TECHNISCHE UNIVERSITÄT WIEN

VIENNA UNIVERSITY OF TECHNOLOGY

Diplomarbeit

Implementation of a General-Purpose Kinematic Estimator in the Software Environment of the International Linear Collider and Application to Pair Production of W Bosons

Ausgeführt am

Atominstitut
der Österreichischen Universitäten

unter der Anleitung von

Prof. DI Dr. Meinhard REGLER

mit Unterstützung von

DI Dr. Wolfgang WALTENBERGER
Institut für Hochenergiephysik der
Österreichischen Akademie der Wissenschaften

durch

Fabian MOSER
Matr. Nr. 0225488
Josefstädterstrasse 33/22
1080 Wien

Wien, im August 2008

Kurzfassung

Thema dieser Diplomarbeit war die Erweiterung einer Programmbibliothek zur Rekonstruktion von Teilchenreaktionen wie sie bei Hochenergiephysikexperimenten untersucht werden. Ziel war es, eine Möglichkeit zu schaffen, kinematische Zwangsbedingungen in diese Rekonstruktion mit einzubeziehen. Außerdem sollte die Bibliothek in eine vorhandenen Softwareumgebung integriert werden, welche zum Einsatz in einem zukünftigen Linearbeschleuniger-Experiment entwickelt wird. Den Abschluß bildete ein Test anhand der Rekonstruktion von W-Boson Zerfällen und die realistische Fragestellung nach der notwendigen integrierten Luminosität zur Verbesserung der bekannten Messungen der W-Masse.

Die Programmbibliothek Rave entstand aus dem Wunsch heraus, eine immer neue Entwicklung derselben Programmteile in typischen Softwarepaketen zur Rekonstruktion von Teilchenreaktionen zu vermeiden. Von dieser Problematik sind jene Teile betroffen, die experimentunabhängig formuliert werden können. Dabei handelt es sich insbesondere um die Rekonstruktion von Interaktionspunkten (Vertices), z.B. von Teilchenzerfällen.

Um die zentrale Idee von Rave von Anfang an hervorzuheben, dient als Kern der Bibliothek jene Implementierung der Algorithmen, welche für das Compact Muon Solenoid (CMS) Experiment erstellt wurde. Rave liefert dazu eine einfache und stabile Schnittstelle. Derselbe Ansatz wurde während dieser Diplomarbeit weiterverfolgt, als jene Teile die es erlauben, die Vertices unter Berücksichtigung von kinematischen Zwangsbedingungen zu rekonstruieren, aus der CMS-Software herausgelöst und in Rave implementiert wurden.

Als nächstes wurde die Integration von Rave in die Rekonstruktions- und Analyse-Software des International Large Detector (ILD) Experiments, welches für den International Linear Collider entwickelt wird, vorgenommen. Dazu war eine entsprechende Erweiterung erforderlich, die sämtliche Funktionen von Rave für die Benutzer dieser Software zugänglich macht.

Zum Abschluß der Arbeit wurde die neue Software anhand einer beispielhaften Rekonstruktion getestet. Die gewählte Reaktion war die Paarproduktion von W-Bosonen durch e^+e^- Kollisionen bei 500 GeV, welche in insgesamt vier so genannte Jets zerfallen. Ziel dieser Analyse war es, die Genauigkeit zu ermitteln, mit der unter Anwendung von kinematischen Zwangsbedingungen die Masse der W-Bosonen rekonstruiert werden kann. Die Jets wurden mit einem weit verbrei-

teten Reaktionsgenerator simuliert und anschließend mit den erwarteten Rekonstruktionsfehlern versehen. Das Ergebnis zeigt die hohe Genauigkeit, mit der die W-Masse im ILD-Experiment bestimmt werden können wird.

Abstract

The topic of this diploma thesis is the extension of a program library for event reconstruction at high energy physics experiments. The extension should enable the use of kinematic constraints during the reconstruction. Moreover the library should be integrated into an existing software environment used for event reconstruction at a future linear collider experiment. A test with the reconstruction of W-boson decays closes this work together with the realistic question for the integrated luminosity necessary to improve on currently known measurements of the W mass.

The Rave library was created to avoid the repeated re-implementation of similar algorithms in every new reconstruction software package, because certain parts can be formulated in an experiment-independent manner. These are in particular the parts doing the reconstruction of interaction vertices, e.g. of particle decays.

To emphasize the central idea of Rave right from the start, the core of the library is the implementation of the algorithms which was designed for the Compact Muon Solenoid (CMS) experiment at the Large Hadron Collider. Rave provides an additional simple and stable interface. The same approach was pursued during this diploma thesis when the parts doing vertex reconstruction with consideration of kinematic constraints were extracted from the CMS software and implemented in Rave.

The next step was the integration into the reconstruction and analysis framework of the International Large Detector (ILD) experiment, which is developed for the International Linear Collider. This required an appropriate extension for making all related functions accessible to the user of the framework.

Finally, the new software was tested with an exemplary reconstruction. The chosen reaction was the pair-production of W bosons by e^+e^- collisions at 500 GeV decaying purely into hadrons and thus forming four jets in the final state. The goal of this analysis was to determine the precision at which the mass of the W boson can be reconstructed when considering kinematic constraints. The jets were simulated by a popular physics generator, thereafter the errors expected for jet reconstruction at that specific experiment were applied. The result shows the high precision which the ILD experiment will provide to determine the W mass.

Acknowledgments

First of all my thanks go to my supervisor Meinhard Regler. As many others before myself he introduced me to the field of high energy physics in the first place. His well-known introductory and hands-on training on particle physics was the starting point of my interest in the field. He generously offered to supervise this diploma thesis which permitted me to continue and deepen my work on the vertex reconstruction toolkit Rave. His comprehensive expertise and commitment were a reliable guidance over the whole duration of this diploma thesis and especially during the studies concerning the last chapter.

Further thanks go to Wolfgang Waltenberger, the author and administrator of the Rave toolkit and (lucky me) the person next to me whom I asked very many questions and who heard and helped me each and every time. His work, ideas and visions are the actual core of Rave. His readiness to dive into any problem and question I issued and to grab a pencil or chalk (or whatever at hand) to throw light on it was always very helpful and also inspiring.

Further thanks go to Winfried Mitaroff who laid a cornerstone with his introductory lecture on data analysis in high energy experiments. He encouraged me to do this diploma thesis and he really enabled me to do it by inviting me to join his ILC division. Also my first publication and my first conference talk are due to him.

Further thanks go to Dietrich Liko for the insights on event reconstruction and simulation and Wolfgang Dungel for his help with ROOT and LaTeX.

Contents

1 Introduction

Progress on high energy physics experiments is often expressed by two paradigms: energy and resolution. The steady increase of the energies at which experiments are performed is of course more obvious and is thus eponymous for the domain. The exploitation of technical and financial limits along this path is profoundly stimulated by theoretical predictions awaiting verification. Letting aside the financial questions, the technical limits are subject to continuous change, but there are a few principles which tend to change only when certain physical limits are met. Today we are at the doorstep of the change of two of those principles.

One thing that will certainly not change in the future is the necessity to favor experiments of colliding beams over those shooting particles at fixed targets. This is because in fixed target experiments the center-of-mass energy (available for the generation of new particles) increases only with the square root of the beam energy in contrast to a linear increase with two times the beam energy at colliders. Nevertheless fixed target experiments are needed to carry out secondary beam experiments, which is not possible at colliders.

For the last decades the accelerators producing and storing particles of the highest energy have been synchrotrons. Those circular accelerators have two major advantages compared to linear accelerators: First, one particle uses the same accelerating devices several times and second, the same group (bunch) of stored particles can be brought to collision many times and far less beam particles are lost missing each other.

On the other hand, circular colliders have one critical disadvantage: Whenever charged particles are bent they generate radiation. Thus the dipole magnets keeping the particles on their circular path cause an energy loss per orbit proportional to γ^4 (synchrotron radiation). Therefore the Large Hadron Collider (LHC) resorted to using the heavier protons instead of electrons which lowers $\gamma = E/m$ by a factor of 1836 for the same particle energy. This benefit does

not come without a downside, however. Because protons are not fundamental particles but are composed of quarks and gluons and the energy is not equally distributed between those constituents, the exact kinematic configuration of each collision and its center-of-mass energy is in principle unknown. This not only reduces the actual center-of-mass energy of the collision by an order of magnitude compared to the beam energy but is also source of an immense amount of background produced by fragmentation reactions within a wide energy range.

The lack of control over the center-of-mass energy of the reactions arises the need of a large number of effectively random reactions to trigger on or later pick out the signal. The measurement of the properties of rarely produced particles quickly resembles the proverbial look for a needle in a haystack. Those difficulties and the recent improvements on the performance, more precisely the electric field gradient, of accelerating radio-frequency (RF) cavities have brought back the concept of a linear accelerator/collider. The inherent bunch-crossing inefficiencies of "one-shot" linear colliders, however, make extreme demands on the bunch density and the beam profiles in order to gain luminosity.

Of course such a machine would mean a tremendous improvement along the second paradigm of progress mentioned in the first paragraph: the resolution. The precisely known collision configuration and low background result in higher detection and reconstruction efficiencies; a high luminosity (and thus event rate) results in smaller statistical errors. Both will contribute to the resolution of the predictions we can make. Desirably at all times energy, luminosity and resolution are pushed to the highest possible level. But while the peak energy and luminosity of an experiment are rather hard facts once the accelerator is built, resolution will increase in time: not only because of the growing number of measurements taken, but also by optimal extraction of information improved through proper understanding. It is this last source of improvement, which will be the topic of this diploma thesis.

The term "event reconstruction" refers to the process of converting electronic signals to observable quantities, and the latter to abstract quantities closer to our physical models. This involves several steps: to convert the shapes and patterns of certain electronic signals to points in space; to associate sets of these points to particle trajectories ("pattern recognition"); to parametrize these trajectories according to a "track model" of the flight path in the detector's magnetic field.

Furthermore to find one or more common points (vertices) where the found paths supposedly originated from. Until this step, the process of event reconstruction is purely based on geometrical and statistical knowledge and methods, although an appropriate choice of the track and vertex parameters already requires some knowledge of the physical processes involved.

This is where the "kinematic" part comes into play. It refers to an improvement of the fitting process which allows for the direct application of physical understanding of the reconstructed event. While both geometrical and kinematic reconstruction are fitting a set of parameters which are believed to govern the investigated event, the kinematic fitting limits the "search space" of these parameters to configurations constrained by known physical laws and conserved quantities. The motivation for the use of sophisticated reconstruction methods like kinematic fitting is not only to improve the yielded resolution, but it also helps with separating signal an background of the data samples under investigation.

During this diploma thesis the vertex reconstruction toolkit Rave (co-developed by the author) was extended for kinematic fitting tools by adopting the respective parts from the framework of software tools used for event reconstruction in the Compact Muon Solenoid (CMS) experiment at the LHC at CERN, and designing a framework-independent programming interface in consistency with the existing Rave programming interface. This extension is shown to improve resolution in one exemplary reconstruction of pair production of W bosons decaying into four jets in the International Large Detector (ILD) experiment at the International Linear Collider (ILC).

2 Fundamentals

2.1 Physics motivation

"What do we consist of? And what does the world around us consist of?" Answering these fundamental questions is one of the longest-standing motivations of science and scientists. There is one concept which most of the answers given to these questions have in common. It is revealed, when we rewrite the questions to: "What is everything composed of?"

Almost as old as the idea that everything can be separated into its components is the idea of a limit of this separation; something indivisible; some fundamental building blocks of everything.

It is the search for those fundamental building blocks that led us to the discovery of the atoms. But they soon turned out to consist of smaller objects: the electron, the proton and the neutron. And the analysis of protons and neutrons showed that they again are composed of smaller objects called quarks and the force carriers called gluons.

The so-called Standard Model of particle physics is a theory which includes all fundamental particles found so far (electron, its neutrino and two quarks of the first family, and their duplicates of the second and third family). It also includes three of the four known fundamental forces (the unified electromagnetic and weak interactions with the carrier particles photon, $W^{\pm}$ and Z^0 bosons, and the strong interaction with eight carrier particles, the gluons), but disregards gravity. The Standard Model predicts only one particle that has not yet been found: the Higgs boson giving all particles their mass.

Naturally one of the central interests in particle physics research concerns the search for the Higgs boson being the most obvious missing piece of the Standard Model puzzle. Another is the search for evidence of "new physics" beyond the Standard Model, as proposed by more general theories (Grand Unified Theory,

Supersymmetry, or further dimensions of space-time). And the search for the origin of "dark matter" and "dark energy", those two mysterious ingredients of what we currently believe the universe predominantly consists of.

2.2 Future accelerators

All particle physics experiments, aiming at the tasks defined above, have one thing in common: the requirement of high center-of-mass energies. On the one hand high energies are required to produce the fundamental particles we want to analyze, on the other hand they are required to explore structures as tiny as fundamental particles. One way to concentrate high amounts of energies in a controllable manner is to accelerate particles to velocities near the speed of light and collide them at a distinct point. This is exactly what is done by the two accelerators described later in this section.

Particles can only be accelerated to higher energies by an electric field and therefore have to be charged. Because the Coulomb force in the electric field is proportional to the charge times the field strength, the energy a particle (with given charge) gains when travelling in such a field is proportional to the field strength times the distance travelled. Consequently when building a linear accelerator one will want to use strong fields over large distances. Alternatively, a circular accelerator forces the particles to follow a circle and reuse the accelerator multiple times: this is achieved by dipole magnets applying a perpendicular Lorentz force proportional to the speed of the particle times the strength of the magnetic field. If the path of the particle should stay the same over many revolutions, obviously the magnetic field has to be variable in order to account for the increased speed of the particle. This kind of accelerator is called synchrotron.

The performance of a particle accelerator/collider is not only defined by the maximum collision energy it is able to achieve, but also by the number of collisions produced over time. This parameter, normalized by the cross-section of the two beam particles' interaction, is called the luminosity of the accelerator. A high luminosity increases the absolute number of events recorded over a certain period and therefore on the one hand increases the probability to record even very rare events, and on the other hand improves the confidence level of more frequent events. Table 2.1 shows some historic, current and future colliders, together with

	Laboratory	Particles	Max. beam energy [GeV]	Peak luminosity [$\mathrm{cm}^{-2}\mathrm{s}^{-1}$]	Remarks
Tevatron II	FermiLab	$p\bar{p}$	980	1.7×10^{32}	running
LHC	CERN	pp	7000	10^{34}	start 2008
BELLE	KEK	$e^- e^+$	$8 + 3.5$	1.6×10^{34}	running
LEP II	CERN	$e^- e^+$	104	10^{32}	shutdown 2000
HERA	DESY	$e^- p$	$27.5e + 920p$	7.5×10^{31}	shutdown 2007
ILC	t.b.d.	$e^- e^+$	$250 \ldots 500$	2×10^{34}	start $\geq$ 2018

Table 2.1: Some colliders [23]

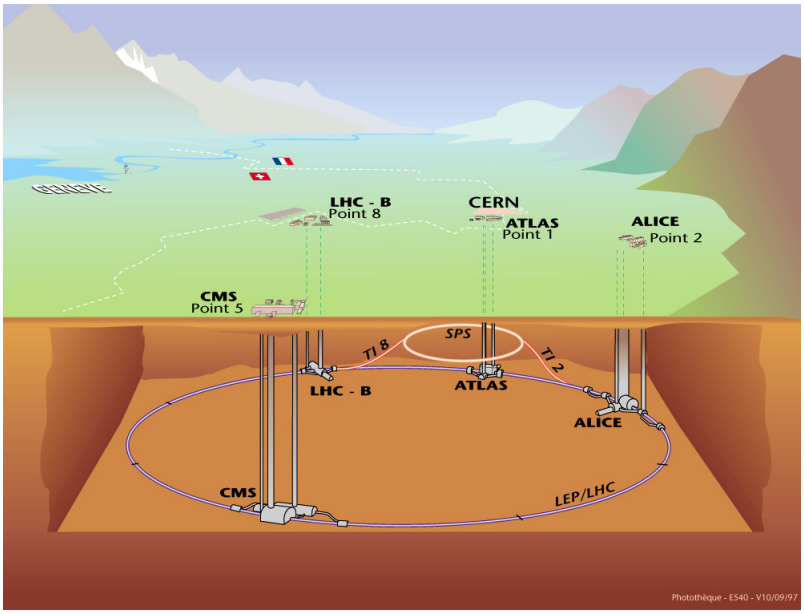


Figure 2.1: Overall view of LHC experiments

their respective maximum beam energy and luminosity [23].

2.2.1 The Large Hadron Collider

The Large Hadron Collider (LHC) is a synchrotron accelerator/collider with a circumference of 27 km, built by the European Organization for Nuclear Research (CERN) in Geneva and extending across the border of France and Switzerland (figure 2.1). It is designed to accelerate protons to an energy of 7 TeV, resulting in a center-of-mass energy at the collision point of 14 TeV. Alternatively heavy ion collisions using Pb^{82+} ions yield center-of-mass energies of 1148 TeV.

Since the particles colliding have the same charge, each beam needs its own

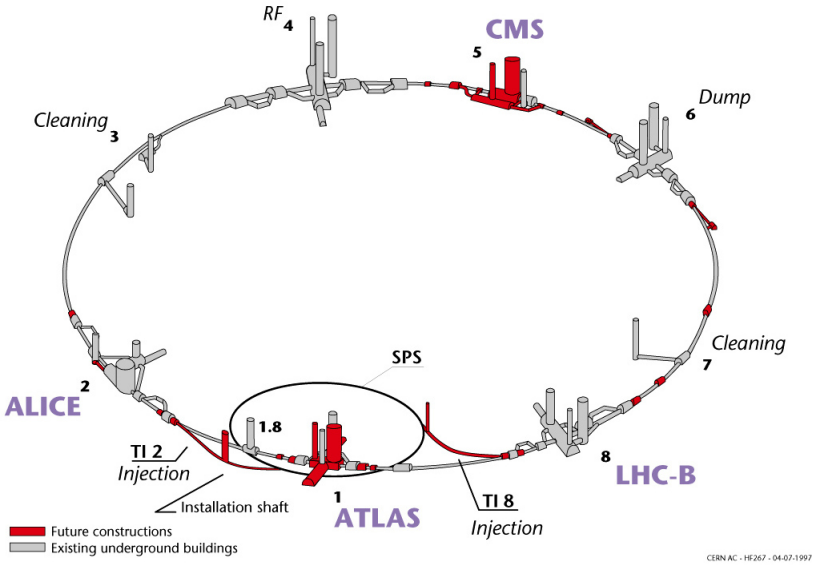


Figure 2.2: Layout of the LHC tunnel including infrastructures

accelerator ring with dipole magnetic fields of opposite polarity. At LHC those fields are required to go up to 8.33 T, and can only be generated by superconducting magnets.

The tunnel together with other local construction has been inherited from the Large Electron-Positron (LEP) collider previously hosted therein. In addition new experimental halls, cooling towers and beam transfer tunnels have been constructed. An overview of the underground layout highlighting new structures is given by figure 2.2.

The protons accelerated in the ring are grouped into 2808 bunches and thus the time between two bunch interactions is at minimum 25 ns. The average event rate R depends on two factors

$$R = L \times \sigma \tag{2.1}$$

where σ is the cross section of the reaction in question, and L is the luminosity of the collider used. For proton-proton collisions the LHC is designed to achieve a maximum luminosity of $2 \times 10^{33}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ during the running, and $10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ when achieving the peak design value. Table 2.2 summarizes the technical characteristics of the LHC at peak design luminosity.

As can be seen from figures 2.1 and 2.2, there are four experiments prepared for operation at the LHC. The two general-purpose detectors ATLAS and CMS

Proton energy at collision	7 TeV
Number of bunches	2808
Number of particles per bunch	1.15×10^{11}
Bunch spacing	25 ns
Average number of events per bunch crossing	19
Average total pp event rate	$0.76 \times 10^{9}\mathrm{s}^{-1}$

Table 2.2: Technical characteristics of the Large Hadron Collider working at peak design luminosity

are both designed for the proton-proton collision mode up to the peak luminosity, as well as for the heavy-ion collision mode. The LHC-b experiment is a dedicated b-physics experiment and will work at $L = 10^{32}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$. On the other hand the ALICE experiment is specialized on the study of heavy-ion collisions.

Because of the relation of the software developed during this diploma thesis to the reconstruction software of the CMS experiment, only this experiment will be covered in detail in section 2.3.1.

2.2.2 The International Linear Collider

The International Linear Collider (ILC) is a linear electron-positron collider. It is designed to be the immediate "consequence" of the LHC. The two follow clearly and explicitly complementary approaches which can be roughly outlined by calling one (LHC) the "discovery machine", and the other (ILC) the "precision machine". The discoveries expected at LHC will point the way ILC will go, and for each discovery the LHC will make, it is up to the ILC to provide comprehensive and precise information upon whatever will have been discovered [4].

The design of ILC is elaborated since 2005 by the international "Global Design Effort" (GDE) under the auspices of the International Committee for Future Accelerators (ICFA). The currently proposed design of the accelerator/collider is illustrated in figure 2.3.

The three most important requirements of the ILC are [17]:

- An initial center-of-mass energy of 500 GeV, with the ability to upgrade to 1 TeV,
- An integrated luminosity of 500 fb^{-1} at 500 GeV or equivalent at lower energies in the first four years,

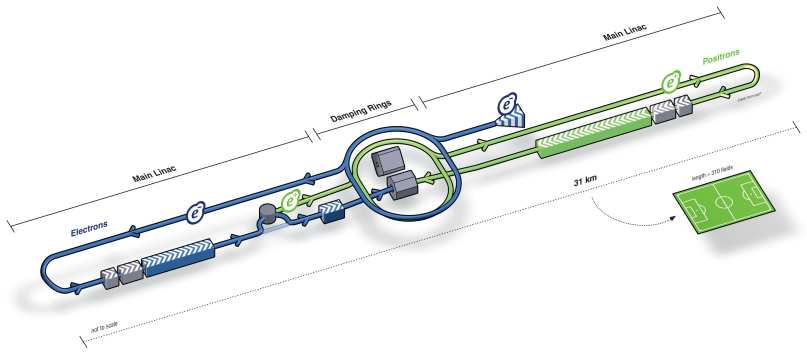


Figure 2.3: A schematic layout of the International Linear Collider. [Graphic courtesy of ILC / form one visual communication]

Parameter	Value	Units
Center-of-mass energy	500	GeV
Peak luminosity	2×10^{34}	$\mathrm{cm}^{-2}\mathrm{s}^{-1}$
Availability	75	%
Repetition rate	5	Hz
Duty cycle	0.5	%
Main Linacs:		
Average accelerating gradient in cavities	31.5	MVm
Length of each Main Linac	11	km
Beam pulse length	1	ms
Average beam current in pulse	9.0	mA
Damping rings:		
Beam energy	5	GeV
Circumference	6.7	km
Length of the Beam Delivery section (2 beams)	4.5	km
Total site length	31	km
Total site power consumption	230	MW

Table 2.3: Global Accelerator Parameters of the International Linear Collider at 500GeV center-of-mass energy [17].

- The ability to scan in energy between 200 and 500 GeV.

