D^+ \to \phi\pi^+$ candidate invariant mass distribution.

Other D species are also observed. $D^{\pm}$ and $D_s^{\pm}$ has been reconstructed in the $\phi\pi$ channel, and their mass difference has been calculated [40] (fig.: 6.8).

$$m(D_s^{\pm}) - m(D^{\pm}) = 99.28 \pm 0.43 \text{ (stat)} \pm 0.27 \text{ (sys) MeV/c}^2$$

The advantage in doing a mass difference measurement in a common final decay state is that most of the systematic effects should cancel in the difference, and the limitations on the measurement then come from statistics.

The next step is to distinguish between direct and secondary production of D mesons. In order to get an idea of the fraction coming from B mesons the core of the impact parameter distribution of the D mesons is fitted with a Gaussian plus additional tails obtained by fitting the prompt K_s^0 in the decay $K_s^0 \to \pi^+\pi^-$ (fig.: 6.9, 6.10). Therefore it is essential to model the tails carefully. The results for the fraction of D mesons from B decays compared to all reconstructed D mesons are: D^0 : 16.43 ± 0.65 %, D^{*+} : 11.41 ± 1.37 %, D^+ : 11.26 ± 0.53 % and D_s^+ : 34.80 ± 2.75 %.

Also hadronic B meson decays have been fully reconstructed. This is a more difficult issue, because the two body decays have very small branching fractions, and the multi-body decays have a larger combinatoric background. A $B^+ \to D^0\pi^+$ signal (fig.: 6.11) and for the first time at a hadron collider a signal of B decays to two long-living hadrons (mainly $\pi^+\pi^-$, $K\pi$, K^+K^-, fig.: 6.12) have been observed. The width of

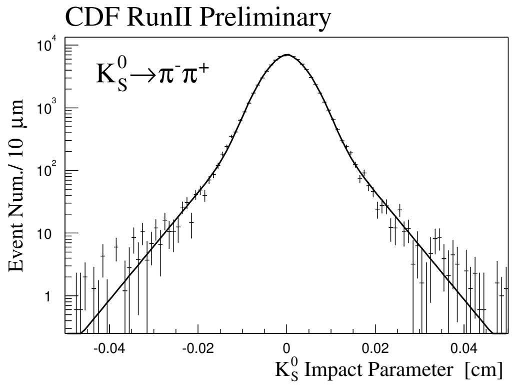


Figure 6.9: Impact parameter distribution for $K^0_s \to \pi^-\pi^+$ decays.

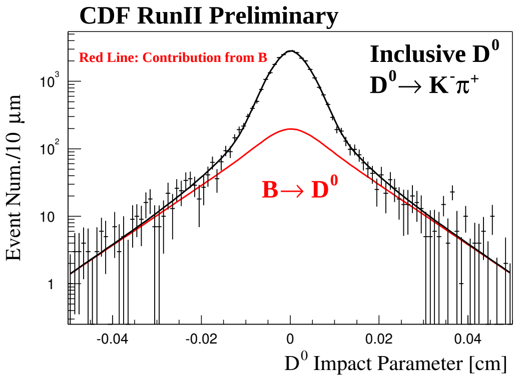


Figure 6.10: Impact parameter distribution of the inclusive D^0 decays.

the $B \to h^+h^-$ mass peak is larger than the mass resolution of the detector, because several decay modes of the B^0_d and the B^0_s contribute to it.

From this snapshot on the first physics results, it can be concluded that CDF II is already taking good quality data. The understanding of the detector is very advanced, especially the silicon is well under control. Surprisingly enough some of these results have proven to be already competitive in spite of the rather limited integrated luminosity so far analyzed. New exciting results for the winter conferences 2003 are expected.

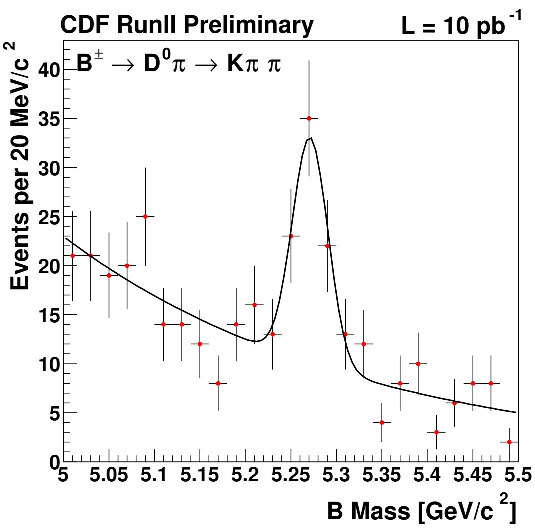


Figure 6.11: $B^+ \to D^0\pi^+$ candidate invariant mass distribution.

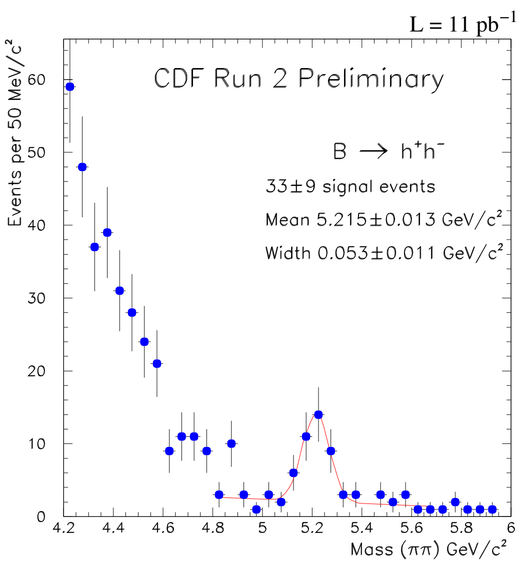


Figure 6.12: $B^0 \to h^-h^+$ candidate invariant mass distribution.

Chapter 7

Conclusion

In this thesis the track reconstruction in the CDF II silicon vertex detector has been presented. The central part of this software is the TrackingKal package which is based on a fast Kalman fitter. The following resolutions are obtained for high transverse momentum tracks:

- transverse momentum resolution:	$\frac{\delta p_T}{p_T^2}$	$\approx$	3.5×10^{-3} GeV^{-1}
- R/ϕ impact parameter resolution:	δD	$\approx$	10 μm
- z resolution:	δz_0	$\approx$	50 μm

The Kalman fitter of the TrackingKal package is about a factor of 50 times faster compared to any other CDF II fitter with comparable resolution.

Due to its progressive character and its high timing performance, the fitter is used inside two different pattern recognition strategies of the TrackingKal package. By studying the performance of the already existing Outside-In strategy and by reimplementing it, it was possible to raise its efficiency to about 90 % and to gain a factor of 40 in timing performance. Even in high combinatoric events such as $t\bar{t}$ with three underlying minimum bias events, about 90 % of its tracks have not picked up a wrong hit at all. The Outside-In tracking is limited to the reconstruction of tracks in the central region. Therefore the Silicon Standalone strategy has been implemented. It reconstructs tracks with silicon information only and thus has to resolve huge combinatorics. It is the first time in CDF that such an algorithm is working reliably. It yields an efficiency of about 85 % and its purity and timing performance are comparable to the Outside-In algorithm.

Due to the TrackingKal package, it has been possible for the first time to perform full event reconstruction at trigger level in a hadronic environment.

Appendix A

Conversion of Parametrization

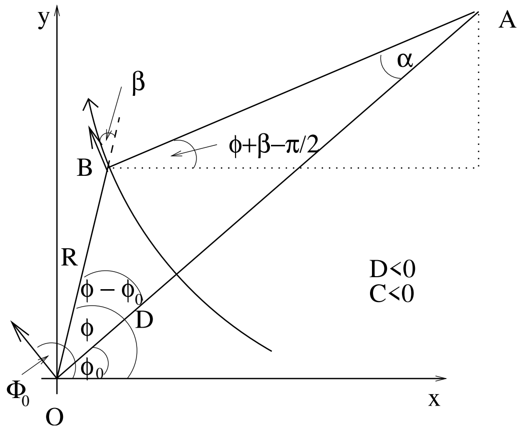
A.1 Perigee Parameters → Fitter Parameters

The calculation of the transformation from the CDF track parameters into the track parametrization used in the fitter is split up in four cases:

- $D < 0$ and $C < 0$ (fig.: A.1),
- $D > 0$ and $C > 0$ (fig.: A.2),
- $D > 0$ and $C < 0$ (fig.: A.3),
- $D < 0$ and $C > 0$ (fig.: A.4).

Before starting with the calculation one variable is introduced:

$$\text{sgn} = \frac{|C|}{C} \tag{A.1}$$



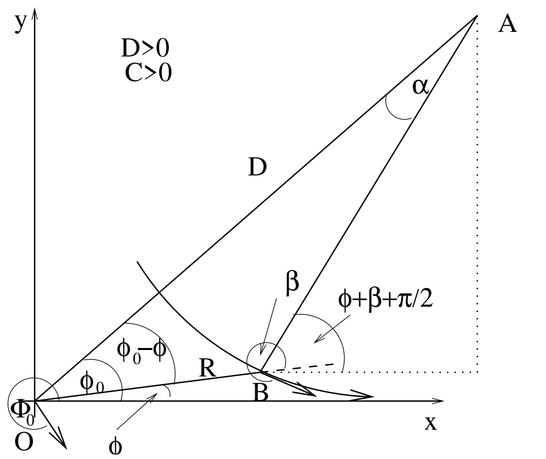
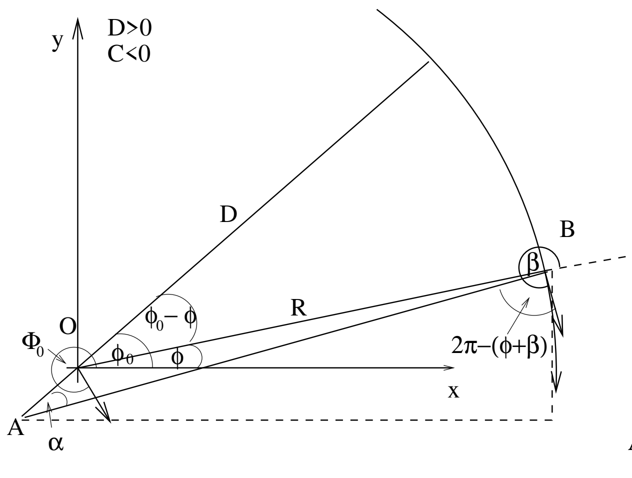


Figure A.1: Helix with D<0 and C<0.

Figure A.2: Helix with D>0 and C>0.



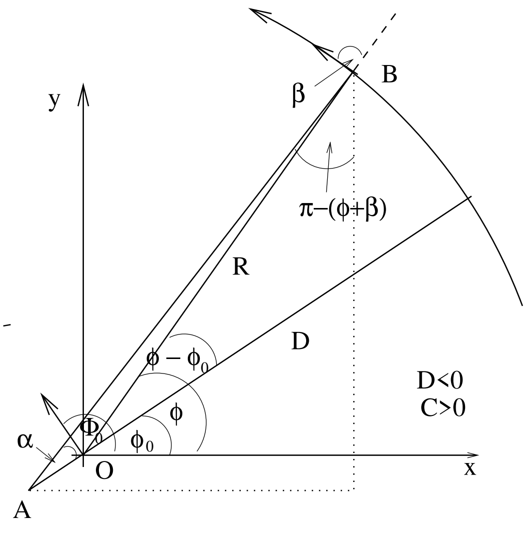


Figure A.3: Helix with D>0 and C<0.