In 2004 an international review panel has decided the main linear accelerator (linac) to be based on 1.3 GHz superconducting radio-frequency (SCRF) accelerating cavities. These cavities are reported to be the most energy efficient way to achieve the 31.5 MV/m baseline average operational accelerating gradient required. The designed peak luminosity is $2 \times 10^{34}\ \mathrm{cm}^{-2}\mathrm{s}^{-1}$, but the maximum luminosity is not needed at the maximum energy. Together with an assumed collider availability of 75% and a realistic operation scenario (maintenance months and luminosity ramping included), this luminosity delivers the required $500\ \mathrm{fb}^{-1}$ in the first four years.

Apart from these values, the designed baseline performance goals of the ILC are the following [2] (table 2.3 summarizes the global parameters of the ILC):

- Beam energy stability and precision should be below the tenth of percent level at any energy.
- A polarisation of 80% of the electron beam within the full energy range must be possible.
- The interaction region should either allow for two simultaneous experiments or, if necessary for design and cost considerations, two experiments should be able to share one interaction region ("push-pull").
- Calibration runs at 91.2 GeV (Z^0 resonance) should be possible. For calibration low luminosity is tolerable.
- The accelerator should be run at modes with low beamstrahlung to make the background manageable, however, quantitative studies are required to work this out.

The site for building the ILC is yet to be chosen and there are several proposed candidates. Cost estimates of the Reference Design Report [9] are based on three sample sites:

- in Northern Illinois near the Fermi National Accelerator Laboratory,
- in Japan (region not disclosed),

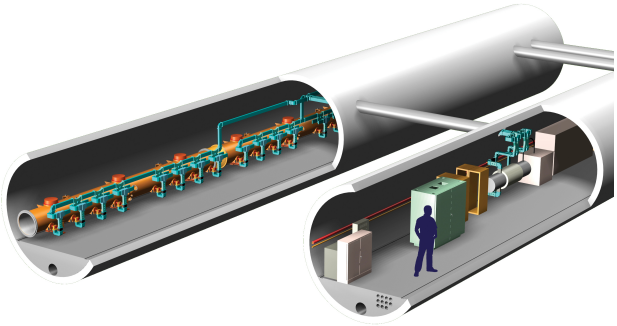


Figure 2.4: Artist's impression of the ILC tunnels. [Graphic courtesy of FermiLab/Sandbox Studio]

- in the Geneva region near the CERN laboratory.

All three use deep underground tunnels. In addition, two shallow tunnel sites are under investigation: one at the former TESLA site near DESY, and the other in the Moscow region near Dubna. Note that the final choice will not necessarily be made from among these sample sites.

Figure 2.4 gives a cutaway view of the deep underground tunnels housing the main linacs. These tunnels usually lie $100 - 150$ m underground and have an interior diameter of 4.5 m. They are separated by $5.0 - 7.5$ m which permits access to the equipment in the service tunnel. The two tunnels need three connections per RF unit: one for the waveguide, one for the signal cable and one for the power and high voltage cables.

2.3 Experiments

2.3.1 The Compact Muon Solenoid experiment

The Compact Muon Solenoid (CMS) experiment is one of four experiments at the LHC. CMS is a multipurpose experiment, which means it has been optimized with a couple of simultaneous goals in mind:

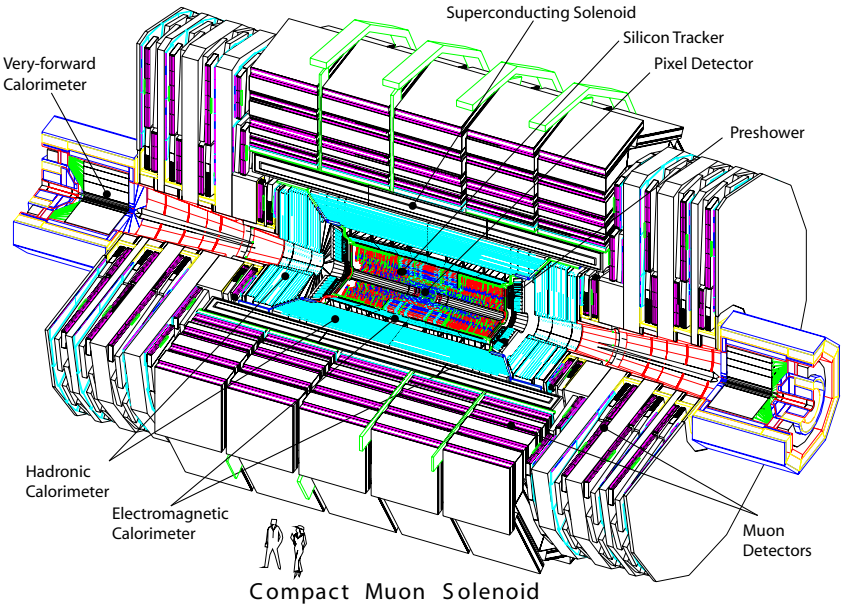


Figure 2.5: Drawing of the complete CMS detector showing scale and complexity.

- The search for the Higgs boson in proton-proton collisions with high luminosity. Even this one goal implies several requirements due to the different decay channels and thus different final states of the the Higgs boson depending on its mass.
- The search for evidence of physics beyond the Standard Model i.e. Supersymmetry or extra dimensions. This requires studies of final states with numerous jets and hard leptons.
- Studies in the field of b-physics, i.e. the oscillations of the neutral $B^0 - \bar{B}^0$ mesons.
- Study of top-quark physics, i.e. a precise measurement of the top mass.

The situations ("signatures") listed require the detector to do precise measurements of photons, muons and electrons over a large energy range. Also precise reconstruction of the tracks of charged particles over a large range of transverse momenta is needed. The importance of muon reconstruction is emphasized by the name of the experiment. Finally, the high event rates at LHC impose strong requirements on the granularity and readout speed of the detector.

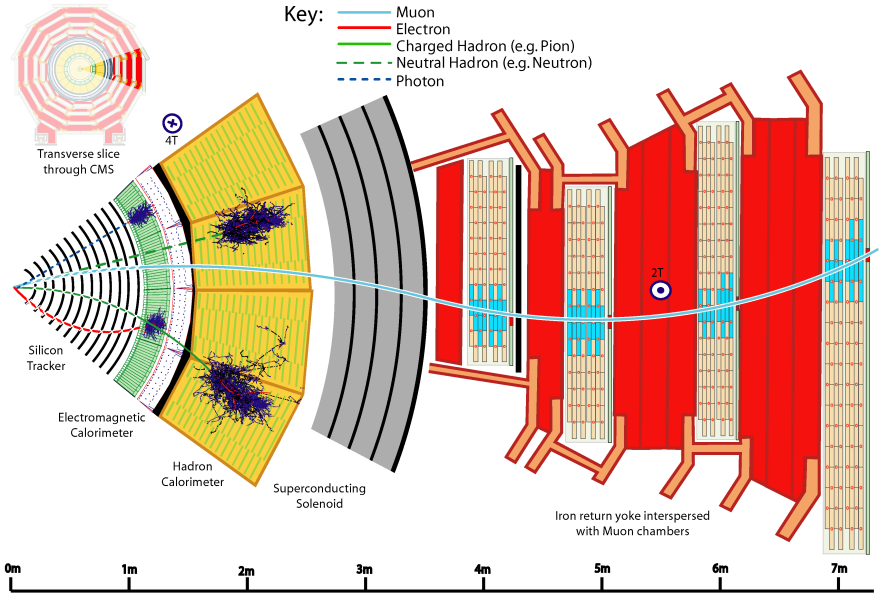


Figure 2.6: The components of the CMS detector and their role in detecting different types of particles.

The general layout of the CMS detector is shown in figure 2.5. The words "Compact" and "Solenoid" in the experiment name are somewhat related, and they outline the starting point of the detector design: the superconducting solenoid of 13 m in length and an inner diameter of 5.9 m generating a uniform magnetic field of 4 T. The tracker (for position and momentum measurement) and the calorimetry (for energy measurement) are compact enough to completely fit inside the solenoid. The only parts outside of the solenoid are the return yoke (returning the magnetic flux) and the muons chambers.

The different components of the detector together with their respective purposes are shown in figure 2.6. The central tracker is itself composed of the pixel detector and the silicon micro-strip tracker.

2.3.2 Experiments at the ILC

The physics studied at ILC will present real challenges to the detectors. The most important ones foreseeable by now being a high-resolution jet mass reconstruction, a high charged-track momentum resolution, and high-performance flavor and quark charge tagging, all well beyond the current state of the art. The

Measured quantity (examples)	Critical system	Critical detector characteristics	Required performance
Triple Higgs coupling Higgs mass BR($H \to WW^*$) $\sigma(e^+e^- \to \nu\bar{\nu}W^+W^-)$	Tracker and Calorimeter	Jet energy resolution, $\Delta E/E$	3 to 4%
Higgs recoil mass Luminosity weighted E_{cm} BR($H \to \mu^+\mu^-$)	Tracker	Charged particle momentum resolution $\Delta p_t/p_t^2$	5×10^{-5}
Higgs branching ratios b quark charge asymmetry	Vertex detector	Impact parameter resolution, δ_b	$\frac{5\mu m \bigoplus 10\mu m}{p\,[\mathrm{GeV/c}]\sin^{3/2}\theta}$
$\bar{\mu}$ mass (SUSY)	Tracker, Calorimeter	Momentum resolution, hermeticity	

Table 2.4: Sub-detector performance needed for key ILC physics measurements [5].

sub-detector performances needed for key ILC measurements are summarized in table 2.4.

Taking the current baseline design of the ILC, it foresees one interaction region equipped with two detectors. It is not yet clear whether the two detectors will be operated in a push-pull mode moving them in and out of the interaction region, or if there will be two beam delivery systems.

The International Large Detector

The International Large Detector (ILD) detector has recently been merged from the two previous Large Detector Concept (LDC) and the GLD concept. Figure 2.7 shows the latest LDC design. There is currently no distinguished ILD concept, but the two previous concepts are under convergence.

The key design motivations of the ILD concept are precision, high reliability and high redundancy. The detector tries to be prepared for the unexpected. The disadvantageous influence of detector material especially on tracking is an important issue, therefore the main tracker is based on a time projection chamber (TPC) minimizing the material budget.

The most obvious difference between the two preceding concepts (LDC and GLD) was their size and magnetic field. While the LDC favored a 4 T magnetic field and a TPC with an outer radius of 1.58 m, the GLD concept suggested a

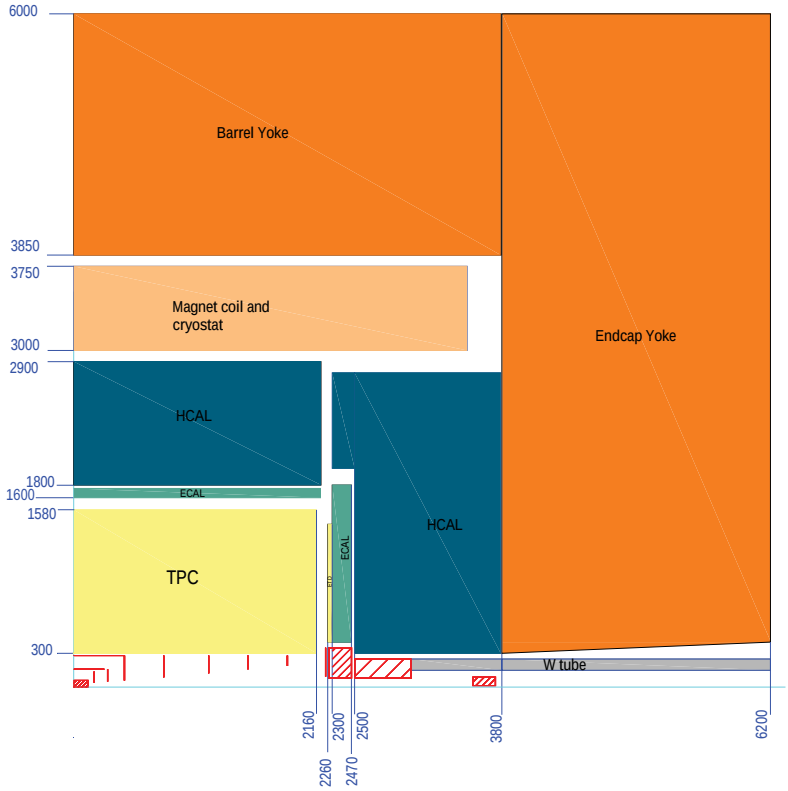


Figure 2.7: Schematic view of the LDC detector showing one quarter of the proposed design. Dimensions in the drawing are in mm. [6]

3.5 T field and a TPC with an outer radius of 2 m.

The cornerstones of the ILD design are a reliable and redundant tracking system, high-precision calorimetry based on Particle Flow Analysis (for best jet energy reconstruction and excellent particle identification) and hermeticity. These goals are met by a high-precision vertex detector, a large-volume TPC, supported by complete silicon tracking, particle flow calorimeters, excellent hermeticity, a 3 . . . 4 T solenoidal field, and an iron return yoke with muon instrumentation. The magnet is based on the CMS design.

A vertex detector and a silicon inner tracker are planned inside the TPC. The inner tracker of the LDC is shown in figure 2.8; in the following this system will be shortly described. Although the number of layers of both, the barrel and the forward region, is different at the GLD inner tracker, the overall system design is rather similar and will not be described separately here.

The tracking system of the LDC tries to optimize pattern recognition per-

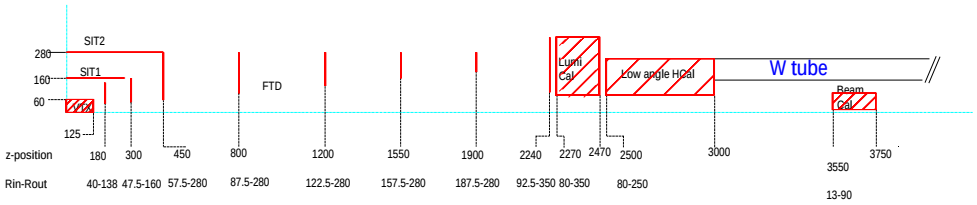


Figure 2.8: Enlarged view of the inner (Si) tracking and the forward detectors of the LDC detector. [6]

formance, momentum resolution and particle identification also in the forward region while minimizing material budget to minimize interference with electron and photon measurements in the calorimeter. The pixel vertex detector starts at an inner radius of 1.55 cm and with its five layers is optimized for excellent point resolution to provide very good bottom and charm tagging capabilities. The silicon tracker is equipped with two layers in the barrel region and six disks in the forward region, the first two of which are implemented as pixel detectors. It provides excellent linkage between the vertex detector and the TPC and extends the coverage of the tracking system to very forward angles.

The ILD collaboration inherited the participants from the LDC (mainly European) and GLD (mainly Asian) concepts. Almost 50 % of them are from Europe, and another third is from Asia, but there are also participants from North America. Overall some 170 groups from 28 countries work for the ILD collaboration.

The Silicon Detector

The Silicon Detector (SiD) concept incorporates silicon-tungsten based electronic calorimetry and all-silicon tracking. It attempts to optimize physics performance while constraining cost, and to be robust against physics and machine background. This last goal is assisted by the high readout speed of silicon detectors, thus most SiD systems will only record background from a single bunch crossing.

Similar to the ILD, the SiD design is guided by the idea to achieve the required jet energy resolution through Particle Flow Analysis. This leads to the choice of a highly pixellated electronic calorimetry and multi-layer and highly segmented hadronic calorimetry. In order to achieve the required resolution, both calorimeters need to be located within the solenoid. To be able to maintain the cost constraints, the solenoid and thus the calorimeters need to be designed as com-

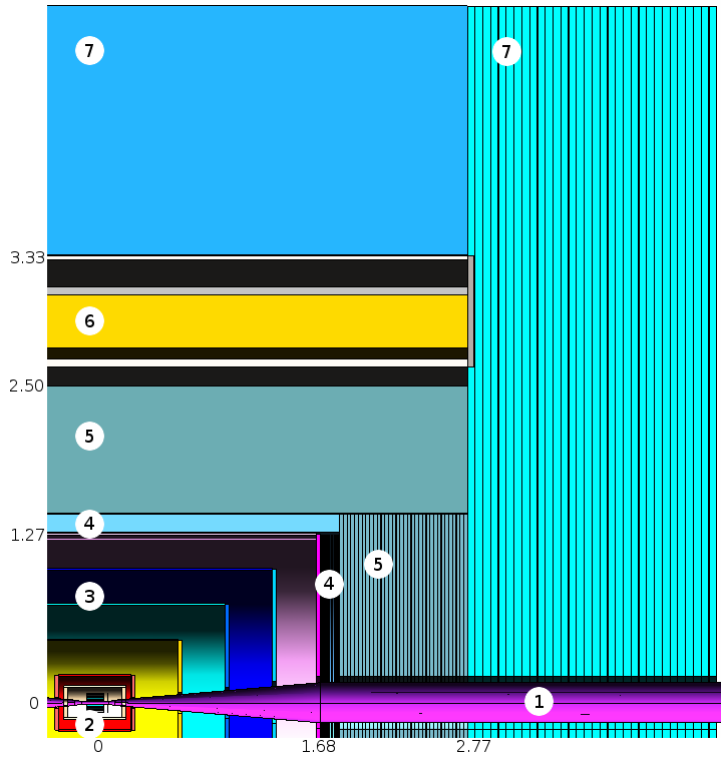


Figure 2.9: Illustration of a quadrant of SiD. The scale shown is in meters. The numbers are indicating the different parts of the detector: the beam pipe (1), the vertex detector (2), the all-silicon tracker (3), the electronic calorimeter (4), the hadronic calorimeter (5), the solenoid (6) and the return yoke instrumented as a muon system (7).

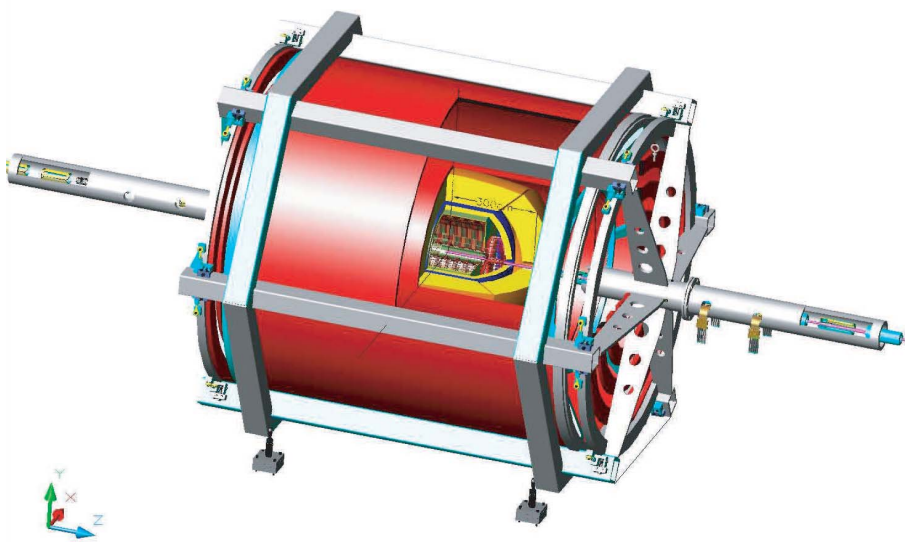


Figure 2.10: Cut-away view of the 4th detector concept. The colors identify the different parts of the detector: the vertex detector is blue, the TPC is green, the calorimeter is yellow and the dual solenoids are red.

pact as possible. A high magnetic field of 5 T should help maintaining separation performance of charged and neutral particles in the calorimeters. The solenoid begins at a radius of 2.50 m and is based on the CMS design. The high field it produces aids in the separation of particles in the calorimeters and provides high momentum resolution in the tracker.

The Fourth Concept

The name of the youngest detector concept originates from times when the LDC and GLD concepts were not yet merged into the current ILD concept. The Fourth (4th) concept is different from the other two in that it does not rely on particle flow reconstruction. Instead it utilizes a novel implementation of compensating calorimetry which is based on a dual read-out of both scintillation light from all charged particles and, separately, Cerenkov light predominately from the electromagnetic particles. A second big difference from the other detectors is the iron-free muon spectrometer inside a dual solenoid flux return.

The calorimeter design of the 4th concept is based on what is called the

DREAM (Dual-REAdout Module). These modules are assembled as towers pointing out of the interaction region. They have good transverse segmentation, no longitudinal segmentation and the read-outs with the photo detectors are located at the outer radius. The transverse segmentation is based on fine spatial sampling of two kinds of fibres. Spatial fluctuations are measured by scintillating fibres while Cerenkov fibres detect predominately the ultra-relativistic electrons. The two separate measurements allow the calorimeter alone to distinguish between particle types. To improve the hadronic energy resolution, the measurement of MeV neutrons liberated in each shower, e.g. by inclusion of a third fibre type, is planned.

The idea behind the dual-solenoid muon system is to return the flux while, at the same time, provide a moderately uniform field for a second muon bending improving the momentum measurement. The low mass of an iron-free system allows for easier detector exchange and also simplifies installation and future modifications of the detector.

3 Vertex reconstruction

The acquisition and analysis of the data produced by the experiments introduced above is a process with a lot of different steps each with its own challenges. For the topic of this writing it is of minor importance to understand all of them, but one should certainly understand the immediate input used for kinematic reconstruction. This is why this section starts with the explanation of how the vertex reconstruction is done. Track reconstruction and the track data used for the vertex reconstruction are explained in appendix A.1.

3.1 Vertex reconstruction

The tracks, reconstructed following the methods presented in appendix A.1, already hold physical data at a level of abstraction that lets us compare it to the decay models at test. Unfortunately the by far biggest part of the particles created in the primary collision is too unstable to traverse the detector layers and let us nicely reconstruct its parameters. The only hints we get from that big group of particles are their decay products, or worse, the decay products of their decay products. That by itself would not impose any major difficulties. The difficulties come from the fact that there is always more than one decay in each event. So before we start thinking about what kind of decay produced the tracks we measured, we have to associate the large amount of tracks usually found in one event to a small sample of decay vertices. And to enable lifetime studies of unstable decay products it would be interesting to know their flight distance; thus we have to reconstruct the vertices with the best precision possible from the tracks associated to them. Those two steps are in fact separate problems.

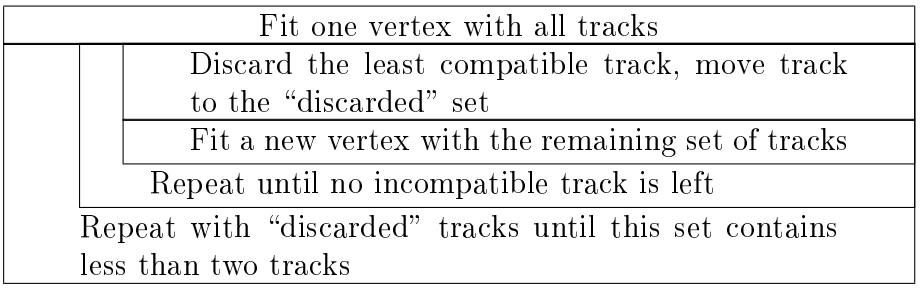


Figure 3.1: The algorithm of the principal vertex reconstructor.

3.1.1 Vertex finding

In the general case of multiple vertices in one event, it is necessary to sort the tracks found in that event into subsets which each share one common point of origin. This task is called vertex finding and the possible strategies to complete it shall be shortly introduced below. The most straightforward of those algorithms, the principal vertex reconstructor will be described in more detail.

The currently known vertex finding algorithms can be divided into hierarchic and non-hierarchic approaches [21]. The hierarchic approach not only sorts the tracks into subsets, but it builds a hierarchy of subsets where the bottom of the hierarchy are the single tracks and the top of it is the set of all tracks. The direction of construction of this hierarchy again subdivides the hierarchic algorithms into those which start from the top and follow a divisive approach and those which start from the bottom and follow an agglomerative approach.

Non-hierarchic approaches avoid a strict clustering hierarchy and instead involve all tracks in an iterative algorithm. Most of them use the notion of vertex prototypes which are attracted by the track data and over the iterations move to a stable point. The last positions of these prototypes are then the vertex candidates.

The principal vertex reconstructor is a hierarchic and divisive vertex finder. The algorithm is defined by the Nassi-Shneiderman diagram shown in figure 3.1. It has one explicit parameter: the cut on the track compatibility. This parameter quantifies whether a track is incompatible or not and is used in the inner loop condition. As the compatibility measure, the output of the vertex fit is the most obvious choice.

For this algorithm to find a secondary vertex, at least two tracks must be incompatible with the primary vertex and they must be compatible with one another.

3.1.2 Vertex fitting

The task of a vertex fitter is to calculate the position of the most probable common origin of a given set of tracks together with its error estimate. The track sets coming from the vertex finder therefore serve as input for a number of independent vertex fitter runs, one for each found vertex.

Linearization point finding

One thing all vertex fitters mentioned herein have in common is the requirement of a rough vertex estimate, a linearization point. Usually the vertex fitters are able to relinearize the input data in case the discrepancy between the linearization point and the vertex candidate gets too large. The quality of the linearization point estimation should therefore not play a crucial role. It is only when it comes to robust fitters that this initial guess plays a much greater role because those fitters are but solving a local optimization problem [21].

Because of their negligible influence on the final result for non-robust fitters, algorithms to estimate the linearization point should be fast. Usual implementations involve simple mode finding techniques like

LMS The one-dimensional Least Median of Squares estimator is the midpoint of the smallest interval that covers at least 50% of all data points

HSM The Half Sample Mode estimator is the result of a recursive application of the LMS estimator.

FSMW The Fraction-of Sample Mode with Weights estimator is a generalization of the HSM estimator to a coverage of $\geq 50\%$ and additionally taking into account the weights associated with each interval where the weight of an interval is defined as the length of the interval divided by the sum of all weights of the contained data points.

which are applied in a coordinate-wise manner to the set of the crossing points of all track pairs.

Global vertex fit

Once the linearization point is chosen, the most straightforward way to fit the vertex position is the application of a least squares fitting method similar to those introduced in section A.1.1. Only the parameters and the derivatives have to be rewritten. The track model is replaced by the vertex model which writes:

$$\mathbf{f}(\mathbf{x},\mathbf{q}) \approx \mathbf{f}(\mathbf{x}_0,\mathbf{q}_0) + \mathbf{A}_i \cdot (\mathbf{x}-\mathbf{x}_0) + \mathbf{B}_i \cdot (\mathbf{q}-\mathbf{q}_0) \tag{3.1}$$

$$\mathbf{A} \equiv \left.\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{q})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_0,\mathbf{q}=\mathbf{q}_0} \qquad \mathbf{B} \equiv \left.\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{q})}{\partial \mathbf{q}}\right|_{\mathbf{x}=\mathbf{x}_0,\mathbf{q}=\mathbf{q}_0} \tag{3.2}$$

Here $\mathbf{x}$ is the estimated vertex position and $\mathbf{q}$ holds the momentum information of the tracks at the vertex. In the case of n tracks, $\mathbf{f}$ has the dimension $5n$, $\mathbf{x}$ has three dimensions and $\mathbf{q}$ has the dimension $3n$.

If there is previous knowledge of the vertex position, e.g. the known beam spot for primary vertices, it can be included at this point by incorporating it into the vertex model as an additional three dimensional virtual measurement $\mathbf{v}$ along with the track parameters $\tilde{\mathbf{p}}$ found during the track fit. The objective function is

$$\begin{aligned} M(\mathbf{x},\mathbf{q}) = &\left[\begin{pmatrix}\mathbf{x}_0\\ \mathbf{f}(\mathbf{x}_0,\mathbf{q}_0)\end{pmatrix} + \begin{pmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{A} & \mathbf{B}\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}-\mathbf{x}_0\\ \mathbf{q}-\mathbf{q}_0\end{pmatrix} - \begin{pmatrix}\mathbf{v}\\ \tilde{\mathbf{p}}\end{pmatrix}\right]^T \mathbf{W} \\ &\left[\begin{pmatrix}\mathbf{x}_0\\ \mathbf{f}(\mathbf{x}_0,\mathbf{q}_0)\end{pmatrix} + \begin{pmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{A} & \mathbf{B}\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}-\mathbf{x}_0\\ \mathbf{q}-\mathbf{q}_0\end{pmatrix} - \begin{pmatrix}\mathbf{v}\\ \tilde{\mathbf{p}}\end{pmatrix}\right] \end{aligned} \tag{3.3}$$

where the weight matrix $\mathbf{W}$ includes the covariance matrix resulting from the track fit $\tilde{\mathbf{C}}$ and the errors of the prior vertex knowledge $\mathbf{C_v}$:

$$\mathbf{W} = \begin{pmatrix}\mathbf{C_v} & \mathbf{0}\\ \mathbf{0} & \tilde{\mathbf{C}}\end{pmatrix}^{-1} \tag{3.4}$$

The objective function is structurally identical with the one in the track fitting case (A.10), thus the same holds for the solution of the minimization:

$$\begin{pmatrix}\tilde{\mathbf{x}}\\ \tilde{\mathbf{q}}\end{pmatrix} = \begin{pmatrix}\mathbf{x}_0\\ \mathbf{q}_0\end{pmatrix} + \left(\mathbf{K}^T\mathbf{W}\mathbf{K}\right)^{-1}\mathbf{K}^T\mathbf{W}\cdot\left[\begin{pmatrix}\mathbf{v}\\ \tilde{\mathbf{p}}\end{pmatrix} - \begin{pmatrix}\mathbf{x}_0\\ \mathbf{f}(\mathbf{x}_0,\mathbf{q}_0)\end{pmatrix}\right] \tag{3.5}$$

with the abbreviation

$$\mathbf{K} \equiv \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{A} & \mathbf{B} \end{pmatrix} \tag{3.6}$$

The covariance matrix of the fitted parameters is again in analogy to the one in the track fitting case (A.14)

$$\operatorname{cov}\left[\begin{pmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{q}} \end{pmatrix}\right] = \left(\mathbf{K}^T \mathbf{W} \mathbf{K}\right)^{-1} \tag{3.7}$$

The Kalman Vertex Filter

The global vertex fitter presented above obviously has the same drawbacks as the global track fitter in section A.1.1: the cost of the inversion needed in equation (3.5) rises with the third power of the number of tracks included in the vertex fit. If used for track association, the performance is even worse, implying an inversion of a full $(3n'+3) \times (3n'+3)$ matrix for each potential association. Luckily, the Kalman filter eliminates these problems providing the possibility to associate the tracks iteratively and to reconstruct the vertex in a progressive manner. In fact, the Kalman Vertex Filter has completely displaced the Global Vertex Fitter in current vertex reconstruction software environments such as Rave [22].

The measurement equation is similar to the vertex model in equation (3.1), but in contrast to the global vertex model, the parameters here are only those of the one track used during the iteration step k:

$$\tilde{\mathbf{p}}_k = \mathbf{f}_k(\hat{\mathbf{x}}_k, \hat{\mathbf{q}}_k) + \epsilon_k \qquad \langle \epsilon_k \rangle = 0 \qquad \operatorname{cov}(\epsilon_k) = \tilde{\mathbf{C}}_k \tag{3.8}$$

To yield a linear problem, the function $\mathbf{f}$ is approximated by the first order Taylor expansion:

$$\mathbf{f}_k(\mathbf{x}_k, \mathbf{q}_k) \approx \mathbf{f}_k(\mathbf{x}_{k,0}, \mathbf{q}_{k,0}) + \mathbf{A} \cdot (\mathbf{x}_k - \mathbf{x}_{k,0}) + \mathbf{B} \cdot (\mathbf{q}_k - \mathbf{q}_{k,0}) \tag{3.9}$$

$$= \mathbf{t}_{k,0} + \mathbf{A} \cdot \mathbf{x}_k + \mathbf{B} \cdot \mathbf{q}_k \tag{3.10}$$

The system equation is particularly simple in this case:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} \tag{3.11}$$

The objective function of the iteration k can therefore be written immediately:

$$M(\mathbf{x}_k, \mathbf{q}_k) = \left[\mathbf{t}_{k,0} + \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{q}_k - \tilde{\mathbf{p}}_k\right]^T \tilde{\mathbf{C}}_k^{-1} \left[\mathbf{t}_{k,0} + \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{q}_k - \tilde{\mathbf{p}}_k\right] + \left[\mathbf{x}_k - \tilde{\mathbf{x}}_{k-1}\right]^T \tilde{\mathbf{R}}_{k-1}^{-1} \left[\mathbf{x}_k - \tilde{\mathbf{x}}_{k-1}\right] \quad (3.12)$$

Each iteration k adds the information of one track with parameters $\tilde{\mathbf{p}}_k$ and corresponding covariance $\mathrm{cov}\,(\tilde{\mathbf{p}}_k) = \tilde{\mathbf{C}}_k$ to the vertex estimate of the previous iteration $\tilde{\mathbf{x}}_{k-1}$ with corresponding covariance $\mathrm{cov}\,(\tilde{\mathbf{x}}_{k-1}) = \tilde{\mathbf{V}}_{k-1}$. The re-estimated track momentum is $\tilde{\mathbf{q}}_k$ with $\mathrm{cov}\,(\tilde{\mathbf{q}}_k) = \tilde{\mathbf{U}}_k$.

The minimization has to be done with respect to both, $\mathbf{x}_k$ and $\mathbf{q}_k$. To write the result, some abbreviations are helpful to maintain clearness:

$$\mathbf{W}_k = \tilde{\mathbf{C}}_k^{-1} \quad (3.13)$$

$$\mathbf{P}_k = \left(\mathbf{B}_k^T \mathbf{W}_k \mathbf{B}_k\right)^{-1} \quad (3.14)$$

$$\mathbf{W}_k^B = \mathbf{W}_k - \mathbf{W}_k \mathbf{B}_k \mathbf{P}_k \mathbf{B}_k^T \mathbf{W}_k \quad (3.15)$$

$$\tilde{\mathbf{x}}_k = \tilde{\mathbf{V}}_k \left[\tilde{\mathbf{V}}_{k-1}^{-1} \tilde{\mathbf{x}}_{k-1} + \mathbf{A}_k \mathbf{W}_k^B \left(\tilde{\mathbf{p}}_k - \mathbf{t}_{k,0}\right)\right] \quad (3.16)$$

$$\tilde{\mathbf{q}}_k = \mathbf{P}_k \mathbf{B}_k^T \mathbf{W}_k \left[\tilde{\mathbf{p}}_k - \mathbf{t}_{k,0} - \mathbf{A}_k \tilde{\mathbf{x}}_k\right] \quad (3.17)$$

$$\tilde{\mathbf{V}}_k = \left(\tilde{\mathbf{V}}_{k-1}^{-1} + \mathbf{A}_k^T \mathbf{W}_k^B \mathbf{A}_k\right)^{-1} \quad (3.18)$$

$$\tilde{\mathbf{U}}_k = \mathbf{P}_k + \mathbf{T}_k^T \tilde{\mathbf{V}}_k^{-1} \mathbf{T}_k \quad (3.19)$$

$$\mathbf{T}_k = \mathrm{cov}(\tilde{\mathbf{x}}_k, \tilde{\mathbf{q}}_k) = -\tilde{\mathbf{V}}_k \mathbf{A}_k^T \mathbf{W}_k \mathbf{B}_k \mathbf{P}_k \quad (3.20)$$

$$(3.21)$$

Robustifications

Although the least squares methods are the ideal choice for perfect data, their performance suffers strongly from real-world problems like misassigned tracks. These problems are addressed by what is called robustification, which summarizes the task of lowering the sensitivity of the vertex fitter with respect to misassigned tracks or, more generally spoken, outlying observations. There are two main approaches to accomplish this task:

- Allow the fitter to fully or partially ignore certain tracks during the fit.
- Change the target of the fitter from minimizing the mean of the residuals to minimizing a statistical measure less sensitive to outliers, e.g. the median of the residuals.

Trimmed vertex fit

The most obvious robustification follows the first approach and uses only a subset of the tracks initially associated with a vertex to actually fit the vertex. This kind of fitter does the following optimization:

$$\hat{\beta}_{LS} = \arg\min_{\beta} \sum_{i=1}^{h<n} r_i^2(\beta) \tag{3.22}$$

The global minimization of this function would imply the separate minimization of all possible combinations of picked tracks, which is not realistically feasible. Alternatively one can use the linearization point as initial guess and recursively select those tracks which are most compatible with the current recursion steps' estimation. This is a version of P. Rousseuw's Fast-LTS algorithm [11].

Weighted vertex fit

The next step in robustification is done by refining the previous approach of "hard" in/out association to "soft" association by weights. The objective function then rewrites to

$$\hat{\beta}_{LS} = \arg\min_{\beta} \sum_{i=1}^{n} w_i \cdot r_i^2(\beta) \tag{3.23}$$

where the weights w_i are defined by

$$w_i(\chi_i^2) = \frac{\Phi(\chi_i^2)}{\Phi(\chi_{\text{cutoff}}^2) + \Phi(\chi_i^2)} = \frac{\exp\left(\frac{-\chi_i^2}{2\,T}\right)}{\exp\left(\frac{-\chi_{\text{cutoff}}^2}{2\,T}\right) + \exp\left(\frac{-\chi_i^2}{2\,T}\right)} \tag{3.24}$$

This method is again applied in an iterative manner lowering the temperature parameter T for each step to steepen the weight function thus providing an annealing schedule for the association. The χ_i^2 in equation (3.24) are calculated

from the vertex displacement at iteration k:

$$\chi_i^2 = (\mathbf{x}_k - \mathbf{x}_{k-1})^T \mathbf{V}_k^{-1} (\mathbf{x}_k - \mathbf{x}_{k-1}) \tag{3.25}$$

The more general objective function (3.23) can be incorporated into a more general linear method relatively simple. But there are two issues one might be tempted to overlook:

- The number of degrees of freedom of the fit and thus the parameter of the expected χ^2 distribution is no longer integral. The number of observations has to be multiplied with their respective weight before adding them up.
- Also the covariance of each observation has to be multiplied with the corresponding weight on every occurrence except upon calculation of the weight parameter in equation (3.25).

The Least Median of Squares

The second approach of robustification mentioned above is that of optimizing a different statistical measure. This is what the Least Median of Squares (LMS) method does. Its objective function minimizes the median of the squared residuals instead of their sum:

$$\hat{\beta}_{\text{LMS}} = \arg\min_{\beta} \text{med}_{i=1}^{n} \, r_i^2(\beta) \tag{3.26}$$

Unfortunately this method has several drawbacks:

- There is neither a simple analytical solution nor fast numerical algorithm to find that minimum.
- There is no proper way to calculate the error of this method.
- The results obtained are not very exact and the algorithm is (statistically) not very efficient.

Nevertheless, the LMS is very robust and it is faster than the linear fitters.

3.1.3 Multivertex reconstruction

A multivertex fitter is different from the algorithms presented above in that the task of assigning tracks to vertices on the one hand and fitting those vertices

from the associated tracks on the other hand can no longer be strictly separated. A multivertex fitter still needs what is called a seeding, which is very similar to what is done by the vertex finders described above, but it takes the full output of such a vertex finder that is a group of track bundles each associated with one vertex candidate. Unlike the vertex fitters mentioned above, a multivertex fitter does not strictly stick with the track association done by the seeding, but it is able to reassign tracks to the vertices found throughout its fitting procedure. Furthermore the Multi-Vertex Fitter (MVF) implemented in Rave uses what is called soft assignment, which allows one track to be assigned to multiple vertices at the same time. The association between tracks and vertices is not done in a discrete manner but rather by fractional weights which are adjusted during the fitting procedure.

The algorithm is very similar to the weighted vertex fitter presented in section 3.1.2. Only the weights (3.24) are generalized to the multivertex situation:

$$w_{i,j} = \frac{\exp\left(\frac{-\chi^2_{i,j}}{2T}\right)}{\exp\left(\frac{-\chi^2_{\text{cutoff}}}{2T}\right) + \sum_{k=1}^{n} \exp\left(\frac{-\chi^2_{i,k}}{2T}\right)} \tag{3.27}$$

Here j is the index of the vertex candidate. Like the weighted vertex fitter, this algorithm works iteratively and the $\chi^2_{i,j}$ at iteration k are naturally defined to be

$$\chi^2_{i,j,k} = (\mathbf{x}_{i,j,k} - \mathbf{x}_{i,j,k-1})^T \mathbf{C}^{-1}_{i,k} (\mathbf{x}_{i,j,k} - \mathbf{x}_{i,j,k-1}) \tag{3.28}$$

3.2 Kinematic vertex reconstruction

The intent of kinematic vertex reconstruction is to incorporate prior knowledge or assumptions about the physics happening into the process of event reconstruction and thus possibly improve the results thereof. The type of this additional information may vary from conservation rules affecting momentum or energy reconstruction to assumptions on properties of involved particles like their masses.

To allow for different configurations of constraints, it is useful to develop a possibility to apply them in a sequential manner. This strategy will be discussed later. For now the most intuitive extension to the above methods is a global strategy.

3.2.1 Global strategy

To make the task of applying constraints to a set of (virtual) measurements accessible in an algorithmic manner, it has to be formulated in solvable equations. This is accomplished using the method of Lagrange multipliers by, rather counter-intuitively, introducing more variables into the fit. For compactness and clearness, the parameters of the vertex fit previously named $\mathbf{x}$ and $\mathbf{q}$ are now combined in the tuple α and accordingly, the virtual measurements $\mathbf{v}$ and $\tilde{\mathbf{p}}$ are combined in the tuple μ. The objective function of the Global Vertex Fitter (3.3), using this convention, writes

$$M(\alpha, \lambda) = [\mathbf{f}(\alpha_0) + \mathbf{K}\eta - \mu]^T \mathbf{W} [\mathbf{f}(\alpha_0) + \mathbf{K}\eta - \mu] \tag{3.29}$$

Let the constraint equation be written as $\mathbf{H}(\alpha) = \mathbf{0}$, where the dimension of $\mathbf{H}$ is r, the number of constraints. Then after the Taylor expansion about α_0, the constraint equation becomes

$$\frac{\partial \mathbf{H}(\alpha_0)}{\partial \alpha} (\alpha - \alpha_0) + \mathbf{H}(\alpha_0) \equiv \mathbf{D}\eta + \mathbf{d} = \mathbf{0} \tag{3.30}$$

$$\text{where} \qquad \mathbf{D} \equiv \frac{\partial \mathbf{H}(\alpha_0)}{\partial \alpha} \qquad , \qquad \mathbf{d} \equiv \mathbf{H}(\alpha_0) \qquad \text{and} \qquad \eta \equiv \alpha - \alpha_0 \tag{3.31}$$

The method of Lagrange multipliers now employs a new objective function $M(\alpha, \lambda)$ which is an extension of equation (3.29) whose minimization with respect to both parameters will give the LMS estimate obeying the constraints implied by the function $\mathbf{H}(\alpha)$:

$$M(\eta, \lambda) = [\mathbf{f}(\alpha_0) + \mathbf{K}\eta - \mu]^T \mathbf{W} [\mathbf{f}(\alpha_0) + \mathbf{K}\eta - \mu] + 2\lambda^T (\mathbf{D}\eta + \mathbf{d}) \tag{3.32}$$

The minimization with respect to λ yields the constraint equation. The solution of the global minimization gives the following expression:

$$\eta_c = -\mathbf{V}_u \mathbf{D}^T \mathbf{V}_D \mathbf{d} - \left(\mathbf{I} - \mathbf{V}_u \mathbf{D}^T \mathbf{V}_D \mathbf{D}\right) \mathbf{V}_u \mathbf{K}^T \mathbf{W} [\mathbf{f}(\alpha_0) - \mu] \tag{3.33}$$

with the abbreviations

$$\mathbf{V}_u \equiv \left(\mathbf{K}^T\mathbf{W}\mathbf{K}\right)^{-1} \qquad \mathbf{V}_D \equiv \left(\mathbf{D}\mathbf{V}_u\mathbf{D}^T\right)^{-1} \tag{3.34}$$

Comparing these results with equations (3.7) and (3.5) shows that $\mathbf{V}_u$ resembles the covariance resulting for the unconstrained vertex fit and that the last part of equation (3.33) resembles the result of the unconstrained fit η_u as presented in equation (3.5). The above result can thus be rewritten to separate the unconstrained and the constrained fit:

$$\eta_c = \eta_u - \mathbf{V}_u\mathbf{D}^T\mathbf{V}_D\left(\mathbf{D}\eta_u + \mathbf{d}\right) \tag{3.35}$$

Then the corresponding covariance matrix becomes

$$\mathbf{V}_c = \mathbf{V}_u - \mathbf{V}_u\mathbf{D}^T\mathbf{V}_D\mathbf{D}\mathbf{V}_u \tag{3.36}$$

Using this separation to replace the overall η in equation (3.32) with its constituents and factorizing the product around the weight matrix $\mathbf{W}$ into those parts containing only η_c and those which do not yields

$$\begin{aligned} M(\eta_u, \eta_c, \lambda) = &\left[\mathbf{f}(\alpha_0) + \mathbf{K}\eta_u - \mu\right]^T \mathbf{W}\left[\mathbf{f}(\alpha_0) + \mathbf{K}\eta_u - \mu\right] + \\ &\left[\eta_c - \eta_u\right]^T \mathbf{V}_u^{-1}\left[\eta_c - \eta_u\right] + 2\lambda^T\left(\mathbf{D}\eta_c + \mathbf{d}\right) \end{aligned} \tag{3.37}$$

The unconstrained fit minimizes only the first term and fixes η_u while the remaining terms are minimized by the constrained fit. The new objective function and the χ^2 resulting from the minimization when substituting equation (3.35) are

$$M(\eta_c, \lambda) = \left[\eta_c - \eta_u\right]^T \mathbf{V}_u^{-1}\left[\eta_c - \eta_u\right] + 2\lambda^T\left(\mathbf{D}\eta_c + \mathbf{d}\right) \tag{3.38}$$

$$\Rightarrow \chi_c^2 = \left(\mathbf{D}\eta_u + \mathbf{d}\right)^T \mathbf{V}_D\left(\mathbf{D}\eta_u + \mathbf{d}\right) \tag{3.39}$$

These equations form the basis of the kinematic fitting done in Rave as discussed in the next chapter. An extension of these results is to allow the constraint equation to include unknown parameters which will be optimized simultaneously [3] [12], but this method is currently not explicitly available within Rave and will not be discussed here.