Figure A.4: Helix with D<0 and C>0.

case 1:

Due to the cosine law in the triangle Δ(BOA) (fig.: A.1) the following relation is valid:

$$\begin{aligned}
\frac{1}{4C^2} &= (\frac{1}{2C}+D)^2 + R^2 + 2R(\frac{1}{2C}+D)\cos(\phi-\phi_0) \\
\cos(\phi-\phi_0) &= -\frac{R^2C+D^2C+D}{R(1+2CD)} && \text{(A.2)} \\
\cos(\pi-(\phi-\phi_0)) &= \frac{R^2C+D^2C+D}{R(1+2CD)} && \text{(A.3)} \\
\phi-\phi_0 &= \pi - \arccos(\frac{R^2C+D^2C+D}{R(1+2CD)}) && \text{(A.4)} \\
\phi &= \phi_0 + \pi - \arccos(\frac{R^2C+D^2C+D}{R(1+2CD)}) \\
&= \Phi_0 - \frac{\pi}{2} + \pi - \arccos(\frac{R^2C+D^2C+D}{R(1+2CD)}) \\
&= \Phi_0 + \frac{\pi}{2} - \arccos(\frac{R^2C+D^2C+D}{R(1+2CD)}) && \text{(A.5)}
\end{aligned}$$

Due to the cosine law in the triangle Δ(OAB) (fig.: A.1) the following relation is valid:

$$\begin{aligned}
R^2 &= \frac{1}{4C^2} + (\frac{1}{2C}+D)^2 - \frac{\frac{1}{2C}+D}{C}\cos(\alpha) \\
\cos(\alpha) &= 1 - 2C^2\frac{R^2-D^2}{1+2CD} \\
\alpha &= \arccos(1-2C^2\frac{R^2-D^2}{1+2CD}) && \text{(A.6)} \\
\phi_0 + \frac{\pi}{2} &= \beta + \phi + \alpha \\
\beta &= \arccos(\frac{R^2C+D^2C+D}{R(1+2CD)}) - \pi + \frac{\pi}{2} - \arccos(1-2C^2\frac{R^2-D^2}{1+2CD}) \\
\beta &= \arccos(\frac{R^2C+D^2C+D}{R(1+2CD)}) - \frac{\pi}{2} + \mathrm{sgn}\arccos(1-2C^2\frac{R^2-D^2}{1+2CD}) && \text{(A.7)} \\
z &= z_0 + \cot(\theta)\Delta s \\
z &= z_0 + \cot(\theta)\arccos(1-2C^2\frac{R^2-D^2}{1+2CD})\frac{\mathrm{sgn}}{2C} && \text{(A.8)} \\
\theta &= \arctan(\cot(\theta)^{-1}) && \text{(A.9)} \\
\kappa &= 2C && \text{(A.10)}
\end{aligned}$$

case 2:

Due to the cosine law in the triangle Δ(BOA) (fig.: A.2) the following relation is valid:

$$
\begin{aligned}
\frac{1}{4C^2} &= (\frac{1}{2C}+D)^2 + R^2 - 2R(\frac{1}{2C}+D)cos(\phi_0-\phi) \\
\cos(\phi_0-\phi) &= \frac{R^2C+D^2C+D}{R(1+2CD)} && \text{(A.11)} \\
\phi_0-\phi &= \arccos(\frac{R^2C+D^2C+D}{R(1+2CD)}) && \text{(A.12)} \\
\phi &= \phi_0 - \arccos(\frac{R^2C+D^2C+D}{R(1+2CD)}) \\
&= \Phi_0 + \frac{\pi}{2} - \arccos(\frac{R^2C+D^2C+D}{R(1+2CD)}) && \text{(A.13)}
\end{aligned}
$$

Due to the cosine law in the triangle Δ(BAO) (fig.: A.2) the following relation is valid:

$$
\begin{aligned}
R^2 &= \frac{1}{4C^2} + (\frac{1}{2C}+D)^2 - \frac{\frac{1}{2C}+D}{C}\cos(\alpha) \\
\cos(\alpha) &= 1 - 2C^2\frac{R^2-D^2}{1+2CD} \\
\alpha &= \arccos(1-2C^2\frac{R^2-D^2}{1+2CD}) && \text{(A.14)} \\
\phi_0 - \frac{\pi}{2} &= \beta + \phi - \alpha \\
\beta &= \arccos(\frac{R^2C+D^2C+D}{R(1+2CD)}) - \frac{\pi}{2} + \arccos(1-2C^2\frac{R^2-D^2}{1+2CD}) \\
&= \arccos(\frac{R^2C+D^2C+D}{R(1+2CD)}) - \frac{\pi}{2} \\
&\quad +\mathrm{sgn}\arccos(1-2C^2\frac{R^2-D^2}{1+2CD}) && \text{(A.15)} \\
z &= z_0 + \cot(\theta)\Delta s \\
&= z_0 + \cot(\theta)\arccos(1-2C^2\frac{R^2-D^2}{1+2CD})\frac{\mathrm{sgn}}{2C} && \text{(A.16)} \\
\theta &= \arctan(\cot(\theta)^{-1}) && \text{(A.17)} \\
\kappa &= 2C && \text{(A.18)}
\end{aligned}
$$

case 3:

Due to the cosine law in the triangle Δ(BOA) (fig.: A.3) the following relation is valid:

$$\begin{aligned}
\frac{1}{4C^2} &= (\frac{1}{2C}+D)^2 + R^2 - 2R(\frac{-1}{2C} - D)cos(\pi - (\phi_0 - \phi)) \\
\cos(\pi - (\phi_0 - \phi)) &= -\frac{R^2C + D^2C + D}{R(1+2CD)} && \text{(A.19)} \\
\cos(-(\phi_0 - \phi)) &= \frac{R^2C + D^2C + D}{R(1+2CD)} && \text{(A.20)}
\end{aligned}$$

$(\phi - \phi_0) > 0$ and the range of values of arccos is $[0, \pi]$, so the sign has to be flipped.

$$\begin{aligned}
\phi - \phi_0 &= -\arccos(\frac{R^2C + D^2C + D}{R(1+2CD)}) \\
\phi &= \Phi_0 + \frac{\pi}{2} - \arccos(\frac{R^2C + D^2C + D}{R(1+2CD)}) && \text{(A.21)}
\end{aligned}$$

Due to the cosine law in the triangle Δ(OAB) (fig.: A.3) the following relation is valid:

$$\begin{aligned}
R^2 &= \frac{1}{4C^2} + (\frac{1}{2C} + D)^2 - \frac{\frac{1}{2C} + D}{C}\cos(\alpha) \\
\cos(\alpha) &= 2C^2\frac{R^2 - D^2}{1+2CD} - 1 \\
\cos(\pi + \alpha) &= 1 - 2C^2\frac{R^2 - D^2}{1+2CD} \\
\alpha &= \arccos(1 - 2C^2\frac{R^2 - D^2}{1+2CD}) - \pi && \text{(A.22)} \\
\phi_0 + \frac{3\pi}{2} &= \beta + \phi + \alpha \\
\beta &= \arccos(\frac{R^2C + D^2C + D}{R(1+2CD)}) - \frac{\pi}{2} + \arccos(1 - 2C^2\frac{R^2 - D^2}{1+2CD}) \\
&= \arccos(\frac{R^2C + D^2C + D}{R(1+2CD)}) - \frac{\pi}{2} && \text{(A.23)} \\
&\quad +\mathrm{sgn}\arccos(1 - 2C^2\frac{R^2 - D^2}{1+2CD}) && \text{(A.24)} \\
z &= z_0 + \cot(\theta)\Delta s \\
&= z_0 + \cot(\theta)\arccos(1 - 2C^2\frac{R^2 - D^2}{1+2CD})\frac{\mathrm{sgn}}{2C} && \text{(A.25)} \\
\theta &= \arctan(\cot(\theta)^{-1}) && \text{(A.26)} \\
\kappa &= 2C && \text{(A.27)}
\end{aligned}$$

case 4:

Due to the cosine law in the triangle Δ(AOB) (fig.: A.4) the following relation is valid:

$$\begin{aligned}
\frac{1}{4C^2} &= (\frac{1}{2C}+D)^2+R^2-2R(\frac{1}{2C}+D)cos(\pi-(\phi-\phi_0)) \\
\cos(\pi-(\phi-\phi_0)) &= \frac{R^2C+D^2C+D}{R(1+2CD)} \qquad \text{(A.28)} \\
\pi-(\phi-\phi_0) &= \arccos(\frac{R^2C+D^2C+D}{R(1+2CD)}) \\
\phi &= \Phi_0+\frac{\pi}{2}-\arccos(\frac{R^2C+D^2C+D}{R(1+2CD)}) \qquad \text{(A.29)}
\end{aligned}$$

Due to the cosine law in the triangle Δ(BAO) (fig:A.4) the following relation is valid:

$$\begin{aligned}
R^2 &= \frac{1}{4C^2}+(\frac{1}{2C}+D)^2-\frac{\frac{1}{2C}+D}{C}\cos(\alpha) \\
\cos(\alpha) &= 2C^2\frac{R^2-D^2}{1+2CD}-1 \\
\cos(\pi+\alpha) &= 1-2C^2\frac{R^2-D^2}{1+2CD} \\
\alpha &= \arccos(1-2C^2\frac{R^2-D^2}{1+2CD})-\pi \qquad \text{(A.30)} \\
\phi_0+\frac{\pi}{2}+\alpha &= \beta+\phi \\
\beta &= \arccos(\frac{R^2C+D^2C+D}{R(1+2CD)})-\frac{\pi}{2}+\arccos(1-2C^2\frac{R^2-D^2}{1+2CD}) \\
&= \arccos(\frac{R^2C+D^2C+D}{R(1+2CD)})-\frac{\pi}{2} \qquad \text{(A.31)} \\
&\quad +\mathrm{sgn}\arccos(1-2C^2\frac{R^2-D^2}{1+2CD}) \qquad \text{(A.32)} \\
z &= z_0+\cot(\theta)\Delta s \\
&= z_0+\cot(\theta)\arccos(1-2C^2\frac{R^2-D^2}{1+2CD})\frac{\mathrm{sgn}}{2C} \qquad \text{(A.33)} \\
\theta &= \arctan(\cot(\theta)^-1) \qquad \text{(A.34)} \\
\kappa &= 2C \qquad \text{(A.35)}
\end{aligned}$$

As shown before the transformation from perigee to fitter parameters can in all four cases be reduced to the same formalism:

$$\phi = \Phi_0 + \frac{\pi}{2} - \arccos(\frac{R^2C + D^2C + D}{R(1+2CD)}) \tag{A.36}$$

$$z = z_0 + \cot(\theta)\arccos(1 - 2C^2\frac{R^2 - D^2}{1+2CD})\frac{\mathrm{sgn}}{2C} \tag{A.37}$$

$$\beta = \arccos(\frac{R^2C + D^2C + D}{R(1+2CD)}) - \frac{\pi}{2} + \arccos(1 - 2C^2\frac{R^2 - D^2}{1+2CD}) \tag{A.38}$$

$$\theta = \arctan(\cot(\theta)^{-1}) \tag{A.39}$$

$$\kappa = 2C \tag{A.40}$$

The covariance matrix can be transformed via:

$$\Sigma^2_{fit} = \mathcal{D} * \Sigma^2_{perigee} * \mathcal{D}^{\mathrm{T}} \tag{A.41}$$

$$\tag{A.42}$$

$$\mathcal{D} = \begin{pmatrix} \frac{d\phi}{d\cot(\theta)} & \frac{d\phi}{dC} & \frac{d\phi}{dz_0} & \frac{d\phi}{dD} & \frac{d\phi}{d\phi_0} \\ \frac{dz}{d\cot(\theta)} & \frac{dz}{dC} & \frac{dz}{dz_0} & \frac{dz}{dD} & \frac{dz}{d\phi_0} \\ \frac{d\theta}{d\cot(\theta)} & \frac{d\theta}{dC} & \frac{d\theta}{dz_0} & \frac{d\theta}{dD} & \frac{d\theta}{d\phi_0} \\ \frac{d\beta}{d\cot(\theta)} & \frac{d\beta}{dC} & \frac{d\beta}{dz_0} & \frac{d\beta}{dD} & \frac{d\beta}{d\phi_0} \\ \frac{d\kappa}{d\cot(\theta)} & \frac{d\kappa}{dC} & \frac{d\kappa}{dz_0} & \frac{d\kappa}{dD} & \frac{d\kappa}{d\phi_0} \end{pmatrix} \tag{A.43}$$

$$= \begin{pmatrix} 0 & \frac{d\phi}{dC} & 0 & \frac{d\phi}{dD} & 1 \\ \frac{dz}{d\cot(\theta)} & \frac{dz}{dC} & 1 & \frac{dz}{dD} & 0 \\ \frac{d\theta}{d\cot(\theta)} & 0 & 0 & 0 & 0 \\ 0 & \frac{d\beta}{dC} & 0 & \frac{d\beta}{dD} & 0 \\ 0 & 2 & 0 & 0 & 0 \end{pmatrix} \tag{A.44}$$

The non trivial elements of $\mathcal{D}$ can be straight forward be derived via formulae A.37 - A.40.