3.2.2 Sequential strategy

In the previous section it has been shown, that the unconstrained and the constrained fit can be separated into two distinct steps. But for a general purpose kinematic fitter, which should be able to apply any combination of constraints to the given parameters, it would be useful to be able to split up the application of those constraints also. It will be shown here, that this is perfectly possible. For this purpose the elements of the second term of the objective function (3.38), which were contributed by the constraints, are split into two:

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{pmatrix}, \qquad \mathbf{d} = \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{pmatrix} \qquad \text{and} \qquad \lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} \tag{3.40}$$

Using this convention, the result of the global fit writes

$$\eta_c = \eta_u - \mathbf{V}_u \begin{pmatrix} \mathbf{D}_1^T & \mathbf{D}_2^T \end{pmatrix} \begin{pmatrix} \mathbf{D}_1\mathbf{V}_u\mathbf{D}_1^T & \mathbf{D}_1\mathbf{V}_u\mathbf{D}_2^T \\ \mathbf{D}_2\mathbf{V}_u\mathbf{D}_1^T & \mathbf{D}_2\mathbf{V}_u\mathbf{D}_2^T \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{D}_1\eta_u + \mathbf{d}_1 \\ \mathbf{D}_2\eta_u + \mathbf{d}_2 \end{pmatrix} \tag{3.41}$$

Now in a first step, only the constraint $\mathbf{D}_1\eta_1 + \mathbf{d}_1 = \mathbf{0}$ is applied. The objective function is

$$M(\eta_1, \lambda_1) = [\eta_1 - \eta_u]^T \mathbf{V}_u^{-1} [\eta_1 - \eta_u] + 2\lambda_1^T (\mathbf{D}_1\eta_1 + \mathbf{d}_1) \tag{3.42}$$

and the solution is like for the global constraint fit

$$\mathbf{V}_{D_1} \equiv \left(\mathbf{D}_1\mathbf{V}_u\mathbf{D}_1^T\right)^{-1} \tag{3.43}$$

$$\eta_1 = \eta_u - \mathbf{V}_u\mathbf{D}_1^T\mathbf{V}_{D_1} (\mathbf{D}_1\eta_u + \mathbf{d}_1) \tag{3.44}$$

$$\mathbf{V}_1 = \mathbf{V}_u - \mathbf{V}_u\mathbf{D}_1^T\mathbf{V}_{D_1}\mathbf{D}_1\mathbf{V}_u \tag{3.45}$$

$$\chi_1^2 = (\mathbf{D}_1\eta_u + \mathbf{d}_1)^T \mathbf{V}_{D_1} (\mathbf{D}_1\eta_u + \mathbf{d}_1) \tag{3.46}$$

The results of this first constrained fit are then used as input for the second step and replace the unconstrained parameters and errors. The solution is again

similar to the one known from the global fit

$$\mathbf{V}_{D_2} \equiv \left(\mathbf{D}_2 \mathbf{V}_1 \mathbf{D}_2^T\right)^{-1} \tag{3.47}$$

$$\eta_2 = \eta_1 - \mathbf{V}_1 \mathbf{D}_2^T \mathbf{V}_{D_2} \left(\mathbf{D}_2 \eta_1 + \mathbf{d}_2\right) \tag{3.48}$$

$$\mathbf{V}_2 = \mathbf{V}_1 - \mathbf{V}_1 \mathbf{D}_2^T \mathbf{V}_{D_2} \mathbf{D}_2 \mathbf{V}_1 \tag{3.49}$$

$$\chi_2^2 = \left(\mathbf{D}_2 \eta_1 + \mathbf{d}_2\right)^T \mathbf{V}_{D_2} \left(\mathbf{D}_2 \eta_1 + \mathbf{d}_2\right) \tag{3.50}$$

To show that the two separate steps are equivalent to the global fit, the results of the first step are expanded within the results of the second step. This yields

$$\begin{aligned} \eta_2 = \eta_u - \mathbf{V}_u \mathbf{D}_1^T \mathbf{V}_{D_1} \left(\mathbf{D}_1 \eta_u + \mathbf{d}_1\right) - \\ \mathbf{V}_1 \mathbf{D}_2^T \mathbf{V}_{D_2} \left(\mathbf{D}_2 \left[\eta_u - \mathbf{V}_u \mathbf{D}_1^T \mathbf{V}_{D_1} \left(\mathbf{D}_1 \eta_u + \mathbf{d}_1\right)\right] + \mathbf{d}_2\right) \end{aligned} \tag{3.51}$$

by switching to the pseudo-vectorized form defined by equation (3.40) one yields

$$\begin{aligned} \eta_2 = \eta_u - \mathbf{V}_u \begin{pmatrix} \mathbf{D}_1^T & \mathbf{D}_2^T \end{pmatrix} \cdot \\ \begin{pmatrix} \mathbf{V}_{D_1} + \mathbf{V}_{D_1} \mathbf{D}_1 \mathbf{V}_u \mathbf{D}_2^T \mathbf{V}_{D_2} \mathbf{D}_2 \mathbf{V}_u \mathbf{D}_1^T \mathbf{V}_{D_1} & -\mathbf{V}_{D_1} \mathbf{D}_1 \mathbf{V}_u \mathbf{D}_2^T \mathbf{V}_{D_2} \\ -\mathbf{V}_{D_2} \mathbf{D}_2 \mathbf{V}_u \mathbf{D}_1^T \mathbf{V}_{D_1} & \mathbf{V}_{D_2} \end{pmatrix} \cdot \\ \begin{pmatrix} \mathbf{D}_1 \eta_u + \mathbf{d}_1 \\ \mathbf{D}_2 \eta_u + \mathbf{d}_2 \end{pmatrix} \end{aligned} \tag{3.52}$$

Comparison of the inverse in equation (3.41) with the large matrix in equation (3.52) shows their equality. Thus the sequential application of constraints is equal to a global application of all constraints at once.

4 Implementation of kinematic fitting in Rave

The previous sections really serve as an introduction to the following two sections which treat the specific approach on kinematic fitting implemented by the Rave toolkit. As was already mentioned in the introduction, most of the kinematic reconstruction code found in Rave is adopted from the corresponding part of the CMSSW framework [18].

4.1 Interface

As the idea behind the creation of the Rave toolkit [20] always has been to be experiment independent [22], it was of particular importance to design the kinematic reconstruction facilities in a general yet not too complicated manner. For the task of kinematic reconstruction this claim is more challenging than for pure geometrical vertex reconstruction because the configuration of an algorithm for kinematic reconstruction has to be flexible not only in the number of its parameters but in its whole structure. A toolkit for kinematic reconstruction claiming experiment independence must allow the user to incorporate different types of hypotheses or knowledge together with an arbitrarily shaped decay model into the fit. The text-string based configuration used by the vertex fitting algorithms provided by Rave could theoretically have been extended to allow that type of configuration also, but that approach would have stretched that concept (which was originally chosen to keep the Rave programming interface as stable as possible) to a complexity neither helpful nor desirable.

Rave is completely implemented in the C++ programming language and pursues a purely object orient approach. One main principle ruling the design of Rave in general and the design of the kinematic fitter in particular is the sepa-

ration of data-classes and algorithmic classes. The classes implemented by Rave can be grouped into these two categories discussed below.

4.1.1 Data classes

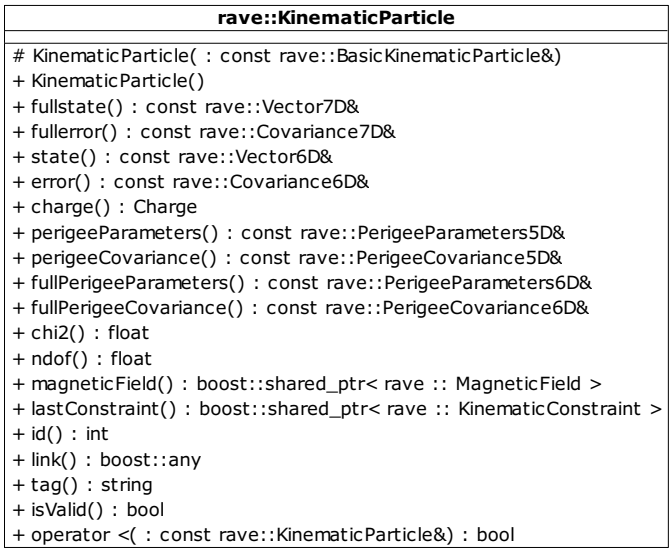
The major choice during the design of the kinematic fitting programming interface was to not artificially complicate the implicitly necessary complex interface, but to favour interface simplicity over interface stability. The result is an interface defined by a small number of easily identified classes. To emphasize the role of kinematic fitting as an information-refining rather than information-transforming process, the basic input data class, named KinematicParticle (figure 4.1(a)), is also the basic output data class.

To distinguish between KinematicParticle instances created by the user from the reconstructed tracks together with a mass hypothesis and those instances created by the kinematic reconstruction code, there is a separate sub-class for each of those cases. The KinematicParticle class itself serves as the abstract base class and is thus not instantiable. To generate the input objects for the kinematic fit, instances of the TransientTrackKinematicParticle class have to be created. To make this task as simple as possible, there are five available constructors allowing the user to either use existing Track instances, easily initialized Vector7D instances or even instances of the PerigeeParameters6D class which is initialized using a common perigee parametrization (see section B.2).

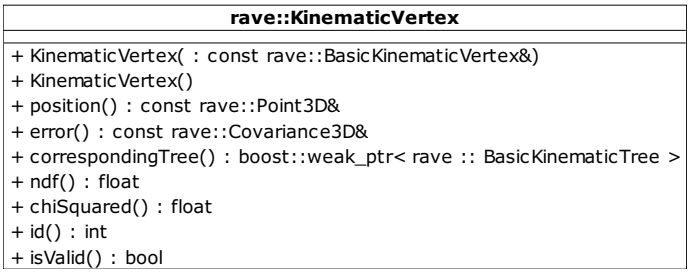
The other sub-class of the KinematicParticle class is the VirtualKinematic-Particle class whose name denotes that no direct measurement has been used to reconstruct it. Instances of this class cannot be created by the user but only by the kinematic reconstruction code.

The second output data class besides the KinematicParticle class is called KinematicTree (figure 4.1(c)) and is actually a structured container allowing for graph-like browsing of the reconstructed decay tree. As with KinematicParticle instances, also KinematicTree instances serve as input for some kinematic fitting methods, but unlike the first, they can not be instantiated manually, but only by invocation of the algorithmic class KinematicTreeFactory introduced in the next section.

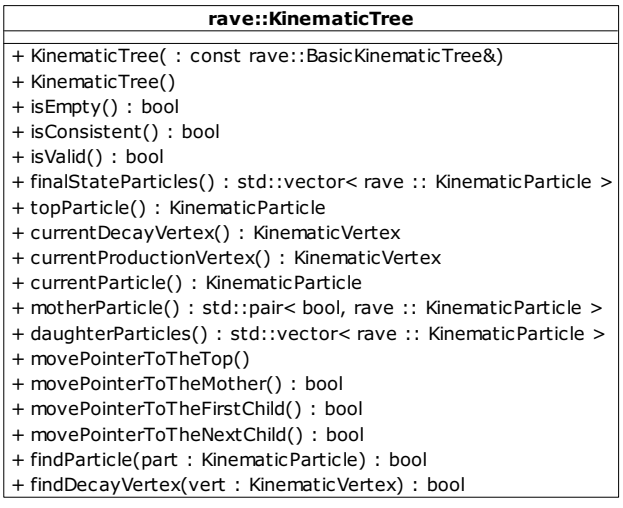
Inside the KinematicTree instance the final states are instances of the Transient-TrackKinematicParticle class initialized by the user while the other edges of



(a) KinematicParticle



(b) KinematicVertex



(c) KinematicTree

Figure 4.1: The Rave data classes

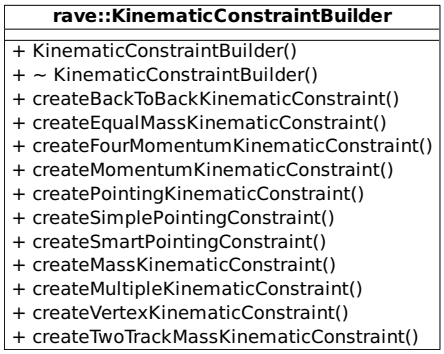


Figure 4.2: The KinematicConstraintBuilder class. The signatures (parameters and return values) of the methods are not shown.

the graph are instances of the VirtualKinematicParticle class initialized by the kinematic reconstruction code.

Those familiar with basic graph ideas will probably have recognized the KinematicParticle instances as being the edges of the graph-like KinematicTree structure, but they are still missing the other major component of graphs: the nodes. This is where the KinematicVertex (figure 4.1(b)) class fits in.

The last input class, which is the only pure input class, is the KinematicConstraint class. Instances of this class can be created by invocation of the different methods of the KinematicConstraintBuilder class (figure 4.2). The fitting methods supplied by the KinematicTreeFactory class have one parameter taking an instance of the KinematicConstraint class. If multiple constraints should be applied during the same fitting step, the user has to take the detour of creating an instance of the MultipleKinematicConstraint class with the KinematicConstraintBuilder and then adding the desired constraints using its addConstraint method.

Internally, the instances of the KinematicConstraint class of course differ if they have been created by different methods of the KinematicConstraintBuilder, but currently there is no possibility to find out which constraint a given instance is actually representing.

4.1.2 Algorithmic classes

The required flexibility of the kinematic fitter with respect to the decay model is only achievable because each specific fit can be sub-divided into a fixed set of basic fitting steps as has been shown in section 3.2.2. The user then con-

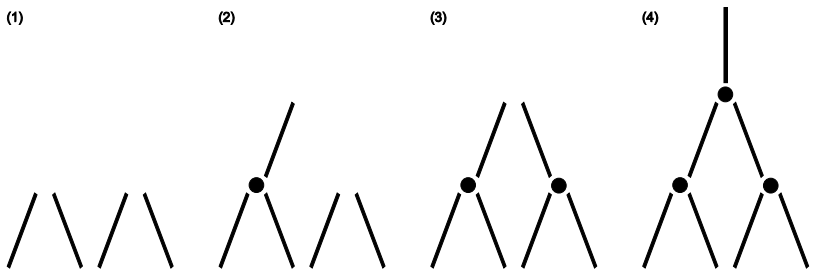


Figure 4.3: The construction of a specific topology from basic topologies.

figures the desired decay topology by composing a decay model from a set of basic sub-models. Both the construction of the basic sub-models as well as their composition to the final model are tasks of the KinematicTreeFactory class.

Figure 4.3 illustrates the process of the construction of a multivertex decay tree. Step (1) shows the input for the kinematic fitting. There are four final states reconstructed by the previous particle reconstruction. They are represented by four instances of the TransientTrackKinematicParticle class created during this first step.

The second step (2) chooses two out of four final states and applies the hypothesis that these two final states originated from a single particle at an unknown position. The two additional shapes represent the output of the kinematic fitting: an instance of the KinematicVertex class represented by the big dot and an instance of the VirtualKinematicParticle class represented by the additional line. For this example, only the constraint of a common vertex of the two final states is applied.

The third step (3) is very similar to the previous one. Again two new objects have been created: another instance of the KinematicVertex and of the VirtualKinematicParticle class.

The fourth and last step (4) uses the two virtual particles reconstructed in the two previous steps and applies the hypothesis that they both originate from one original particle themselves. The output of the fit is again a reconstructed vertex and a reconstructed particle. If the hypothesis is more detailed e.g. saying that the two secondary particles are the same, this can by included into the fit by applying a mass constraint. The resulting χ^2 can then be used to resolve the combinatorial problem of the first step, to do the correct association of two out of four final state particles in the first two steps.

To use the kinematic fitting facilities of Rave, an instance of the KinematicTreeFactory class has to be created. The available constructors take optional arguments. If no arguments are given, an artificial scenario with a homogeneous 4 T magnetic field and no matter (only vacuum) will be assumed. The arguments allow the user to specify an arbitrary magnetic field by implementing a sub-class of the abstract MagneticField class. Furthermore an arbitrary matter configuration is possible thanks to the Propagator concept. By implementing a sub-class of the abstract Propagator class, the user can inject an arbitrary propagation algorithm into the Rave reconstruction process. The default (the VacuumPropagator class) is adequate if the final states are already reached inside the beam pipe. The verbosity argument controls the level of detail of the produced output. This serves debugging and tracing purposes and usually can be left at the default. The second constructor takes additional beam spot information.

An instance of the KinematicTreeFactory class provides two methods for the actual kinematic fitting. The useVertexFitter method takes a set of particles and reconstructs an original particle assuming that the input particles were the decay products of the reconstructed particle. It uses the global kinematic fit described in section 3.2.1. The constraints parameter is optional. If it is not given, an unconstrained vertex fit is done. The result is an instance of the KinematicTree class containing the newly created virtual particle and the given input particles.

The useParticleFitter method implements the sequential strategy described in section 3.2.2. It takes a set of trees or particles as the input and performs a refit obeying the constraints given as the second argument. If the input is a set of trees, only their top particles are considered in the fit.

4.1.3 General remarks

All Rave interface classes are implemented as reference counting proxy classes. This means, that the data classes themselves are not carrying any data, but rather reference-counted pointers to the internal base classes. Copying of these classes is therefore no memory overhead because only the internal pointer is copied. Since alteration of data objects by the user after their initialization is generally not supported, duplication of the data would be unnecessary in either case. Thus memory management difficulties are generally avoided as long as no pointer to these classes are used. The use of pointers in connection with Rave classes would

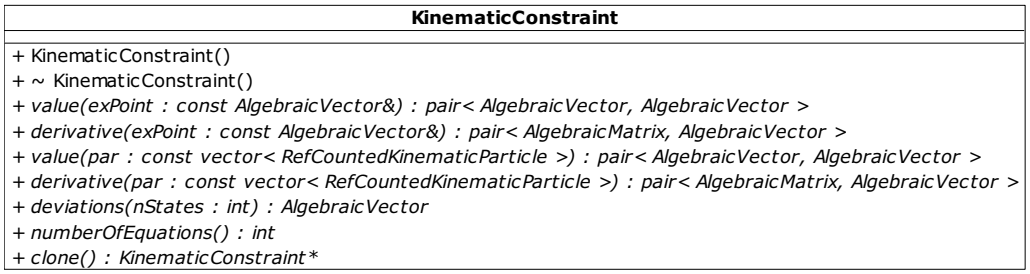


Figure 4.4: The abstract KinematicConstraint base class used internally.

annul the internal reference counting and is therefore strongly discouraged.

4.2 Implementation

The implementation of the kinematic fitting is designed to be modular and extensible. The modularity is achieved by using common abstract base classes for the algorithmic classes possibly receiving extension. This concerns mainly those parts of the algorithm involving constraints as new types of constraints need to be easily added, namely those classes involved in the sequential constraint fit and accessed through the useParticleFitter interface method.

4.2.1 Constraints

The constraints involved in the kinematic fit are internally represented by subclasses of the abstract KinematicConstraint class. The KinematicConstraintBuilder interface class hides this structure by providing construction methods for each constraint type.

The methods implemented by each constraint provide the fitter with the $\mathbf{d}$ tuple and the $\mathbf{D}$ matrix as they are defined in equation (3.30) along with a method returning the number of constraints and thus the expected impact on the χ^2 distribution and a method for duplication of the class instance.

4.2.2 Particle fitter

The constraint refit via the useParticleFitter method of the KinematicTreeFactory class is split into two parts treated by different algorithmic classes. The ParentParticleFitter class does the actual constrained refit involving only the

top particles of the given trees while the task of the ChildUpdator class is to propagate the additional information gained from the constraints to any decay products referenced by the given trees. The latter is currently not implemented.

At the time of this writing there are two ParentParticleFitter sub-classes. One uses the mass as the seventh parameter of each particle (in addition to three position and three momentum parameters) while the other uses its energy. The constraints have to be chosen such as to account for this difference if they involve mass or energy information. The Rave user can select between the two by invoking useParticleFitter with the optional parameters string set to either "ppf:lppf" for the mass-oriented fitter and "ppf:leppf" for the energy-oriented fitter. There are currently no constraints available for the second.

4.2.3 Algebra

During this diploma thesis the CMSSW framework was undergoing a change of the primary algebra interface. This change also affected the code included in Rave. The previously used matrix-multiplication algorithms provided by the CLHEP library were replaced by those provided by the ROOT SMatrix class and its relatives. Since that evolution had to be done in a smooth manner, by the time of this writing Rave depends on both, CLHEP and ROOT classes.

4.3 Debugging facilities

The debugging facilities of the kinematic fitting code in Rave are based on structured logging. Dependent on the required detail of debugging information, Rave is able to stream arbitrary debugging data to the standard output. This debugging information is tagged with location information down to a source-line level for the full-detail configuration. Together with input data reproducibility this provides the possibility to trace errors also without run-time debugging.

To ease the implementation of data logging, all data classes come with streaming operators and thus provide an intuitive streaming interface for logging and debugging purposes.

4.4 Validation in Vertigo

As already mentioned in the introduction of this chapter, Rave is a toolkit, which means, it is not directly usable by itself. Instead it ought to be included in a reconstruction framework. Modern reconstruction frameworks, like Marlin [13], allow for easy inclusion of tools like Rave. This modularity decouples the different parts of the reconstruction chain and simplifies decentralized development of those parts.

To be independent of any experiment-specific reconstruction frameworks and to be able to fully test the interface under development and changes to it, a mini-framework called Vertigo has been developed. Vertigo serves two roles: First it is a fully functional but pure vertex reconstruction framework. The Vertigo user provides the reconstructed track data in one of the accepted formats and Vertigo reads that data, uses Rave to do the reconstruction and outputs the results in one of the provided formats. As a console-based application, Vertigo is configurable purely by command line parameters.

4.4.1 Python bindings

The second role of Vertigo is that of a helper-class and language interface provider for the Python scripting language. That interface allows access to the complete Rave programming interface without the need to compile anything. The possibility to use Rave from a Python script is especially useful for testing and debugging because even complex test cases are implemented quickly with Python. For the kinematic fitting interface that advantage is even bigger because it relies on configuration through iterative method calls and a sophisticated class tree. The use of Rave for kinematic reconstruction through Vertigo is only possible using the scripting interface for the same reason it is not configurable through a text string-based interface.

4.4.2 Event generation

Another feature of Vertigo is its ability to generate appropriate input for reconstruction testing. It thus removes the hassle of looking for and managing input data files if a rudimentary event generation suffices. The event generators provided by Vertigo are called guns and there is one gun generating $Z \to \mu\mu$ type

events with full kinematic information attached. The example 4.1 is a Python script using this Vertigo gun for event generation and then using the Rave kinematic reconstruction with a constraint on the Z mass to reconstruct the original Z boson.