$$\begin{aligned}
\frac{d\phi}{dC} &= \frac{1}{\sqrt{1-(\frac{R^2C+D^2C+D}{R(1+2CD)})^2}}\frac{R^2-D^2}{R(1+2CD)} && \text{(A.45)}\\
\frac{d\phi}{dD} &= \frac{1}{\sqrt{1-(\frac{R^2C+D^2C+D}{R(1+2CD)})^2}}(\frac{1}{R}-\frac{2(D+D^2C+R^2C)C}{R(1+2CD)^2}) && \text{(A.46)}\\
\frac{dz}{d\cot(\theta)} &= \frac{\text{sgn}}{2C}\arccos(1-2C^2\frac{R^2-D^2}{1+2CD}) && \text{(A.47)}\\
\frac{dz}{dC} &= -\cot(\theta)(\frac{dz}{d\cot(\theta)}\frac{1}{C} \\
&\quad -\text{sgn}\frac{2(R^2-D^2)(1+CD)}{(1+2CD)^2} * \frac{1}{\sqrt{1-(1-2C^2\frac{R^2-D^2}{1+2CD})^2}}) && \text{(A.48)}\\
\frac{dz}{dD} &= -\text{sgn} * \cot(\theta)\frac{2C(D+D^2C+R^2C)}{(1+2CD)^2} \\
&\quad * \frac{1}{\sqrt{1-(1-2C^2\frac{R^2-D^2}{1+2CD})^2}} && \text{(A.49)}\\
\frac{d\theta}{d\cot(\theta)} &= -\frac{1}{1+\cot^2(\theta)} && \text{(A.50)}\\
\frac{d\beta}{dC} &= \text{sgn}\frac{4C(1+CD)(R^2-D^2)}{(1+2CD)^2} \\
&\quad * \frac{1}{\sqrt{1-(1-2C^2\frac{R^2-D^2}{1+2CD})^2}} - \frac{d\phi}{dC} && \text{(A.51)}\\
\frac{d\beta}{dD} &= \frac{4C(D+D^2C+R^2C)}{(1+2CD)^2} * \frac{1}{\sqrt{1-(1-2C^2\frac{R^2-D^2}{1+2CD})^2}} - \frac{d\phi}{dD} && \text{(A.52)}
\end{aligned}$$

A.2 Fitter Parameters → Perigee Parameters

In this section the perigee parameters are expressed in terms of the fitter parameters. We use again the distinction into the four cases, we made above and refer again to the same sketches.

case 1:

$$
\begin{aligned}
B_x &= R\cos(\phi) && \text{(A.53)}\\
B_y &= R\sin(\phi) && \text{(A.54)}\\
A_x &= B_x - \frac{\cos(\phi+\beta-\frac{\pi}{2}}{\kappa}) && \text{(A.55)}\\
A_y &= B_y - \frac{\sin(\phi+\beta-\frac{\pi}{2})}{\kappa} && \text{(A.56)}\\
\tan(\phi_0) &= \frac{A_y}{A_x} && \text{(A.57)}\\
\Phi_0 &= \arctan(\frac{R\sin(\phi)+\cos(\phi+\beta)\kappa^{-1}}{R\cos(\phi)-\sin(\phi+\beta)\kappa^{-1}}) - \mathrm{sgn}\frac{\pi}{2} && \text{(A.58)}
\end{aligned}
$$

Due to the cosine law in the triangle Δ(ABO) (fig.: A.1) the following relation is valid:

$$
\begin{aligned}
(-D-\frac{1}{\kappa})^2 &= R^2+\frac{1}{\kappa^2}+2\frac{R}{\kappa}\cos(\frac{\pi}{2}+\beta) &&\\
D &= -\frac{1}{\kappa}+\mathrm{sgn}\sqrt{R^2+\frac{1}{\kappa^2}-\frac{2R}{\kappa}\sin(\beta)} && \text{(A.59)}\\
z_0 &= z-\cot(\theta)*\Delta s &&\\
&= z-\cot(\theta)*\frac{\alpha}{\kappa} && \text{(A.60)}\\
\beta+\phi &= \Phi_0-\alpha && \text{(A.61)}\\
z_0 &= z-\frac{\cot(\theta)}{\kappa}((\phi+\beta-\Phi_0+\pi)\mathrm{mod}(2\pi)-\pi) && \text{(A.62)}\\
\cot(\theta) &= \cot(\theta) && \text{(A.63)}\\
C &= \frac{\kappa}{2} && \text{(A.64)}
\end{aligned}
$$

case 2:

$$B_x = R\cos(\phi) \tag{A.65}$$
$$B_y = R\sin(\phi) \tag{A.66}$$
$$A_x = B_x + \frac{\cos(\phi+\beta+\frac{\pi}{2})}{\kappa} \tag{A.67}$$
$$A_y = B_y + \frac{\sin(\phi+\beta+\frac{\pi}{2})}{\kappa} \tag{A.68}$$

$$\tan(\phi_0) = \frac{A_y}{A_x} \tag{A.69}$$
$$\Phi_0 = \arctan(\frac{R\sin(\phi)+\cos(\phi+\beta)\kappa^{-1}}{R\cos(\phi)-\sin(\phi+\beta)\kappa^{-1}}) - \mathrm{sgn}\frac{\pi}{2} \tag{A.70}$$

Due to the cosine law in the triangle Δ(OBA) (fig.: A.2) the following relation is valid:

$$(D+\frac{1}{\kappa})^2 = R^2 - \frac{1}{\kappa^2} + 2\frac{R}{\kappa}\cos(\frac{\pi}{2}+\beta) \tag{A.71}$$
$$D = -\frac{1}{\kappa} + \mathrm{sgn}\sqrt{R^2 + \frac{1}{\kappa^2} - \frac{2R}{\kappa}\sin(\beta)} \tag{A.72}$$
$$z_0 = z - \cot(\theta) * \Delta s$$
$$= z - \cot(\theta) * \frac{\alpha}{\kappa} \tag{A.73}$$
$$\beta + \phi = \Phi_0 + \alpha \tag{A.74}$$
$$z_0 = z - \frac{\cot(\theta)}{\kappa}((\phi+\beta-\Phi_0+\pi)\mathrm{mod}(2\pi)-\pi) \tag{A.75}$$
$$\cot(\theta) = \cot(\theta) \tag{A.76}$$
$$C = \frac{\kappa}{2} \tag{A.77}$$

case 3:

$$
\begin{aligned}
B_x &= R\cos(\phi) && \text{(A.78)}\\
B_y &= R\sin(\phi) && \text{(A.79)}\\
A_x &= B_x + \frac{\sin(2\pi - (\phi+\beta))}{\kappa} && \text{(A.80)}\\
A_y &= B_y + \frac{\cos(2\pi - (\phi+\beta))}{\kappa} && \text{(A.81)}\\
\tan(\phi_0) &= \frac{A_y}{A_x} && \text{(A.82)}\\
\Phi_0 &= \arctan(\frac{R\sin(\phi) + \cos(\phi+\beta)\kappa^{-1}}{R\cos(\phi) - \sin(\phi+\beta)\kappa^{-1}}) - \mathrm{sgn}\frac{\pi}{2} && \text{(A.83)}
\end{aligned}
$$

Due to the cosine law in the triangle Δ(OBA) (fig.: A.3) the following relation is valid:

$$
\begin{aligned}
(-D - \frac{1}{\kappa})^2 &= R^2 - \frac{1}{\kappa^2} + 2\frac{R}{\kappa}\cos(\frac{\pi}{2} + \beta) && \text{(A.84)}\\
-D - \frac{1}{\kappa} &= \sqrt{R^2 + \frac{1}{\kappa^2} - \frac{2R}{\kappa}\sin(\beta)} &&\\
D &= -\frac{1}{\kappa} + \mathrm{sgn}\sqrt{R^2 + \frac{1}{\kappa^2} - \frac{2R}{\kappa}\sin(\beta)} && \text{(A.85)}\\
z_0 &= z - \cot(\theta) * \Delta s &&\\
&= z - \cot(\theta) * \frac{\alpha}{\kappa} && \text{(A.86)}\\
\beta + \phi &= \Phi_0 + \alpha && \text{(A.87)}\\
z_0 &= z - \frac{\cot(\theta)}{\kappa}((\phi + \beta - \Phi_0 + \pi)\mathrm{mod}(2\pi) - \pi) && \text{(A.88)}\\
\cot(\theta) &= \cot(\theta) && \text{(A.89)}\\
C &= \frac{\kappa}{2} && \text{(A.90)}
\end{aligned}
$$

case 4:

$$
\begin{aligned}
B_x &= R\cos(\phi) && \text{(A.91)}\\
B_y &= R\sin(\phi) && \text{(A.92)}\\
A_x &= B_x - \frac{\sin(\pi-(\phi+\beta))}{\kappa} && \text{(A.93)}\\
A_y &= B_y - \frac{\cos(\pi-(\phi+\beta))}{\kappa} && \text{(A.94)}\\
\tan(\phi_0) &= \frac{A_y}{A_x} && \text{(A.95)}\\
\Phi_0 &= \arctan(\frac{R\sin(\phi)+\cos(\phi+\beta)\kappa^{-1}}{R\cos(\phi)-\sin(\phi+\beta)\kappa^{-1}}) - \mathrm{sgn}\frac{\pi}{2} && \text{(A.96)}
\end{aligned}
$$

Due to the cosine law in the triangle Δ(AOB) (fig.: A.4) the following relation is valid:

$$
\begin{aligned}
(D+\frac{1}{\kappa})^2 &= R^2 - \frac{1}{\kappa^2} - 2\frac{R}{\kappa}\cos(\frac{\pi}{2}-\beta) && \text{(A.97)}\\
D+\frac{1}{\kappa} &= \sqrt{R^2+\frac{1}{\kappa^2}-\frac{2R}{\kappa}\sin(\beta)} &&\\
D &= -\frac{1}{\kappa} + \mathrm{sgn}\sqrt{R^2+\frac{1}{\kappa^2}-\frac{2R}{\kappa}\sin(\beta)} && \text{(A.98)}\\
z_0 &= z - \cot(\theta) * \Delta s &&\\
&= z - \cot(\theta) * \frac{\alpha}{\kappa} && \text{(A.99)}\\
\beta+\phi &= \Phi_0 + \alpha && \text{(A.100)}\\
z_0 &= z - \frac{\cot(\theta)}{\kappa}((\phi+\beta-\Phi_0+\pi)\mathrm{mod}(2\pi)-\pi) && \text{(A.101)}\\
\cot(\theta) &= \cot(\theta) && \text{(A.102)}\\
C &= \frac{\kappa}{2} && \text{(A.103)}
\end{aligned}
$$

As shown before the transformation from fitter to perigee parameters can in all 4 cases be reduced to the same formalism:

$$
\begin{aligned}
D &= -\frac{1}{\kappa} + \mathrm{sgn}\sqrt{R^2 + \frac{1}{\kappa^2} - \frac{2R}{\kappa}\sin(\beta)} && \text{(A.104)}\\
\Phi_0 &= \arctan\left(\frac{R\sin(\phi) + \frac{\cos(\phi+\beta)}{\kappa}}{R\cos(\phi) - \frac{\sin(\phi+\beta)}{\kappa}}\right) - \mathrm{sgn}\frac{\pi}{2} && \text{(A.105)}\\
\cot(\theta) &= \cot(\theta) && \text{(A.106)}\\
z_0 &= z - \frac{\cot(\theta)}{\kappa}((\phi + \beta - \Phi_0 + \pi)\mathrm{mod}(2\pi) - \pi) && \text{(A.107)}\\
C &= \frac{\kappa}{2} && \text{(A.108)}
\end{aligned}
$$

The covariance matrix can be transformed via:

$$
\begin{aligned}
\Sigma^2_{perigee} &= \mathcal{D} * \Sigma^2_{fit} * \mathcal{D}^{\mathrm{T}} && \text{(A.109)}\\
& && \text{(A.110)}\\
\mathcal{D} &= \begin{pmatrix}
\frac{d\cot(\theta)}{d\phi} & \frac{d\cot(\theta)}{dz} & \frac{d\cot(\theta)}{d\theta} & \frac{d\cot(\theta)}{d\beta} & \frac{d\cot(\theta)}{d\kappa}\\
\frac{dC}{d\phi} & \frac{dC}{dz} & \frac{dC}{d\theta} & \frac{dC}{d\beta} & \frac{dC}{d\kappa}\\
\frac{dz_0}{d\phi} & \frac{dz_0}{dz} & \frac{dz_0}{d\theta} & \frac{dz_0}{d\beta} & \frac{dz_0}{d\kappa}\\
\frac{dD}{d\phi} & \frac{dD}{dz} & \frac{dD}{d\theta} & \frac{dD}{d\beta} & \frac{dD}{d\kappa}\\
\frac{d\phi_0}{d\phi} & \frac{d\phi_0}{dz} & \frac{d\phi_0}{d\theta} & \frac{d\phi_0}{d\beta} & \frac{d\phi_0}{d\kappa}
\end{pmatrix} && \text{(A.111)}\\
&= \begin{pmatrix}
0 & 0 & \frac{d\cot(\theta)}{d\theta} & 0 & 0\\
0 & 0 & 0 & 0 & \frac{1}{2}\\
0 & 1 & \frac{dz_0}{d\theta} & \frac{dz_0}{d\beta} & \frac{dz_0}{d\kappa}\\
0 & 0 & 0 & \frac{dD}{d\beta} & \frac{dD}{d\kappa}\\
1 & 0 & 0 & \frac{d\phi_0}{d\beta} & \frac{d\phi_0}{d\kappa}
\end{pmatrix} && \text{(A.112)}
\end{aligned}
$$

The non trivial elements of $\mathcal{D}$ can be derived straight forward via formulae A.104 - A.108.

$$\frac{d\cot(\theta)}{d\theta} = -\frac{1}{\sin^2(\theta)} \tag{A.113}$$

$$\frac{dz_0}{d\theta} = \frac{1}{\sin^2(\theta)} * \frac{1}{\kappa}((\Phi_0 - \phi - \beta + \pi)\mathrm{mod}(2\pi) - \pi) \tag{A.114}$$

$$\frac{dz_0}{d\beta} = \frac{\cot(\theta)}{\kappa}(\frac{d\Phi_0}{d\beta} - 1) \tag{A.115}$$

$$\frac{dz_0}{d\kappa} = -\frac{\cot(\theta)}{\kappa^2}((\Phi_0 - \phi - \beta + \pi)\mathrm{mod}(2\pi) - \pi) \tag{A.116}$$

$$\frac{dD}{d\beta} = -\mathrm{sgn}\frac{R\cos(\beta)}{\kappa\sqrt{R^2 + \frac{1}{\kappa^2} - \frac{2R}{\kappa}\sin(\beta)}} \tag{A.117}$$

$$\frac{dD}{d\kappa} = \frac{1}{\kappa^2} - \mathrm{sgn}\frac{\kappa^{-1} - R\sin(\beta)}{\kappa^2\sqrt{R^2 + \frac{1}{\kappa^2} - \frac{2R}{\kappa}\sin(\beta)}} \tag{A.118}$$

$$\frac{d\phi_0}{d\beta} = \frac{\kappa^{-1} - R\sin(\beta)}{\kappa(R^2 + \kappa^{-2} - 2R\kappa^{-1}\sin(\beta))} \tag{A.119}$$

$$\frac{d\phi_0}{d\kappa} = \frac{-R(\cos(\phi + \beta)\cos(\phi) + \sin(\phi + \beta)\sin(\phi))}{\kappa^2(R^2 + \kappa^{-2} + 2R\kappa^{-1}\sin(\beta))} \tag{A.120}$$

Appendix B

Formulae for Describing the Fitter

B.1 Transport Matrix

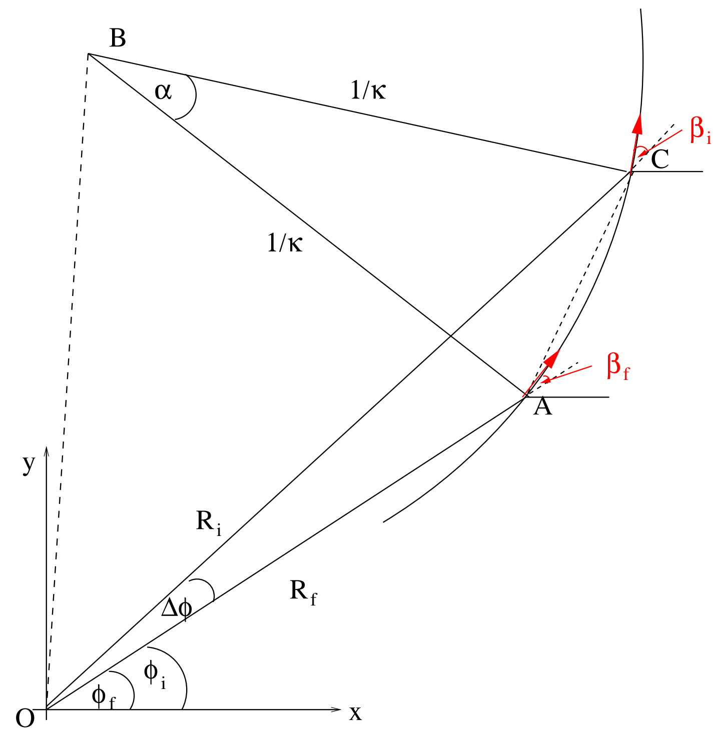


Figure B.1: Sketch of helix parameters.

$$\begin{aligned}
\delta\vec{x}(R_f) &= \mathcal{T}(R_i,R_f) * \delta\vec{x}(R_i) && \text{(B.1)}\\
\Sigma^2_{\vec{x}}(R_f) &= \mathcal{T}(R_i,R_f) * \Sigma^2_{\vec{x}}(R_i) * \mathcal{T}^{\mathrm{T}}(R_i,R_f) && \text{(B.2)}\\
\Omega_{\vec{x}}(R_f) &= (\Sigma^2_{\vec{x}}(R_f))^{-1}\\
&= (\mathcal{T}(R_i,R_f) * \Sigma^2_{\vec{x}}(R_i) * \mathcal{T}^{\mathrm{T}}(R_i,R_f))^{-1}\\
&= (\mathcal{T}^{\mathrm{T}}(R_i,R_f))^{-1} * (\Sigma^2_{\vec{x}}(R_i))^{-1} * (\mathcal{T}(R_i,R_f))^{-1}\\
&= \mathcal{T}^{\mathrm{T}}(R_f,R_i) * \Omega^2_{\vec{x}}(R_i) * \mathcal{T}(R_f,R_i) && \text{(B.3)}
\end{aligned}$$

A lot of the elements in the transport matrix are trivial either 1 or 0. All non trivial elements are derived in this section:

$$\mathcal{T}(R_i,R_f) = \begin{pmatrix}
\frac{d\phi_f}{d\phi_i} & \frac{d\phi_f}{dz_i} & \frac{d\phi_f}{d\theta_i} & \frac{d\phi_f}{d\beta_i} & \frac{d\phi_f}{d\kappa_i}\\
\frac{dz_f}{d\phi_i} & \frac{dz_f}{dz_i} & \frac{dz_f}{d\theta_i} & \frac{dz_f}{d\beta_i} & \frac{dz_f}{d\kappa_i}\\
\frac{d\theta_f}{d\phi_i} & \frac{d\theta_f}{dz_i} & \frac{d\theta_f}{d\theta_i} & \frac{d\theta_f}{d\beta_i} & \frac{d\theta_f}{d\kappa_i}\\
\frac{d\beta_f}{d\phi_i} & \frac{d\beta_f}{dz_i} & \frac{d\beta_f}{d\theta_i} & \frac{d\beta_f}{d\beta_i} & \frac{d\beta_f}{d\kappa_i}\\
\frac{d\kappa_f}{d\phi_i} & \frac{d\kappa_f}{dz_i} & \frac{d\kappa_f}{d\theta_i} & \frac{d\kappa_f}{d\beta_i} & \frac{d\kappa_f}{d\kappa_i}
\end{pmatrix} \quad \text{(B.4)}$$

$$= \begin{pmatrix}
1 & 0 & 0 & \mathrm{der1} & \mathrm{der2}\\
0 & 1 & \mathrm{der3} & \mathrm{der4} & \mathrm{der5}\\
0 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & \mathrm{der6} & \mathrm{der7}\\
0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

$$\begin{aligned}
\alpha &= -\Delta\phi\\
&= \phi_i + \beta_i - \phi_f - \beta_f && \text{(B.5)}\\
\Delta\Phi &= \phi_i - \phi_f; && \text{(B.6)}
\end{aligned}$$

Due to the cosine law in $\Delta(OAB)$ and $\Delta(OCB)$ the following relation is valid:

$$\begin{aligned} R_i^2 + \frac{1}{\kappa^2} + \frac{2R_i}{\kappa}\cos(\pi - \beta_i) &= R_f^2 + \frac{1}{\kappa^2} + \frac{2R_f}{\kappa}\cos(\pi - \beta_f) && \text{(B.7)} \\ \sin(\beta_f) &= \frac{\kappa}{2R_f}(R_i^2 - R_f^2) + \frac{R_i}{R_f}\sin(\beta_i) && \text{(B.8)} \end{aligned}$$

Derive for β_i or κ:

$$\begin{aligned} \text{der6} &= \frac{\delta\beta_f}{\delta\beta_i} \\ &= \frac{R_i\cos(\beta_i)}{R_f\cos(\beta_f)} && \text{(B.9)} \\ \text{der7} &= \frac{\delta\beta_f}{\delta\kappa} && \text{(B.10)} \\ &= \frac{R_i^2 - R_f^2}{2R_f\cos(\beta_f)} && \text{(B.11)} \end{aligned}$$

Due to the cosine law in $\Delta(OAC)$ and $\Delta(BAC)$ the following relation is valid:

$$R_i^2 + R_f^2 - 2R_iR_f\cos(\Delta\Phi) = \frac{1}{\kappa^2} + \frac{1}{\kappa^2} - \frac{2}{\kappa^2}\cos(\Delta\phi) \qquad \text{(B.12)}$$

Derive for β_i or κ:

$$\begin{aligned} \text{der1} &= \frac{\delta\phi_f}{\delta\beta_i} \\ &= \frac{\frac{\sin(\Delta\phi)}{\kappa^2}(1 - \frac{\delta\beta_f}{\delta\beta_i})}{\frac{\sin(\Delta\phi)}{\kappa^2} + R_iR_f\sin(\Delta\Phi)} \\ &= \frac{\frac{\sin(\Delta\phi)}{\kappa^2}(1 - \text{der6})}{\frac{\sin(\Delta\phi)}{\kappa^2} + R_iR_f\sin(\Delta\Phi)} && \text{(B.13)} \\ \text{der2} &= \frac{\delta\phi_f}{\delta\kappa} \\ &= \frac{\frac{2}{\kappa^3}(1 - \cos(\Delta\phi)) + \frac{\sin(\Delta\phi)}{\kappa^2}\frac{\delta\beta_f}{\delta\kappa}}{\frac{\sin(\Delta\phi)}{\kappa^2} + R_iR_f\sin(\Delta\Phi)} \\ &= \frac{\frac{2}{\kappa^3}(1 - \cos(\Delta\phi)) + \frac{\sin(\Delta\phi)}{\kappa^2}\text{der7}}{\frac{\sin(\Delta\phi)}{\kappa^2} + R_iR_f\sin(\Delta\Phi)} && \text{(B.14)} \end{aligned}$$

$$\begin{aligned} z_f &= z_i + \Delta s \cot(\theta) \\ &= z_i + \frac{\Delta\phi}{\kappa}\cot(\theta) \end{aligned} \tag{B.15}$$

Derive for θ, β_f or κ:

$$\begin{aligned} \text{der3} &= \frac{\delta z_f}{\delta\theta} \\ &= -\frac{\Delta\phi}{\kappa}(1+\cot^2(\theta)) \\ \text{der4} &= \frac{\delta z_f}{\delta\beta_i} \end{aligned} \tag{B.16}$$

$$\begin{aligned} &= \frac{\frac{\delta\phi_f}{\delta\beta_i} + \frac{\delta\beta_f}{\delta\beta_i} - 1}{\kappa}\cot(\theta) \\ &= \frac{\text{der1} + \text{der6} - 1}{\kappa}\cot(\theta) \end{aligned} \tag{B.17}$$

$$\begin{aligned} \text{der5} &= \frac{\delta z_f}{\delta\kappa} \\ &= \frac{\frac{\delta\phi_f}{\kappa} + \frac{\delta\beta_f}{\delta\kappa} - \Delta\phi}{\kappa}\cot(\theta) \\ &= \frac{\text{der2} + \text{der7} - \Delta\phi}{\kappa}\cot(\theta) \end{aligned} \tag{B.18}$$

B.2 Extrapolating Track Parameters to Different Radii

Again we start with distinguishing four different cases which can later be shown to end up in the same formulae. The different cases are:

- outward extrapolation and $\kappa > 0$ (fig.: B.2),
- inward extrapolation and $\kappa > 0$ (fig.: B.3),
- outward extrapolation and $\kappa < 0$ (fig.: B.4),
- inward extrapolation and $\kappa < 0$ (fig.: B.5).