Listing 4.1: Kinematic reconstruction using python

```
1 import vertigo
2 eventfactory = vertigo.EventFactory ("gun:kinematics")
3 treefactory = vertigo.RaveKTreeFactory()
4 builder = vertigo.RaveKConstraintBuilder()
5 event = eventfactory.next()
6 zmass = 91.187
7 constraint = builder.createTwoTrackMassKinematicConstraint(zmass)
8 tree = treefactory.useVertexFitter(event.particles(), constraint)
9 print tree.topParticle().fullstate()
```

4.5 Application of Rave within the ILC framework

It has been an explicit goal of this diploma thesis not only to implement the kinematic reconstruction facilities in Rave but also to demonstrate their usability for the International Linear Collider (ILC) event reconstruction process. Because the different experiments have come to develop different software frameworks for their needs, only one of them (ILD) has been chosen to be extended to use the Rave library. Thanks to the common data interface of all ILC software frameworks, a subsequent inclusion with other experiments should be easy. This chapter first gives an overview of the most important components of the ILD software framework and will then show how Rave has been embedded with this framework.

4.5.1 The ILD software framework

At the heart of the ILD software framework are two distinct components: The event simulation is done by Mokka and the reconstruction by Marlin. Between these two components the data is transferred in database files. Because they are completely separate and Rave is a pure reconstruction library, Mokka will not be discussed herein. However as it is the common data format shared by all ILC experiments, first the LCIO persistency framework will be introduced.

LCIO

The Linear Collider Input/Output (LCIO) persistency system has been designed to provide a common data model for detector studies. All groups have come to use LCIO as their simulation output and analysis input format, which was probably also due to its native support for both C++ and Java bindings. Having a common input format for data analysis is especially useful to compare the performance and share results between detector concepts.

The LCIO data model is based on the typical structure of high-energy physics data-sets. The first level of structure is the file-system level. LCIO databases are saved and deployed in files with the slcio extension. The second level of structure is the event level. Each file contains a number of events which can be read from the file in a sequential manner. The events are structured into so-called collections where each collection contains an arbitrary number of data objects but all of the same type.

Directly after the simulation, the events typically contain collections for the simulated detector hits and the particle information from the physics generator. During reconstruction, more and more collections are added to the events containing the results of the tracking, the vertexing or other reconstruction steps.

The collection types used for the purposes of Rave are

- EVENT::Track
- EVENT::Vertex and
- EVENT::ReconstructedParticle.

They contain parameters and errors but also information about previous fit results and references to the data objects used during previous fits. Detailed documentation to the whole LCIO programming interface can be found at the LCIO website [1]. The track parametrization used in LCIO is described in [15]. The conversion matrices for conversion between LCIO and Rave conventions are derived in appendix B.

Marlin

The Marlin framework is the analysis framework used with the ILD collaboration. It is based on the LCIO data model and defines itself as general processing

framework for LCIO data. The idea behind Marlin is, that each reconstruction or analysis task can be applied to the data separately and sequentially and can thus be encapsulated in a module called a Processor. Furthermore, the events stored in a LCIO file are independent. Marlin allows to configure a stack of processors and operates them on the events read form a given LCIO file one by one.

Each processor defines a set of collections he needs for input and another set of collections he will append to each processed event. The Marlin configuration accommodates the modularity gained by the processor concept by using a hierarchical XML structure. Marlin allows each processor to register a set of parameters that will be read from the configuration file and handed over back to him at the beginning of the reconstruction run. This per-processor configuration is encapsulated in an XML block of the tag "processor". It usually holds the names of the collections to be read and written together with some parameters configuring details of the processor's behaviour.

4.5.2 Rave processors

This section documents the implementation of a Marlin plug-in providing Marlin users with the vertex and kinematic reconstruction possibilities of Rave. The plug-in developed is called MarlinRave and it provides two new processors to the Marlin framework: the RaveVertexing processor and the RaveKinematics processor.

The RaveVertexing processor

The vertexing capabilities of Rave are accessed from Marlin through the RaveVertexing processor. The configuration of the processor requires an input collection containing EVENT::Track objects. It takes a text string configuring the vertexing algorithm to be used together with its parameters. Further the names of the two output collections, one containing the reconstructed vertices and the other containing the refitted tracks. The last two parameters called HistFile and Verbose, serve debugging purposes and are not needed for usual operation. Below is an example of how the steer file section configuring the RaveVertexing processor may look like:

Listing 4.2: Marlin steer file section for vertex reconstruction

```
<processor name="MyRaveVertexing" type="RaveVertexing">
  <parameter name="HistFile" type="string"> </parameter>
  <parameter name="Method" type="string">default </parameter>
  <parameter name="RefittedTracks" type="string"> </parameter>
  <parameter name="Tracks" type="string">Tracks </parameter>
  <parameter name="Verbose" type="int">0 </parameter>
  <parameter name="Vertices" type="string">Vertices </parameter>
</processor>
```

The RaveKinematics processor

Because the configuration interface of Marlin does not allow for arbitrary nesting, the RaveKinematics processor has been designed to include a flexible number of predefined kinematic decay topologies. It facilitates the inclusion of new topologies as needed by the user. From the Marlin configuration file only the topology is chosen and a text string allows the passing of parameters from the configuration file to the selected topology.

A new topology To facilitate comparison, the same topology reconstructed by the Python example in section 4.4 shall now be reconstructed using Marlin. First, the RaveKinematics plug-in has to be extended for the new topology. To do so, a topology file including the following source code has to be dropped inside the topology directory of MarlinRave and MarlinRave has to be recompiled.

Listing 4.3: Topology file for MarlinRave

```
#include "KinematicTopology.h"
#include <rave/KinematicTreeFactory.h>
#include <rave/KinematicConstraintBuilder.h>

class TopologyTwoTrackMass :
  public KinematicTopology
{
  public:
    void setup(const std::string & parameters) {
      return;
    };
    rave::KinematicTree build(
      const rave::KinematicTreeFactory & factory,
      const std::vector< rave::KinematicParticle > & particles,
      const int verbose = 0) const
    {
      rave::KinematicConstraintBuilder builder;
```

```
18        rave::KinematicConstraint cs =
19            builder.createTwoTrackMassKinematicConstraint(91.187);
20        return factory.useVertexFitter(particles, cs);
21      };
22      bool valid() const { return true; };
23  };
24
25  #include "TopologyBuilder.h"
26  namespace {
27    TopologyBuilder< TopologyTwoTrackMass > t(
28      "TwoTrackMass",
29      "Applies Z-mass constraint.");
30  }
```

The first line is needed in every topology file because topologies are defined by inheritance of the KinematicTopology class as shown in line 6. The second and third lines make the Rave interface accessible to the topology. The class name of each topology can be freely chosen but should be unique. The processor learns about the new topology from the TopologyBuilder which is a template accepting any topology class as shown in line 27. Each working topology has to implement at least the build method (line 12). It is this method where the actual reconstruction happens. The setup method (line 9) is optional. It takes a text string and may initialize private variables with parameters read from that string. The valid method is checked before every use of a topology and has to return **true** if the topology is to be used.

As mentioned above, the TopologyBuider registers the topology with the RaveKinematics processor. It takes two arguments: The name in line 28 is used in the Marlin steer file to identify the topology to be used for reconstruction. The description in line 29 should summarize the hypothesis on which the topology relies to do the kinematic reconstruction.

Configuration All source files inside the topology directory will be included into the RaveKinematics processor. The choice of the topology actually used for reconstruction is done by the Marlin steer file. The section in the Marlin steer file configuring the kinematic reconstruction could look like the following:

Listing 4.4: Marlin steer file section for kinematic reconstruction

```
1 <processor name="MyRaveKinematics" type="RaveKinematics">
2   <parameter name="Particles" type="string">
3     Particles
```

```
4   </parameter>
5   <parameter name="Topology" type="string">
6     TwoTrackMass
7   </parameter>
8   <parameter name="Parameters" type="string"> </parameter>
9   <parameter name="Verbose" type="int">0 </parameter>
10  <parameter name="KinematicParticles" type="string">
11    KinematicParticles
12  </parameter>
13  <parameter name="KinematicVertices" type="string">
14    KinematicVertices
15  </parameter>
16 </processor>
```

The topology used for reconstruction is selected in line 6. The name given here must be the name of one of the topologies RaveKinematics provides. Line 3 must hold the name of the input collection. The type of the collection has to be LCIO::ReconstructedParticle. The content of line 8 is passed to the setup method of the selected topology and can be used to allow reconfiguration of the topology without recompilation. Lines 11 and 14 hold the names of the output collections of the kinematic fit. All reconstructed particles form a new collection named by line 11 while the vertices found form another collection named by line 14.

5 An exemplary reconstruction

5.1 Preamble

In this final chapter, the kinematic fitting with Rave is demonstrated for an exemplary reconstruction. The background present in every realistic data sample will not be simulated, however, to eliminate related issues. In practice, background reduction will be of major importance and will happen at an early stage. Due to the huge primary data rate it is done in progressive stages with increasing computing time, where the first few stages are done online and thus can not use computationally intensive algorithms ("multi-level trigger"). The first filtering stages usually apply very basic cuts e.g. on the phase-space population. Then constraints are used for diffuse empirical cuts and later also well-defined statistical tests with empirical cuts can be applied. The whole process effectively defines event masks using Monte-Carlo simulated data samples and applies them on the real data samples. Of course statistical equivalence of simulated and real data is required for this to work. It is common practice to always produce two independent Monte-Carlo samples where one is used for the cut determination and the other is used to test the statistical influence of the cuts on the analysis.

The two aims of the reconstruction presented below are the correct association of the final states with the expected W bosons and the optimal extraction of information from the selected configuration. The strategy to achieve these goals can be split into two separate steps: jet association and kinematic reconstruction.

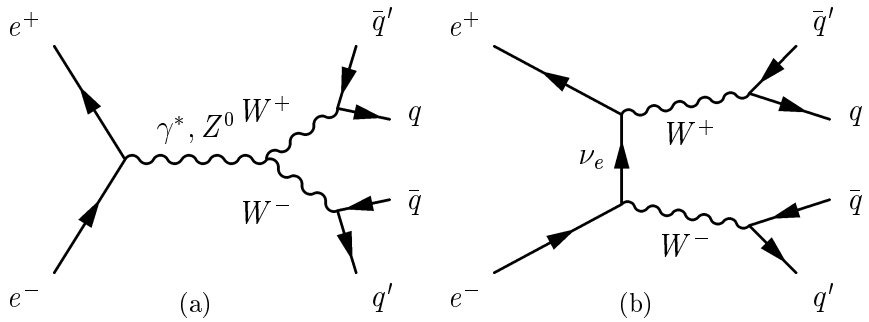


Figure 5.1: Tree-level Feynman diagrams of the decay channel selected for the exemplary reconstruction ($q = u$ or c, $q' = d$ or s or b).

5.2 Physics

5.2.1 General

The channel treated here will be one of the channels possibly analyzed at the ILC experiments (see section 2.3.2). It is defined by a pair of W bosons decaying into four jets in the final state. The tree-level (first order) Feynman diagrams of the process are shown in figure 5.1. The quantity of interest is the mass of the W boson.

The first thing to do is of course to think about what the expected outcome of this analysis will be and thus to justify the feasibility of this analysis. To be able to reference the particles involved in the decay in the following, they are labeled in an unambiguous manner and the decay writes

$$e_1\, e_2 \rightarrow W_A\, W_B \rightarrow j_1 j_2 j_3 j_4$$

The law of momentum conservation then claims that

$$p_1 + p_2 = q = k_A + k_B = \underbrace{k_1 + k_2}_{k_A} + \underbrace{k_3 + k_4}_{k_B} \tag{5.1}$$

Even though in our case both electrons have the same energy ($q = 0$) and hence the momenta of the Ws must be diametrical, in general their energies need

not be equal:

$$k_A = -k_B \qquad\qquad E_A \neq E_B \Leftrightarrow m_A \neq m_B \tag{5.2}$$

5.2.2 Correlation of W masses

It follows immediately that also the masses of the W bosons need not be equal. Their expected distribution is given by the following cross section [10]:

$$\sigma(s) = \int_0^s \mathrm{d}s_A \int_0^{s-s_A} \mathrm{d}s_B \sigma^*(s, s_A^2, s_B^2)\rho(s_A)\rho(s_B) \tag{5.3}$$

$$\text{where} \qquad \rho(s_i) = \frac{1}{\pi}\frac{\Gamma_W(s_i)\sqrt{s_i}}{\left(s_i - M_W^2\right)^2 + s_i\left(\Gamma_W(s_i)\right)^2} \tag{5.4}$$

$$\text{and} \qquad \sigma^*(s, s_A^2, s_B^2) \propto \kappa^{1/2}(s, s_A^2, s_B^2) \tag{5.5}$$

The contributions to equation (5.3) closely resemble the relativistic Breit-Wigner distribution. For high energies s, σ^* depends only weakly on s_A and s_B due to the definition of κ (5.6) hence little or no correlation between the found W masses is expected at the 500 GeV operating energy of the ILC.

$$\kappa(a, b, c) = \sqrt{(a - b - c)^2 - 4bc} \tag{5.6}$$

The probability of pair-produced W bosons to fully hadronize, i.e. to result in four jets in the final state, is the square of the branching ratio of a single W to hadronize, which can be found in [23], and thus

$$\frac{p\left(W^+W^- \to 4\,\text{jets}\right)}{p\left(W^+W^- \to \text{anything}\right)} = \left(\frac{\Gamma(\text{hadrons})}{\Gamma_{\text{total}}}\right)^2 \approx 46\% \tag{5.7}$$

5.2.3 The jet "mass" parameter

Special care has to be taken when using jet parameters in conjunction with kinematic conservation constraints. It must be accounted for the fact that a jet is not a physical object and thus does not have a physical mass[1]. However the jet's

[1] The γ factor of a jet does not define a Lorentz transformation because the jet is actually a mixture of constituent with different values for γ. Strictly spoken, a jet does not have a γ

energy and momentum both being simply the sum of the energies and momenta of its constituents, are to this extent physical quantities.

If the construct of a "jet mass" is interpreted as a parameter of its Lorentz boost, it reflects the mass of the quark initiating the jet. Due to the small values in this picture and the non-linearity of the energy conservation in terms of masses, using mass parameters for kinematic constraint reconstruction of jets should generally be avoided.

5.3 Input

5.3.1 The relativistic Breit-Wigner distribution

At the limit of high total energy a first approximation of the mass distribution of short-lived particles like the W boson is the relativistic Breit-Wigner distribution. The two parameters of this distribution are the location M and the scale Γ:

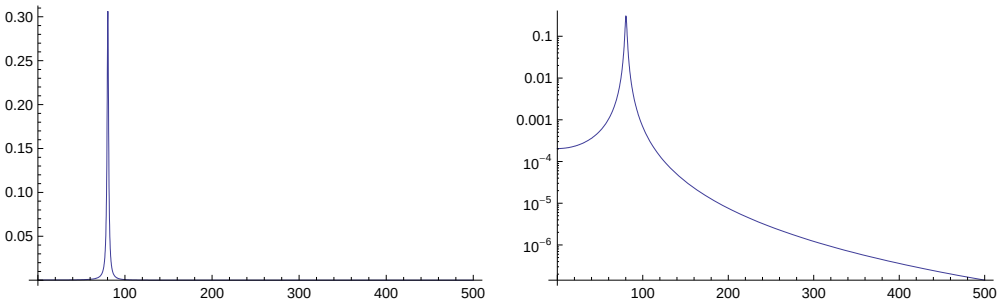
$$p(E) \propto \frac{1}{\left(E^2 - M^2\right)^2 + M^2\Gamma^2} \tag{5.8}$$

Since this distribution is a special case of the Cauchy distribution, its expectation is undefined and all other moments diverge [14]. It only becomes manageable by truncation. To be treated as a probability density function, this distribution has to be normalized for a fixed value range because the global integral diverges. Due to its asymmetry it is sensitive not only to the chosen value range but especially to the location parameter M. To illustrate the tails and the asymmetry of the relativistic Breit-Wigner distribution, it is plotted in figure 5.2 on a linear and a logarithmic scale.

5.3.2 The generated W masses

The masses generated by the Pythia [19] event generator however draw a different picture of the W mass distribution than the expected Breit-Wigner distribution shown above. The found mass distribution is plotted as a histogram in figure 5.3 and it can be seen, that the tails on both sides behave different from what one would expect.

factor and thus it has nothing like a mass.



(a) Linear axis (b) Logarithmic axis

Figure 5.2: The normalized relativistic Breit-Wigner distribution

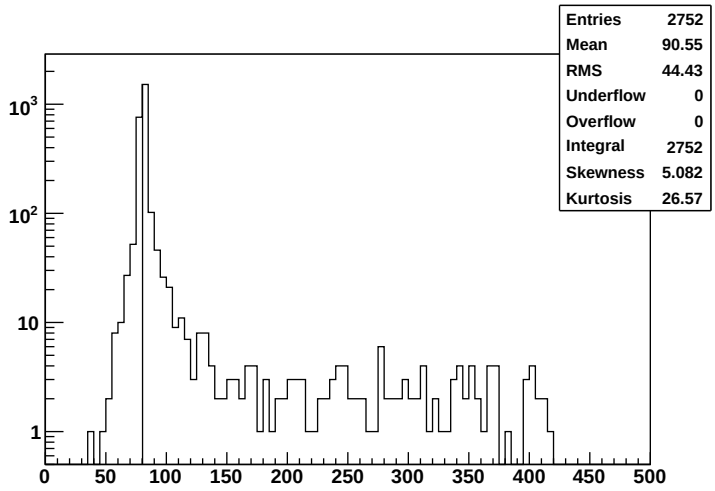


Figure 5.3: The W masses generated by Pythia are plotted as a histogram on a logarithmic scale

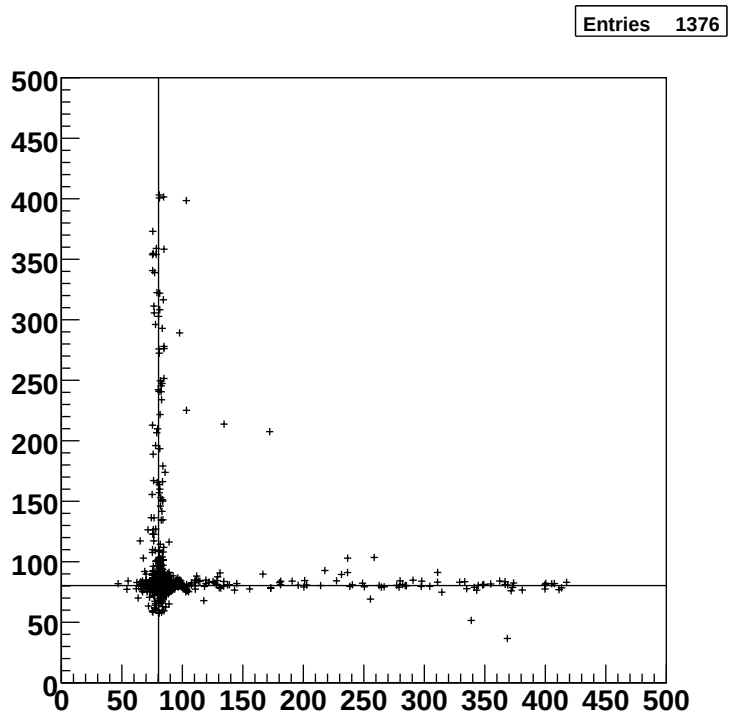


Figure 5.4: All generated W masses. Each cross represents one event. The axis are the masses of the two generated W particles.

It is instructive to compare the relative number of masses found above a certain limit for both, Breit-Wigner distributed masses and Pythia generated masses. It turns out, that about 1 % of the masses would be expected to be above 100 GeV but about 7 % of the Pythia generated masses are in that region. To check if this behaviour is due to an accident during Pythia configuration, the Pythia generator was invoked directly to produce another set of W masses from electron-positron collisions. The result was similar to the earlier distribution and a configuration error seems highly improbable.

However the approximation of negligible correlation of the masses of two W particles generated during one collision is supported by the output of Pythia as can be seen in figure 5.4. The correlation computed from this scatter plot is $\rho \approx 0.04$.

Because detailed study of the event generating mechanism would have been out of the scope of this diploma thesis, it was decided to sacrifice a smaller number of events and drop those events found in the tails of the mass distribution.

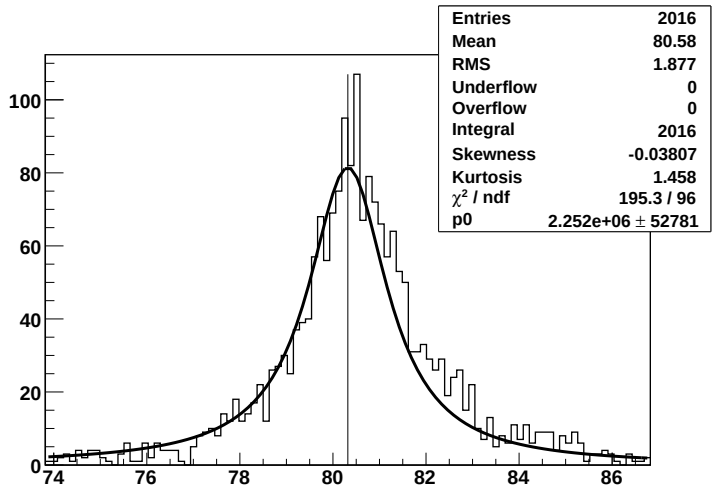


Figure 5.5: The generated W masses contained in all events selected for reconstruction with the expected Breit-Wigner distribution drawn over the histogram.

5.3.3 Statistics and event selection

The event selection of the finally used events was done using the generator information about the W masses. The selected range was chosen such as to include 90 % of the masses expected from a Breit-Wigner shape. All events containing masses outside 80.32 ± 6.50 GeV were dropped because they originated from different physical processes. This cut is not possible under real-world conditions, but helped to limit the scope of this work to regions where the generated masses were understood and seemed realistically distributed. The resulting mass distribution is shown in figure 5.5 together with the shape of the expected Breit-Wigner distribution and they show pretty good agreement.

Thanks to the event selection also the mean and standard deviation of the found distribution found in the upper right corner of all shown histograms get comparable to the expected values of 80.26 for the mean and 1.92 for the standard deviation when normalizing the Breit-Wigner distribution to the same range.

5.3.4 Error model

To model the projected performance of the jet reconstruction done at ILC experiments, the errors presented in table 5.1 have been applied to all four generated

Energy resolution	$\sigma_E/E = 30\%/\sqrt{E}$
Direction resolution	$\sigma_\theta = \sin(\theta)\sigma_\phi = 10$ mrad

Table 5.1: Projected performance for jet reconstruction at the ILD experiment.

jets using a Gaussian random function. The direction resolution affects the θ and ϕ measurements and the azimuth error is scaled to the polar angle θ.

The total momenta of the jets are unknown because physically the jet has no mass (see section 5.2.3). The implicitly defined masses used later for calculation of the W masses are virtual quantities obtained from momentum and energy conservation. If jet reconstruction later is able to reconstruct and assign the energies and masses of all (heavy) jet constituents, there will be a measurement for this virtual quantity as well and the outcome of this example will have to be reevaluated. To initialize the values of the total momenta, the energy seemed to be a sensible choice because the implicit jet mass is expected to be small compared to the jet energy. The errors on the total momentum is given nine times the error of the energy measurement to suppress the influence of the initialization during the fit. An alternative approach would have been to assign complete correlation between energy and total momentum, but this has not been used.