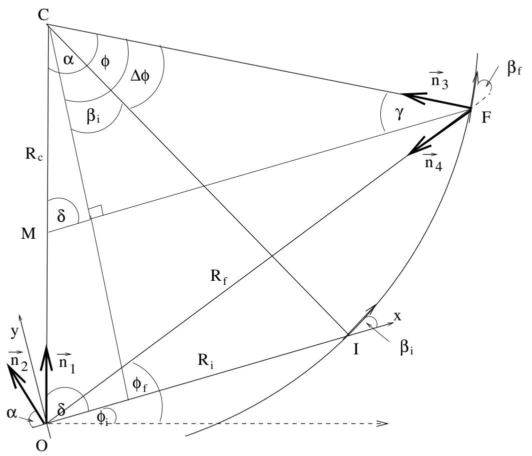


Figure B.2: Outward extrapolation with helix curvature $\kappa > 0$.

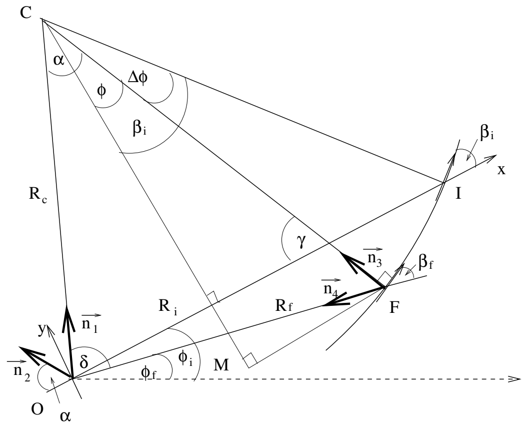


Figure B.3: Inward extrapolation with helix curvature $\kappa > 0$.

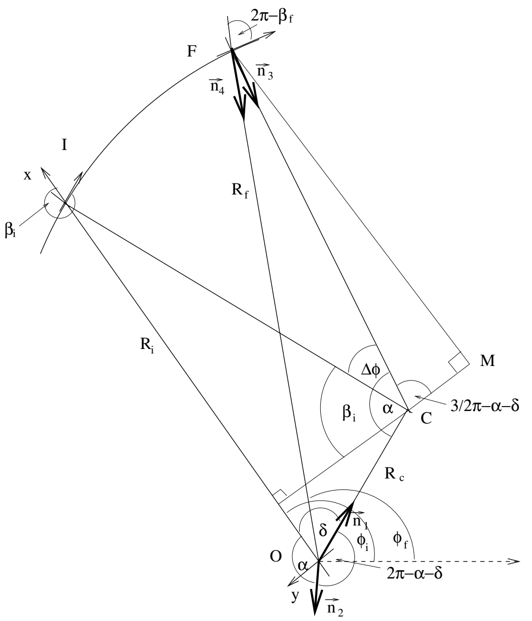


Figure B.4: Outward extrapolation with helix curvature $\kappa < 0$.

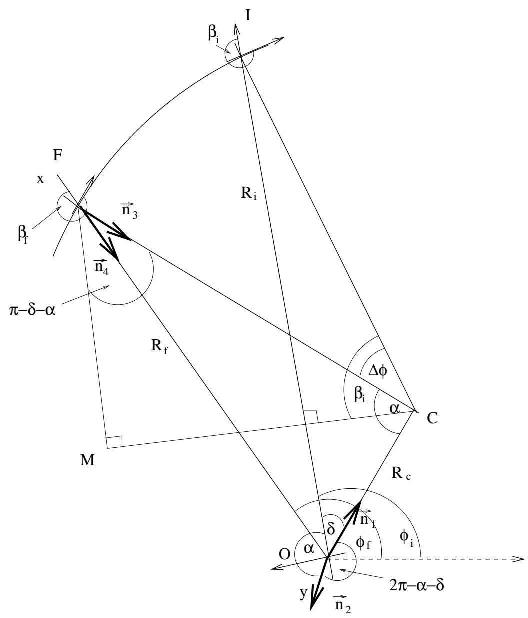


Figure B.5: Inward extrapolation with helix curvature $\kappa < 0$.

In all the four cases the calculations are derived in a rotated coordinate system. The origin remains fix, and the axis are rotated by the angle ϕ of the starting set of parameters (ϕ_i).

case1/case2:

The coordinates of the center of the circle C are the following:

$$x_c = r_i - \frac{\sin(\beta_i)}{\kappa} \tag{B.19}$$

$$y_c = \frac{\cos(\beta_i)}{\kappa} \tag{B.20}$$

$$r_c = \sqrt{x_c^2 + y_c^2} \tag{B.21}$$

Due to the law of cosine in the triangle $\Delta(OFC)$ (fig.: B.2/B.3) the following relation is valid:

$$R_f^2 = r_c^2 + \frac{1}{\kappa^2} - 2\frac{r_c}{\kappa}\cos(\alpha) \tag{B.22}$$

$$-\cos(\alpha) * r_c = (R_f^2 - r_c^2 - \frac{1}{\kappa^2})/(\frac{2}{\kappa}) \tag{B.23}$$

$$\sin(\alpha) * r_c = \sqrt{r_c^2 - (\cos(\alpha) * r_c)^2} \tag{B.24}$$

By calculating the scalar product of the normalized vectors $\vec{n}_1$ and $\vec{n}_2$ and of $\vec{n}_1$ and $\vec{n}_2^{\perp}$ (fig.: B.2/B.3), $\sin(\phi)$, $\cos(\phi)$, $\sin(\gamma)$ and $\cos(\gamma)$ can be derived.

$$\begin{aligned} \cos(\gamma) &= \vec{n}_1 * \vec{n}_2 \\ &= \frac{(-x_c * \cos(\alpha) * r_c) + (y_c * \sin(\alpha) * r_c)}{r_c^2} \end{aligned} \tag{B.25}$$

$$\sin(\phi) = \cos(\gamma) \tag{B.26}$$

$$\begin{aligned} \cos(\gamma - \frac{\pi}{2}) &= \vec{n}_1 * \vec{n}_2^{\perp} \\ &= \frac{(x_c * \sin(\alpha) * r_c) + (y_c * \cos(\alpha) * r_c)}{r_c^2} \end{aligned} \tag{B.27}$$

$$\sin(\gamma) = \cos(\gamma - \frac{\pi}{2}) \tag{B.28}$$

$$\cos(\phi) = \sin(\gamma) \tag{B.29}$$

The coordinates of F (x_f, y_f) can now be expressed in the following way:

$$x_f = x_c + \frac{\sin(\phi)}{\kappa} \tag{B.30}$$

$$y_f = y_c - \frac{\cos(\phi)}{\kappa} \tag{B.31}$$

By calculating the scalar product of the normalized vectors $\vec{n}_3$ and $\vec{n}_4$ (fig.: B.2/B.3), $\sin(\beta_f)$ can be derived.

$$\begin{aligned} \sin(\beta_f) &= \cos(\beta_f - \frac{\pi}{2}) \\ &= \vec{n}_3 * \vec{n}_4 \\ &= \frac{\sin(\phi) * x_f - \cos(\phi) * y_f}{R_f} \\ &= \end{aligned} \tag{B.32}$$

$$\phi = \arctan(\sin(\phi), \cos(\phi)) \tag{B.33}$$

$$\Delta\phi = \phi - \beta_i \tag{B.34}$$

$$\Delta s = \frac{\Delta\phi}{\kappa} \tag{B.35}$$

$$z_f = z_i + \cot(\theta) * \Delta s \tag{B.36}$$

$$\phi_f - \phi_i = \arctan(y_f, z_f) \tag{B.37}$$

$$\phi_f = \phi_i + \arctan(y_f, z_f) \tag{B.38}$$

$$\theta_f = \theta_i \tag{B.39}$$

$$\beta_f = \phi - \arctan(y_f, z_f) \tag{B.40}$$

$$\kappa_f = \kappa_i \tag{B.41}$$

case3/case4:

The coordinates of the center of the circle C are the following:

$$x_c = r_i - \frac{\sin(\beta_i)}{\kappa} \tag{B.42}$$

$$y_c = \frac{\cos(\beta_i)}{\kappa} \tag{B.43}$$

$$r_c = \sqrt{x_c^2 + y_c^2} \tag{B.44}$$

Due to the law of cosine in the triangle $\Delta(OFC)$ (fig.: B.4/B.5) the following relation is valid:

$$R_f^2 = r_c^2 + \frac{1}{\kappa^2} + 2\frac{r_c}{\kappa}\cos(\alpha) \tag{B.45}$$

$$\cos(\alpha) * r_c = (R_f^2 - r_c^2 - \frac{1}{\kappa^2})/(\frac{2}{\kappa}) \tag{B.46}$$

$$\sin(\alpha) * r_c = \sqrt{r_c^2 - (\cos(\alpha) * r_c)^2} \tag{B.47}$$

By calculating the scalar product of the normalized vectors $\vec{n}_1$ and $\vec{n}_2$ and of $\vec{n}_1$ and $\vec{n}_2^{\perp}$ (fig.: B.4/B.5) $\cos(2\pi - \alpha - \delta)$ and $\sin(2\pi - \alpha - \delta)$ can be derived.

$$\begin{aligned} \cos(\alpha+\delta) &= \vec{n}_1 * \vec{n}_2 \\ &= \frac{(-x_c * \cos(\alpha) * r_c) + (y_c * \sin(\alpha) * r_c)}{r_c^2} \qquad \text{(B.48)} \\ \sin(\alpha+\delta) &= \vec{n}_1 * \vec{n}_2^{\perp} \\ &= \frac{(x_c * \sin(\alpha) * r_c) + (y_c * \cos(\alpha) * r_c)}{r_c^2} \qquad \text{(B.49)} \\ & \qquad \text{(B.50)} \end{aligned}$$

The coordinates of F (x_f, y_f) can now be expressed in the following way:

$$\begin{aligned} x_f &= x_c - \frac{\sin(\frac{3\pi}{2} - \alpha - \delta)}{\kappa} \\ &= x_c + \frac{\cos(\alpha+\delta)}{\kappa} \qquad \text{(B.51)} \\ y_f &= y_c - \frac{\cos(\frac{3\pi}{2} - \alpha - \delta)}{\kappa} \\ y_f &= y_c - \frac{\sin(\alpha+\delta)}{\kappa} \qquad \text{(B.52)} \\ & \qquad \text{(B.53)} \end{aligned}$$

By calculating the scalar product of the normalized vectors $\vec{n}_3$ and $\vec{n}_4$ (fig.: B.4/B.5), $\sin(\beta_f)$ can be derived.

$$\begin{aligned}
\sin(\beta_f) &= \cos(\beta_f - \frac{\pi}{2}) \\
&= \vec{n}_3 * \vec{n}_4 \\
&= \frac{\cos(\alpha+\delta) * x_f - \sin(\alpha+\delta) * y_f}{R_f} && \text{(B.54)} \\
&= \\
\alpha + \delta - \frac{\pi}{2} &= \arctan(\cos(\alpha+\delta), \sin(\alpha+\delta)) && \text{(B.55)} \\
\Delta\phi &= \arctan(\cos(\alpha+\delta), \sin(\alpha+\delta)) - \beta_i && \text{(B.56)} \\
\Delta s &= \frac{\Delta\phi}{\kappa} && \text{(B.57)} \\
z_f &= z_i + \cot(\theta) * \Delta s && \text{(B.58)} \\
\phi_f - \phi_i &= \arctan(y_f, z_f) && \text{(B.59)} \\
\phi_f &= \phi_i + \arctan(y_f, z_f) && \text{(B.60)} \\
\theta_f &= \theta_i && \text{(B.61)} \\
\beta_f &= \arctan(\cos(\alpha+\delta), \sin(\alpha+\delta)) - \arctan(y_f, z_f) && \text{(B.62)} \\
\kappa_f &= \kappa_i && \text{(B.63)}
\end{aligned}$$

B.3 Matrix Inversion

The matrix inversion are calculated following the Cholesky decomposition. This is a special LU decomposition for matrices which are symmetric and positive definite. This algorithm is described e.g. in reference [25, 41].

B.4 Multiple Scattering through Small Angles

A charged particle traversing a medium is deflected by many small-angle scatters. Most of this deflection is due to Coulomb scattering and can be described via a Gaussian distribution for small scattering angels [27].

$$\sigma_{ms} = \frac{13.6MeV}{\beta cp} z \sqrt{\frac{x}{X_0}} (1 + 0.038 \ln(\frac{x}{X_0})) \tag{B.64}$$

where:

βc	:	velocity of the incident particle,
p	:	momentum of the incident particle,
z	:	charge number of the incident particle,
x	:	path length of the incident particle in the medium,
X_0	:	radiation length of the medium.

$$p = \frac{p_t}{\sin(\theta)} = \frac{B_{Field} * CC}{\sin(\theta)\kappa} \tag{B.65}$$

where:

p_t	:	transverse momentum,
B_{Field}	:	-1.4116 T,
CC	:	curvature constant 0.0029979 $\frac{GeV cm}{T}$.

$$\kappa = \frac{B_{Field} * CC}{\sin(\theta) p} \tag{B.66}$$

$$\frac{\delta\kappa}{\delta\theta} = \frac{B_{Field} CC \cos(\theta)}{\sin^2(\theta) p} \tag{B.67}$$

B.5 Ionization Energy Loss by Heavy Particles

Particles passing through a material, lose energy primarily by ionization. The mean rate of energy loss is given by the Bethe Bloch equation [27]:

$$-\frac{dE}{dx} = Kz^2 \frac{Z}{A} \frac{1}{\beta^2} (\frac{1}{2} \ln(\frac{2m_e c^2 \beta^2 \gamma^2 T_{max}}{I^2}) - \beta^2 - \frac{\rho}{2}) \tag{B.68}$$

K	:	$0.307075\ \frac{MeVcm^2}{g}$,
z	:	charge number of the incident particle,
Z	:	atomic number of the medium,
A	:	atomic mass of the medium,
$m_e c^2$	:	electron rest energy (511 keV),
T_{max}	:	maximum kinetic energy [MeV],
I	:	mean excitation energy [eV],
β	:	speed of the incident particle divided by the speed of light,
γ	:	relativistic factor,
δ	:	density-effect correction.

All factors are put in the so called material dE/dx constant, and so the mean energy loss of a particle with $z = 1$ in a material with the mean excitation energy I is given by

$$E_{loss} = x\frac{dEdxConst}{\beta^2}\ln(\frac{2m_e\beta^2\gamma^2}{I}) - \beta^2 - \frac{\rho}{2}); \tag{B.69}$$

The errors of the dE/dx correction and its contribution to the error on κ are neglected for the Kalman fitter of the TrackingKal package due to its relative smallness.

Appendix C

Formulae for Silicon Standalone Tracking

C.1 Creating of the Reference

According to the sketch C.1 the center (x_c, y_c) of the circle can be calculated:

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} = \frac{1}{2}\begin{pmatrix} x_2 + x_1 \\ y_2 + y_1 \end{pmatrix} + m * \begin{pmatrix} y_2 - y_1 \\ x_2 - x_1 \end{pmatrix} \tag{C.1}$$

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} = \frac{1}{2}\begin{pmatrix} x_2 + x_0 \\ y_2 + y_0 \end{pmatrix} + n * \begin{pmatrix} y_2 - y_0 \\ x_2 - x_0 \end{pmatrix} \tag{C.2}$$

$$m = \frac{x_0 - x_1}{2(y_2 - y_1)} + n\frac{y_2 - y_0}{y_2 - y_0} \tag{C.3}$$

$$n = \frac{(y_0 - y_1)(y_1 - y_2) - (x_2 - x_1)(x_0 - x_1)}{2((x_0 - x_2)(y_1 - y_2) + (x_2 - x_1)(y_0 - y_2))} \tag{C.4}$$

The radius of the circle can then be deduced:

$$R = \sqrt{(x_1 - x_c)^2 + (y_1 - y_c)^2} \tag{C.5}$$

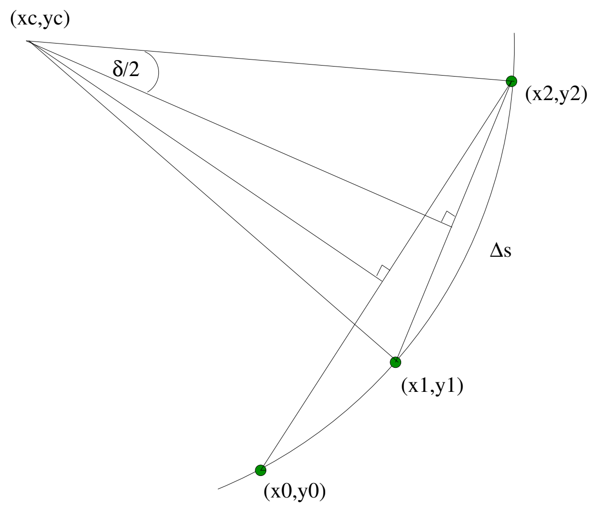


Figure C.1: Calculating fit reference out of two 3D hits and the beam position.

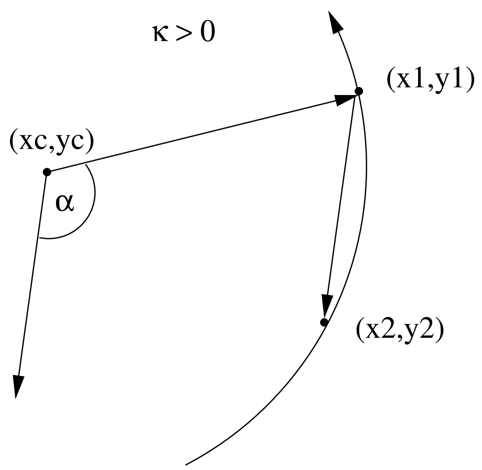


Figure C.2: Determination of the sign of the curvature for positively charged particles.

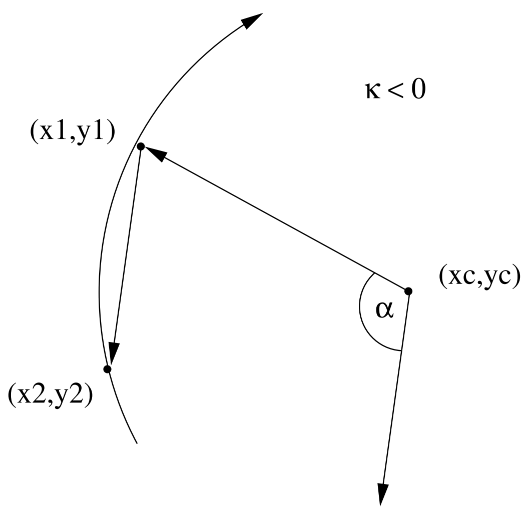


Figure C.3: Determination of the sign of the curvature for negatively charged particles.

The sign of the curvature is the inverse of the sign of the following cross product (fig.: C.2/C.3):

$$\sin(\alpha) = \begin{pmatrix} x_1 - x_c \\ y_1 - y_c \end{pmatrix} \times \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \end{pmatrix} \tag{C.6}$$

$$sgn = -\frac{\sin(\alpha)}{|\sin(\alpha)|} \tag{C.7}$$

$$C = \frac{sgn}{2R} \tag{C.8}$$

The path length in r/ϕ between hit1 and hit2 and then the $\cot(\theta)$ of the helix can be calculated in the following way (fig.: C.1):

$$\sin(\frac{\delta}{2}) = \frac{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}{2R} \tag{C.9}$$

$$\cot(\theta) = \frac{\delta * R}{z_1 - z_2} \tag{C.10}$$

The impact parameter D is defined as:

$$D = sgn(\sqrt{x_c^2 + y_c^2} - R); \tag{C.11}$$

With the impact parameter D, the curvature C and the radius of hit2 (r_2) β of the reference at r_2 can be calculated following the formulae (eqn. A.38):

$$\beta = \arccos(\frac{R^2C + D^2C + D}{R(1 + 2CD)}) - \frac{\pi}{2} + \arccos(1 - 2C^2\frac{R^2 - D^2}{1 + 2CD}) \tag{C.12}$$

$$\tag{C.13}$$

Now we have calculated all parameters at the reference at radius r_2:

$$p[0] = r_2 \tag{C.14}$$

$$p[1] = \phi_2 \tag{C.15}$$

$$p[2] = z_2 \tag{C.16}$$

$$p[3] = \arctan(\cot(\theta)) \tag{C.17}$$

$$p[4] = \beta \tag{C.18}$$

$$p[5] = 2C \tag{C.19}$$

C.2 Transverse Momentum of Hit Doublets

In the Silicon Standalone strategy, a measure of the transverse momentum of the 3D hit doublets is already needed before the full calculation of the helix is performed. In order to save CPU time we use some approximation to sort the doublets by there transverse momenta. The first one assumes that the tracks come from (0,0), which is true for MC but not for data. In addition we assume, that all radii for the outermost hit of the doublet are the same. This is valid because we want to sort the doublets only within the same combinations of layers.

$$\begin{aligned}
\gamma &= \pi - |\Delta\phi| - |\alpha| \\
&\approx \pi - |\Delta\phi| - |\arcsin(\frac{\Delta\Phi R_2}{\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}})| && \text{(C.20)} \\
\Delta\phi &= \phi_1 - \phi_2 && \text{(C.21)}
\end{aligned}$$

Due to the generalized theorem of Thales (fig.: C.5) γ is for a fixed size of R_1 independently of R_2 a measure for the radius of the circle. For larger γ the radius and so the transverse momentum gets smaller.

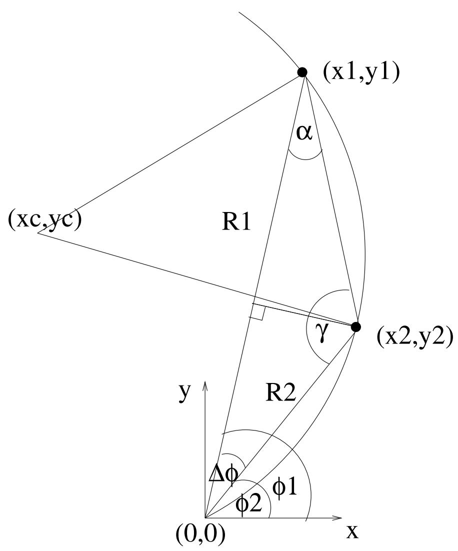


Figure C.4: Fast calculation of transverse momenta for hit doublets.

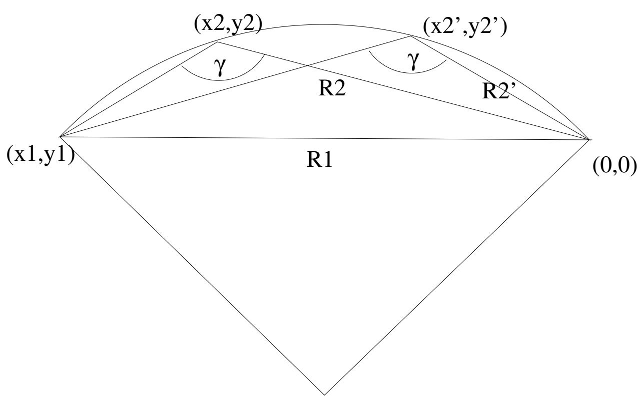


Figure C.5: Generalized theorem of Thales.

Appendix D

Interfaces to Fitter and Strategies

D.1 How to Access the Kalman Fitter?

Besides the extensive use of the fitter inside the TrackingKal package, the fitter can also be easily used from outside without being an expert of the TrackingKal package at all.
The standard use of the Kalman fitter is to use its backward fit in order to get the best estimate of the track at the production vertex. The interfaces for the use of the backward fit are defined in `TrackingSI/TrackingSI/TrackFitting/SiKalmanFitter.hh`.
Other possibilities of the Kalman fitter are to calculate unbiased residuals or to use the forward fit in order to extrapolate the track to some detector components outside the silicon, e.g. the COT or the calorimeters. Those functionalities are defined in `TrackingKal/TrackingKal/KalUtils`.