5.4 Kinematics

5.4.1 The unconstrained data

During the previous section, any position information within the input data has been left unmentioned. That is because it is not available. Although the jets could be assumed to start at the interaction point due to the negligible lifetime and thus flight path of the W bosons, these assumptions will not be included in the kinematic reconstruction. However if Rave is to be used for kinematic reconstruction, its interface will expect position parameters and errors within the input data. To solve this problem, two steps are necessary: First, any kinematic constraints involving position parameters must be decoupled from those involving momentum or energy parameters and can be dropped. Second, to avoid any influence of position parameters on the fit, they are assigned huge errors. Due to the weighting of parameters with their inverse error, all position information is

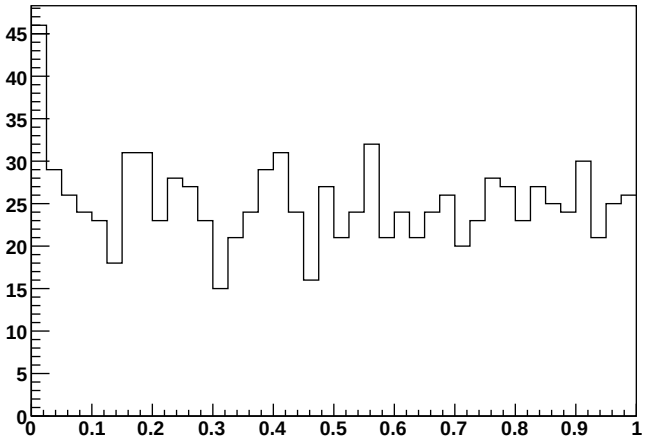


Figure 5.6: The χ^2 probability distribution of the input parameters calculated with 12 degrees of freedom.

thus deleted from the fit and has no influence on the resulting χ^2.

To analyze the effect constraints have on the data, first the unconstrained data is inspected. Figure 5.6 shows the χ^2 probability distribution of the unconstrained (input) data calculated assuming 12 degrees of freedom because of the unknown total momentum parameters.

5.4.2 Application of constraints

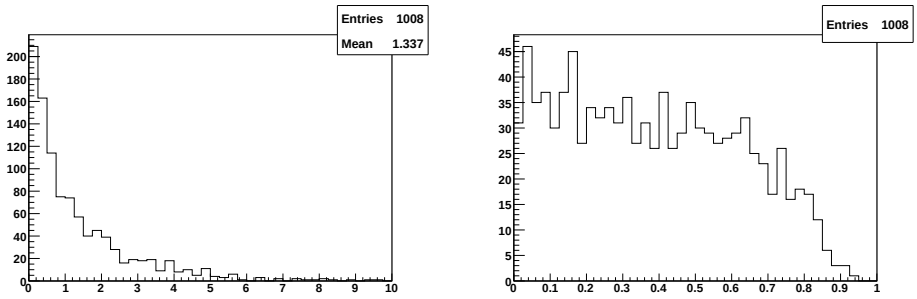
The constraints applied to the input parameters are those of energy and momentum conservation. They are explicitly written as:

$$E_1 + E_2 + E_3 + E_4 = 500\,\text{GeV} \tag{5.9}$$

$$\mathbf{p}_1 + \mathbf{p}_2 + \mathbf{p}_3 + \mathbf{p}_4 = \mathbf{0} \tag{5.10}$$

where the subscripts identify the four jets.

Taking these $1+3 = 4$ constraint equations and the error model from table 5.1 it is obvious, that the energy parameters and the momentum parameters do not interfere during the kinematic reconstruction. This means that the energy reconstruction and the momentum reconstruction can be safely assumed to be independent. The number of degrees of freedom of the fits can be calculated from the number of measurements n_m, the number of constraints n_c and the number



(a) χ^2 distribution (b) χ^2 probability distribution

Figure 5.7: The distribution of the χ^2 resulting from the kinematic fit and its probability distribution for one degree of freedom.

of fit parameters n_p as $n_{\text{dof}} = n_m + n_c - n_p$. From the above assumption it is clear that the energy reconstruction is a fit with no unknowns and one constraint and thus has $n_m + n_c - n_p = 4 + 1 - 4 = 1$ degrees of freedom. The momentum reconstruction on the other hand has $2 \times 4 + 3 - 12 = -1$ degrees of freedom and is thus under-determined. The total χ^2 of the kinematic fit is then expected to resemble a distribution with 1 degree of freedom. It has to be noted, that due to the under-determined fit of the momenta, there is still a hyper-surface of possible outcomes. However the chosen initialization of the total momenta has proven to allow for sensible fit results. The fit can be understood to reduce the under-determination of the parameters from 4 to 1 dimension thus increasing the level of information. Figure 5.7 shows the χ^2 distribution and the corresponding probability distribution for the assumed degree of freedom.

The number of iterations has been designed to be flexible and is determined by comparison of the returned χ^2 between two iterations. The maximum number of iterations as well as the break condition can be configured. It turns out that two iterations generally suffice for reasonable break condition values.

5.4.3 The influence of unknowns

The obvious influence of the unknown total jet momenta is that these four free parameters consume the information gain of the momentum conservation constraints. However their effect cannot be clearly separated because the total momenta themselves are no parameters in this fit but rather are participating the three direction momentum parameters. Nevertheless the covariance matrices

preserve the momentum direction information which is actually assumed to be measured.

5.4.4 Validation

The whole kinematic reconstruction has been reimplemented for this specific scenario in the Python programming language. This independent implementation of the kinematic fitting algorithms in a different language allowed for comparative checks between the Rave C++ source code and the Python source code and speeded up the debugging process. That speedup was primarily due to a much faster turnaround cycle for changes in the reconstruction and more flexible data handling within Python.

5.5 Association

Up until this point, no difference was made between the four jets. The applied kinematic constraints acted on all final states equally and did not take into account what particles they could have originated from. But because it is the goal of this reconstruction to provide a measurement of the W mass, the jets now will be grouped into two pairs, each of which will be assumed to have decayed from a W boson. This task of associating two out of four jets with one of two W bosons is not obvious because there are of course three different possibilities to do this association. It is the topic of this section to illustrate how the most likely of those three possibilities is chosen.

As with every statistical decision process, there are two types of errors. One is the error of rejecting a correct association, which is called a type 1 error. The other is the error of not rejecting an incorrect association, which is called a type 2 error. The measures described in the following will be characterized by showing the values of both of those error types in the final subsection 5.5.5.

5.5.1 Filtering

It is common practice to drop data which does not match certain criteria often formulated by a statistical test quantity. This is called filtering and in the context of jet association it denotes the process of rejecting (dropping) a set of possible

jet combinations. The advantage of a filter is, that it allows for potential data reduction while being cheap by means of processor time. Sensible filtering thus can make a reconstruction significantly more efficient.

There are different conditions upon which the dropped data-set can be selected. Without any measure distinguishing the possible combinations, every filter will be able to drop only whole events. This may be advantageous if the event has been incorrectly reconstructed and the correct association would have been confused with incorrect ones by later association steps. Such a condition could be based on the χ^2 values of the kinematic fit. It is common practice to accept a certain percentage of reconstructed events based on an empirical χ^2 cut calculated from the expected number of degrees of freedom and drop the rest.

During this reconstruction, a χ^2 cut has been applied on the output of the kinematic fitter.

5.5.2 Equal mass hypothesis

One possibility to distinguish the associations is to assume that each of them is valid and to calculate the W masses resulting from this assumption. Figure 5.8 shows a scatter plot of all resulting events. Each cross represents one event and the two W masses of the event are indicated by its position.

In a second step, a constrained fit could be run requiring the two resulting masses to be equal because they are both masses of W bosons. Such a requirement would favor the combinations along a diagonal line over those along the borders which does not agree with the true distribution of the W masses shown in figure 5.4. For this reason, the equal mass hypothesis was not used during this reconstruction.

5.5.3 Similar mass hypothesis

The most obvious improvement of the equal mass hypothesis would be to model a selection requirement from the uncorrelated Breit-Wigner distribution. That is of course not possible in real-world scenarios because the position parameter of the expected distribution is exactly what this reconstruction should deliver and any previous knowledge of it must not be used during the reconstruction.

There is however a compromise between the equal mass requirement and the

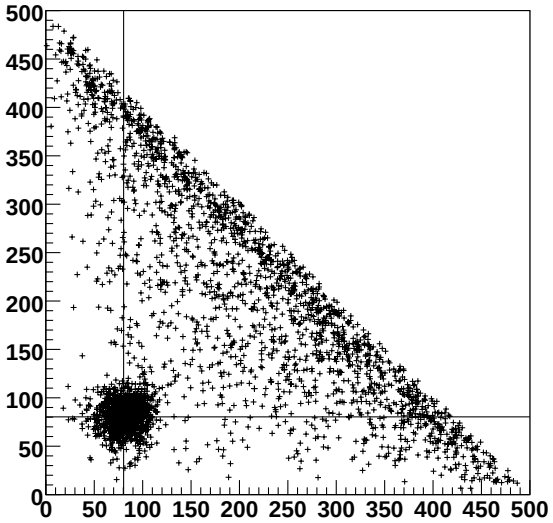


Figure 5.8: The W masses resulting from all possible jet associations. Each cross represents one event, i.e. one W pair and the masses of the two W bosons are indicated by the position of the cross.

above. The two W masses are known to not be equal so the requirement of equal masses is withdrawn. But they are still picked from the same distribution. This means that the condition selecting the jet association could involve a distribution shared by the two W masses. Then a maximum likelihood fit is derived to include the new requirement. The combination being most probable by means of likelihood is chosen. Of course in this case, the scale of the distribution has to be predefined because otherwise it would diverge to maximize the likelihood of any combination. The position of the distribution on the other hand is estimated during the optimization leaving one (fractional) degree of freedom. For the following fit is is useful to approximate the assumed Breit-Wigner distribution of the masses by a Gaussian with the same first momentum and an unknown mean.

The likelihood of a configuration is written as

$$L(\alpha_c|\alpha_m, \bar{m}) = L(\alpha_c|\alpha_m) \cdot L(m_1(\alpha_m), m_2(\alpha_m)|\bar{m}) \tag{5.11}$$

The first term holds the results of the kinematic fit. It models a multivariate Gaussian distribution according to the parameters (α_c) and covariances ($\mathbf{V}_c$) obtained from the kinematic reconstruction. The second term holds the new

similar-mass hypothesis. It models the distribution of the two masses dependent on the position parameter $\bar{m}$. It can be split into two symmetric terms if the masses are approximated to be uncorrelated. Taking the logarithm of equation (5.11) and multiplying with -2 results in a modified Least Squares method:

$$\begin{aligned} -2\ln L(\alpha_c|\alpha_m,\bar{m}) = &\left[\alpha_m-\alpha_c\right]^T \mathbf{V}_c^{-1}\left[\alpha_m-\alpha_c\right] \\ &-2\ln p\left(m_1\left(\alpha_m\right)|\bar{m}\right)-2\ln p\left(m_2\left(\alpha_m\right)|\bar{m}\right)+\text{const} \end{aligned} \tag{5.12}$$

Although any probability density function can be used for p, the Gaussian distribution is certainly the easiest choice. The new objective function then is

$$M(\alpha_m,\bar{m}) = \left[\alpha_m-\alpha_c\right]^T \mathbf{V}_c^{-1}\left[\alpha_m-\alpha_c\right] + \sum_{i=1,2}\frac{\left(m_i\left(\alpha_m\right)-\bar{m}\right)^2}{\sigma_t^2} \tag{5.13}$$

where σ_t is the predefined scale.

The objective function M is minimized with respect to α_m and $\bar{m}$ and yields a pseudo-χ^2 which was used as selection criterion. Using this method to select the best jet association out of all possible ones shown in figure 5.8 results in the distribution shown in figure 5.9.

Although the approximation of Gaussian distributed masses is not correct, the presented selection mechanism using the pseudo-χ^2 has shown to be reliable and introduces only small errors of both first and second kind as will be discussed in section 5.5.5.

5.5.4 Low correlation hypothesis

Already from comparison of figures 5.8 and 5.4 it is clear that a filter dropping the combinations where both masses exceed a certain limit would significantly improve the performance of the association. Such an upper limit could be derived from theoretical predictions or previous measurements. For this reconstruction this cut was made setting the limit to a value of 130 GeV. At this value only two events of the full data sample would be dropped and no events of the preselected data sample because the preselection cuts are much stronger (see section 5.3.3).

Figure 5.10 shows the distribution of the masses before and figure 5.12 shows the same distribution after the association. The distribution used for the associ-

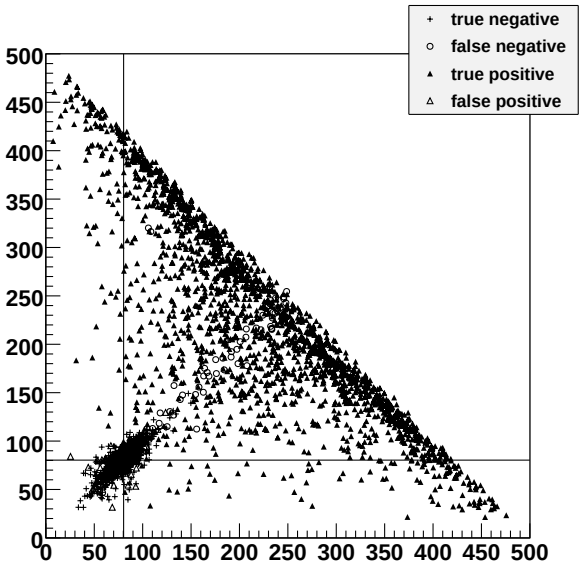


Figure 5.9: Association by probability of similar-mass-constrained fit. The density of empty markers (false positives and negatives) around the diagonal illustrate the weakness of the method in this region.

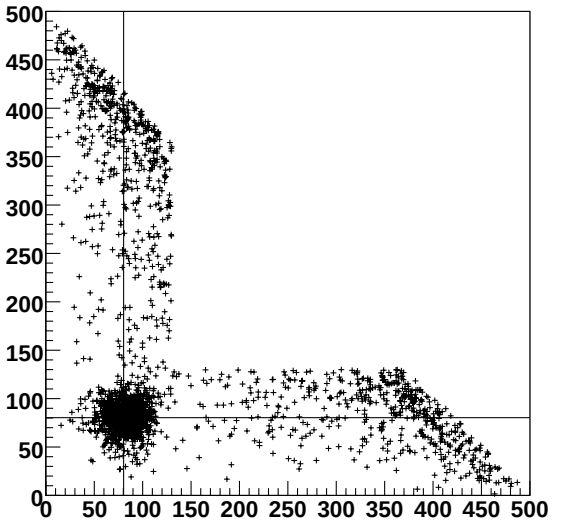


Figure 5.10: The distribution of all possible W masses with a cut on the maximum mass only one of the W bosons may exceed at 130 GeV, motivated by the low correlation hypothesis.

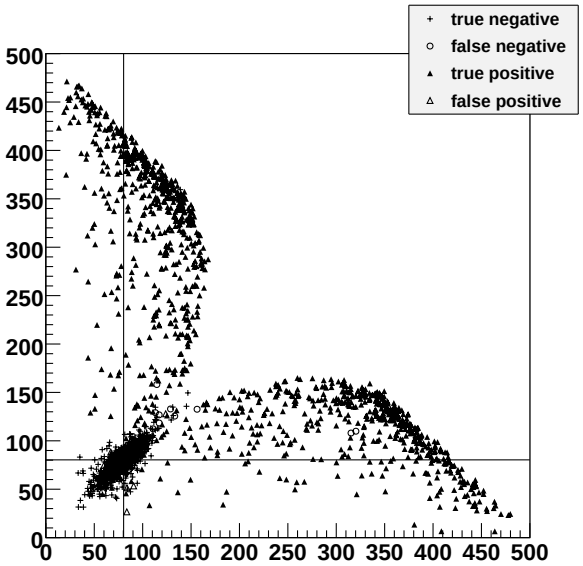


Figure 5.11: Association by probability of similar-mass-constrained fit which has been applied to the reduced set of all possible combination shown in figure 5.10

ation is shown in figure 5.11.

5.5.5 Performance analysis

Table 5.2 illustrates the performance of the association for the treated data sample. The two columns show the values for the same association process without and with the initial event selection discussed in section 5.3.3.

5.6 Global evaluation

All the W masses reconstructed during the single-event reconstruction are used to calculate an overall value for the position parameter of the W mass distribution.

5.6.1 The central limit theorem

The central limit theorem states that for a sequence of independent random variables X_i, each from a distribution with mean μ_i and variance σ_i, then the

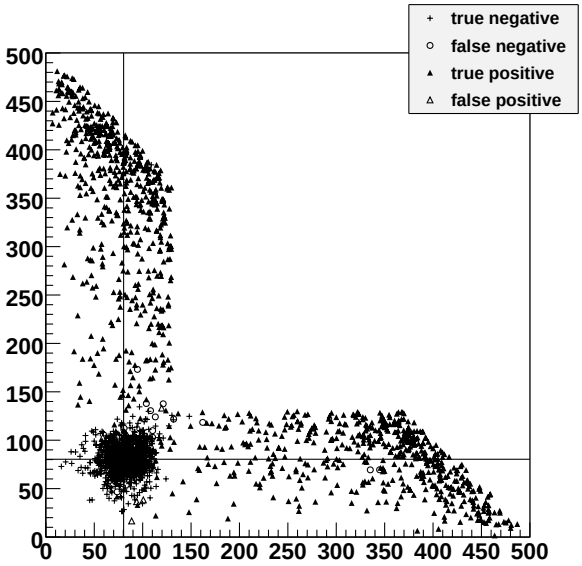


Figure 5.12: The distribution of all possible W masses after application of kinematic constraints to the final states. The results from the similar-mass hypothesis are removed from the parameters. They are only used for association purposes. The association indicated by the different markers is done using an additional similar-mass-constrained fit. The negatives (true and false) are used for the final fit.

Association step	error type	full sample	selected events
Similar Mass hypothesis only	type 1	13.5 %	6.0 %
	type 2	7.0 %	3.2 %
Low Correlation hypothesis only	type 1	0.1 %	0.1 %
	type 2	55.8 %	55.6 %
Both combined	type 1	6.2 %	**1.6** %
	type 2	3.2 %	**0.9** %

Table 5.2: The false positive (error type 1) and false negative (error type 2) rate of the association steps introduced above. The bold numbers indicate the error introduced upon the data-set for the following global evaluation. All ratios are also calculated for the full data sample without initial event selection to demonstrate the practical usability of the presented association method.

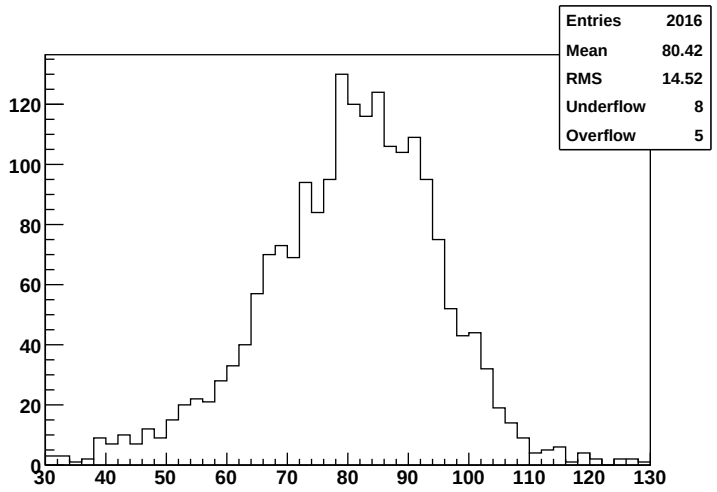


Figure 5.13: The distribution of the reconstructed W masses.

distribution of the sum $S = \sum X_i$ will have the mean $\sum \mu_i$ and the variance $\sum \sigma_i^2$. Even stronger, it says that as $N \to \infty$:

$$\frac{S - \sum_{i=1}^{N} \mu_i}{\sqrt{\sum_{i=1}^{N} \sigma_i^2}} \to N(0,1) \tag{5.14}$$

With that said, that the final result obtained from a reconstruction like the one done here will be normally distributed and the variance of that result will decrease with the number of reconstructed events like $1/N$.

5.6.2 Final result

The histogram of the reconstructed W masses is shown in figure 5.13. It contains all W masses obtained by the previous association step. The final result of this reconstruction of the position parameters of the W mass distribution has to be taken from the histogram:

$$\bar{m}_W = 80.42\,\mathrm{GeV/c} \qquad \sigma_{\bar{m}_W} = \frac{\sigma_{m_W}}{\sqrt{N}} = 0.32\,\mathrm{GeV/c} \tag{5.15}$$

The mean for the position is close to the expected value of 80.32 used for simulation (the vertical line in figure 5.3) and its offset is quite small compared with

the associated σ.

The deconvolution of the found distribution with a Breit-Wigner distribution succeeds in decreasing the mean value but has not been pursued because of an unsatisfactory χ^2 result.

For an ultimate evaluation of the possible precision, the constituents of the jets have to be reconstructed. This would allow for independent measurements of the associated jet momenta. According work is on the way. For now, the histogram mentioned above is the final conclusion.

5.6.3 The stability of the results

Because of the reduced data sample used for this reconstruction, the result is relatively insensitive to energy cuts applied on the final states. On the other hand it is obvious from the mass distribution shown in figure 5.3 that in the case of the reconstruction of the full sample, maximum energy cuts would affect the results. In that case the limit in equation (5.14) does not converge for the sample size used herein and other estimators have to be used for the final parameter estimation.

5.7 Discussion

5.7.1 Current status and physical limits

The current status of knowledge on the position parameter of the W mass distribution as summarized by the Particle Data Group [23] is

$$m_W = 80.398\ \mathrm{GeV/c} \qquad \sigma_{m_W} = 0.025\ \mathrm{GeV/c} \tag{5.16}$$

which is the result of a fit including data from six different experiments and an overall of approximately 260000 events. Some of the experiments report the systematic error separately and approximate it at the order of 0.03 GeV/c.

There is however a physical limit to the precision of measurements of this kind and the measurements at the ILC are likely to be dominated by this inevitable systematic error. Sources of the systematic limit are correlations between the two W bosons, like Colory Connection effects or Bose-Einstein correlations [16].

5.7.2 The impact of the ILC

The reconstruction demonstrated in this chapter gives a realistic precision for real measurements at the ILC. Together with the operation parameters presented in section 2.2.2 it is possible to foresee the precision the ILC will deliver for similar W mass reconstruction scenarios dependent on the actual luminosity.

Although the integrated luminosity of the ILC in the first four years is an explicit design goal at 500 fb^{-1}, this value will not be used in the following calculation. Instead the actual operation luminosity will be left as an open parameter for the number of events of the same type as used in the above reconstruction.

$$N_{WW} = \mathcal{L} \cdot \sigma \cdot \left(\frac{\Gamma_{\text{hadrons}}}{\Gamma_{\text{total}}} \right)^2 \cdot \epsilon \qquad \text{with} \qquad \mathcal{L} = \int L dt \tag{5.17}$$

Here σ is the cross section of the $e^+e^- \to WW$ process at 500 GeV center-of-mass energy. The value for this cross section has been calculated by Pythia to be 7238.0 fb. $\Gamma_{\text{hadrons}}/\Gamma_{\text{total}}$ is the probability of one W boson to decay into hadrons. From the decay modes of the W particle given in [23] that ratio is given to be 0.676 and its square is thus 0.457. Finally ϵ is the detector and jet reconstruction efficiency for four-jet events which is approximated with 40 %. This is the value with the biggest uncertainty in this calculation. Since there is no common value for this efficiency, it is derived from some assumptions.