D.1.1 Backward Fitting with the Kalman Fitter

In order to use the backward fit of the Kalman Fitter in some of your code you have first to include the header file:

```
# include "TrackingSI/TrackingSI/TrackFitting/SiKalmanFitter.hh"
```

The Kalman fitter uses its own geometry description, which has to be initialized. Due to the different alignment from run to run, an instance of the `SiKalmanFitter` has to be defined in the `beginJob` method of the module, in which the fitter is used. Then it automatically picks up the correct geometry run by run. So add the following lines in the `beginJob` method:

```
SiKalmanFitter* kalfitter = SiKalmanFitter::instance();
```

Depending on the way the informations used by the fitter are available, there are different interfaces to use:

If all informations as e.g. the silicon hits or the parameters measured in the COT are stored in a `Cdftrack` the fit can be simply called via:

```
kalfitter → fit( CdfTrack & track, double innerRadius = 0 );
```

- The results of the fit are the parameters, the weight matrix, the χ^2 and the dof of the fit. They are stored in the member variable fit (of type `HelixFit`) of the input track.
- The results of former fits of this track are overwritten.
- The fit works only for tracks with the following algorithm values:
  - `CdfTrack::CotStandAloneAlg`
  - `CdfTrack::OutsideInAlg`
  - `CdfTrack::OutsideIn3DAlg`
  - `CdfTrack::OutsideInStereoAlg`
  - `CdfTrack::KalSvxStandAloneAlg`
  - `CdfTrack::KalOutsideIn3DAlg`
  - `CdfTrack::KalOutsideInStereoAlg`
  - `CdfTrack::KalOutsideInAlg`

  Although there are still a number of other possible algorithm values, those are the ones for tracks coming out of reasonable tracking algorithms.
- The return value of this routine is `TrackFitter::OK` or `TrackFitter::FAILED`. In the first case the fit has been processed properly, and the results are o.k., in the other case the fit has not been performed most probably to some inconsistent input.

- If no value for the inner radius is set the fit is performed till the innermost layer, which is intersected by the track. This is in most of the cases the beampipe. If the particle is supposed to be originated further outside the variable `innerRadius` is the minimum radius, to which the backward fit is performed. The material and also hits at smaller radii are not taken into account for the fit.

If you do not want to overwrite the old fit results or you do not have them packed in a `CdfTrack` e.g. for unpacking the pads you have to use different interfaces.
The input this time is:

- `const HepVector& siliconRef`: some precursor for the fit, this can be either the old fit parameters, the parameters of the COT track or a rough calculation of the reference by using some of the silicon hits
- `std::vector<const SiHit*>& hits` : list of silicon hits, this can also be empty, e.g. in case of propagating a COT track through the material in the silicon
- `HepVector& fitParam`: the parameters of the fit result are written in here
- `HepSymMatrix& fitCov`: the covariance matrix of the fit is written in here
- `double& fitChi2`: the chi2 of the fit is written in here
- `int& fitDof`: the number of degrees of freedom of the fit are written in here
- `double innerRadius = 0`: same as in the above case
- `const HepVector& cotParam`: the parameters of the associated COT track if available
- `const HepSymMatrix& fitCov`: the covariance matrix of the associated COT track if available

In case of available COT information the fitter is called via:

```
kalfitter -> fit( constHepVector& cotParam,
                  HepSymMatrix& cotCov,
                  const HepVector& siliconRef,
                  std::vector < const SiHit* > & hits,
                  HepVector& fitParam,
                  HepSymMatrix& fitCov,
                  double& fitChi2,
                  int& fitDof,
                  double innerRadius = 0 );
```

else via:

```
kalfitter -> fit( const HepVector& siliconRef,
                  std::vector < const SiHit* > & hits,
                  HepVector& fitParam,
                  HepSymMatrix& fitCov,
                  double& fitChi2,
                  int& fitDof,
                  double innerRadius = 0 );
```

The output of the fitter are again `TrackFitter::OK` or `TrackFitter::FAILED`.

There are some additional options to set:

- use of dE/dx correction in the fitter, the default is true:

  `kalfitter-> setDEdx(bool dEdx);`

- set a mass for the fitter, the default is the mass of the pion:

  `kalfitter-> setMass(double m);`

- set a scaling factor for the COT covariance matrix, the default is currently 2.25 and will be once the COT errors are understood 1:

  `kalfitter-> setCOTErrorScale(double scale);`

D.1.2 Additional Features of the Kalman Fitter

In order to use one of the additional features of the Kalman Fitter in your code you have first to include the header file:

```
# include "TrackingKal/TrackingKal/KalUtils.hh"
```

and also to define an instance of `KalUtils` in the `beginJob` function of your modul.

```
KalUtils* kalutils = KalUtils::instance();
```

Calculating the Unbiased Residual

The best estimate of a fit at a given layer is calculated by taking the weighted mean of the forward and the backward fit up to this layer. The hit at this layer is for both fits not taken into account. The unbiased residual is the distance of the best estimate of this fit to the hit in this layer. For a given track with a corresponding `CdfTrack_clnk t` and a given layer l (numbering of canonical layers 0-7) the unbiased residual is calculated via calling:

```
kalutils → unbiasedresid( CdfTrack_clnk t,
                          int l,
                          std::vector < results > & res )
```

Due to overlap regions, there might be more than one result for a canonical layer. So the results are stored in a vector. The results vector is cleared whenever this method is invoked, therefore clients can not accidently accumulate results belonging to different layers. The results are stored in the struct `results` also defined in `KalUtils.hh`. The struct contains the following members:

- `std::vector <SiDigiCode> siDigi` :
  this is a list of digi codes for the wafers the track is going through within an error window of 2 cm in z and 0.15 rad in ϕ in this layer. The appearance of more than one `SiDigiCode` in this list is not due to the overlap regions but due to the error windows. E.g. in z the distance between the wafers of a barrel can be smaller than the resolution of the track.

- `double estimR` :
  radius of the best etimate of the fit.

- `double estimPhi` :
  best estimate of the global ϕ position of the fit
- `double estimZ` :
  best estimate of the global z position of the fit
- `bool phiHitAv` :
  true if track has a ϕ hit at this layer
- `double deltaPhi` :
  unbiased residual in global ϕ
- `bool zHitAv` :
  true if track has a z hit at this layer
- `double deltaZ` :
  unbiased residual in global z

The return value of the function is true in the case of success. The function is working for the same track types as the `SiKalmanFitter`, although it does not make sense to use tracks of the type `CdfTrack::OutsideInAlg` as input, because they do not have any silicon information.

Calculating the Best Estimate at a Given Radius

Best estimate of track perigee parameters at a given radius R is somewhat a contradiction in terms. What really happens is the following. A forward fit is performed from the perigee to a given radius R, taken all material and multiple scattering into account. A backward fit is performed form the radius of the outermost measurement of the track to the radius R. If R coincidence with the radius of a hit only one of both fits incorporate this hit. The weighted mean of the forward and backward fit is taken. Finally we compute the perigee parameters from the best estimate at R. This last step is simply a transformation to a different parameter space. The baseline is that although the perigee parameters suggest, they give the best estimate at the closest approach to the origin in the x/y plane, this is not true for the result of this method. The Kalman Fitter uses a material description, which contains the silicon and the inner wall of the COT. Every material outside the inner wall is not described. This implies the energy loss and multiple scattering will be wrong when you extrapolate to a radius outside the COT inner wall. In this case the "outside"flag is true.
The input track has to have the same algorithm value as for the calculation of the unbiased residual.
The parameters of the best estimate at radius R are stored in `HepVector& param`,

the corresponding covariance matrix in `HepSymMatrix& cov`. `CdfTrack_clnk t` is the link to the track, we want to know about.
The function is called via:

```
kalutils → bestEstimateAtR( CdfTrack_clnk t,
                             double R,
                             HepSymMatrix& cov,
                             HepVector& param,
                             bool& outside )
```

The return value of this function is true in case of success.

Forward Fit to the COT Inner Wall

This is just a special case of the routine described above. The idea is to extrapolate tracks found in the standalone tracking to the COT inner wall in order to pick up additional COT hits. This function is somehow preparing the input to an Inside-Out tracking which is planed but not yet in place. The link to the track (`CdfTrack_clnk t`), which we want to extrapolate is fitted from the perigee versus the COT inner wall. The last material taken into account for multiple scattering and dE/dx is the COT inner wall. The track parameters are again given in perigee parametrization, although they describe the track at the radius of about 41 cm.
Due to this specialized functionality of this function only tracks with the algorithm value `CdfTrack::KalSvxStandAloneAlg` are accepted as input. If the track does not intersect with the COT inner wall the "interactCOT"flag is set to false.
The results of the function are:

- `HepSymMatrix& cov`:
  the covariance matrix of the forward fit

- `HepVector& param`:
  the result parameters

- `double& chi2`:
  the χ^2 of the fit

- `int& dof`:
  the number of degrees of freedom of the fit

The function can be called via:

```
kalutils  → forwardFit2COT(  CdfTrack_clnk t,
                             HepSymMatrix& cov,
                             HepVector& param,
                             double& chi2,
                             int& dof,
                             bool& interactCOT )
```

The return value of this function is true in case of success.

D.2 How to Access the Tracking Strategies?

All silicon tracking strategies are interfaced by the `SiPatternRecModule` located in the `TrackingMods` package. Once the module is plugged into the executable, different strategies can be chosen via tcl switches.
There are quite a lot of obsolete tracking algorithms, which can be called by the `SiPatternRecModule`, too. But the only currently reasonable and maintained algorithms are:

- the RUN I like OI tracking which is split up in r/ϕ tracking and z tracking,
- the Kalman fitter based OI algorithm of the TrackingKal package,
- the Kalman fitter based Silicon Standalone algorithm of the TrackingKal package.

For those three strategies, their tuning parameters can be set by tcl switches. In the following all the switches are explained, and their default settings are listed.

D.2.1 Choice of the Tracking Strategies

The following algorithms are obsolete ones. It is heavily recommended not to switch them on.

```
PerformOutsideInTrackingCR    set  false
AddZToOI                      set  false
PerformRPhiHL                 set  false
PerformRPhiHL                 set  false
Perform3DOutsideInTracking    set  false
PerformStandaloneTracking     set  false
PerformSvxStandaloneTracking  set  false
PerformSiSoloRPhi             set  false
PerformPerfectSiTracking      set  false
```

In order to use the RUN I like r/ϕ algorithm you have to set the following tcl switch true.

```
PerformOutsideInTracking  set  true
```

To add z and SAS hits to the tracks found in the RUN I like r/ϕ algorithm the following switch has to be set to true.

```
PerformOutsideInZTracking  set  true
```

The TrackingKal OI strategy and Silicon Standalone strategy in combination can be used via:

```
PerformKalOISvxStandaloneTracking  set  true
```

If only the TrackingKal OI should be processed, then the standalone has to be switched off from this combination explicitly.
This switch has only an impact if `PerformKalOISvxStandaloneTracking` is switched on.

```
KalStandaloneOff  set  false
```

If only the TrackingKal Silicon Standalone tracking without the TrackingKal OI tracking is required, the following switch has to be set to true.

```
PerformKalSvxStandaloneTracking  set  false
```

There are two version of the TrackingKal OI tracking: the 2 loop version, which picks up first ϕ and SAS hits and then $90°z$ hits, and the 3 loop version which first picks up ϕ, then SAS and finally $90°z$ hits. The 3 loop version is recommended to use. If TrackingKal OI tracking is switched on, it can be chosen between both versions via:

```
Do3LoopOIInKal  set  true
```

The tracking group has decided to use both OI algorithms and to run them in combination. The order in which they are processed is determined by the following switches. The second algorithm only uses the unused COT tracks of the first algorithm.

```
KALOIonUnusedCOTTracksOnly  set  true
WEIOIonUnusedCOTTracksOnly  set  false
```

D.2.2 Tuning Parameters

There are some tcl switches, which affect all the tracking strategies called by the `SiPatternRecModule`.
If running on data the zigzag bonding of LAYER 6 has to be corrected for. The zigzag bonding is for all releases younger than 4.8.4 per default included in the simulation. So the switch has to correspond to the setting of the simulation.

```
zigzagbonding  set  true
```

The minimum transverse momentum for tracks searched in the strategies can be set via the following switch:

```
MinimumPt  set  0.2
```

Do not set this switch to 0.0 when running Silicon Standalone tracking. Otherwise it will take quite some CPU time due to the high combinatorics.
In order to calculate unbiased residuals or to simulate detector inefficiencies it might be helpful to exclude certain layers from the reconstruction. This can be done by the following switch:

```
missingMultiLayers  set  00000000
```

An earlier switch was designed to skip one layer from reconstruction, but it is not working anymore since the `missingMultLayers` switch has been introduced, just ignore it.

```
missingLayer  set  −1
```

Tuning Parameters for the RUN I like Strategy

The following switches have only an impact on the RUN I like OI tracking strategy and not on the TrackingKal strategies.