The resulting precision is therefore:

$$\sigma_{m_W} \approx \frac{14.52}{\sqrt{\mathcal{L} \cdot 7238.0 \cdot 0.457 \cdot 0.4}} \approx \frac{14.52}{\sqrt{\mathcal{L} \cdot 1328.1}} \approx 0.4 \cdot \mathcal{L}^{-1/2}\ \text{fb}^{1/2}\text{GeV/c} \tag{5.18}$$

which would significantly improve the current total precision of the known W mass shown in section 5.7.1 during the first four year of operation if the designed value of $\mathcal{L}_{4y} = 500$ fb^{-1} is realized. If the reconstruction would have been done without kinematic constraints, the resulting precision can be approximated to be $\sigma_{m_W} = 0.0273$ GeV/c, which would not beat the current precision.

6 Conclusion and perspective

6.1 Conclusion

The Rave library, an effort to decouple the detector-independent parts of the CMS reconstruction framework and make them reusable by other experiments, has been extended for kinematic fitting facilities. As with the existing vertex reconstruction code, it was a major goal to maintain source code compatibility with the CMS code. This was accomplished by wrapping the CMS code into interface classes designed specifically for that purpose. Those new interface classes had to integrate into the existing set of Rave interface classes while providing full support for the possibilities of the underlying kinematic reconstruction code.

To demonstrate its usability in a real-world environment, Rave has been integrated into the existing reconstruction framework of the ILD detector concept. The vertex and kinematic reconstruction facilities of Rave are fully exposed to the user of that system allowing extensive testing and performance comparison with other solutions.

An exemplary reconstruction has been done to demonstrate the status of kinematic fitting in Rave. Events generated by the Pythia software have been used in conjunction with a Gaussian error model of the projected jet reconstruction precision of the ILD detector to produce an event sample. The simulated jets were then refitted with kinematic constraints and associated in pairs to their original W particles. The masses of the reconstructed W particles were then plotted as a histogram and the yielded mass resolution was used to calculate the W mass resolution the ILC probably will provide.

6.2 Perspective

Improvements of the presented kinematic reconstruction are possible and feasible at several levels:

- The kinematic reconstruction formalism may be extended to include second order terms of constraint equations. This could be especially useful for the behaviour of energy or mass constraints at small energies or masses.
- The algorithmic class to propagate the constraint information applied at lower levels in the decay tree up to the final states (LagrangeChildUpdator) needs implementation. This could be contributed back to the CMS software as well.
- Test cases for all existing constraints should be developed and included in routine testing.
- ILC-typical decay channels should be analyzed for the need of additional constraints, which in consequence should be implemented. Comparison with alternative solutions [8] would be interesting,
- A representative number of frequently used topologies should be implemented for MarlinRave.
- The integration into the Marlin default setup should be pursued. As a first step, MarlinRave needs to be adopted to the installation requirements of Marlins AFS setup.
- Kinematic reconstruction done in the context of ILC should be reproduced and the performance should be compared with the Rave performance.

It is planned that especially the last three points will get the attention of the author in the time after this diploma thesis because a successful integration of Rave into the default setup of the ILD reconstruction chain will probably result in more attention for the remaining points as an immediate consequence.

A Additional remarks

A.1 Track reconstruction

The data recorded by the detector, at least of the tracking detectors, usually consists of what is called "hits". A hit is the representation of the signal the detector received when a particle traversed it. Together with the knowledge about the detector resolution, it is a measurement of the position of such a traversal.

The path the traversing particle followed is called a track and it can be reconstructed by connecting the hits the particle caused in the detector. As for current experiments of course the task of track reconstruction is much more complicated than that. Due to the high collision energies today's high energy physics experiments operate at high multiplicities meaning that often 20 to 100 tracks per event need to be separated and reconstructed. Together with large background from beamstrahlung and from secondary activities of the particles, this complicates the task of associating measured hits to tracks. This problem is commonly targeted by sophisticated pattern recognition algorithms. For the purpose of this section, correct associations are assumed with exception of few remaining ambiguities.

Apart from the correctly associated hits containing also the resolution information of each measured coordinate, what is needed for successful track reconstruction is a theoretical model of the path the particle may have followed given the magnetic field it was moving in. This model, which will be referred to as the track model from now on, defines a track as a function of a few parameters such as a reference point and the particle momentum. More formally, the track model is a function mapping from five parameters to n measurements:

$$\mathbf{f} : \mathbf{p} \to f_i(\mathbf{p}), i = 1, \ldots, n \tag{A.1}$$

The actually measured coordinates then include a randomly distributed experimental error:

$$\mathbf{c} = \mathbf{f}(\hat{\mathbf{p}}) + \epsilon \tag{A.2}$$

where $\hat{\mathbf{p}}$ are the true parameters. It is then the task of the track reconstruction to find a mapping

$$\mathbf{F} : \mathbf{c} \to \mathbf{F}(\mathbf{c}) \equiv \tilde{\mathbf{p}} \tag{A.3}$$

which has no bias (A.4) and minimum variance (A.5).

$$\langle \tilde{\mathbf{p}} \rangle = \hat{\mathbf{p}} \tag{A.4}$$

$$\left\langle (\tilde{\mathbf{p}} - \hat{\mathbf{p}})^2 \right\rangle \equiv \sigma^2(\tilde{\mathbf{p}}) \to \text{Minimum} \tag{A.5}$$

A.1.1 Global track fit

The Least Squares Method (LSM), being commonly familiar, is a natural choice for the task (A.3) and if the track model is to a good approximation linear, the LSM turns out to have minimum variance among the linear and unbiased estimates. The linear expansion of the track model around the expansion point $\mathbf{p}_0$ is

$$\mathbf{f}(\mathbf{p}) = \mathbf{f}(\mathbf{p}_0) + \mathbf{A} \cdot (\mathbf{p} - \mathbf{p}_0) + O((\mathbf{p} - \mathbf{p}_0)^2) \qquad \mathbf{A} \equiv \left. \frac{\partial \mathbf{f}(\mathbf{p})}{\partial \mathbf{p}} \right|_{\mathbf{p}=\mathbf{p}_0} \tag{A.6}$$

Then with the general weight matrix notation

$$\mathbf{W}_\mathbf{x} \equiv \mathbf{V}_\mathbf{x}^{-1} \qquad \text{where} \qquad \mathbf{V}_\mathbf{x} \equiv \left\langle (\mathbf{x} - \langle \mathbf{x} \rangle) (\mathbf{x} - \langle \mathbf{x} \rangle)^T \right\rangle \tag{A.7}$$

the LSM minimizes the objective function

$$M(\mathbf{p}) = [\mathbf{f}(\mathbf{p}) - \mathbf{m}]^T \mathbf{W}_\mathbf{m} [\mathbf{f}(\mathbf{p}) - \mathbf{m}] \tag{A.8}$$

where $\mathbf{m}$ is a realization of the random quantity $\mathbf{c}$ defined in equation (A.2). Using the linearized track model (A.6) together with the abbreviations

$$\eta \equiv \mathbf{p} - \mathbf{p}_0 \qquad \text{and} \qquad \mathbf{f}_0 \equiv \mathbf{f}(\mathbf{p}_0) \tag{A.9}$$

the target function (A.8) gets

$$M(\mathbf{p}) = \left[\mathbf{f}_0 + \mathbf{A}\eta - \mathbf{m}\right]^T \mathbf{W}_\mathbf{m} \left[\mathbf{f}_0 + \mathbf{A}\eta - \mathbf{m}\right] \tag{A.10}$$

Minimization by demanding $\partial M(\mathbf{p})/\partial \mathbf{p} = \partial M(\mathbf{p})/\partial \eta = \mathbf{0}$ yields

$$\left.\frac{\partial M(\mathbf{p})}{\partial \eta}\right|_{\eta=\tilde{\eta}} = \mathbf{A}^T \left(\mathbf{W}_\mathbf{m} + \mathbf{W}_\mathbf{m}^T\right) \left[\mathbf{f}_0 + \mathbf{A}\tilde{\eta} - \mathbf{m}\right] = \mathbf{0} \tag{A.11}$$

$$\Rightarrow \tilde{\eta} = \left(\mathbf{A}^T \mathbf{W}_\mathbf{m} \mathbf{A}\right)^{-1} \mathbf{A}^T \mathbf{W}_\mathbf{m} \left(\mathbf{m} - \mathbf{f}_0\right) \tag{A.12}$$

$$\tilde{\mathbf{p}} = \mathbf{p}_0 + \tilde{\eta} \tag{A.13}$$

The covariance of the fitted parameters is [12]:

$$\mathrm{cov}\left(\tilde{\mathbf{p}}\right) = \left(\mathbf{A}^T \mathbf{W}_\mathbf{m} \mathbf{A}\right)^{-1} \tag{A.14}$$

Of course for this to work and to reveal the desired properties, several assumptions must hold:

- The track model must be correct i.e. the magnetic field must be known, the traversed material must be known and the equation of motion must be solved accurately (all with sufficient precision).
- The association of hits to tracks has to be correct, otherwise the track model can certainly not be correct.
- The covariance $\mathbf{V} = \left\langle \epsilon \epsilon^T \right\rangle$ must be known (i.e. the detector resolution has to be understood) and it must be non-singular.
- The measurement (A.2) must itself be unbiased i.e. $\langle \epsilon \rangle = \mathbf{0}$

A.1.2 Multiple scattering

At the precision of today's experiments and considering that tracked particles may traverse significant amounts of matter especially for low angles or muon tracking

in calorimeters one has to account for the effects of multiple scattering. As will be shown in this section, any scattering which has to a good approximation only linear effects on the track model with respect to an arbitrary scattering parameter can be incorporated into the weight matrix and the algorithm introduced above will work without modification.

First the track model (A.1) is revisited such as to account for the scattering parameters $\mathbf{s}$:

$$\mathbf{f} : \mathbf{p}, \mathbf{s} \to f_i(\mathbf{p}, \mathbf{s}) \qquad\qquad i = 1 \ldots N \tag{A.15}$$

The characteristics of the scattering material are subsumed in the diagonal weight matrix for the scattering parameters:

$$(\mathbf{W_s})_{l,l} = \frac{1}{\sigma(s_l)} \qquad \sigma(s_l) \approx \frac{15}{p\beta}\sqrt{L/L_{\mathrm{RAD}}} \qquad l = 1 \ldots M \tag{A.16}$$

Here p is the momentum of the particle, β is as usual the velocity of the particle relative to the speed of light and the term below the square root is the thickness of the material layer in unit of the radiation length. M is the number of layers the particle is traversing.

Using this additional information, the target function (A.8) is rewritten to

$$M(\mathbf{p}, \mathbf{s}) = [\mathbf{f}(\mathbf{p}, \mathbf{s}) - \mathbf{m}]^T \mathbf{W_m} [\mathbf{f}(\mathbf{p}, \mathbf{s}) - \mathbf{m}] + \mathbf{s}^T \mathbf{W_s} \mathbf{s} \tag{A.17}$$

Linear expansion of the new track model (A.15) with respect to the scattering parameters around $\mathbf{s} = \mathbf{0}$ gives

$$\mathbf{f}(\mathbf{p}, \mathbf{s}) = \mathbf{f}(\mathbf{p}, \mathbf{0}) + \mathbf{B}\mathbf{s} + O(\mathbf{s}^2) \qquad \mathbf{B} \equiv \left.\frac{\partial \mathbf{f}(\mathbf{p}, \mathbf{s})}{\partial \mathbf{s}}\right|_{\mathbf{s}=\mathbf{0}} \tag{A.18}$$

and with the abbreviation $\mathbf{f_p} = \mathbf{f}(\mathbf{p}, \mathbf{0})$ equation (A.17) becomes

$$\begin{aligned} M(\mathbf{p},\mathbf{s}) &= [\mathbf{f_p} + \mathbf{Bs} - \mathbf{m}]^T \mathbf{W_m} [\mathbf{f_p} + \mathbf{Bs} - \mathbf{m}] + \mathbf{s}^T \mathbf{W_s} \mathbf{s} \\ &= [\mathbf{f_p} - \mathbf{m}]^T \mathbf{W_m} [\mathbf{f_p} - \mathbf{m}] + 2\mathbf{s}^T \mathbf{B}^T \mathbf{W_m} [\mathbf{f_p} - \mathbf{m}] \\ &\quad + \mathbf{s}^T \left[\mathbf{B}^T \mathbf{W_m} \mathbf{B} + \mathbf{W_s}\right] \mathbf{s} \end{aligned} \tag{A.19}$$

or

$$M(\mathbf{p},\mathbf{s}) = M(\mathbf{p},\mathbf{0}) + 2\mathbf{s}^T\mathbf{d} + \mathbf{s}^T\mathbf{D}\mathbf{s} \tag{A.20}$$

with

$$\mathbf{d} = \mathbf{B}^T \mathbf{W_m} [\mathbf{f_p} - \mathbf{m}] \qquad \mathbf{D} = \mathbf{B}^T \mathbf{W_m} \mathbf{B} + \mathbf{W_s} \tag{A.21}$$

Minimizing M with respect to $\mathbf{s}$ yields

$$\left.\frac{\partial M(\mathbf{p},\mathbf{s})}{\partial \mathbf{s}}\right|_{\mathbf{s}=\mathbf{0}} = 2\mathbf{d} + 2\mathbf{D}\mathbf{s} = \mathbf{0} \tag{A.22}$$

$$\Rightarrow \mathbf{D}\mathbf{s} = -\mathbf{d} \Rightarrow \mathbf{s} = -\mathbf{D}^{-1}\mathbf{d} \tag{A.23}$$

Substituting this solution in equation (A.20) and expanding for $\mathbf{d}$ gives

$$M(\mathbf{p},\mathbf{s})|_{\mathbf{s}=\text{const}} = M(\mathbf{p},\mathbf{0}) - \mathbf{d}^T\mathbf{D}^{-1}\mathbf{d} \tag{A.24}$$

$$= [\mathbf{f_p} - \mathbf{m}]^T \left(\mathbf{W_m} - \mathbf{W_m}\mathbf{B}\mathbf{D}^{-1}\mathbf{B}^T\mathbf{W_m}\right) [\mathbf{f_p} - \mathbf{m}] \tag{A.25}$$

The middle term then is the new weight matrix $\mathbf{W}'$ and it can be shown that

$$\mathbf{W}' \equiv \mathbf{W_m} - \mathbf{W_m}\mathbf{B}\mathbf{D}^{-1}\mathbf{B}^T\mathbf{W_m} = \left(\mathbf{W_m}^{-1} + \mathbf{B}\mathbf{W_s}^{-1}\mathbf{B}^T\right)^{-1} \tag{A.26}$$

Using this new weight matrix equation (A.17) can be rewritten to

$$M(\mathbf{p},\mathbf{s})|_{\mathbf{s}=\text{const}} = [\mathbf{f}(\mathbf{p},\mathbf{0}) - \mathbf{m}]^T \mathbf{W}' [\mathbf{f}(\mathbf{p},\mathbf{0}) - \mathbf{m}] \tag{A.27}$$

which resembles equation (A.8) from the previous section and the minimization procedure is therefore the same.

A.1.3 The Kalman filter

While the error matrix $\mathbf{V}$ and thus also the weight matrix $\mathbf{W}$ in equation (A.14) were diagonal because the detector errors are usually uncorrelated, the weight matrix $\mathbf{W}'$ in equation (A.26) is in general no longer diagonal. However the LSM requires the inversion of the weight matrix. Unfortunately the computing time for this inversion grows with the third power of the number of measurements which itself can get as large as 100 in complex detectors. Moreover outlier detection would require fast in- or exclusion of single measurements which in the case of the LSM triggers an entire recalculation of the weight matrix inversion.

An alternative to the LSM and the de-facto baseline in track fitting is the Kalman filter (i.e. a special case of it). This method follows the track from one measurement to the next updating the track parameters including the additional information. Being a recursive method, the whole filter is defined in a step-wise manner accounting for the measurements one by one.

What has been one global track model is replaced by one "measurement equation" per detector:

$$\mathbf{c}_k = \mathbf{f}_k(\hat{\mathbf{p}}_k) + \epsilon_k \qquad \langle \epsilon_k \rangle = 0 \qquad \mathrm{cov}(\epsilon_k) = \mathbf{V}_k = \mathbf{W}_k^{-1} \tag{A.28}$$

Furthermore the "system equation" (A.29) defines the prediction of the track parameters measured by detector k based on the fit including the measurements by all previous detectors up until $k-1$ and using the propagation function $\mathbf{h}_k$ between detector $k-1$ and detector k.

$$\hat{\mathbf{p}}_k = \mathbf{g}_k(\hat{\mathbf{p}}_{k-1}) + \mathbf{S}_k \delta_k \qquad \langle \delta_k \rangle = 0, \qquad \mathrm{cov}(\delta_k) = \mathbf{R}_k \tag{A.29}$$

δ_k together with its covariance matrix $\mathbf{R}_k$ describes expected random effects along the trajectory which are mainly due to multiple scattering. And $\mathbf{S}_k$ accounts for the different sensitivity of the parameters to that noise.

The Kalman filter in its basic form assumes both the measurement equation (A.28) and the system equation (A.29) to be linear. Otherwise they are approximated by their first-order Taylor expansion in the following:

$$\mathbf{g}_k(\mathbf{p}_{k-1}) = \mathbf{G}_k\mathbf{p}_{k-1} + \mathbf{g}_k(\mathbf{0}) \tag{A.30}$$

$$\mathbf{f}_k(\mathbf{p}_k) = \mathbf{A}_k\mathbf{p}_k + \mathbf{f}_k(\mathbf{0}) \tag{A.31}$$

For convenience, the constant terms are dropped in the following. Corresponding to those two equations, the Kalman filter is a two-step recursive method. On each iteration the first step generates a predicted estimate based on the information included during previous iterations. The second step merges that prediction with the information of the current iteration. To distinguish between the estimates representing different amounts of information, they are denoted by $\tilde{\mathbf{p}}_{k|j}$ where k identifies the filter iteration and j is the number of measurements $\{\mathbf{m}_1, \ldots, \mathbf{m}_j\}$ used for this estimation. Here $\mathbf{m}_k$ are realizations of $\mathbf{c}_k$ in equation (A.28).

Using this notation, the first step uses the system equation and does linear error propagation to include the preceding covariance matrix together with the process noise into the prediction of the current covariance matrix:

$$\tilde{\mathbf{p}}_{k|k-1} = \mathbf{G}_k\tilde{\mathbf{p}}_{k-1|k-1} \tag{A.32}$$

$$\tilde{\mathbf{C}}_{k|k-1} = \mathbf{G}_k\tilde{\mathbf{C}}_{k-1|k-1}\mathbf{G}_k^T + \mathbf{S}_k\mathbf{R}_k\mathbf{S}_k^T \tag{A.33}$$

The second step uses least-squares estimation to include the measurement $\mathbf{m}_k$ into the estimate of $\mathbf{p}_k$. Thus it minimizes the objective function

$$\begin{aligned} M(\mathbf{p}_k) = {} & (\mathbf{A}_k\mathbf{p}_k - \mathbf{m}_k)^T \mathbf{W}_k (\mathbf{A}_k\mathbf{p}_k - \mathbf{m}_k) \\ & + \left(\mathbf{p}_k - \tilde{\mathbf{p}}_{k|k-1}\right)^T \tilde{\mathbf{C}}_{k|k-1}^{-1} \left(\mathbf{p}_k - \tilde{\mathbf{p}}_{k|k-1}\right) \end{aligned} \tag{A.34}$$

The result of the minimization of M with respect to $\mathbf{p}_k$ solved for $\mathbf{p}_k = \tilde{\mathbf{p}}_{k|k}$ is

$$\tilde{\mathbf{p}}_{k|k} = \tilde{\mathbf{C}}_{k|k}\left[\tilde{\mathbf{C}}_{k|k-1}^{-1}\tilde{\mathbf{p}}_{k|k-1} + \mathbf{A}_k^T\mathbf{W}_k\mathbf{m}_k\right] \tag{A.35}$$

$$\tilde{\mathbf{C}}_{k|k} = \left[\tilde{\mathbf{C}}_{k|k-1}^{-1} + \mathbf{A}_k^T\mathbf{W}_k\mathbf{A}_k\right]^{-1} \tag{A.36}$$

which, after some matrix algebra, can be written in the more common form

$$\tilde{\mathbf{p}}_{k|k} = \tilde{\mathbf{p}}_{k|k-1} + \mathbf{K}_k \left(\mathbf{m}_k - \mathbf{A}_k \tilde{\mathbf{p}}_{k|k-1} \right) \tag{A.37}$$

$$\mathbf{K}_k = \tilde{\mathbf{C}}_{k|k-1} \mathbf{H}_k^T \left(\mathbf{V}_k + \mathbf{A}_k \tilde{\mathbf{C}}_{k|k-1} \mathbf{A}_k^T \right)^{-1} \tag{A.38}$$

$$\tilde{\mathbf{C}}_{k|k} = \left(\mathbf{1} - \mathbf{K}_k \mathbf{A}_k \right) \tilde{\mathbf{C}}_{k|k-1} \tag{A.39}$$

To initiate the recursion, a choice for the initial estimate $\mathbf{p}_{1|1}$ and its covariance matrix $\mathbf{C}_{1|1}$ has to be made. If the first observation provides sufficient information, it can be used as starting point. Otherwise unmeasured components are given arbitrary but large errors in order to minimize their influence on the subsequent steps.

Of course there are several assumptions made on the way to equation (A.37), most of them in the first few steps of linearization. It is therefore legitimate to ask for indications, that the result obtained is actually meaningful. This is where the test for goodness of fit are introduced. This test consists of two separate parts: analysis of pull quantities and evaluation of the resulting χ^2.

The pull quantities (or reduced residuals) (A.40) provide evidence that covariance matrix of the measurement errors are correct, that the track model is correct and that the reconstruction program work properly. If the variance of the pulls quantities differs significantly from 1, at least one of these conditions is not met.

$$p_i = \frac{m_i - \tilde{c}_i}{\left[\sigma^2(c_i) - \sigma^2(\tilde{c}_i) \right]^{\frac{1}{2}}} \tag{A.40}$$

Testing the χ^2 resulting from the fit is a more global check. In the case of Gaussian errors, the whole χ^2 distribution can be observed, while in other cases only its mean value may be of use. The parameter of the expected χ^2 distribution, which at the same time is its mean value, is usually called the number of degrees of freedom of the fit. In the case of five fitted parameters, this number is the total number of measured coordinates minus five. In section 3.2 the influence of constraints on the resulting χ^2 distribution will be discussed. Given the expected parameter it is convenient to plot the value of the cumulative distribution function instead of the χ^2 values themselves. In the ideal case, this plot shows a uniform distribution.

In the case of the Kalman filter, the residuals $\mathbf{r}_{k|k}$ along with the χ^2 can be continuously evaluated in the following way:

$$\mathbf{r}_{k|k} = \mathbf{m}_k - \mathbf{H}_k \mathbf{p}_{k|k} \tag{A.41}$$

$$\mathbf{Q}_{k|k} = \mathbf{C}(\mathbf{r}_{k|k}) = \mathbf{V}_k - \mathbf{H}_k \mathbf{C}_{k|k} \mathbf{H}_k^T \tag{A.42}$$

$$\chi^2_{k,F} = \mathbf{r}^T_{k|k} \mathbf{Q}^{-1}_{k|k} \mathbf{r}_{k|k} \tag{A.43}$$

$$\tag{A.44}$$

Here $\chi^2_{k,F}$ is χ^2-distributed with the parameter being the dimension of $\mathbf{m}_k$. The total χ^2 of the fit, χ^2_k, is the sum of those of the individual filter steps:

$$\chi^2_k = \chi^2_{k-1} + \chi^2_{k,F} \tag{A.45}$$

where the distribution of χ^2_k is, as before, dependent on the total number of measurements and the number of fitted parameters.

A.1.4 Robust filtering

The conditions under which the algorithms discussed so far have optimal properties are:

- The linear track model is at least a good approximation of the actual track.
- The measurement errors and also the process noise follow a Gaussian distribution.

While the first condition can be met by adjusting the track model accordingly, the second condition is hardly ever met in real detectors. In those situations nonlinear estimators, i.e. estimators not based on the linear least-squares method, may perform better.

There are several possible choices for such an estimator, the M-estimator and the Gaussian-sum filter being among the most common ones. Both introduce means to respond to a significant number of so called outliers, observations lying beyond the core Gauss distribution.

B Parameter conversions

B.1 Units

When talking about different sets of parameters and their (mutual) conversion, one has to define the units they are stored in. Luckily, in the area this document is treating, there has been the consent of a specific set of units which is presented in table B.1.

Energy	[GeV]
Momentum	[GeV/c]
Mass	[GeV/c^2]
Distance and position	[cm]
Time	[ns]
Magnetic field	[T]
Electric charge	[e]

Table B.1: Common units, used with *CMSSW*

Nevertheless there are slight deviations from this consent in some cases. For full transparency, the units used with the respective perigee parameters mentioned below are indicated in each case. The Cartesian coordinates in this document always use the units defined in table B.1.

The two parameters for the charge q and the mass m are not explicitly mentioned, but since the charge is sometimes embedded with perigee parameters, it will be used with some of the conversions.

B.2 CMSSW perigee parametrization

This section is based on [7] and can be understood as an extract of that document. The naming convention will be preserved an is presented in table B.2. The

order of the parameters in the table corresponds to their storage position in the AlgebraicVector.

ρ	$[\mathrm{cm}^{-1}]$	transverse curvature for charged particles inverse magnitude of transverse momentum otherwise
θ	$\in [0, \pi]$	polar angle of the trajectory with respect to the z-axis
ϕ_p	$\in (-\pi, \pi]$	azimuthal angle of the momentum in the xy-plane
ϵ	[cm]	signed transverse impact parameter
z_p	[cm]	signed longitudinal impact parameter

Table B.2: CMSSW perigee parameters commented

B.2.1 Conversion to Cartesian coordinates

Parameters

The *CMSSW* perigee parameters are converted to Cartesian coordinates ($\vec{r}$, $\vec{p}$) in the following way if $\vec{R}$ is the reference point (given in units of [cm]):

$$
\begin{aligned}
r_x &= R_x + \epsilon \sin(\phi_p) && \text{(B.1)}\\
r_y &= R_y - \epsilon \cos(\phi_p) && \text{(B.2)}\\
r_z &= R_z + z_p && \text{(B.3)}\\
p_x &= p_t \cos(\phi_p) && \text{(B.4)}\\
p_y &= p_t \sin(\phi_p) && \text{(B.5)}\\
p_z &= p_t \cot(\theta) && \text{(B.6)}
\end{aligned}
$$

where

$$
\begin{aligned}
p_t &= \begin{cases} a\left|\frac{qB_z}{\rho}\right| & \forall q \neq 0 \\ \frac{1}{\rho} & \forall q = 0 \end{cases} && \text{(B.7)}\\
a &= c \times 10^{-15} \approx 3 \times 10^{-3} && \text{(B.8)}
\end{aligned}
$$

See section B.5 for the derivation of the factor a.

Covariance

The Jacobian in this case is

$$\mathbf{J} = \frac{\partial(r_x, r_y, r_z, p_x, p_y, p_z)}{\partial(\rho, \theta, \phi_p, \epsilon, z_p)} \tag{B.9}$$

$$= \begin{pmatrix} 0 & 0 & \epsilon\cos(\phi_p) & \sin(\phi_p) & 0 \\ 0 & 0 & \epsilon\sin(\phi_p) & -\cos(\phi_p) & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \cos(\phi_p)\frac{\partial p_t}{\partial \rho} & 0 & -p_t\sin(\phi_p) & 0 & 0 \\ \sin(\phi_p)\frac{\partial p_t}{\partial \rho} & 0 & p_t\cos(\phi_p) & 0 & 0 \\ \cot(\theta)\frac{\partial p_t}{\partial \rho} & p_t[-1-\cot^2(\theta)] & 0 & 0 & 0 \end{pmatrix} \tag{B.10}$$

where

$$\frac{\partial p_t}{\partial \rho} = \begin{cases} -a|\frac{qB_z}{\rho^2}| & \forall q \neq 0, \rho > 0 \\ a|\frac{qB_z}{\rho^2}| & \forall q \neq 0, \rho < 0 \\ -\frac{1}{\rho^2} & \forall q = 0 \end{cases} \tag{B.11}$$

Then the Cartesian error $\mathbf{C}^{\text{Cartesian}}$ calculated form the perigee error $\mathbf{C}^{\text{perigee}}$ is

$$\mathbf{C}^{\text{Cartesian}} = \mathbf{J}^T\mathbf{C}^{\text{perigee}}\mathbf{J} \tag{B.12}$$

B.2.2 Conversion from Cartesian coordinates

Parameters

The *CMSSW* perigee parameters can be calculated from Cartesian coordinates in the following way if $\vec{R}$ is the reference point (given in units of [cm]):

$$\begin{aligned}
\rho &= \begin{cases} \operatorname{sgn}(-q)\, a \frac{qB_z}{p_t} & \forall q \neq 0 \\ \frac{1}{p_t} & \forall q = 0, \end{cases} && \text{(B.13)} \\
&= \frac{1}{p_t}\left\{1 - \operatorname{sgn}(q)\left[aqB_z + \operatorname{sgn}(q)\right]\right\} = \frac{\alpha}{p_t} && \text{(B.14)} \\
\theta &= \arctan\left(\frac{p_t}{p_z}\right) && \text{(B.15)} \\
\phi_p &= \arctan\left(\frac{p_y}{p_x}\right) && \text{(B.16)} \\
\epsilon &= d_t \operatorname{sgn}(\pi - \delta\phi) && \text{(B.17)} \\
z_p &= d_z && \text{(B.18)}
\end{aligned}$$

where q is the charge of the particle and

$$\begin{aligned}
\alpha &= 1 - \operatorname{sgn}(q)\left[aqB_z + \operatorname{sgn}(q)\right] && \text{(B.19)} \\
p_t &= \sqrt{p_x^2 + p_y^2} && \text{(B.20)} \\
\vec{d} &= \vec{r} - \vec{R} && \text{(B.21)} \\
d_t &= \sqrt{d_x^2 + d_y^2} && \text{(B.22)} \\
\phi_d &= \arctan\left(\frac{d_y}{d_x}\right) && \text{(B.23)} \\
\delta\phi &= \left[(\phi_p + 2\pi) \mod 2\pi\right] - \left[(\phi_d + 2\pi) \mod 2\pi\right] && \text{(B.24)}
\end{aligned}$$

Covariance

The Jacobian in this case is

$$\begin{aligned}
\mathbf{J} &= \frac{\partial(\rho, \theta, \phi_p, \epsilon, z_p)}{\partial(r_x, r_y, r_z, p_x, p_y, p_z)} && \text{(B.25)} \\
&= \begin{pmatrix}
0 & 0 & 0 & -\alpha\frac{p_x}{p_t^3} & -\alpha\frac{p_y}{p_t^3} & 0 \\
0 & 0 & 0 & \beta\frac{p_x}{p_t p_z} & \beta\frac{p_y}{p_t p_z} & -\beta\frac{p_t}{p_z^2} \\
0 & 0 & 0 & -\gamma\frac{p_y}{p_x^2} & \gamma\frac{1}{p_x} & 0 \\
\zeta\frac{d_x}{d_t} & \zeta\frac{d_y}{d_t} & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0
\end{pmatrix} && \text{(B.26)}
\end{aligned}$$

where

$$\beta = \left[1 + \left(\frac{p_t}{p_z}\right)^2\right]^{-1} \tag{B.27}$$

$$\gamma = \left[1 + \left(\frac{p_y}{p_x}\right)^2\right]^{-1} \tag{B.28}$$

$$\zeta = \operatorname{sgn}(\pi - \delta\phi) \tag{B.29}$$

Then the perigee error $\mathbf{C}^{\text{perigee}}$ calculated form the Cartesian error $\mathbf{C}^{\text{Cartesian}}$ is

$$\mathbf{C}^{\text{perigee}} = \mathbf{J}^T \mathbf{C}^{\text{Cartesian}} \mathbf{J} \tag{B.30}$$

B.3 LCIO perigee parametrization

This section is based on [15]. The naming conventions are preserved and the order of presentation in table B.3 is such as to correspond with the order proposed in the cited document, which is also used within the current version of the *LCIO* implementation.

d_0	[mm]	signed transverse impact parameter
ϕ_0	$\in [0, \pi]$	azimuthal angle of the momentum in the xy-plane
Ω	[mm^{-1}]	transverse curvature for charged particles inverse magnitude of transverse momentum otherwise
z_0	[mm]	signed longitudinal impact parameter
$\tan\lambda$	$\in (-\infty, \infty)$	slope $\frac{dz}{ds}$ where s is the path integral in the xy-plane

Table B.3: *LCIO* perigee parameters commented

B.3.1 Conversion to Cartesian coordinates

Parameters

The *LCIO* perigee parameters are converted to Cartesian coordinates ($\vec{r}$, $\vec{p}$) in the following way if $\vec{R}$ is the reference point (given in units of [cm]):

$$
\begin{aligned}
r_x &= R_x + d_0 \sin(\phi_0) \times 10^{-1} && \text{(B.31)}\\
r_y &= R_y - d_0 \cos(\phi_0) \times 10^{-1} && \text{(B.32)}\\
r_z &= R_z + z_0 \times 10^{-1} && \text{(B.33)}\\
p_x &= p_t \cos(\phi_0) && \text{(B.34)}\\
p_y &= p_t \sin(\phi_0) && \text{(B.35)}\\
p_z &= p_t \tan(\lambda) && \text{(B.36)}
\end{aligned}
$$

where

$$
p_t = \begin{cases} a\left|\frac{qB_z}{\Omega}\right| \times 10^{-1} & \forall q \neq 0 \\ \frac{1}{\Omega} \times 10^{-1} & \forall q = 0 \end{cases} \qquad \text{(B.37)}
$$

$$
\text{(B.38)}
$$

The factor a is defined in equation (B.8).

Covariance

The Jacobian in this case is

$$
\mathbf{J} = \frac{\partial(r_x, r_y, r_z, p_x, p_y, p_z)}{\partial(d_0, \phi_0, \Omega, \tan\lambda)} \qquad \text{(B.39)}
$$

$$
= \begin{pmatrix}
10^{-1}\sin(\phi_0) & d_0 10^{-1}\cos(\phi_0) & 0 & 0 & 0 \\
-10^{-1}\cos(\phi_0) & d_0 10^{-1}\sin(\phi_0) & 0 & 0 & 0 \\
0 & 0 & 0 & 10^{-1} & 0 \\
0 & -p_t \sin\phi_0 & \cos\phi_0 \frac{\partial p_t}{\partial \Omega} & 0 & 0 \\
0 & p_t \cos\phi_0 & \sin\phi_0 \frac{\partial p_t}{\partial \Omega} & 0 & 0 \\
0 & 0 & \tan\lambda \frac{\partial p_t}{\partial \Omega} & 0 & p_t
\end{pmatrix} \qquad \text{(B.40)}
$$

where

$$\frac{\partial p_t}{\partial \Omega} = \begin{cases} -a|\frac{qB_z}{\Omega^2}| \times 10^{-2} & \forall q \neq 0, \rho > 0 \\ a|\frac{qB_z}{\Omega^2}| \times 10^{-2} & \forall q \neq 0, \rho < 0 \\ -\frac{1}{\Omega^2} \times 10^{-2} & \forall q = 0 \end{cases} \tag{B.41}$$

Then the Cartesian error $\mathbf{C}^{\text{Cartesian}}$ calculated form the perigee error $\mathbf{C}^{\text{perigee}}$ is

$$\mathbf{C}^{\text{Cartesian}} = \mathbf{J}^T \mathbf{C}^{\text{perigee}} \mathbf{J} \tag{B.42}$$

B.3.2 Parameter conversion from Cartesian coordinates

The *LCIO* perigee parameters can be calculated from Cartesian coordinates in the following way if $\vec{R}$ is the reference point (given in units of [cm]):

$$\begin{aligned} d_0 &= [d_x \sin(\phi_0) - d_y \cos(\phi_0)] \times 10 && \text{(B.43)} \\ \phi_0 &= \arctan\left(\frac{p_y}{p_x}\right) && \text{(B.44)} \\ \Omega &= \begin{cases} \text{sgn}(-q) a \frac{qB_z}{p_t} & \forall q \neq 0 \\ \frac{1}{p_t} & \forall q = 0 \end{cases} && \text{(B.45)} \\ z_0 &= d_z \times 10 && \text{(B.46)} \\ \tan\lambda &= \frac{p_z}{p_t} && \text{(B.47)} \end{aligned}$$

where

$$\begin{aligned} p_t &= \sqrt{p_x^2 + p_y^2} && \text{(B.48)} \\ \vec{d} &= \vec{r} - \vec{R} && \text{(B.49)} \end{aligned}$$

B.4 Conversion between the two parameter sets

This section summarizes the conversion rules between the two perigee parametrizations discussed above. First, the conversion from the LCIO convention to the CMSSW convention:

$$\rho = \Omega \times 10 \tag{B.50}$$
$$\theta = \frac{\pi}{2} - \arctan(\tan\lambda) \tag{B.51}$$
$$\phi_p = \phi_0 \tag{B.52}$$
$$\epsilon = d_0 \times 10^{-1} \tag{B.53}$$
$$z_p = z_0 \times 10^{-1} \tag{B.54}$$

and second, the conversion from *CMSSW* to *LCIO*:

$$d_0 = \epsilon \times 10 \tag{B.55}$$
$$\phi_0 = \phi_p \tag{B.56}$$
$$\Omega = \rho \times 10^{-1} \tag{B.57}$$
$$z_0 = z_p \times 10 \tag{B.58}$$
$$\tan\lambda = \cot(\theta) \tag{B.59}$$

B.5 The factor "a" or the conversion from [e T cm] to [GeV / c]

To derive the factor a shown in equation (B.8), one can take the detour for the basic SI units. We obtain

$$[\mathrm{e}\cdot\mathrm{T}\cdot\mathrm{cm}] = 1.6\times10^{-19}[\mathrm{A}\cdot\mathrm{s}][\mathrm{kg}\cdot\mathrm{s}^{-2}\cdot\mathrm{A}^{-1}]10^{-2}[\mathrm{m}] \tag{B.60}$$
$$= 1.6\times10^{-21}[\mathrm{m}\cdot\mathrm{kg}\cdot\mathrm{s}^{-1}] \tag{B.61}$$
$$[\mathrm{GeV}/c] = \frac{10^9\times1.6\times10^{-19}[\mathrm{m}^2\cdot\mathrm{kg}\cdot\mathrm{s}^{-2}]}{3\times10^8[\mathrm{m}\cdot\mathrm{s}^{-1}]} \tag{B.62}$$
$$= \frac{1.6}{3}\times10^{-18}[\mathrm{m}\cdot\mathrm{kg}\cdot\mathrm{s}^{-1}] \tag{B.63}$$

dividing the two we get a as follows:

$$a = \frac{[\mathrm{e}\cdot\mathrm{T}\cdot\mathrm{cm}]}{[\mathrm{GeV}/c]} = \frac{1.6\times10^{-21}}{\frac{1.6}{3}\times10^{-18}} = 3\times10^{-3} \tag{B.64}$$

Bibliography

[1] Lcio homepage. Online. http://lcio.desy.de.

[2] Parameters for the linear collider. Online, November 2006. http://www.linearcollider.org/newsline/pdfs/20061207_LC_Parameters_Novfinal.pdf.

[3] P. Avery. Applied Fitting Theory I: General Least Squares Theory. CBX 91-72, October 1991.

[4] J. Bagger, B. Barish, N. Calder, A. de Roeck, J. L. Feng, F. Gilman, J. Hewett, J. Huth, J. Jackson, Y.-F. Kim, E. Kolb, J. Lykken, K. Matchev, H. Murayama, J. Siegrist, and R. Weiss. Discovering the Quantum Universe, 2005.

[5] T. Behnke, C. Damerell, J. Jaros, and A. Miyamoto. ILC Reference Design Report Volume 4 - Detectors. Technical report, ILC Global Design Effort and World Wide Study, 2007.

[6] T. Behnke et al. Detector Outline Document For The Large Detector Concept. Technical report, DESY, Germany, 2006.

[7] P. Billoir and S. Quian. Fast vertex fitting with a local parametrization of tracks. *Nuclear Instruments & Methods in Physics Research*, A311:139–150, 1992.

[8] J. Böhme and B. List. An Object-Oriented Kinematic Fitting Package. via private communication.

[9] J. Brau, Y. Okada, and N. Walker. ILC Reference Design Report Volume 1 - Executive Summary. Technical report, ILC Global Design Effort and World Wide Study, 2007.

[10] A. Denner. Techniques for the calculation of electroweak radiative corrections at the one-loop level and results for W-physics at LEP200. *Fortschritte der Physik*, 41:4, 1993.

[11] K. V. Driessen and P. Rousseuw. Computing LTS regression for large data sets. Technical report, University of Antwerp, 1999.

[12] R. Frühwirth, M. Regler, R. K. Bock, H. Grote, and D. Notz. *Data Analysis Techniques for High-Energy Physics, Second Edition*. Cambridge Monographs on Particle Physics, Nuclear Physics and Cosmology, 2000.

[13] F. Gaede. Marlin and LCCD - Software tools for the ILC. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 559:177–180, April 2006.

[14] F. James. *Statistical Methods in Experimental Physics, 2nd Edition*. World Scientific, 2006.

[15] T. Krämer. Track Parameters in LCIO. LC-DET-2006-004, August 2006.

[16] D. Liko. Limits of W mass measurements, July 2008. Private communication.

[17] N. Phinney, N. Toge, and N. Walker. ILC Reference Design Report Volume 3 - Accelerator. Technical report, ILC Global Design Effort and World Wide Study, 2007.

[18] K. Prokofiev. *Study of the $B_s^0 \to (J/\psi)\phi \to \mu^+\mu^- K^+ K^-$ Decay with the CMS Detector at LHC*. PhD thesis, Universität Zürich, 2005.

[19] T. Sjöstrand, L. Lönnblad, S. Mrenna, and P. Skands. PYTHIA 6.321. http://cepa.fnal.gov/psm/simulation/mcgen/lund/.

[20] W. Waltenberger. Homepage of the Rave toolkit. Online. http://projects.hepforge.org/rave.

[21] W. Waltenberger. *Development of Vertex Finding and Vertex Fitting Algorithms for CMS*. PhD thesis, Technische Universität Wien, November 2004.

[22] W. Waltenberger and F. Moser. RAVE - an open, extensible, detector-independent toolkit for reconstruction of interaction vertices. In B. Philips, editor, *Proc. IEEE Nuclear Science Symposium (NSS '06)*, volume 06. IEEE, 2006.

[23] W.-M. Yao, C. Amsler, D. Asner, R. Barnett, J. Beringer, P. Burchat, C. Carone, C. Caso, O. Dahl, G. D'Ambrosio, A. DeGouvea, M. Doser, S. Eidelman, J. Feng, T. Gherghetta, M. Goodman, C. Grab, D. Groom, A. Gurtu, K. Hagiwara, K. Hayes, J. Hernández-Rey, K. Hikasa, H. Jawahery, C. Kolda, Y. Kwon, M. Mangano, A. Manohar, A. Masoni, R. Miquel, K. Mönig, H. Murayama, K. Nakamura, S. Navas, K. Olive, L. Pape, C. Patrignani, A. Piepke, G. Punzi, G. Raffelt, J. Smith, M. Tanabashi, J. Terning, N. Törnqvist, T. Trippe, P. Vogel, T. Watari, C. Wohl, R. Workman, P. Zyla, B. Armstrong, G. Harper, V. Lugovsky, P. Schaffner, M. Artuso, K. Babu, H. Band, E. Barberio, M. Battaglia, H. Bichsel,

O. Biebel, P. Bloch, E. Blucher, R. Cahn, D. Casper, A. Cattai, A. Ceccucci, D. Chakraborty, R. Chivukula, G. Cowan, T. Damour, T. DeGrand, K. Desler, M. Dobbs, M. Drees, A. Edwards, D. Edwards, V. Elvira, J. Erler, V. Ezhela, W. Fetscher, B. Fields, B. Foster, D. Froidevaux, T. Gaisser, L. Garren, H.-J. Gerber, G. Gerbier, L. Gibbons, F. Gilman, G. Giudice, A. Gritsan, M. Grünewald, H. Haber, C. Hagmann, I. Hinchliffe, A. Höcker, P. Igo-Kemenes, J. Jackson, K. Johnson, D. Karlen, B. Kayser, D. Kirkby, S. Klein, K. Kleinknecht, I. Knowles, R. Kowalewski, P. Kreitz, B. Krusche, Y. Kuyanov, O. Lahav, P. Langacker, A. Liddle, Z. Ligeti, T. Liss, L. Littenberg, L. Liu, K. Lugovsky, S. Lugovsky, T. Mannel, D. Manley, W. Marciano, A. Martin, D. Milstead, M. Narain, P. Nason, Y. Nir, J. Peacock, S. Prell, A. Quadt, S. Raby, B. Ratcliff, E. Razuvaev, B. Renk, P. Richardson, S. Roesler, G. Rolandi, M. Ronan, L. Rosenberg, C. Sachrajda, S. Sarkar, M. Schmitt, O. Schneider, D. Scott, T. Sjöstrand, G. Smoot, P. Sokolsky, S. Spanier, H. Spieler, A. Stahl, T. Stanev, R. Streitmatter, T. Sumiyoshi, N. Tkachenko, G. Trilling, G. Valencia, K. van Bibber, M. Vincter, D. Ward, B. Webber, J. Wells, M. Whalley, L. Wolfenstein, J. Womersley, C. Woody, A. Yamamoto, O. Zenin, J. Zhang, and R.-Y. Zhu. Review of particle physics. *Journal of Physics G*, 33:1+, 2006.