The COT error matrix is not yet fully understood. A scaling factor is introduced to normalize the pulls of the COT tracks. The default value is `1.5`.

```
cotSF  set  1.5
```

The RUN I like OI strategy currently uses a quite simple material description. The entire Geant 3 description also used for simulation can be plugged in by the following switch. It slows awfully down the tracking algorithm. It is not recommended to use it.

```
useG3XIntegrator  set  false
```

If the RUN I like strategy should only use SAS hits and no $90°z$ in its z path, the following tcl switch has to be set to false:

```
use90Z  set  true
```

Tuning Parameters for the TrackingKal strategies

The following switches have only an impact on the TrackingKal strategy and not on the RUN I like strategy.
Also for the TrackingKal OI strategy, the COT covariance matrix scaling factor can be set. In contrast to the RUN I like strategy it do not describe the scaling of the error matrix but of the covariance matrix. So the default is `2.25`.

```
KalCOTErrorScale  set  2.25
```

Picking up SAS and z hits can be switched off for the TrackingKal strategies. There are two switches, which concern both OI and the standalone tracking.

```
UseStereoInKal  set  true
   UseZ90InKal  set  true
```

If `UseStereoInKal` is switched off, there are automatically no $90°z$ hits attached to the tracks, too, independently of the `UseZ90InKal` switch.

Tuning Parameters for the TrackingKal 3 Loop OI Strategy

The meaning of the following switches are explained in detail in chapter 4. They only have an impact on the 3 loop version of the TrackingKal OI strategy. There are no setters for the tuning parameters of the 2 loop version.

```
KalOI3LoopChi2Cut1       set  40.0
KalOI3LoopChi2Cut2       set  30.0
KalOI3LoopChi2Cut3       set  20.0
KalOI3LoopChi2Cut4       set  10.0
KalOI3LoopChi2Cutz       set  40.0
KalOI3LoopCleanUpPhi     set  true
KalOI3LoopCleanUpZ       set  true
KalOI3LoopPhiSigmaRoad1  set  8.0
KalOI3LoopPhiSigmaRoad2  set  6.0
KalOI3LoopPhiSigmaRoad3  set  3.0
KalOI3LoopZSigmaRoad     set  4.0
KalOI3LoopSASSigmaRoad   set  4.0
KalOI3LoopMinPhiHits     set  3
```

Tuning Parameters for the TrackingKal Silicon Standalone Strategy

The meaning of the following switches are explained in detail in chapter 5.

```
KalSVXMaxChi2pdofCut1    set  20.0
KalSVXPhiWindow          set  0.05
KalSVXZWindow            set  0.5
KalSVXPVz1               set  5
KalSVXPvz2               set  0.8
KalSVXCleanUpPhi         set  true
KalSVXCleanUpZ           set  false
KalSVXSigmaPhiWindow     set  6.0
KalSVXSigmaZWindow       set  6.0
KalSVXSigmaStereoWindow  set  6.0
KalSVXMinPhiHits         set  4
BeamX0                   set  -0.16695
BeamY0                   set  0.43157
```

The default for the beam position is the mean beam position measured on data. For MC studies, `BeamX0` and `BeamY0` have both to be set to 0.0.

Bibliography

[1] F. Abe et all. (CDF Collaboration),
Evidence for Top Production in $\mathrm{p\bar{p}}$ *Collisons at* $\sqrt{s}$*=1.8 TeV,*
Phys. Rev. Lett. 73, 225 (1994)

[2] T. Affolder et al. (CDF Collaboration),
Measurement of the Top Quark Mass with the Collider Detector at Fermilab,
FERMILAB-PUB-00/127-E, submitted to Phys. Rev. D June 29, 2000

[3] T. Affolder et al. (CDF Collaboration),
Measurement of the W Boson Mass with the Collider Detector at Fermilab,
FERMILAB-PUB-00/127-E, submitted to Phys. Rev. D July 26, 2000

[4] D. Acosta et all. (CDF Collaboration),
Measurement of B-meson lifetimes using fully reconstructed B decays produced in $p\bar{p}$ *collisions at* $\sqrt{s}$ *= 1.8 TeV,*
Phys. Rev. D 65, 092009 (2002)

[5] CDF Collaboration,
CDF II Technical Design Report,
FERMILAB-PUB-96/390-E, Nov 1996, 318pp,
`http://www-cdf.fnal.gov/upgrades/tdr/tdr.html`

[6] E. James [CDF Collaboration],
Electroweak Physics Prospects for CDF in Run II,
CDF/PUB/CDF6221

[7] The LEP Working Group for Higgs Boson Search,
Search for the Standard Model Higss Boson at LEP, **LHWG Note/2002-01**

[8] A. Stocchi, *Experimental results on heavy quarks*,
arXiv:hep-ph/0211245.

[9] CDF B Physics Group,
Mixing and Lifetimes in Run II, Report from Working Group 3,
CDF/DOC/BOTTOM/CDFR/5740

[10] H. Niu, et al.,
Limit on $B_s^0\bar{B}_s^0$ Oscillation Frequency in $p\bar{p}$ Collisions at $\sqrt{s}$=1.8 TeV,
CDF/ANAL/BOTTOM/CDFR/5815

[11] M. Tanaka (CDF Collaboration), *CDF: Run II Physics Projections*,
Pub. Proceedings Beauty 2000

[12] B. Aubert *et al.* [BABAR Collaboration],
Measurement of the CP-violating asymmetry amplitude $\sin(2\beta)$,
Phys. Rev. Lett. 89 (2002) 201802

[13] K. Abe *et al.* [BELLE Collaboration],
Improved Measurement of Mixing-induced CP Violation in the Neutral B Meson System,
hep-ex/0208029

[14] R. Fleischer, *New strategies to extract beta and gamma from* $B_d \rightarrow \pi^+ \, pi^-$ *and* $B_s \rightarrow K^+K^-$, **Phys. Lett. B 459 (1999) 306**

[15] P. Gatti, *Performance of the new tracking system at CDF II*,
CDF/THESIS/TRACKING/PUBLIC/5561

[16] K. Rinnert,
A Fast-Access Material Map for the SVXII/ISL Silicon Tracking Volume at CDF 2, **CDF/DOC/TRACKING/CDFR/5846**

[17] P. Schemitz,
Eine Hard- und Software Umgebung für Physik-Analysen bei CDF II,
internal note IEKP-KA/2003-1, University of Karlsruhe

[18] J. O. Coplien,
Multi-Paradigm Design for C++,
ISBN 0-201-82467-1

[19] *The CDF Run II Event Data Model Page*,
http://www-cdf.fnal.gov/upgrades/computing/projects/edm/edm.html

[20] K. Bloom,
Getting Started with CDF Run 2 Offline,
CDF/DOC/COMP_UPG/CDFR/5294

[21] M. Feindt, C. Lecci, S. Menzemer, K. Rinnert, P. Schemitz, A. Skiba,
TrackingKal - A Tracking and Alignment Software Package for the CDFII Silicon Detector,
CDF/DOC/TRACKING/PUBLIC/5968

[22] H. Wenzel,
Tracking in the SVX,
CDF/DOC/SEC_VXT/PUBLIC/1790

[23] N. Bachetta, et al.,
Results of the KEK Beam Test of SVX II Protoype Detectors,
CDF/DOC/TACKING/PUBLIC/3547

[24] P. S. Maybeck,
Stochastic Models, Estimation, and Control,
Vol. 1

[25] V. Blobel, E. Lohrmann,
Statistische und numerische Methoden der Datenanalyse,
Teubner, 1998

[26] M. Milnik,
Diploma thesis in progress

[27] *The Europen Physical Journal C, Review of Particle Physics*

[28] K. Anikeev, G. Bauer, I. Furic, S. Gromoll, A. Korn, I. Kravchenko, Ch. Paus, A. Rakitin, J. Tseng,
Calibration of Energy Loss and Magnetic Field using J/Ψ Events in Run II,
CDF/DOC/BOTTOM/CDFR/5958

[29] P. Azzi, G. Busetto, P. Gatti, A. Ribon,
Histogram Tracking in the COT,
CDF/DOC/TACKING/CDFR/5562

[30] A. Mukherjee,
CTC and VTX Tracking,
CDF/ANAL/TRACKING/PUBLIC/5490

[31] W.-M. Yao, K. Bloom,
"Outside-In"Silicon Tracking at CDF,
CDF/TRACKING/DOC/CDFR/CDF5591

[32] C. Lecci, S. Menzemer, K. Rinnert, M. Feindt,
Pre-Tracking PVz Finder,
CDF/DOC/TRACKING/PUBLIC/5988

[33] P. Azzi, A. Sidoti,
Performance Evaluation of Outside-In Silicon Tracking Algorithms on Simulated Events,
CDF/ANAL/TRACKING/GROUP/5897

[34] P. Azzi, A. Sidoti,
Comparator Studies - The bottom line,
`http://www-cdf.fnal.gov/internal/upgrades/computing/projects/reconstruction/tracking`

[35] K. Anikeev, F. Azfar, G. Bauer, I. K. Furic, S. Gromoll, A. Korn, I. Kravchenko, M. Mulhearn, Ch. Paus, A. Rakitin, J. Tseng,
The Inclusive B Livetime using Run II Data at CDF,
CDF/DOC/BOTTOM/CDFR/6023

[36] F. Abe et al., Phys. Rev. **D57** (1998) 5383

[37] K. Anikeev, G. Bauer, I. K. Furic, S. Gromoll, A. Korn, I. Kravchenko, M. Mulhearn, Ch. Paus, A. Rakitin, J. Tseng,
Exclusive B Lifetime Measurement with Run II Data at CDF,
CDF/DOC/BOTTOM/CDFR/6022

[38] K. Anikeev, G. Bauer, I. K. Furic, S. Gromoll, A. Korn, I. Kravchenko, M. Mulhearn, Ch. Paus, A. Rakitin, J. Tseng,
Measurement of B Mesons Masses in the Exclusive J/Ψ Channels,
CDF/DOC/BOTTOM/CDFR/6022

[39] S. De Cecco, S. Giagu, M. Rescigno, *A Measurement of the Relative Branching Fractions $\Gamma(D^0 \to K^+K^-)/\Gamma(D^0 \to K\pi)$ and $\Gamma(D^0 \to \pi^+\pi^-)/\Gamma(D^0 \to K\pi)$*,
CDF/DOC/BOTTOM/CDFR/6018

[40] K. Anikeev, G. Bauer, I. K. Furic, S. Gromoll, A. Korn, I. Kravchenko, M. Mulhearn, Ch. Paus, A. Rakitin, J. Tseng,
Measurement of the Mass Difference $m(D_s^\pm) - m(D^\pm)$ in the $\phi\pi$ Decay Channel,
CDF/DOC/BOTTOM/CDFR/6021

[41] W. Press, B. Flannery, S. Teukolsky, W. Vetterling,
Numerical Recipes: the art of scientific computing,
Cambridge Univ. Press, 1989

Acknowledgements

A lot of people have supported me in various ways over the last few years and I like to say "Thank you" to all of them.

Especially, I like to thank my supervisor Michael Feindt, for all his advises and support. I enjoyed all the discussions about statistical problems where I profited a lot from his huge knowledge. Thanks also to Günter Quast who co-supervised this thesis.

I like to thank the Institute of Experimental Nuclear Physics (IEKP) in Karlsruhe and the CDF experiment for giving me the opportunity to undertake this research and the "Land Baden-Württemberg" and the "Graduiertenkolleg für Teilchenphysik und Astrophysik" for providing funding in the form of a graduate scholarship.

I also like to thank the CDF tracking group, especially their (ex)conveners Matt Herndon and Avi Yagil for their support of the work "on long distance scale".

The TrackingKal package has mainly been developed in cooperation with Kurt Rinnert. I like to thank him for being a reliable source of encouragement, humor and C++ knowledge. I really enjoyed working together with him.

I also like to thank Marcel Stanitzki and Markus Moch for their patience in reading drafts of this thesis, and even more for always being available for a cup of tea.